



Overview

Casibase is an open-source [Domain Knowledge](#) Database & IM & Forum Software powered by [ChatGPT](#).

You need to enable JavaScript to run this app.

Casibase features

1. With a separate front-end and back-end architecture developed in Golang, Casibase supports high concurrency, provides web-based management UI and supports multiple languages including Chinese, English, French, German, Spanish, Japanese, Korean, Russian, Indonesian, Malay, Portuguese, Italian, and Vietnamese.
2. Casibase supports third-party application login, such as GitHub, Google, QQ, WeChat, etc., and supports the extension of third-party login with plugins.
3. Based on embedding and prompt engineering for knowledge management, Casibase supports customized embedding methods and language models.
4. Casibase supports integration with existing systems by db sync, so users can transition to Casibase smoothly.
5. Casibase supports mainstream databases: MySQL, PostgreSQL, SQL Server, etc., and supports the extension of new databases with plugins.
6. Casibase is a powerful tool for asset management, enabling easy connection to assets via RDP, VNC, and SSH protocols and efficient handling of remote connections to machines.
7. Casibase's Security Log Auditing feature allows you to effortlessly track and monitor remote connections with detailed audit logging, recording connection start time, duration, and other relevant details, and also enables you to capture and analyze API logs for Casdoor operations, enhancing security and operational transparency.
8. Casibase supports database management. Casibase's Database Management

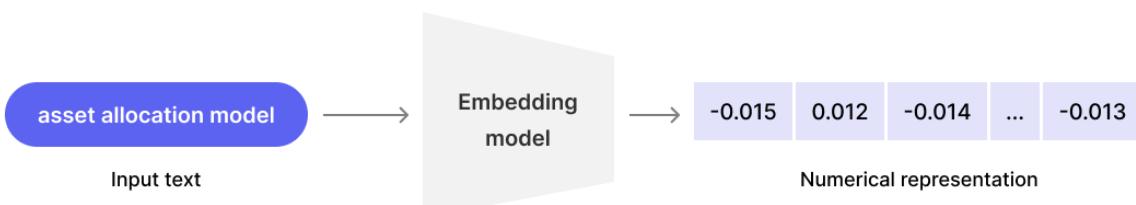
feature allows you to easily connect, manage, and organize databases while controlling access and simplifying user management and authorization for database resources.

9. Casibase is an open-source container cloud platform based on Docker and Kubernetes. It is suitable for individuals or organizations to build their own dedicated container cloud environment. Based on the Casbin permission management engine, Casibase implements fine-grained access control policies. Users can easily create, orchestrate, and manage container applications on Casibase. The project focuses on optimizing Casibase's application orchestration, service governance, and platform visualization core functions, improving platform usability and manageability, making it a leading lightweight container cloud platform.

How it works

Step 0 (Pre-knowledge)

Casibase's knowledge retrieval process is based on embedding and prompt engineering, so it is highly recommended that you take a brief look at how embedding works. An [introduction](#) to Embedding.



Step 1 (Importing Knowledge)

To get started with Casibase, users need to follow these steps to import knowledge and create a domain-specific knowledge database:

1. **Configure Storage:** In the Casibase dashboard, users should first configure the storage settings. This involves specifying the storage system to be used for storing knowledge-related files, such as documents, images, or any other relevant data. Users can choose from a variety of storage options based on their preferences and requirements.
2. **Upload Files to Storage:** Once the storage is set up, users can proceed to upload files containing domain-specific knowledge into the configured storage system. These files can be in various formats, such as text documents, images, or structured data files like CSV or JSON.
3. **Select Embedding Method for Knowledge Generation:** After the files are uploaded, users have the option to choose the embedding method for generating knowledge and corresponding vectors. Embeddings are numerical representations of textual or visual content, which facilitate efficient similarity search and data analysis.

TIP

How knowledge is embedded?

- For textual data: Users can choose from various embedding methods, such as Word2Vec, GloVe, or BERT, to convert the textual knowledge into meaningful vectors.
- For visual data: If the uploaded files contain images or visual content,

users can select image embedding techniques like CNN-based feature extraction to create representative vectors.

- More methods coming soon...

By following these steps, users can populate their domain knowledge database with relevant information and corresponding embeddings, which will be used for effective searching, clustering, and retrieval of knowledge within Casibase. The embedding process allows the system to understand the context and relationships between different pieces of knowledge, enabling more efficient and insightful knowledge management and exploration.

Step 2 (Retrieving Knowledge)

After importing your `domain knowledge`, Casibase transforms it into `vectors` and stores these vectors in a `vector database`. This vector representation enables powerful functions like `similarity search` and `efficient retrieval of related information`. You can quickly find relevant data based on context or content, enabling advanced querying and uncovering valuable insights within your domain knowledge.

Step 3 (Building the Prompt)

Casibase performs a similarity search on the stored knowledge vectors to find the closest match to the user's query. Using the search results, it creates a `prompt template` to frame a specific question for the `language model`. This ensures accurate and contextually relevant responses, delivering comprehensive answers based on the domain knowledge in Casibase.

Step 4 (Achieving the Goal)

At this stage, using Casibase, you have successfully acquired the knowledge you sought. Through the innovative combination of domain knowledge transformed into vectors and powerful language models like ChatGPT, Casibase provides you with accurate and relevant responses to your inquiries. This enables you to efficiently access and utilize the domain-specific information stored in Casibase, meeting your knowledge requirements with ease.

Step 5 (Optional Fine-tuning)

If you find that the results are not entirely satisfactory, you can try to get better results by doing the following:

- Tweaking Language Model Parameters
- Asking multiple questions
- Optimizing the original files

By utilizing these fine-tuning options, you can improve the efficiency of your knowledge management in Casibase, ensure that the system is better aligned with your goals, and provide more accurate and insightful information.

HINTS

Other ways to optimize results (may require source code changes):

- Updating `Embedding` Results: Refine the knowledge representation by adjusting the embedding results of your domain knowledge.

- Modifying **Prompt** Templates: By customizing the prompts, you can elicit more precise responses from the language model.
- Exploring Different **Language Models**: Experiment with different models to find the one that best suits your requirements for generating responses.

Online demo

Read-only site (any modification operation will fail)

- Chat bot (<https://ai.casibase.com>)
- Admin UI (<https://ai-admin.casibase.com>)

Writable site (original data will be restored for every 5 minutes)

- Chat bot (<https://demo.casibase.com>)
- Admin UI (<https://demo-admin.casibase.com>)

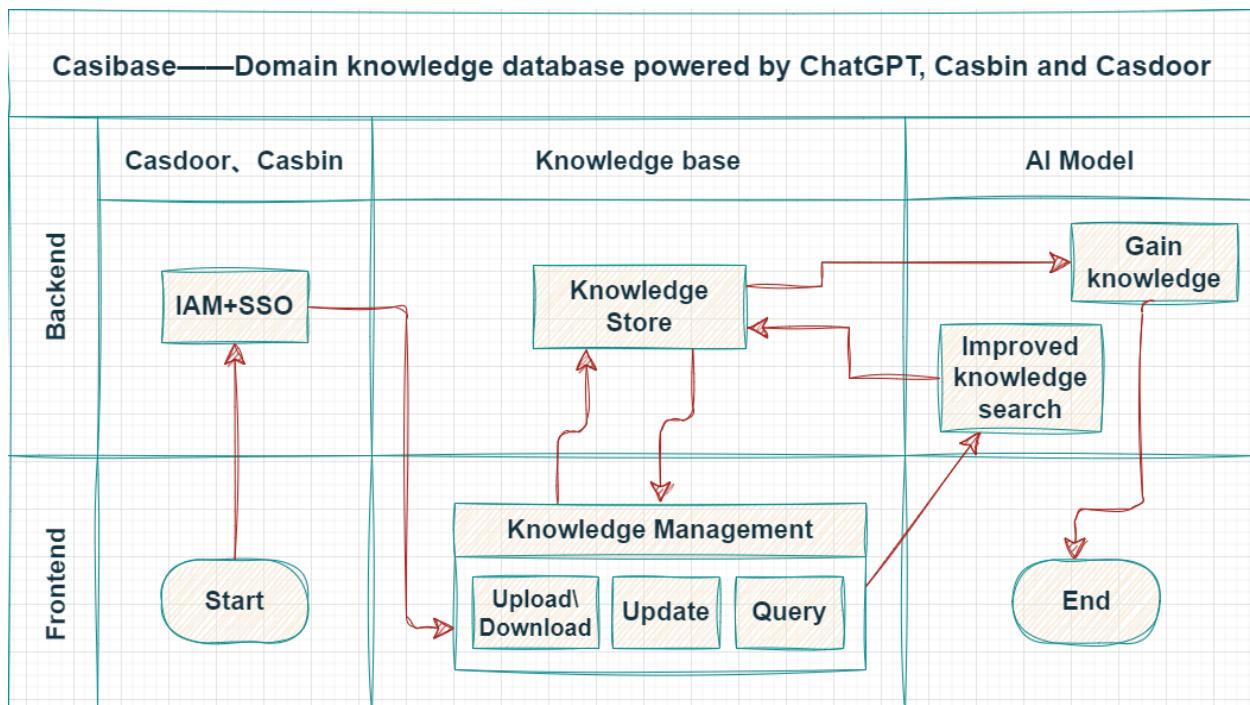
Global admin login:

- Username: **admin**
- Password: **123**

Architecture

Casibase contains 2 parts:

Name	Description	Language	Source code
Frontend	User interface for the casibase application	JavaScript + React	https://github.com/casibase/casibase/tree/master/web
Backend	Server-side logic and API for casibase	Golang + Beego + MySQL	https://github.com/casibase/casibase



Supported Models

Language Model

Model	Sub Type	Link
OpenAI	gpt-4-32k-0613 ◇ gpt-4-32k-0314 ◇ gpt-4-32k ◇ gpt-4-0613 ◇ gpt-4-0314 ◇ gpt-4 ◇ gpt-3.5-turbo-0613 ◇ gpt-3.5-turbo-0301 ◇ gpt-3.5-turbo-16k ◇ gpt-3.5-turbo-16k-0613 ◇ gpt-3.5-turbo ◇ text-davinci-003 ◇ text- davinci-002 ◇ text-curie-001 ◇ text- babbage-001 ◇ text-ada-001 ◇ text- davinci-001 ◇ davinci-instruct-beta ◇ davinci ◇ curie-instruct-beta ◇ curie ◇ ada ◇ babbage	OpenAI
Hugging Face	meta-llama/Llama-2-7b, tiiuae/falcon-180B, bigscience/bloom, gpt2, baichuan-inc/ Baichuan2-13B-Chat, THUDM/chatglm2-6b	Hugging Face
Claude	claude-2, claude-v1, claude-v1-100k, claude- instant-v1, claude-instant-v1-100k, claude-v1.3, claude-v1.3-100k, claude-v1.2, claude-v1.0, claude-instant-v1.1, claude-instant-v1.1-100k, claude-instant-v1.0	Claude
OpenRouter	google/palm-2-codechat-bison, google/ palm-2-chat-bison, openai/gpt-3.5-turbo, openai/gpt-3.5-turbo-16k, openai/gpt-4,	OpenRouter

Model	Sub Type	Link
	openai/gpt-4-32k, anthropic/claude-2, anthropic/claude-instant-v1, meta-llama/llama-2-13b-chat, meta-llama/llama-2-70b-chat, palm-2-codechat-bison, palm-2-chat-bison, gpt-3.5-turbo, gpt-3.5-turbo-16k, gpt-4, gpt-4-32k, claude-2, claude-instant-v1, llama-2-13b-chat, llama-2-70b-chat	
Ernie	ERNIE-Bot, ERNIE-Bot-turbo, BLOOMZ-7B, Llama-2	Ernie
iFlytek	spark-v1.5, spark-v2.0	iFlytek
ChatGLM	chatglm2-6b	ChatGLM
MiniMax	abab5-chat	MiniMax
Local	custom-model	Local Computer

Embedding Model

Model	Sub Type	Link
OpenAI	AdaSimilarity, BabbageSimilarity, CurieSimilarity, DavinciSimilarity, AdaSearchDocument, AdaSearchQuery, BabbageSearchDocument, BabbageSearchQuery, CurieSearchDocument, CurieSearchQuery, DavinciSearchDocument,	OpenAI

Model	Sub Type	Link
	DavinciSearchQuery, AdaCodeSearchCode, AdaCodeSearchText, BabbageCodeSearchCode, BabbageCodeSearchText, AdaEmbeddingV2	
Hugging Face	sentence-transformers/all-MiniLM-L6-v2	Hugging Face
Cohere	embed-english-v2.0, embed-english-light-v2.0, embed-multilingual-v2.0	Cohere
Ernie	default	Ernie
Local	custom-embedding	Local Computer

Core Concepts

As Casibase's user, you should get familiar with at least 4 core concepts:

Provider, Storage, Chat and Vector.

Providers

Providers are the backbone of Casibase, offering essential services and integration with external systems. The Provider class definition is shown as follows:

```
type Provider struct {
    Owner      string `xorm:"varchar(100) notnull pk" json:"owner"`
    Name       string `xorm:"varchar(100) notnull pk" json:"name"`
    CreatedTime string `xorm:"varchar(100)" json:"createdTime"`

    DisplayName string `xorm:"varchar(100)" json:"displayName"`
    Category    string `xorm:"varchar(100)" json:"category"`
    Type        string `xorm:"varchar(100)" json:"type"`
    ClientId    string `xorm:"varchar(100)" json:"clientId"`
    ClientSecret string `xorm:"varchar(2000)" json:"clientSecret"`
    ProviderUrl string `xorm:"varchar(200)" json:"providerUrl"`
}
```



There are two primary types of providers in Casibase:

- Storage Providers. The Storage Providers facilitates the storage and

retrieval of data within Casibase. It supports various storage options, including:

- AWS
- Azure
- Local File System

- **AI Providers.** The AI Providers are responsible for handling AI-related tasks and services in Casibase. It supports multiple AI models and technologies, including:
 - OpenAI
 - ChatGLM
 - InternLM

Vectors

Vectors in Casibase represent numerical representations of different types of data. These vectors enable efficient processing and analysis of information. Some of the vector types available are:

- Text Vector
- Image Vector
- ... (other vector types)

The Vector class definition is shown as follows:

```
type Vector struct {
    Owner      string      `xorm:"varchar(100) notnull pk"
```

Chats

Chats are at the core of interactive communication between users and the AI models in Casibase. They consist of three essential components:

- Question: The user's input or query, seeking information or assistance.
- Query Prompt: A formatted version of the user's question, prepared for processing by the AI models.
- Answer: The AI-generated response to the user's question, providing relevant information or solutions.

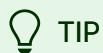
The Chat class definition is shown as follows:

```
type Chat struct {
    Owner      string    `xorm:"varchar(100) notnull pk"`
    json:"owner"`
    Name       string    `xorm:"varchar(100) notnull pk"`
    json:"name"`
    CreatedTime string   `xorm:"varchar(100)" json:"createdTime"`
    UpdatedTime string   `xorm:"varchar(100)" json:"updatedTime"`

    DisplayName string   `xorm:"varchar(100)" json:"displayName"`
    Category    string   `xorm:"varchar(100)" json:"category"`
    Type        string   `xorm:"varchar(100)" json:"type"`
    User1       string   `xorm:"varchar(100)" json:"user1"`
    User2       string   `xorm:"varchar(100)" json:"user2"`
    Users       []string `xorm:"varchar(100)" json:"users"`
    MessageCount int      `json:"messageCount"`
}
```

Embedding

Embedding is the process of transforming various types of data, such as text and images, into dense vector representations. This step is crucial for facilitating efficient data processing and analysis within Casibase.



TIP

- By embedding, the questions in chat and the knowledge files in storage will be turned into vectors and used in the next step of knowledge search.
- Casibase's default embedding method is provided by OpenAI at a rate of up to three calls per minute. We recommend minimizing coupling between knowledge files to facilitate embedding and further processing.

Transactions

Transactions track AI token usage and costs through Casdoor integration. Each message generates a transaction record for billing and usage monitoring. See [Billing & Usage](#) for details.

Records

Records support data aggregation through a `Count` field, allowing consolidated logging without storing individual entries. See [Records](#) for more information.

Scans

Scans represent network and security scanning operations executed against infrastructure assets. The Scan object enables automated discovery, security auditing, and system assessment through integrated scan providers.

Each scan targets a specific asset (such as a virtual machine) or IP address, executes using a configured scan provider (like Nmap or OS Patch), and captures results in both raw and structured formats. Scans integrate with the asset inventory to automatically resolve target addresses and maintain historical scanning records for security analysis.

Server Installation

Requirements

OS

All major operating systems including Windows, Linux and macOS are supported.

Environment

- Go 1.20+
- Node.js LTS (18)
- Yarn 1.x

❗ INFO

The use of Casibase is divided into two steps:

- step1: Deploy and run Casdoor
- step2: Deploy and run Casibase (this docs)

We strongly suggest you use Yarn 1.x to run & build Casdoor&Casibase frontend, using NPM might cause UI styling issues, see more details at: [casdoor#294](#)

⚠ CAUTION

For Chinese users, in order to download the Go dependency packages successfully, you need to use a Go proxy by Configuring the GOPROXY environment variable. We strongly recommend: <https://goproxy.cn/>

Database

Casibase uses [XORM](#) to talk to the database. Based on [Xorm Drivers Support](#), Casibase currently provides support for the following databases:

- [MySQL](#)
- [MariaDB](#)
- [PostgreSQL](#)
- [CockroachDB](#)
- [SQL Server](#)
- [Oracle](#)
- [SQLite 3](#)
- [TiDB](#)

guacd

Casibase uses guacamole-server to provide remote desktop access. If you want to use this feature, you need to install guacamole-server first. If you haven't installed guacamole-server, please refer to [guacamole-server Installation](#).

You can also run guacd in docker with the following command:

```
docker run -d --name guacd -p 4822:4822 guacamole/guacd
```

Download

Casibase can be installed using pre-built binaries or by building from source.

Pre-built Binaries

For production deployments, we recommend using pre-built binaries. Download the latest release from the [GitHub Releases page](#). Casibase provides binaries for:

- Linux: x86_64 (amd64) and ARM64
- Windows: x86_64 (amd64) and ARM64
- macOS: x86_64 (amd64) and ARM64

Extract the downloaded archive and you'll find the Casibase binary along with the web frontend and configuration files ready to use.

Build from Source

The source code of Casibase is hosted at GitHub: <https://github.com/casibase/casibase>. Both the Go backend code and React frontend code are inside the single repository.

Name	Description	Language	Source code
Frontend	Web frontend UI for Casibase	JavaScript + React	https://github.com/casibase/casibase/tree/master/web
Backend	RESTful API backend for Casibase	Golang + Beego + XORM	https://github.com/casibase/casibase

Casibase supports [Go Modules](#). To download the code, you can just simply clone the code via git:

```
cd path/to/folder  
git clone https://github.com/casibase/casibase
```

Configuration

Configure Casdoor

Please refer to [Casdoor-SSO](#) section to configure Casdoor.

Remember your `clientId`, `clientSecret`, `organization`, `application` and so on in Casdoor configuration, we will use them later.

Configure Database

Casibase supports mysql, mssql, sqlite3, postgres. Casibase uses mysql by default.

MySQL

Casibase will store its users, nodes and topics information in a MySQL database named: `casibase`. If the database does not exist, it needs to be created manually. The DB connection string can be specified at: <https://github.com/casibase/casibase/blob/master/conf/app.conf>

```
driverName = mysql  
dataSourceName = root:123456@tcp(localhost:3306)/  
dbName = casibase
```

PostgreSQL

Since we must choose a database when opening Postgres with xorm, you should prepare a database manually before running Casibase.

Let's assume that you have already prepared a database called `casibase`, then you should specify `app.conf` like this:

```
driverName = postgres
dataSourceName = "user=postgres password=postgres host=localhost
port=5432 sslmode=disable dbname=casibase"
dbName =
```

 INFO

For PostgreSQL, make sure `dataSourceName` has non-empty `dbName` and leave the standalone `dbName` field empty like the above example.

CockroachDB

You can also use Cockroachdb with postgres driver. It has same configuration as PostgreSQL.

```
driverName = postgres
dataSourceName = "user=postgres password=postgres host=localhost
port=5432 sslmode=disable dbname=casibase
serial_normalization=virtual_sequence"
dbName =
```

 INFO

For CockroachDB, don't forget to add `serial_normalization=virtual_sequence` to the `dataSourceName` like the above example. otherwise you will get error regarding existed database, whenever the service starts or restarts. Notice, this must be added before the database created.

Sqlite3

You should specify `app.conf` like this:

```
driverName = sqlite
dataSourceName = "file:casibase.db?cache=shared"
dbName = casibase
```

Custom configuration

Casibase supports custom configuration, you can modify the configuration file `conf/app.conf` to change the configuration.

```
casdoorEndpoint = <Your Casdoor endpoint>
clientId = <Your Casdoor application's client ID>
clientSecret = <Your Casdoor application's client secret>
casdoorOrganization = <Your Casdoor organization name>
casdoorApplication = <Your Casdoor application name>
```

Run

 CAUTION

Casibase requires Casdoor to provide access control and some back-end services, so you must make sure Casdoor is running properly before running Casibase.

How to install and run Casdoor:

- [Casdoor Installation](#)

Using Pre-built Binary

If you downloaded a pre-built binary, extract the archive and run Casibase directly. The binary already includes the compiled frontend.

For Linux/macOS:

```
# Example for Linux x86_64
tar -xzf casibase_Linux_x86_64.tar.gz
cd casibase
./casibase
```

For Windows (PowerShell):

```
# Example for Windows x86_64
Expand-Archive casibase_Windows_x86_64.zip -DestinationPath .
cd casibase
.\casibase.exe
```

The server will start on port 14000 by default.

Development mode

Backend

Casibase's Go backend runs at port 14000 by default. You can start the Go backend with the following command:

```
go run main.go
```

After the server is successfully running, we can start the frontend part.

Frontend

Casibase's frontend is a very classic [Create-React-App \(CRA\)](#) project. It runs at port `13001` by default. Use the following commands to run the frontend:

```
cd web
yarn install
yarn start
```

Building from Source

If you're building from source, follow these steps to create a production build.

Backend

Build Casibase Go backend code into executable and start it.

For Linux:

```
go build  
.casibase
```

For Windows:

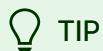
```
go build  
casibase.exe
```

Frontend

Build Casibase frontend code into static resources (.html, .js, .css files):

```
cd web  
yarn install  
yarn build
```

Nginx



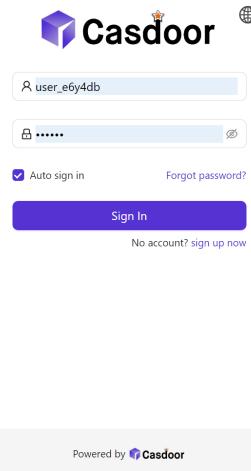
If you use nginx as a reverse proxy, you need to add the following configuration to the nginx configuration file:

```
location / {  
    *** your configuration ***  
    proxy_set_header Upgrade $http_upgrade;  
    proxy_set_header Connection "upgrade";  
}
```

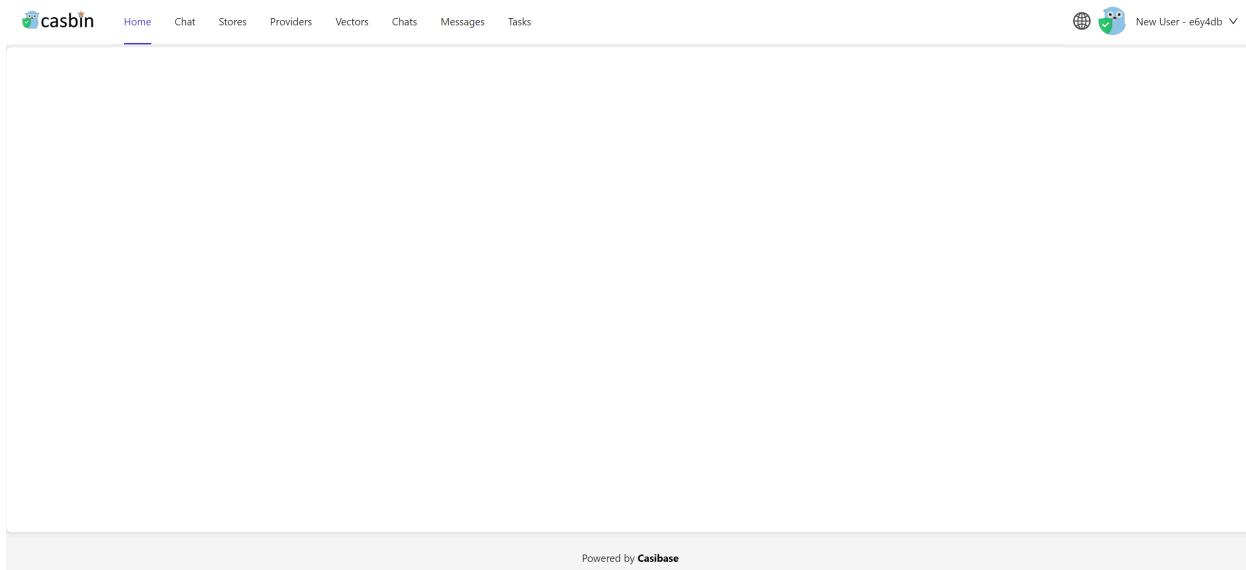
Because Casibase uses websocket to communicate with guacd.

Preview

Visit: `http://localhost:13001` in your browser. Login into Casibase dashboard with the user account you have just registered in Casdoor:



Then you will go to the home page of Casibase:





TIP
To use another port, please edit `conf/app.conf` and modify `httpport`, then restart the Go backend.

(Optional) Try with Docker

Requirements

Hardware

If you want to build the Docker image yourself, please ensure that your machine has at least 2GB of memory. Casibase's frontend is an NPM project of React. Building the frontend requires at least 2GB of memory. Having less than 2GB of memory may result in a frontend build failure.

If you only need to run the pre-built image, please ensure that your machine has at least 100MB of memory.

OS

All operating systems (Linux, Windows, and macOS) are supported.

Docker

You can use Docker (docker-engine version \geq 17.05) in Linux or Docker Desktop in Windows and macOS.

- [Docker](#)

Regardless of the operating system, users must ensure that they have **docker-engine version \geq 17.05**. This is because we utilize the multi-stage build feature in the docker-compose.yml, which is supported in versions 17.05 and above. For more information, see <https://docs.docker.com/develop/develop-images/multistage-build/>.

If you use docker-compose, please ensure you have docker-compose version >= 2.2. For Linux users, note that docker-compose needs to be installed separately from docker-engine.

Get the image

We have provided two DockerHub images:

Name	Description	Suggestion
casibase-all-in-one	Both Casibase and a MySQL database are included in the image	This image already includes a toy database and is only for testing purposes
casibase	Only Casibase is included in the image	This image can be connected to your own database and used in production

1. casbin/casibase-all-in-one: This image includes the casibase binary, a MySQL database, and all the necessary configurations. It is designed for new users who want to try Casibase quickly. With this image, you can start Casibase immediately with just one or two commands, without any complex configuration. However, please note that we **do not recommend** using this image in a production environment.

Option-1: Use the toy database

Run the container with port `14000` exposed to the host. The image will be automatically pulled if it doesn't exist on the local host.

```
docker run -p 14000:14000 casbin/casibase-all-in-one
```

Visit <http://localhost:14000> in your browser. Log into the Casibase dashboard with the default global admin account: `built-in/admin`

```
admin  
123
```

Option-2: Try with docker-compose

Create a `conf/app.conf` directory in the same directory level as the `docker-compose.yml` file. Then, copy `app.conf` from Casibase. For more details about `app.conf`, you can see [Via Ini file](#).

Below is a minimal but complete `docker-compose.yml` example that starts a MySQL database and the Casibase service. It configures Casibase to connect to the database using MySQL. Save this file as `docker-compose.yml` (next to a `conf` folder if you want to mount a custom `app.conf`).

```
services:  
  db:  
    image: mysql:8.0  
    restart: always  
    environment:  
      MYSQL_ROOT_PASSWORD: 123456  
      MYSQL_DATABASE: casibase  
    volumes:  
      - db_data:/var/lib/mysql  
    ports:  
      - "3306:3306" # optional: expose DB to host  
  
  casibase:
```

What does the above compose file do:

- The Casibase container connects to the database using the Compose service name `db` (i.e. `db:3306`). When both services run in the same Docker network (default for compose), using the service name as host is the simplest and most reliable approach.
- The `dataSourceName` above uses the MySQL root account for simplicity. For production use please create a dedicated DB user and a strong password.
- Mounting `./conf/app.conf` into `/conf/app.conf` is optional. If you prefer environment variables, you can remove the mount and rely on the `driverName` and `dataSourceName` variables.
- If both `app.conf` and environment variables are provided, the environment variables take precedence and will override the corresponding settings in `app.conf`.

 NOTE

Casdoor: By default Casibase uses the hosted Casdoor instance at `https://door.casdoor.com` for user authentication. If you need to manage users, applications, or customize the authentication flow, you must deploy your own Casdoor instance and update Casibase's `app.conf` (or the equivalent environment variables) to point to your Casdoor server. You can look at [Casdoor configuration](#) for more details.

RUNNING_IN_DOCKER: By default `RUNNING_IN_DOCKER` is enabled in docker image. When enabled, Casibase replaces `localhost` with the Docker bridge address (for example, `host.docker.internal` or the equivalent bridge hostname) so that the container can reach services running on the host.

Bring up the services:

```
docker-compose up -d
```

Check logs (follow):

```
docker-compose logs -f casibase
```

Visit <http://localhost:14000> in your browser. Log into the Casibase dashboard with the default global admin account: `built-in/admin`

admin

123

Stop and remove containers and volumes (data removed):

```
docker-compose down -v
```

Option-3: Try directly with the standard image



If it is not convenient to mount the configuration file to a container, using environment variables is also a possible solution.

example

```
docker run \
-e driverName=mysql \
-e dataSourceName='user:password@tcp(x.x.x.x:3306)/' \
```

Create `conf/app.conf`. You can copy it from `conf/app.conf` in Casibase. For more details about `app.conf`, you can see [Via Ini file](#).

Then run

```
docker run -p 14000:14000 -v /folder/of/app.conf:/conf casbin/casibase:latest
```

Anyway, just mount the app.conf to `/conf/app.conf` and start the container.

Visit <http://localhost:14000> in your browser. Log into the Casibase dashboard with the default global admin account: `built-in/admin`

```
admin  
123
```

(Optional) Try with K8s Helm

Introduction

This guide shows how to deploy Casibase on Kubernetes using Helm for easy and scalable management. Helm simplifies the deployment process and allows for easy configuration management.

Prerequisites

- A running Kubernetes cluster
- Helm v3 installed
- kubectl configured to connect to your cluster
- A MySQL/PostgreSQL database (recommended for production)
- A Casdoor instance for authentication

Configuration

Before installation, you must create an application configuration file. The Helm chart will not work with default values.

Pre step: Create app.conf File

Create an `app.conf` file with your specific settings:

You can view more details about the configuration options in the [Casibase Configuration Documentation](#).

Or check the latest configuration options in [app.conf example](#).

```
appname = casibase
httpport = 14000
runmode = prod
SessionOn = true
copyRequestBody = true

# Database Configuration - REQUIRED
driverName = mysql
dataSourceName = your-username:your-password@tcp(your-db-host:3306)-
dbName = your-database

# Casdoor Authentication - REQUIRED
casdoorEndpoint = https://door.casdoor.com
clientId = your-client-id
clientSecret = your-client-secret
casdoorOrganization = "your-organization"
casdoorApplication = "your-application"
redirectPath = /callback

# Optional Settings
redisEndpoint =
guacamoleEndpoint = 127.0.0.1:4822
isDemoMode = false
disablePreviewMode = false
logPostOnly = true
landingFolder =
cacheDir = "C:/casibase_cache"
appDir = ""
isLocalIpDb = false
audioStorageProvider = ""
providerDbName = ""
socks5Proxy = "127.0.0.1:10808"
publicDomain = ""
adminDomain = ""
enableExtraPages = false
shortcutPageItems = []
usageEndpoints = []
iframeUrl = ""
forceLanguage = ""
defaultLanguage = "en"
```

Optional: Using Secrets for Sensitive Data

For production environments, create a Kubernetes secret with your configuration:

```
# Create secret from app.conf file
kubectl create secret generic casibase-config --from-
file=app.conf=./app.conf
```

Installation Steps

Step 1: Prepare Configuration Files

Ensure you have created both:

- `app.conf` - Application configuration

Step 2: Install with Configuration File



INFO

Visit the [Casbin Helm Chart](#) to find the latest version.

Install Casibase by passing the app.conf file directly:

```
# Method 1: Pass app.conf content as appConfig parameter
helm install casibase oci://registry-1.docker.io/casbin/casibase-helm-
chart \
--version v1.549.0 \
--set-file appConfig=./app.conf
```

Alternative Installation with Secret

If using secrets for sensitive data:

```
# Create secret first
kubectl create secret generic casibase-config --from-
file=app.conf=./app.conf

# Install with secret reference (no additional values file needed)
helm install casibase oci://registry-1.docker.io/casbin/casibase-helm-
chart \
  --version v1.549.0 \
  --set appConfig="" \
  --set appConfigFromSecret=casibase-config
```

Step 3: Verify Installation

Check the deployment status:

```
kubectl get pods
kubectl get services
kubectl logs -l app.kubernetes.io/name=casibase
```

Step 4: Access Casibase

Once installed, Casibase will be accessible through the Kubernetes service on port 14000. If you enabled ingress, it will be available at your configured domain.

Configuration Options Reference

The following table shows the main configuration parameters available in the Helm chart:

Parameter	Description	Default Value
<code>replicaCount</code>	Number of Casibase replicas to run	<code>1</code>
<code>image.repository</code>	Docker image repository	<code>casbin</code>
<code>image.name</code>	Docker image name	<code>casibase</code>
<code>image.pullPolicy</code>	Image pull policy	<code>IfNotPresent</code>
<code>image.tag</code>	Image tag (defaults to chart appVersion)	<code>""</code>
<code>appConfig</code>	Application configuration (app.conf content)	See values.yaml
<code>appConfigFromSecret</code>	Mount app.conf from secret instead	<code>""</code>

Parameter	Description	Default Value
<code>service.type</code>	Kubernetes service type	<code>ClusterIP</code>
<code>service.port</code>	Service port	<code>14000</code>
<code>ingress.enabled</code>	Enable ingress	<code>false</code>
<code>ingress.hosts</code>	Ingress hosts configuration	<code>[]</code>
<code>resources</code>	CPU/ Memory resource requests and limits	<code>{}</code>
<code>autoscaling.enabled</code>	Enable horizontal pod autoscaler	<code>false</code>
<code>autoscaling.minReplicas</code>	Minimum number of replicas	<code>1</code>
<code>autoscaling.maxReplicas</code>	Maximum number of replicas	<code>100</code>

Parameter	Description	Default Value
<code>autoscaling.targetCPUUtilizationPercentage</code>	CPU utilization threshold	80
<code>nodeSelector</code>	Node labels for pod assignment	{}
<code>tolerations</code>	Toleration labels for pod assignment	[]
<code>affinity</code>	Affinity settings for pod assignment	{}

Advanced Configuration Options

For production deployments, consider these additional options:

```
# Autoscaling
autoscaling:
  enabled: true
  minReplicas: 2
  maxReplicas: 10
  targetCPUUtilizationPercentage: 70
```

Managing the Deployment

Upgrading Casibase

To upgrade your Casibase deployment to a new version:

```
helm upgrade casibase oci://registry-1.docker.io/casbin/casibase-helm-  
chart --version <new-version>
```

To upgrade with custom values:

```
helm upgrade casibase oci://registry-1.docker.io/casbin/casibase-helm-  
chart --version <new-version> \  
-f custom-values.yaml
```

Checking Deployment Status

Monitor your deployment:

```
# Check pod status  
kubectl get pods -l app.kubernetes.io/name=casibase  
  
# Check service status  
kubectl get svc -l app.kubernetes.io/name=casibase  
  
# View logs  
kubectl logs -l app.kubernetes.io/name=casibase  
  
# Describe deployment  
helm status casibase
```

Uninstalling Casibase

To completely remove Casibase from your cluster:

```
helm uninstall casibase
```

Troubleshooting

Common Issues

1. Pod not starting: Check logs with `kubectl logs <pod-name>`
2. Service not accessible: Verify service configuration and ingress setup
3. Database connection issues: Ensure database credentials and connectivity are correct
4. Configuration errors: Validate your `appConfig` syntax
5. Casdoor authentication failures: Verify Casdoor endpoint and credentials
6. Domain/URL issues: Check domain configuration and DNS resolution

Configuration-Related Issues

Problem: Casdoor authentication not working

- Verify `casdoorEndpoint` is accessible from the cluster
- Check `clientId` and `clientSecret` are correct
- Ensure `redirectPath` matches your Casdoor application configuration

Problem: Configuration syntax errors

```
# Validate YAML syntax before deployment
```

Getting Help

- Check pod events: `kubectl describe pod <pod-name>`
- View Helm release info: `helm status casibase`
- Review configuration: `helm get values casibase`

Conclusion

Using Helm to deploy Casibase on Kubernetes provides a robust, scalable solution for managing your knowledge base platform. The chart offers flexible configuration options to suit various deployment scenarios, from development environments to production clusters.

Key benefits of this approach:

- Easy deployment and updates through Helm commands
- Flexible configuration through values files
- Kubernetes-native scaling and management
- Production-ready with proper resource management and health checks

For more advanced configurations and troubleshooting, refer to the [Kubernetes documentation](#) and [Helm documentation](#).

Casibase Public API

Casibase frontend web UI is a [SPA \(Single-Page Application\)](#) developed in React. The React frontend consumes the Casibase RESTful API exposed by the Go backend code. This RESTful API is referred to as the [Casibase Public API](#). In other words, with HTTP calls, you can do everything just like how the Casibase web UI itself does. There's no other limitation. The API can be utilized by the following:

- Casibase's frontend
- Casibase client SDKs (e.g., casibase-java-sdk)
- Any other customized code from the application side

The full reference for the [Casibase Public API](#) can be found on Swagger: <https://ai-admin.casibase.com/swagger>. These Swagger docs are automatically generated using Beego's Bee tool. If you want to generate the Swagger docs by yourself, see: [How to generate the swagger file](#)

How to authenticate with [Casibase Public API](#)

Casibase Public API supports two application-level authentication methods: [Bearer Token](#) and [Basic Auth](#). The [Bearer Token](#) method is recommended as it is more secure.

SDK Authentication Example (Java)

To illustrate how authentication is handled in practice, here is an example from the Casibase Java SDK. The following code shows how the SDK constructs the

credential for API requests. This process authenticates the SDK with application-level permissions, effectively acting as an admin.

```
// ...

protected void Service(Config config, AuthTypeEnum authType)
throws Exception {
    this.config = config;
    switch (authType){
        case BASIC:
            this.credential =
Credentials.basic(config.clientId, config.clientSecret);
            break;
        case BEARER:
            String token = config.clientId + ":" +
config.clientSecret;
            this.credential = "Bearer " +
DigestUtils.md5Hex(token);
            break;
        default:
            throw new Exception("Invalid auth type");
    }
}
```

The example above demonstrates how to prepare the credential for both authentication types:

- **BASIC:** It uses a helper (`okhttp3.Credentials`) to perform the standard Base64 encoding for Basic Authentication.
- **BEARER:** It constructs the token by taking the MD5 hash of `clientId:clientSecret` (using `org.apache.commons.codec.digest.DigestUtils`) and prepending the result with "Bearer ".

1. By **Bearer Token** (Recommended)

This method is more secure because it uses a static access token, which is a hashed value of your `clientId` and `clientSecret`.

How to get the access token?

The access token is calculated using the following formula: `md5(clientId + ":" + clientSecret)`

How to authenticate?

The access token must be provided in the `Authorization` header as a Bearer Token.

```
Authorization: Bearer <The access token>
```

2. By **Basic Auth**

This method uses the `clientId` and `clientSecret` directly for authentication. It is considered less secure because the `clientSecret` might be exposed. It is supported for convenience and compatibility purposes.

How to authenticate?

HTTP Basic Authentication: This is the standard way.

```
Authorization: Basic <The Base64 encoding of  
"clientId:clientSecret">
```

If you are not familiar with Base64 encoding, you can use a library for this, as

`HTTP Basic Authentication` is a widely supported standard.

Where to find the Client ID and Secret?

Both authentication methods require a `clientId` and `clientSecret`. You can find these values for your application in the Casibase configuration file: `conf/app.conf`.

System Info

Casibase provides real-time system monitoring to help administrators track resource usage and performance. The System Info page displays key metrics about your Casibase instance.

Accessing System Info

Global administrators can access the system monitoring dashboard through the **SysInfo** menu item in the navigation bar. This page automatically refreshes metrics every 3 seconds.

Available Metrics

The system monitoring dashboard displays key metrics in an organized layout that spans the full page width for better visibility.

CPU Usage: Displays the current CPU utilization percentage for each processor core. This helps identify if the system is experiencing high computational load.

Memory Usage: Shows the memory consumption of the Casibase process relative to total system memory. Monitor this to ensure sufficient memory is available for smooth operation.

Disk Usage: Tracks the storage space used by Casibase's data directory. This metric helps you monitor growth and plan for storage capacity.

Network Usage: Displays cumulative network I/O for the Casibase process, including bytes sent, received, and total throughput. This helps track data transfer patterns and identify potential network bottlenecks.

API Performance Metrics

For deployments with Prometheus integration, the System Info page also shows:

- **API Latency:** Response time distribution across different API endpoints
- **API Throughput:** Request count per endpoint to identify high-traffic routes

These metrics help optimize performance and troubleshoot slow requests.

Container Cloud

Casibase is an open-source Container Cloud Platform built on the foundations of Docker and Kubernetes. It is designed for individuals and organizations to easily build, manage, and operate their own private cloud environments with a focus on simplicity and application-centric management.

The Challenge: Complexity in the Cloud-Native Era

In the world of modern software, Kubernetes has become the standard for running applications. However, its power comes with significant complexity. Deploying even a seemingly simple application, like a WordPress blog, requires orchestrating a multitude of distinct Kubernetes resources:

- **Deployments** to manage the application pods (the WordPress server itself).
- **Services** to expose the application to the network.
- **PersistentVolumeClaims** to request storage for the database and uploads.
- **StatefulSets** to manage the database pods (like MySQL).
- **ConfigMaps** and **Secrets** to handle configuration and sensitive data.

Managing these individual components manually is often called a "resource-centric" approach. This approach presents several challenges:

1. **High Learning Curve:** Users must have a deep understanding of various Kubernetes resources and how they interact.
2. **Operational Burden:** Manually creating, updating, and deleting these resources is tedious and prone to human error.
3. **Lack of Atomicity:** There is no way to treat the entire "WordPress application"

as a single, atomic unit. You cannot simply "install" or "uninstall" it with one action.

4. **Inconsistency:** Ensuring that the application is deployed identically across development, testing, and production environments is difficult.

The Casibase Approach: From Managing Resources to Managing Applications

Casibase fundamentally simplifies this process by shifting the focus from managing individual resources to managing the application as a whole. We believe you should be able to manage your applications without getting lost in the weeds of Kubernetes YAML configurations.

To achieve this, Casibase introduces a powerful, application-centric model built on two core concepts:

1. Templates: The Reusable Blueprint

A Template in Casibase is a complete, reusable blueprint for an application. It encapsulates all the necessary Kubernetes resource manifests required to deploy a service. Think of it as a "package" for a cloud application.

- **What it contains:** A template holds the base YAML configurations for all the components of an application (Deployments, Services, etc.), structured for use with Kustomize.
- **The Goal:** To make the underlying complexity transparent. Once a template for WordPress is created, anyone can use it to deploy WordPress without needing to know the details of its Kubernetes architecture.

2. Applications: The Live Instance

An Application is a live, running instance created from a Template. It represents a specific deployment of that template in your cluster.

- **Customization:** When you create an Application, you select a Template and then provide your own specific configurations, such as the number of replicas, a custom domain name, or a specific database password.
- **How it works:** These custom configurations are treated as "patches" or "overlays." Casibase uses Kustomize to intelligently merge your custom parameters with the base manifests from the template, generating the final, complete configuration.
- **Lifecycle Management:** The Application becomes the single unit you interact with. You can deploy, monitor, update, and delete the entire application with single clicks in the UI.

By adopting this model, Casibase transforms the complex task of cloud-native deployment into a streamlined, intuitive workflow. Instead of wrestling with `kubectl` and YAML files, you can manage the entire lifecycle of your applications through a clean web interface: select a template, fill in a few parameters, and click deploy.

Beginner Guide

Add a Storage Provider

Discover how to integrate a storage provider with Casibase.

Add an AI Model Provider

Learn how to add a model provider to enhance Casibase functionality.

Add an Embedding Provider

Explore how to integrate an embedding provider with Casibase.

Add a Text-to-Speech Model Provider

Learn how to add a text-to-speech provider to enhance Casibase functionality.

Add a Speech-to-Text Provider

Learn how to add a speech-to-text provider to enhance Casibase functionality.

Add a Store

Learn how to add a store to your Casibase knowledge base system.

Chats with AI

Implement AI chat functionality in your Casibase knowledge base system.

Add a Storage Provider

This document is a step-by-step tutorial designed for beginners. It will guide you through the process of integrating a storage provider with Casibase, our powerful knowledge base system.

Introduction

Adding a storage provider to Casibase enables you to efficiently manage and store data, making it an essential component for your knowledge base system.

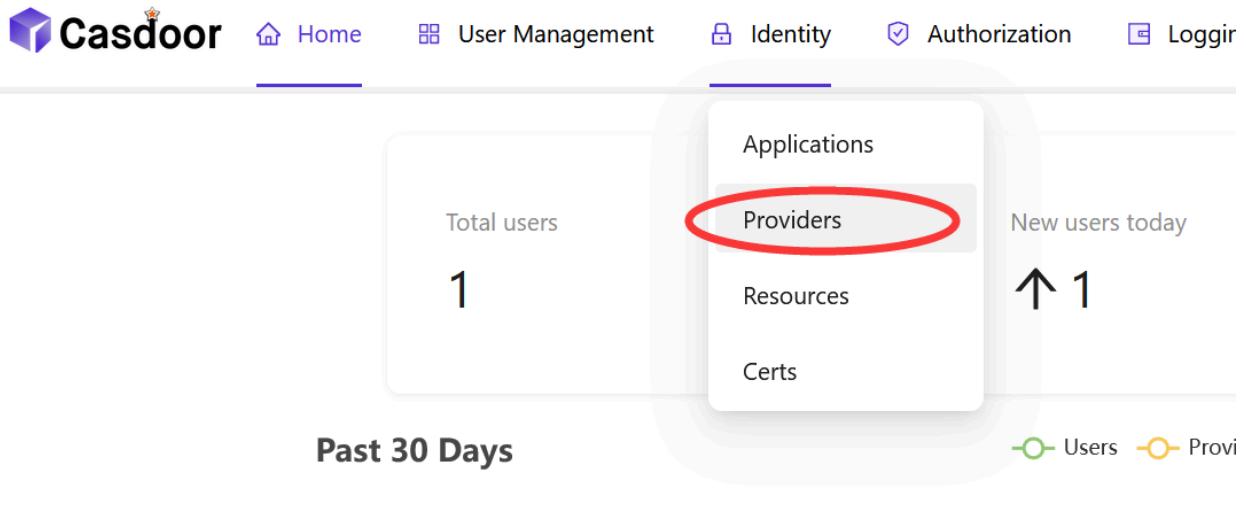
If you're new to integrating storage providers, don't worry. We've broken down the process into simple steps that anyone can follow.

Step 1: Deploy Casdoor and Casibase

If you haven't done, please refer to the [Deploy Casdoor and Casibase](#) tutorial.

Step 2: Add a New Storage Provider

Storage providers are used to store data. They can be added in Casdoor by clicking the `Identity - Providers` button on the home page.



Step 2.1: Add a storage provider

Click the **Add** button to add a storage provider.

The screenshot shows the "Providers" list page in Casdoor. At the top, there is a navigation bar with Home, User Management, Identity, and Authorization. Below the navigation, there is a table with the following data:

Name	Organization	Created time	Actions
provider_captcha_default	admin (Shared)	2023-09-10 19:31:50	Edit Delete

At the top left of the table, there is a blue "Add" button, which is highlighted with a red circle. The table has search and sort icons for each column.

Step 2.2: Fill in the storage provider information

Fill in the storage provider information and click the **Save & Exit** button.

New Provider

Save

Save & Exit

Cancel

Name ②:

provider_storage_1

Display name ②:

Provider_storage_1

Organization ②:

admin (Shared)

Category ②:

Storage

Type ②:

aws AWS S3

Client ID ②:

Alibaba Cloud OSS

aws AWS S3

Client secret ②:

Azure Blob

Endpoint ②:

Google Cloud Storage

Endpoint (Intranet) ②:

Local File System

:

MinIO

Bucket ②:

Qiniu Cloud Kodo

 TIP

Casdoor supports many storage providers, including:

- [AWS S3](#)
- [Azure Blob](#)
- [Google Cloud Storage](#)
- [MinIO](#)

- [Qiniu Cloud Kodo](#)
- [Alibaba Cloud OSS ...](#)

Example

Add an Aliyun OSS storage provider

CAUTION

- Client ID: The AccessKey ID of your Aliyun OSS account.
- Client Secret: The AccessKey Secret of your Aliyun OSS account.

 is the placeholder for your Aliyun OSS account information.

Category [?](#) : Storage

Type [?](#) : Alibaba Cloud OSS

Client ID [?](#) : LTA***NLf

Client secret [?](#) : Vo6***pi8

Endpoint [?](#) : oss-cn-beijing.aliyuncs.com

Endpoint (Intranet) [?](#) :

Bucket [?](#) : xx-bucket-0

Path prefix [?](#) :

Domain [?](#) : https://xx-bucket-0.oss-cn-beijing.aliyuncs.com

Provider URL [?](#) : https://github.com/organizations/xxx/settings/applications/1234567

[Save](#) [Save & Exit](#) [Cancel](#)

Step 2.3: View the storage provider

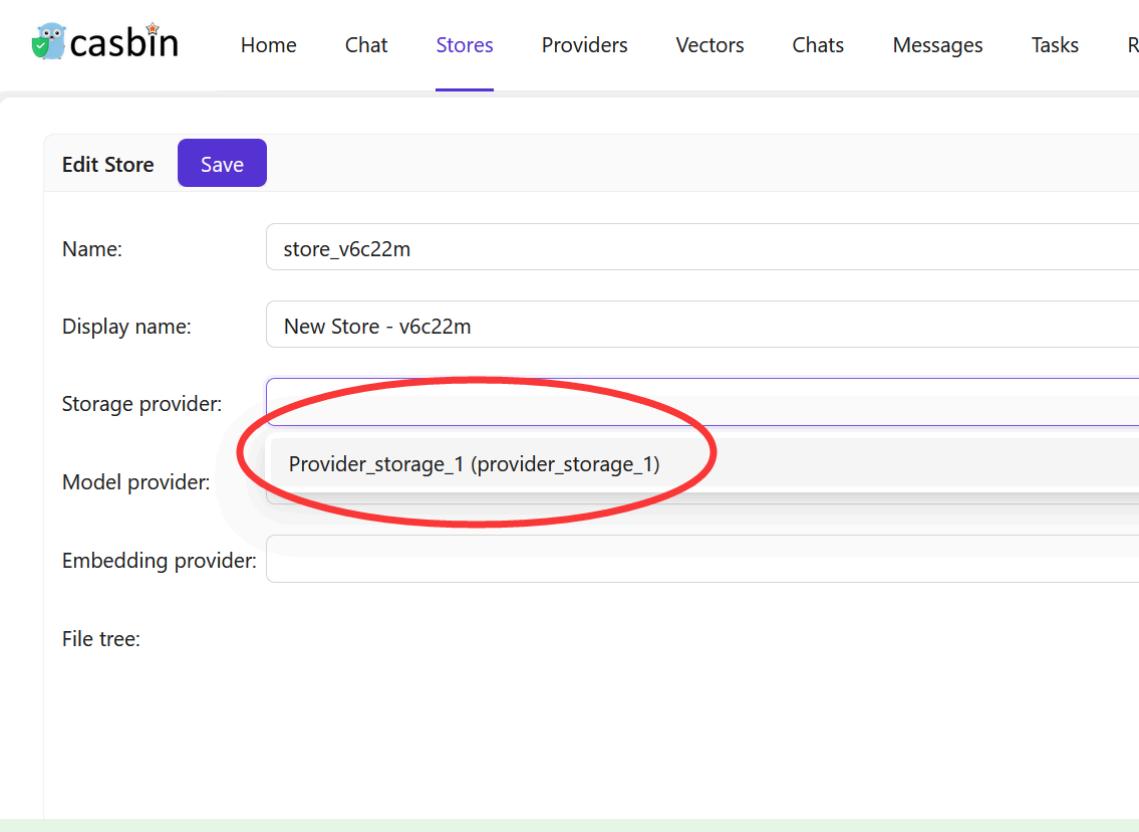
After adding the storage provider, you can view the storage provider information.

Name	Organization	Created time	Display name	Category	Type	Client ID	Provider URL	Action
provider_storage_1	admin (Shared)	2023-09-10 21:23:02	Provider_storage_1	Storage	Alibaba Cloud OSS	...	https://github.com/organizations/xxx...	Edit Delete



Storage providers come from Casdoor. You can add a storage provider in Casdoor and then add it to Casibase.

Refer to [Step 2: Add a New Storage Provider](#) for more information.



The screenshot shows the Casbin web interface with the 'Stores' tab selected. The 'Edit Store' form is displayed, with the 'Save' button highlighted in purple. The form fields are as follows:

- Name: store_v6c22m
- Display name: New Store - v6c22m
- Storage provider: (highlighted with a red circle, showing the option 'Provider_storage_1 (provider_storage_1)')
- Model provider: (empty)
- Embedding provider: (empty)
- File tree: (empty)

Store Example

[Home](#)[Chat](#)[Stores](#)[Providers](#)[Vectors](#)[Chats](#)[M](#)[Edit Store](#)[Save](#)

Name:

my_store

Display name:

My_Store

Storage provider:

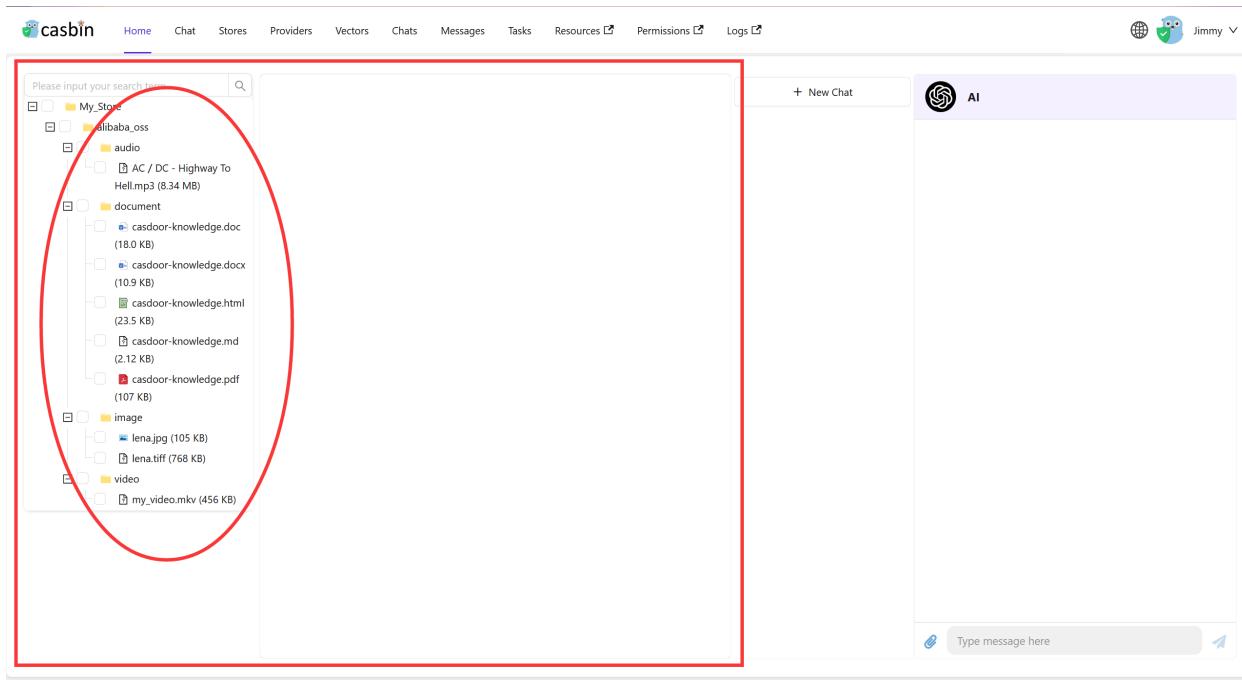
Provider_storage_1 (provider_storage_1)

Model provider:

Embedding provider:

File tree:

Save the configuration and navigate to the [Stores](#) page to see the file-tree of the storage provider.



The screenshot shows the Casibin web interface. On the left, there is a file tree with a red box highlighting the 'My_Store' folder. The tree structure is as follows:

- My_Store
 - alibaba_oss
 - audio
 - AC / DC - Highway To Hell.mp3 (8.34 MB)
 - document
 - casdoor-knowledge.doc (18.0 KB)
 - casdoor-knowledge.docx (10.9 KB)
 - casdoor-knowledge.html (23.5 KB)
 - casdoor-knowledge.md (2.12 KB)
 - casdoor-knowledge.pdf (107 KB)
 - image
 - lena.jpg (105 KB)
 - lena.tif (768 KB)
 - video
 - my_video.mkv (456 KB)

Now you can manage your data in Casibase.

In the next chapter, we'll learn how to add an [AI model provider](#) to Casibase.

Add an AI Model Provider

This document is a step-by-step tutorial designed for beginners. It will guide you through the process of integrating a model provider with Casibase, our powerful knowledge base system.

Introduction

Adding a model provider to Casibase enables you to enhance its functionality by incorporating machine learning models and AI capabilities. Model providers allow you to analyze and process data within your knowledge base system, making it more intelligent and efficient.

If you're new to integrating model providers, don't worry. We've broken down the process into simple steps that anyone can follow.

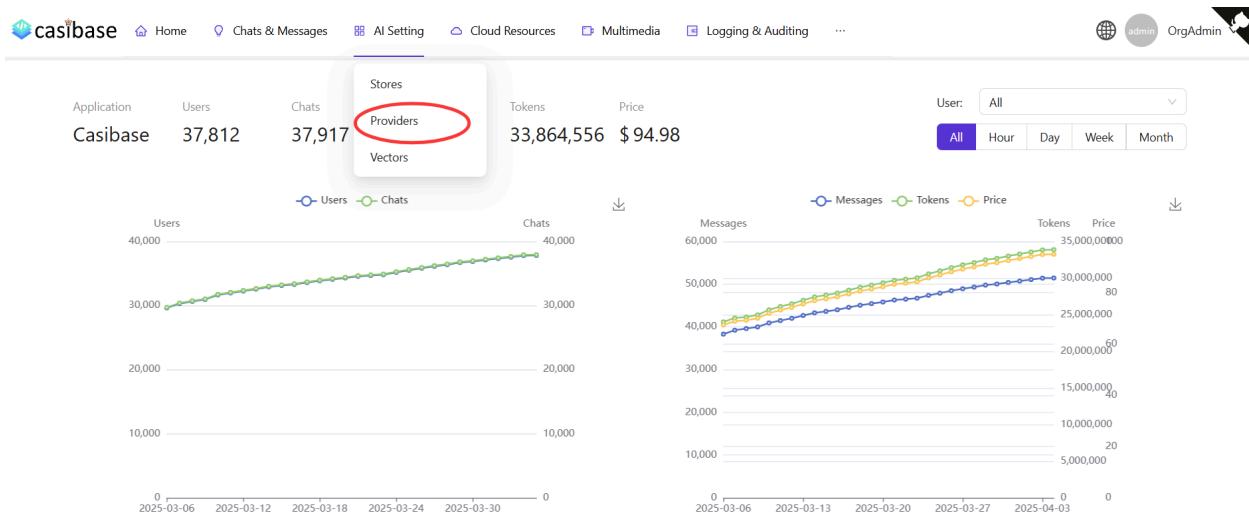
Step 1: Deploy Casdoor and Casibase

Before you can add an AI model provider, make sure you have Casdoor and Casibase deployed. If you haven't done, please refer to the [Deploy Casdoor and Casibase](#) tutorial.

Step 2: Add a New Model Provider

Model providers are used to integrate LLM into Casibase. You can add them by following these steps:

Click the `Providers` button on the home page.



Step 2.1: Add a Model Provider

Click the **Add** button to add a model provider.

The table lists existing model providers and shows the **Add** button highlighted with a red circle.

Name	Display name	Category	Type	Sub type	API key	Secret key	Region
provider tts_alibabacloud_cosyvoice	Provider TTS AlibabaCloud Cosyvoice	Text-to-Speech	Alibaba Cloud	cosyvoice-v1		***	
provider_blockchain_chainmaker	Provider Blockchain ChainMaker	Blockchain	Tencent ChainMaker (Demo Network)	text-davinci-003	AKIDObZHjqUxnNltkKdIQyAWcS1JK4XidG	***	ap-beijing
provider_model_alibabacloud_deepseek_r1	Provider Model AlibabaCloud DeepSeek-R1	Model	Alibaba Cloud	deepseek-r1		***	
provider_cloud_alibabacloud	Provider Cloud AlibabaCloud	Public Cloud	Aliyun	text-davinci-003	LTAI5thkUsopAioN6xi2LMg	***	cn-beijing
dall-e-3	dall-e-3	Model	OpenAI	dall-e-3		***	
provider_model_azure_gpt4_1	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_model_azure_gpt4_2	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_model_azure_gpt4_3	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_embedding_openai_v3	Provider Embedding OpenAI V3	Embedding	OpenAI	text-embedding-ada-002		***	
provider_model_openai_gpt4_vision	Provider Model OpenAI GPT4 Vision	Model	OpenAI	gpt-4-vision-preview		***	

Step 2.2: Fill in Model Provider Details

Fill in the model provider details and click the **Save & Exit** button.

[Home](#)[Chat](#)[Stores](#)[Providers](#)[Vectors](#)[Chats](#)[Me](#)[Edit Provider](#)[Save](#)

Name:

provider_openai_model

Display name:

OpenAI model

Category:

Model

Type:

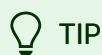
OpenAI

Sub type:

text-davinci-003

Secret key:

Provider URL:

<https://platform.openai.com/account/api-keys>[Save](#)

Casibase supports many model providers, including:

- [Hugging Face](#)
 - meta-llama/Llama-2-7b

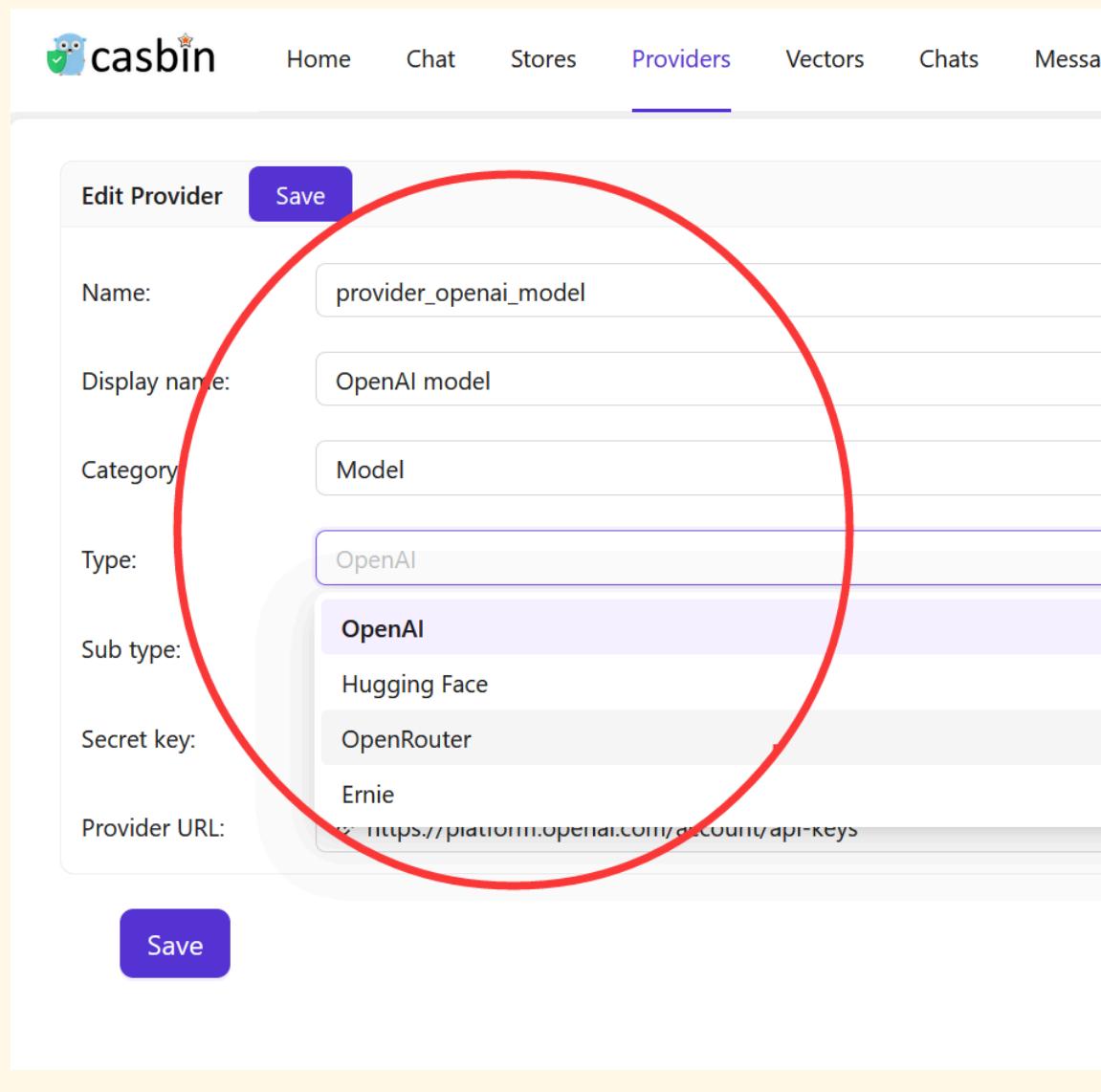
- THUDM/chatglm2-6b
- baichuan-inc/Baichuan2-13B-chat
- gpt2
-
- OpenRouter
 - anthropic/claude-2
 - palm-2-chat-bison
 - palm-2-codechat-bison
 - openai/gpt-4
 -
- OpenAI
 - text-davinci-003
 - gpt-3.5-turbo
 - gpt-4
 -

⚠ CAUTION

- Category: The first-level category of the model provider. For example, **Model** and **Embedding**.
- Type: The second-level category of the model provider. For example, **OpenAI** and **Hugging Face**.
- API key: Your API key from the provider.

Example

Add an OpenAI model provider



casbin

Home Chat Stores Providers Vectors Chats Message

Edit Provider Save

Name: provider_openai_model

Display name: OpenAI model

Category: Model

Type: OpenAI

Sub type: OpenAI
Hugging Face
OpenRouter
Ernie

Provider URL: <https://platform.openai.com/account/api-keys>

Save

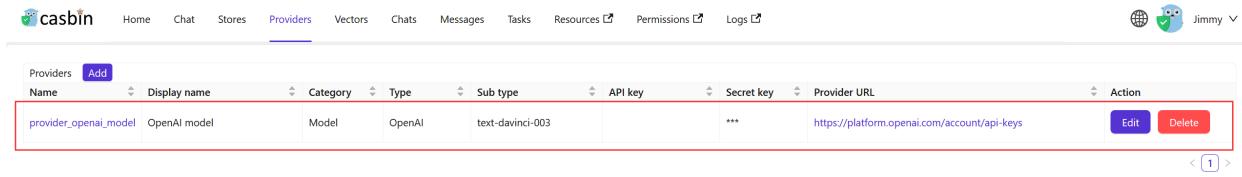
⚠ CAUTION

Some models don't support streaming-output. Known models that support streaming-output include:

- gpt-3.5-turbo-0613

After adding a model provider, you can use it to analyze and process data in Casibase using chatbots, question answering, and other AI capabilities.

Return to the model provider list page:



The screenshot shows the Casibase interface with the 'Providers' tab selected. A single provider entry is listed in the table:

Name	Display name	Category	Type	Sub type	API key	Secret key	Provider URL	Action
provider_openai_model	OpenAI model	Model	OpenAI	text-davinci-003	***		https://platform.openai.com/account/api-keys	<button>Edit</button> <button>Delete</button>

A red box highlights the entire table row for the provider entry.

Now that you've added a model provider, you can use it to analyze and process data in Casibase using chatbots, question answering, and other AI capabilities.

In the next chapter, we'll learn how to add an embedding provider to Casibase.

Add an Embedding Provider

This document is a step-by-step tutorial designed for beginners. It will guide you through the process of integrating an embedding provider with Casibase, our powerful knowledge base system.

Introduction

Embedding is a technique used to represent words and documents as vectors. Embedding providers allow you to analyze and process data within your knowledge base system, making it more intelligent and efficient.

Refer to the [Core Concepts](#) section of our previous documentation for more information about embedding.

In Casibase, you can add an embedding provider by following these steps:

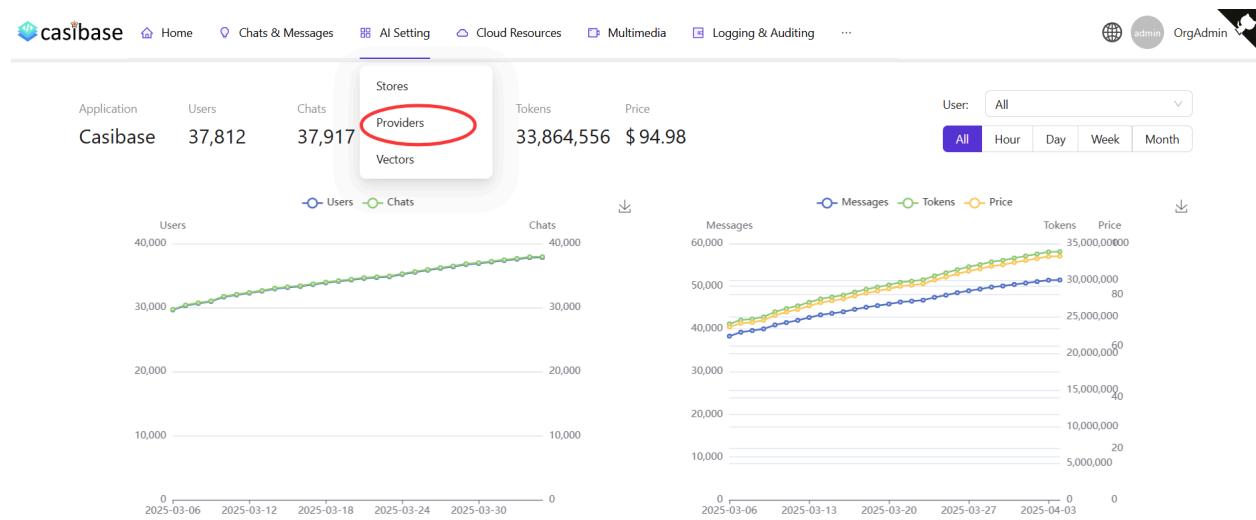
Step 1: Deploy Casdoor and Casibase

Before you can add an embedding model provider, make sure you have Casdoor and Casibase deployed. If you haven't done, please refer to the [Deploy Casdoor and Casibase](#) tutorial.

Step 2: Add a New Embedding Provider

Embedding providers are used to integrate embedding into Casibase. You can add them by following these steps:

Click the **Providers** button on the home page.



Step 2.1: Add an Embedding Provider

Click the **Add** button to add an embedding provider.

The screenshot shows the 'Add Storage Provider' table. The 'Providers' button is highlighted with a red circle. The table has the following columns:

Name	Display name	Category	Type	Sub type	API key	Secret key	Region
provider tts_alibabacloud_cosyvoice	Provider TTS AlibabaCloud Cosyvoice	Text-to-Speech	Alibaba Cloud	cosyvoice-v1		***	
provider_blockchain_chainmaker	Provider Blockchain ChainMaker	Blockchain	Tencent ChainMaker (Demo Network)	text-davinci-003	AKIDObZHjvqUrxnNltkKdfQvAWcs1JK4XidG	***	ap-beijing
provider_model_alibabacloud_deepseek_r1	Provider Model AlibabaCloud DeepSeek-R1	Model	Alibaba Cloud	deepseek-r1		***	
provider_cloud_alibabacloud	Provider Cloud AlibabaCloud	Public Cloud	Aliyun	text-davinci-003	LTAI5tHkUsopAioN6xi2LMg	***	cn-beijing
dall-e-3	dall-e-3	Model	OpenAI	dall-e-3		***	
provider_model_azure_gpt4_1	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_model_azure_gpt4_2	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_model_azure_gpt4_3	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_embedding_openai_v3	Provider Embedding OpenAI V3	Embedding	OpenAI	text-embedding-ada-002		***	
provider_model_openai_gpt4_vision	Provider Model OpenAI GPT4 Vision	Model	OpenAI	gpt-4-vision-preview		***	

Step 2.2: Fill in Embedding Provider Details

Fill in the embedding provider details and click the **Save & Exit** button.

[Home](#)[Chat](#)[Stores](#)[Providers](#)[Vectors](#)[CI](#)[Edit Provider](#)[Save](#)

Name:

embedding_openai_adasimilarity

Display name:

Embedding_OpenAI_AdaSimilarity

Category:

Embedding

Type:

OpenAI

Sub type:

AdaSimilarity

Secret key:

Provider URL:

<https://platform.openai.com/account/api-keys>[Save](#)

TIP

Same as the [Model Provider](#) section, Casibase supports many embedding providers, including:

- OpenAI
 - AdaSimilarity
 - DavinciSimilarity
 - AdaEmbedding2
 -
- Hugging Face
 - sentence-transformers/paraphrase-MiniLM-L6-v2
 -

Return providers list page:

Name	Display name	Category	Type	Sub type	API key	Secret key	Provider URL	Action
embedding_openai_adasimilarity	Embedding_OpenAI_AdaSimilarity	Embedding	OpenAI	1		***	https://platform.openai.com/account/api-keys	Edit Delete
model_openai_text_davinci_003	Model OpenAI text-davinci-003	Model	OpenAI	text-davinci-003		***	https://platform.openai.com/account/api-keys	Edit Delete

Now, you can use the embedding provider to convert text to vectors.

After adding an embedding provider, you can use it to retrieve similar documents in Casibase. For more information, please refer to the [Core Concepts](#) section of our previous documentation.

In the next chapter, we will learn how to integrate storage providers, model providers, and embedding providers with Casibase.

Add a Text-to-Speech Model Provider

This document is a step-by-step tutorial designed for beginners. It will guide you through the process of integrating a text-to-speech provider with Casibase, our powerful knowledge base system.

Introduction

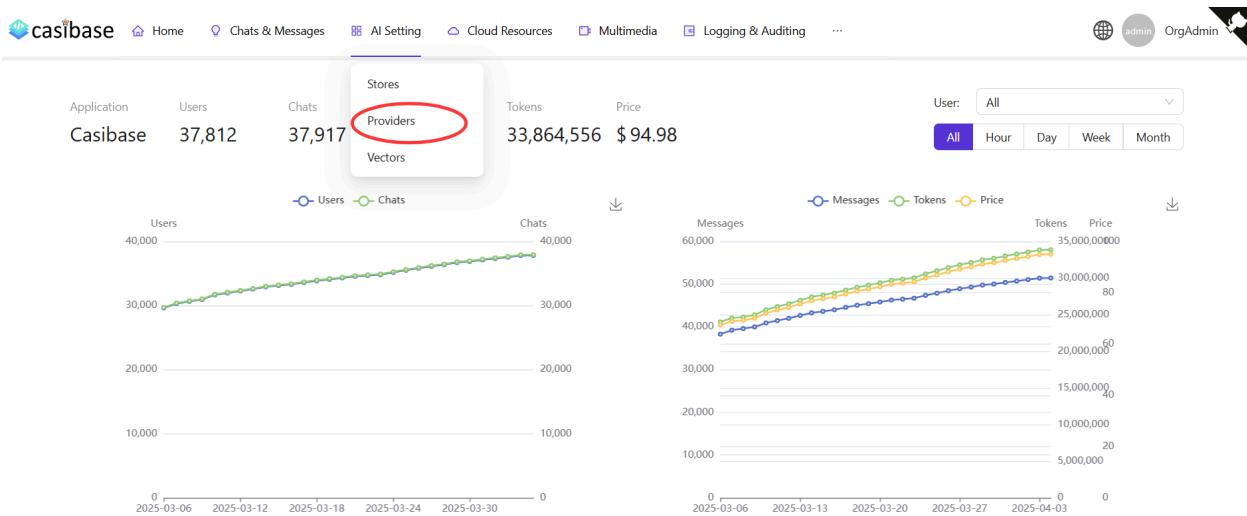
Text-to-Speech (TTS) is a technology that converts text into spoken voice output. TTS providers allow your Casibase applications to communicate with users through synthesized speech, enhancing the user experience and accessibility of your knowledge base system.

In Casibase, integrating a TTS provider enables your AI applications to verbally respond to queries, creating more interactive and engaging user experiences.

Add a New Text-to-Speech Provider

Text-to-Speech providers are used to integrate voice synthesis capabilities into Casibase. You can add them by following these steps:

Click the **Providers** button on the page.



Add a Text-to-Speech Provider

Click the **Add** button to add a Text-to-Speech provider.

The table lists existing providers and has an **Add** button highlighted with a red circle.

Name	Display name	Category	Type	Sub type	API key	Secret key	Region
provider_tts_alibabacloud_cosyvoice	Provider TTS AlibabaCloud Cosyvoice	Text-to-Speech	Alibaba Cloud	cosyvoice-v1		***	
provider_blockchain_chainmaker	Provider Blockchain ChainMaker	Blockchain	Tencent ChainMaker (Demo Network)	text-davinci-003	AKIDObZHjqUxnNltkKdIQyAWcS1JK4XidG	***	ap-beijing
provider_model_alibabacloud_deepseek_r1	Provider Model AlibabaCloud DeepSeek-R1	Model	Alibaba Cloud	deepseek-r1		***	
provider_cloud_alibabacloud	Provider Cloud AlibabaCloud	Public Cloud	Aliyun	text-davinci-003	LTAI5thkUsopAioN6xi2LMg	***	cn-beijing
dall-e-3	dall-e-3	Model	OpenAI	dall-e-3		***	
provider_model_azure_gpt4_1	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_model_azure_gpt4_2	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_model_azure_gpt4_3	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_embedding_openai_v3	Provider Embedding OpenAI V3	Embedding	OpenAI	text-embedding-ada-002		***	
provider_model_openai_gpt4_vision	Provider Model OpenAI GPT4 Vision	Model	OpenAI	gpt-4-vision-preview		***	

Fill in Text-to-Speech Provider Details

Fill in the embedding provider details and click the **Save & Exit** button.

casibase

Home Chats & Messages AI Setting Cloud Resources Multimedia Logging & Auditing ...

admin OrgAdmin

Edit Provider

Save Save & Exit

Name: provider_tts_alibabacloud_cosyvoice

Display name: Provider TTS AlibabaCloud Cosyvoice

Category: Text-to-Speech

Type: Alibaba Cloud

Sub type: cosyvoice-v1

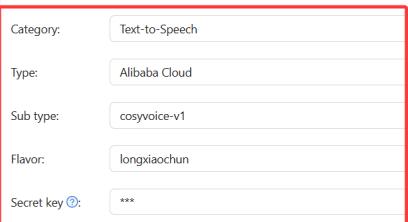
Flavor: longxiaochun

Secret key: ***

Provider URL: <https://bailian.console.aliyun.com/?apiKey=1#/api-key>

State: Active

Save Save & Exit



 TIP

Casibase currently supports the following Text-to-Speech provider:

- Alibaba Cloud
 - cosyvoice-v1 (with multiple voice options)

Testing Your Text-to-Speech Provider

You can test your TTS provider by clicking the `Read it out` button. This will allow you to enter text and hear the synthesized speech output.

The screenshot shows the 'Edit Provider' page in the casibase AI Setting. The provider is named 'provider_r7fdnn' with a display name 'New Provider - r7fdnn'. It is categorized as 'Text-to-Speech' and is of type 'Alibaba Cloud' (cosyvoice-v1). The flavor is '龙小淳, 女, 中英双语'. The provider test message is 'Hello, I'm casibase AI.' with a 'Read it out' button. The provider URL is 'https://platform.openai.com/account/api-keys'. The state is 'Active'. There are 'Save' and 'Save & Exit' buttons at the bottom.

Provider test: Hello, I'm casibase AI. Read it out

This testing feature allows you to verify your TTS configuration before implementing it in your applications, ensuring the voice quality and settings meet your requirements.

Voice Options for Alibaba Cloud

When using Alibaba Cloud's `cosyvoice-v1`, you can choose from various voice options:

- longwan
- longcheng
-

Using Text-to-Speech in Stores

After adding a Text-to-Speech provider, you can select this provider in your store settings and choose whether to enable TTS streaming.

Edit Store

Name:

Display name:

Title:

Avatar:

Storage provider:

Image provider:

Split provider:

Model provider:

Embedding provider:

Text-to-Speech provider:

Enable TTS streaming:

Frequency:

Now, your store can convert text responses to speech, providing a more interactive experience for users.

Add a Speech-to-Text Provider

This document is a step-by-step tutorial designed for beginners. It will guide you through the process of integrating a speech-to-text provider with Casibase, our powerful knowledge base system.

Introduction

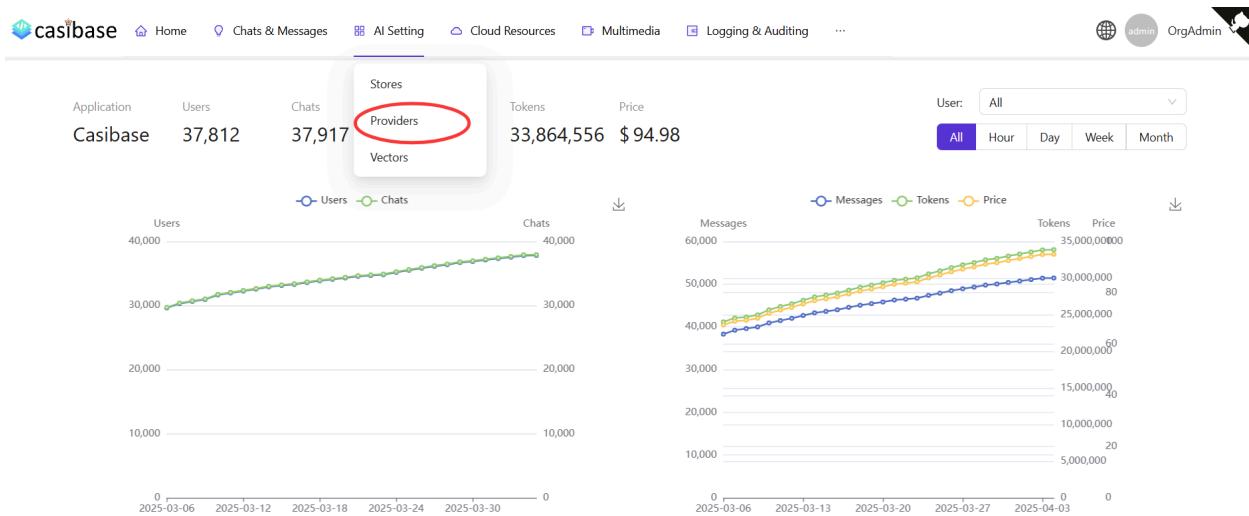
Speech-to-Text (STT) is a technology that converts spoken language into written text. STT providers allow your Casibase applications to understand and process spoken user input, enhancing the user experience and accessibility of your knowledge base system.

In Casibase, integrating an STT provider enables your AI applications to receive and process voice queries, creating more interactive and natural user interactions.

Add a New Speech-to-Text Provider

Speech-to-Text providers are used to integrate voice recognition capabilities into Casibase. You can add them by following these steps:

Click the **Providers** button on the page.



Add a Speech-to-Text Provider

Click the **Add** button to add a Speech-to-Text provider.

Providers **Add** Add Storage Provider

Name	Display name	Category	Type	Sub type	API key	Secret key	Region
provider_tts_alibabacloud_cosyvoice	Provider TTS AlibabaCloud Cosyvoice	Text-to-Speech	Alibaba Cloud	cosyvoice-v1		***	
provider_blockchain_chainmaker	Provider Blockchain ChainMaker	Blockchain	Tencent ChainMaker (Demo Network)	text-davinci-003	AKIDObZHjqUxnNltkKdIQyAWcS1JK4XidG	***	ap-beijing
provider_model_alibabacloud_deepseek_r1	Provider Model AlibabaCloud DeepSeek-R1	Model	Alibaba Cloud	deepseek-r1		***	
provider_cloud_alibabacloud	Provider Cloud AlibabaCloud	Public Cloud	Aliyun	text-davinci-003	LTAI5thkUsopAioN6xi2LMg	***	cn-beijing
dall-e-3	dall-e-3	Model	OpenAI	dall-e-3		***	
provider_model_azure_gpt4_1	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_model_azure_gpt4_2	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_model_azure_gpt4_3	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_embedding_openai_v3	Provider Embedding OpenAI V3	Embedding	OpenAI	text-embedding-ada-002		***	
provider_model_openai_gpt4_vision	Provider Model OpenAI GPT4 Vision	Model	OpenAI	gpt-4-vision-preview		***	

Fill in Speech-to-Text Provider Details

Fill in the speech-to-text provider details and click the **Save & Exit** button.

The screenshot shows the Casibase AI Setting interface with the 'Edit Provider' section active. The provider is named 'provider_njowpc' and has a display name 'New Provider - njowpc'. It is categorized as 'Speech-to-Text' and is of type 'Alibaba Cloud'. The sub-type is 'parafomer-realtime-v1'. The secret key is marked as '***'. The provider URL is set to <https://platform.openai.com/account/api-keys>. The state is set to 'Active'. At the bottom, there are 'Save' and 'Save & Exit' buttons. A red box highlights the provider configuration section.

Name: provider_njowpc

Display name: New Provider - njowpc

Category: Speech-to-Text

Type: Alibaba Cloud

Sub type: parafomer-realtime-v1

Secret key: ***

Provider URL: <https://platform.openai.com/account/api-keys>

State: Active

Save Save & Exit

Powered by casibase

Using Voice Recognition

When you click the voice recognition button in your Casibase application, the following process occurs:

1. The browser will request permission to access your microphone
2. Once granted, the system will begin listening and automatically convert your speech to text
3. After you finish speaking, the recognized text will be automatically sent as a message

Store: New Store - dir... ▾

Model: New Provider - 8mi1... ▾



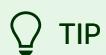
Hello, I'm Casibase AI Assistant

I'm here to help answer your questions

Type message here



This feature enables hands-free interaction with your Casibase applications, making them more accessible and convenient to use.



TIP

Casibase currently supports the following Speech-to-Text provider:

- [Alibaba Cloud](#)
 - paraformer-realtime-v1

Add a Store

We have added a storage provider, a model provider, and an embedding provider. Now we need to configure a store to use these providers.

 CAUTION

This guide assumes that you have already deployed a Casibase knowledge base system. If you haven't done, please refer to the [Deploy Casdoor and Casibase tutorial](#).

Besides, this guide assumes that you have already added a storage provider, a model provider, and an embedding provider. If you have not, please follow the [Add a Storage Provider](#), [Add a AI Model Provider](#), and [Add an Embedding Provider](#) guides.

Step 1: Add a New Store

Stores are used to integrate storage, model, and embedding providers into Casibase. You can add them by following these steps:

Click the `Stores` button on the home page and then click the `Add` button to add a store.

[Home](#)[Chat](#)[Stores](#)[Providers](#)[Vectors](#)[Chats](#)[Messages](#)

Stores

Add

Name	Display name	Storage provider
my_store	My_Store	provider_storage

Step 2: Fill in Store Details

Select the storage provider, model provider, and embedding provider you added before.

Fill in the store details and click the [Save & Exit](#) button.

casbin

Home Chat Stores Providers Vectors Chats Messages Tasks Resources  P

Edit Store Save

Name: my_store

Display name: My_Store

Storage provider: Provider_storage_1 (provider_storage_1)

Model provider: Model OpenAI text-davinci-003 (model_openai_text_davinci_003)

Embedding provider:

File tree:

Embedding_OpenAI_Adasimilarity (embedding_openai_adasimilarity)

My_Store

- alibaba_oss
 - audio
 - AC / DC - Highway To Hell.mp3 (8.34 MB)
 - document
 - casdoor-knowledge.doc (18.0 KB)
 - casdoor-knowledge.docx (10.9 KB)
 - casdoor-knowledge.html (23.5 KB)
 - casdoor-knowledge.md (2.12 KB)
 - casdoor-knowledge.pdf (107 KB)
 - image
 - lena.jpg (105 KB)
 - lena.tiff (768 KB)
 - video
 - my_video.mkv (456 KB)

Click the Save & Exit button and return to the stores list page:

casbin

Home Chat Stores Providers Vectors Chats Messages Tasks Resources  Permissions  Logs 

Jimmy

Stores Add

Name	Display name	Storage provider	Model provider	Embedding provider	Action
my_store	My_Store	provider_storage_1 	model_openai_text_davinci_003	embedding_openai_adasimilarity	View Refresh Vectors Edit Delete

< 1 >

Now, you can use the store to store knowledge base data, convert text to vectors, and chat with the chatbot.

In the next section, we will learn how to chat with the chatbot in Casibase.

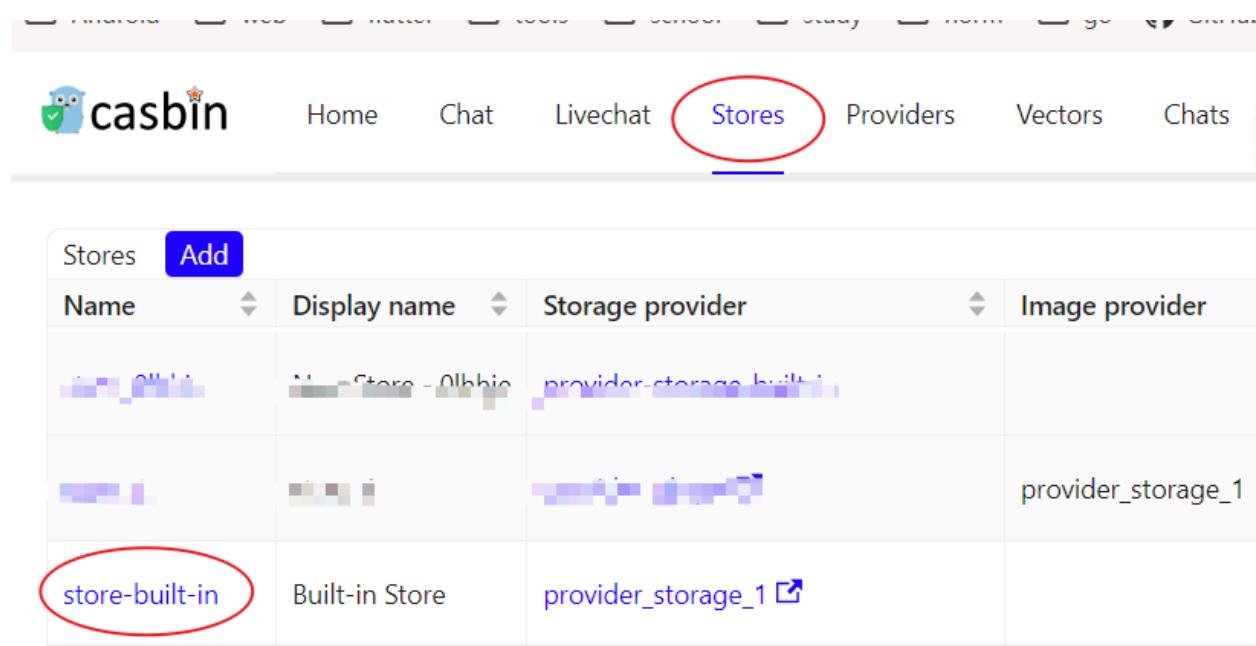
Support Multi-store

The multi-store mode provides users with different models, suggestions, and more within each distinct store.

Step 1: Enable Multi-store

First, you should enable multi-store mode in the built-in store.

Click the `Stores` button on the home page and then click the `store-built-in` button to enter the store-built-in store.



The screenshot shows the Casibase interface with the 'Stores' button highlighted by a red oval. Below the header, there is a table titled 'Stores' with an 'Add' button. The table has columns: Name, Display name, Storage provider, and Image provider. There are three rows in the table. The first row has a red oval around the 'store-built-in' name. The second row has a red oval around the 'provider_storage_1' storage provider. The third row has a red oval around the 'provider_storage_1' storage provider.

Stores		Add	
Name	Display name	Storage provider	Image provider
store-built-in	Store - Olbbic	provider_storage_builtin	
		provider_storage_1	
store-built-in	Built-in Store	provider_storage_1	

Scroll down and find the `Can Select Store` field, tick it.

The screenshot shows a software interface with the following elements:

- Suggestion count:** 3
- Theme color:** A color swatch showing blue.
- Can Select Store:** A checkbox with a checked mark, circled in red.
- File tree:** A hierarchical tree view of files and folders. The root folder is "casibase", which contains "user_iaycd9" and "chat_0edmw". "user_iaycd9" contains "chat_0edmw" (which has several image and text files), "chat_1gs4l9" (which has one image file), and "chat_1ikor4".
- Search bar:** "Please input your search term" with a magnifying glass icon.
- Message bar:** "No data"

Step 2: Add Usable Stores

The multi-store mode only provides usable stores. To make a store usable, you need to configure its storage provider, model provider, and embedding provider.

Step 3: Select For Conversation

Casibase provides a very convenient method for selecting a store.



[+ New Chat](#)

New Chat - 7

store_1

store-built-in

New Chat - 8

New Chat - 5

New Chat - 8

New Chat - 7

Chat

C

L

T

D

...

You are an expert in your field

Thank you for recognizing my expertise. Whether it's related to my specific area of knowledge or expertise to provide insightful answers and solutions to any problems you may have.

Just hover your mouse over "New Chat" and then you can select the Store you wish to use from the list that appears below.

If you click the "New Chat" button, the system will assign you a default Store.

Chats with AI

This document is a step-by-step tutorial designed for beginners. It will guide you through the process of implementing AI chat functionality in your Casibase knowledge base system.

Introduction

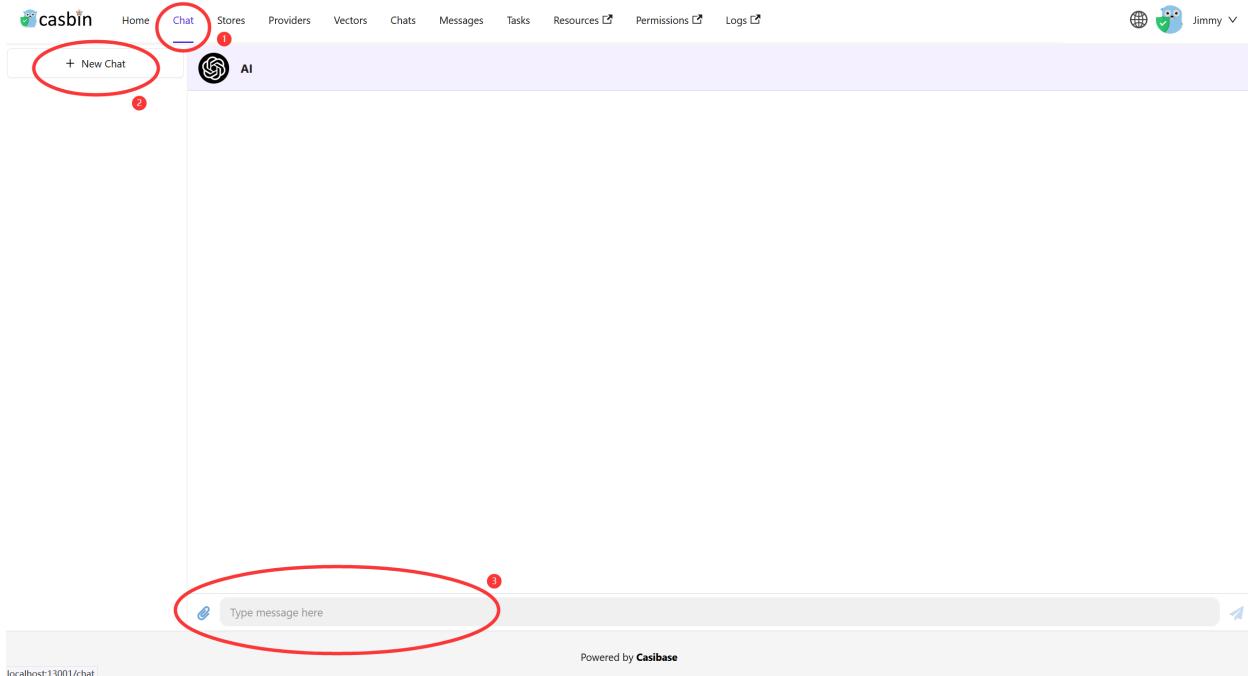
In previous sections, we have deployed Casdoor and Casibase, integrated a storage provider, a model provider, and a embedding provider with Casibase, and added a store to use these providers.

Refer to the [Add a Store](#) section of our previous documentation for more information about stores.

Now, let's implement AI chat functionality in Casibase.

Step 1: Access the Chat Interface

When you log in to Casibase, you'll land directly on the chat interface. This gives you immediate access to start conversations with AI. If you need to create a new chat session, click the `New Chat` button.



Step 2: Send a Message

Write a message and click the **Send** button to send it.



Step 3: Knowledge Base Chat

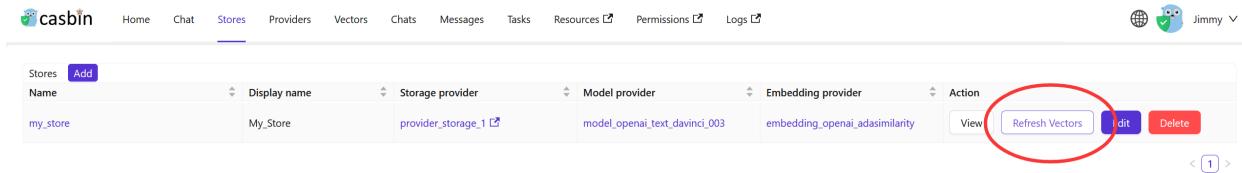
Additionally, you can chat with the chatbot in the knowledge base.

There are some requirements for the knowledge base chat:

- The knowledge base must have a store.
- The store must have a model provider.
- The store must have an embedding provider.

- The store must have a storage provider.
- The storage provider must have a readable document (e.g. a markdown file, docx file and pdf file).

Once you have met these requirements, you can return to the **Stores** page and click the **Refresh Vectors** button to embedding the knowledge base data.

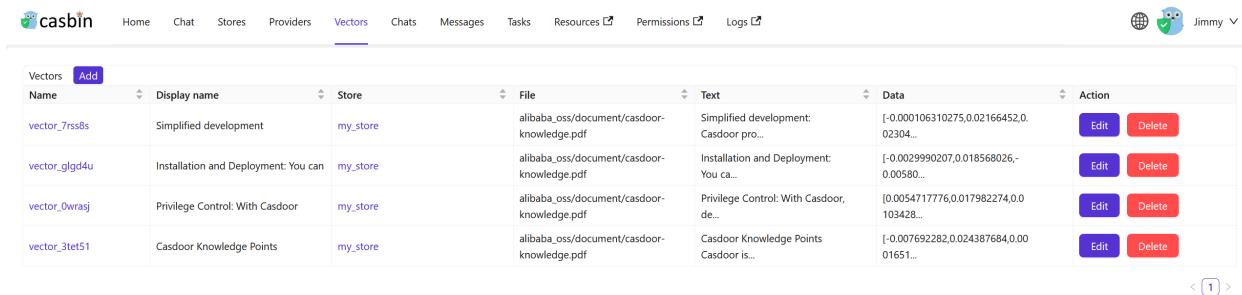


Name	Display name	Storage provider	Model provider	Embedding provider	Action
my_store	My_Store	provider_storage_1	model_openai_text_davinci_003	embedding_openai_adasimilarity	View Refresh Vectors Edit Delete

The button will be disabled when the embedding is in progress.

After the embedding is complete, you can click the **Vectors** button in the navigation bar to view the vectors.

Result:



Name	Display name	Store	File	Text	Data	Action
vector_7rss8s	Simplified development	my_store	alibaba_oss/document/casdoor-knowledge.pdf	Simplified development: Casdoor pro...	[-0.000106310275,0.02166452,0.02304...	Edit Delete
vector_glgd4u	Installation and Deployment: You can	my_store	alibaba_oss/document/casdoor-knowledge.pdf	Installation and Deployment: You ca...	[-0.0029990207,0.018568026,-0.00580...	Edit Delete
vector_0wrasj	Privilege Control: With Casdoor	my_store	alibaba_oss/document/casdoor-knowledge.pdf	Privilege Control: With Casdoor, de...	[0.0054717776,0.017982274,0.010342...	Edit Delete
vector_3tet51	Casdoor Knowledge Points	my_store	alibaba_oss/document/casdoor-knowledge.pdf	Casdoor Knowledge Points Casdoor is...	[-0.007692282,0.024387684,0.0001651...	Edit Delete

Let's chat with the chatbot in the knowledge base.



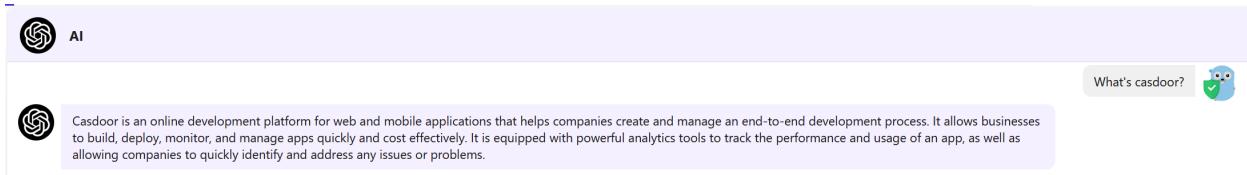
AI

What's casdoor?

Answer: Casdoor is an OAuth2 and OIDC based authentication portal designed to help developers easily add user authentication and authorization features to their applications.

When the AI responds using your knowledge base, you'll see a "Knowledge sources" button showing how many document fragments were referenced. Click it to view each source with its relevance score, and click any source to navigate directly to that document in your store.

Compare the results with non-knowledge base chat:



⚠ CAUTION

The embedding rate is related to two factors:

- The documents in the knowledge base:
 - Number of documents: The more documents, the longer the embedding time.
 - Size of documents: The larger the document size, the longer the embedding time.
- The embedding provider:
 - API rate limit: The more API rate limit, the faster the embedding speed.
 - API concurrency: The more API concurrency, the faster the embedding speed.

For example, if you use the [OpenAI API](#) as the embedding provider, the embedding rate is related to the [OpenAI API](#) rate limit and concurrency.

Conclusion

In this guide, we have learned how to implement AI chat functionality in Casibase.

Now, you can chat with the chatbot in Casibase. Enjoy it!

More information about Casibase can be found in the [Core Concepts](#) section of our documentation.

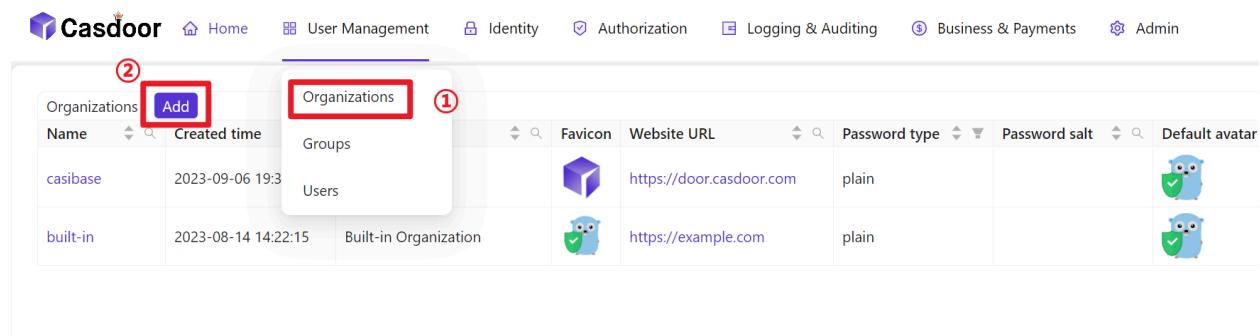
Casdoor-SSO

Casibase uses Casdoor as its identity and single-sign-on (SSO) provider. Make sure to deploy it in advance.

Please refer to [Casdoor Server Installation](#) to install and configure Casdoor.

Follow these steps to setup Casdoor for casibase:

- Create an Organization



Name	Created time	Groups	Favicon	Website URL	Password type	Password salt	Default avatar
casibase	2023-09-06 19:3	Users		https://door.casdoor.com	plain		
built-in	2023-08-14 14:22:15	Built-in Organization		https://example.com	plain		

- Configure information about the Organization

Casdoor Home User Management Identity Authorization Logging & Auditing Business & Payments Admin

Edit Organization

Save **Save & Exit** (2)

Name (2): casibase

Display name (1): Casibase

Favicon (2): URL (2): <https://cdn.casbin.org/img/favicon.png>

Preview: 

Website URL (2): <https://door.casdoor.com>

Password type (2): plain

- Create a new Application

Casdoor Home User Management Identity Authorization Logging & Auditing Business & Payments Admin

Applications (2) **Add**

Name	Created time	Display name
app-casibase	2023-09-06 19:38:54	Casibase
app-built-in	2023-08-14 14:22:15	Casdoor

Applications (1)

Logo	Organization	Providers
	casibase	provider_captcha_default
	built-in	provider_captcha_default

- Configuring Application Information (Remember Name, ClientID and ClientSecret)

Casdoor Home User Management Identity Authorization Logging & Auditing Business & Payments Admin All

Edit Application Save **Save & Exit** ④

Name ①: app-casibase ①

Display name ②: Casibase

Logo ③: URL ④: https://cdn.casbin.org/img/casdoor-logo_1185x256.png

Preview: 

Home ⑤: ⑤

Description ⑥: ⑥

Organization ⑦: casibase ⑦

Tags ⑧: ⑧

Client ID ⑨: 548c8b9c7431d2621db1 ⑨

Client secret ⑩: 2bc7640d487fc4dea6f4b77f07f1bf4433e4ad40 ⑩

Cert ⑪: cert-built-in

- Create a Certificate: In the Casdoor dashboard, choose Cert → Add, keep **Algorithm** as **RS256** (default), enter a name, and click Save.

Casdoor Home User Management Identity Authorization Logging & Auditing Business & Payments Admin

New Cert Save **Save & Exit** Cancel

Organization ⑪: casbin

Name ⑫: cert_casbin ⑫

Display name ⑬: New Cert - casbin

Scope ⑭: JWT

Type ⑮: x509

Crypto algorithm ⑯: RS256 (RSA + SHA256)

Bit size ⑰: 4096

Expire in years ⑱: 20

Certificate ⑲: **Copy certificate** **Download certificate** Private key: **Copy private key** **Download private key**

```
-----BEGIN CERTIFICATE-----  
MIIE2zCCAS0gAwIBAgIDAjEAMA0GCSqGSIb3DQEBCwUAMCoxD2ANBgNVBAoTBrmNh  
c2JpbEUUMBIGA1UEAwv1Y2Vydj94dGhry3iwfhNMjUwNzAyM1TzMiQ1WhhNNdUlw  
NzAyMT2mQ1WjAnQ8QwVQKwEzyXNjAw4x4fDASg8MVAMC2hcnfReGx0  
b2NvMIIcJiANBpkjkhkG9w0fAOFFAAOCAgBAMILCcKCAgEa11WZblk2c7B13y  
xpVdb+ikyqf5+eqnX0f07qK2slqMSWAu1CWSKsMULiOGLfpmrlhWpevkL  
WfWjpH4IAUfPcG+L4zTcaQfB89gu5fRYVWtgWeUra4hSP87c1b20UNQe8  
zdn0XDAgePldZXqoWHzlL6Gu12yQcJch1XawhHvhnhGW02W184XxJSLjq  
b63U1C8005Cw09ug5bgfWSo174VQXjutjSCR29d6yrlx143l0hapNkAT  
wqav18fge1Z7QzU1Mvzgnbill8Gf06EZOMMNNS4EcNrrR8AUJup552f1n0  
6j8nxzbeBMSG86Zt+J+D9P0tzlaxsLKMhM7ifGEZ739kXMuflsNtNgfEr  
XccQzQqybpzl32vx3iVNuJgOnlylFAOrmLEkjsgogm+j2zdc1Bz+dv3h
```

Cert ⑲: cert_casbin

- Bind the Certificate to the Application: Open the *Config* tab of your newly

created Application, select the certificate you just created from the Cert dropdown, and click Save.

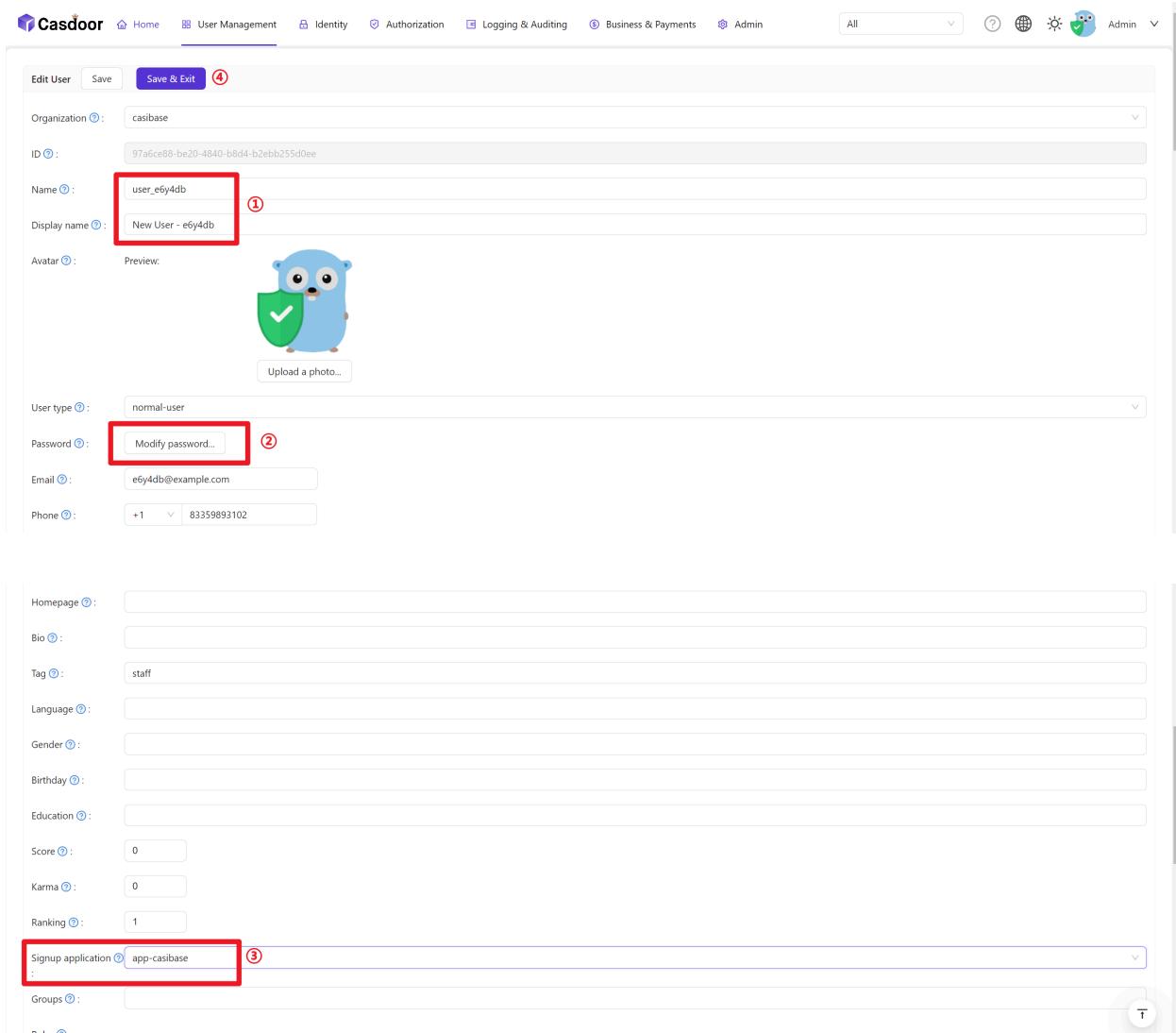


- Add a member to the newly created organization

Name	Created time	Display name	Favicon	Website URL	Password type	Password salt	Default avatar	Soft deletion	Action
casibase	2023-09-06 19:34:53	Casibase		https://door.casdoor.com	plain			<input type="radio"/>	Groups Users Edit Delete
built-in	2023-08-14 14:22:15	Built-in Organization		https://example.com	plain			<input type="radio"/>	Groups Users Edit Delete

Users	Add	Upload (xlsx)						
Organization	Application	Name	Created time	Display name	Avatar	Email	Phone	Affiliation
casibase	app-casibase	user_e6y4db	2023-09-06 19:37:26	New User - e6y4db		e6y4db@example.com	83359893102	Example Inc.

- Configure member information (remember its Name as well as Password)



The screenshot shows the Casdoor 'Edit User' interface. At the top, there are buttons for 'Edit User', 'Save', and 'Save & Exit'. The 'Save & Exit' button is highlighted with a red box and the number '4'. Below this, the 'Organization' field is set to 'casibase'. The 'ID' field contains a UUID. The 'Name' field is 'user_e6y4db' (highlighted with a red box and '1'), and the 'Display name' field is 'New User - e6y4db'. The 'Avatar' section shows a blue cartoon character holding a shield with a checkmark, with a 'Upload a photo...' button. The 'User type' is 'normal-user'. The 'Password' field is 'Modify password...' (highlighted with a red box and '2'). The 'Email' field is 'e6y4db@example.com' and the 'Phone' field is '+1 83359893102'. In the 'Additional Information' section, the 'Signup application' field is set to 'app-casibase' (highlighted with a red box and '3'). Other fields like 'Homepage', 'Bio', 'Tag', 'Language', 'Gender', 'Birthday', 'Education', 'Score', 'Karma', and 'Ranking' are present but empty. The 'Groups' section has a plus sign icon.

User Data Synchronization

Casibase automatically keeps user information synchronized with Casdoor. When a user's account is accessed through the application, Casibase fetches the latest user data directly from Casdoor rather than relying on cached session data. This ensures that any updates to user profiles in Casdoor—such as changes to name, email, avatar, or permissions—are immediately reflected in Casibase without requiring users to log out and log back in.

This real-time synchronization happens transparently in the background whenever the application needs to access the current user's account information.

Billing Integration

Casibase integrates with Casdoor's transaction system for AI usage billing. When users interact with AI models, transaction records are automatically created in Casdoor to track token consumption and costs. This enables centralized billing management across all your applications. For details on how transactions work, see the [Billing & Usage](#) section.

Permissions

Permissions Overview

Understanding Casibase's permission system

Frontend Permissions

Frontend permission utilities in Casibase

Backend Permissions

Backend permission utilities in Casibase

Permissions Overview

Casibase uses a role-based permission system to control access to features and resources. The system distinguishes between different user types, each with specific capabilities and access levels.

User Roles

Casibase supports several user roles:

System Admin - Full administrative access to all features, including system configuration, user management, and all resources. This is the highest privilege level.

Chat Admin - Administrative access focused on chat and knowledge base operations. Can manage stores, vectors, chats, activities, and view all users' data.

Chat User - Regular users who can interact with AI chat, access their assigned stores, and view their own usage data.

Video Normal User - Users with restricted video permissions. Can only modify videos in "Draft" state when no remarks are present.

Permission Hierarchy

The permission system follows this hierarchy (from highest to lowest):

1. System Admin (built-in admin or users with `isAdmin` flag)
2. Chat Admin (users with `type: "chat-admin"`)
3. Chat User (users with `type: "chat-user"`)

4. Anonymous/Guest users (limited read-only access)

Higher-level roles inherit the permissions of lower-level roles.

How Permissions Work

Permissions are checked throughout the application:

Frontend checks determine what UI elements users see and which client-side operations they can perform. These checks prevent unauthorized access to admin panels, configuration pages, and sensitive features.

Backend checks enforce permissions on API endpoints and data operations. Even if frontend checks are bypassed, backend validation ensures security at the server level.

This dual-layer approach provides defense in depth - the frontend offers a smooth user experience by hiding inaccessible features, while the backend guarantees security enforcement.

Authentication Integration

Casibase integrates with [Casdoor](#) for user authentication and SSO. User roles and permissions are managed through Casdoor and synchronized automatically with Casibase.

Implementation

Casibase centralizes permission logic to maintain consistency and simplify updates:

- Frontend utilities in `Setting.js` provide reusable permission checking functions
- Backend utilities in `util/permission.go` handle server-side authorization

For implementation details, see [Frontend Permissions](#) and [Backend Permissions](#).

Frontend Permissions

Casibase's React frontend uses centralized utility functions to check user permissions. These functions are defined in `web/src/Setting.js` and ensure consistent permission checking across all components.

Permission Utility Functions

`isAdminUser(account)`

Checks if a user is a system administrator.

```
if (Setting.isAdminUser(account)) {  
    // Show admin-only features  
}
```

Returns `true` for:

- Built-in admin account (`owner === "built-in"`)
- Users with the `isAdmin` flag set to true

`isChatAdminUser(account)`

Checks if a user has chat-admin privileges.

```
if (Setting.isChatAdminUser(account)) {  
    // Show chat admin features  
}
```

Returns `true` for users with `type === "chat-admin"`.

canViewAllUsers(account)

Checks if a user can view data for all users (not just their own).

```
if (Setting.canViewAllUsers(account)) {  
  // Enable "All users" option in dropdown  
}
```

Returns `true` for:

- The admin account (`name === "admin"`)
- Chat admin users

This function is commonly used in usage pages and activity logs where admins need to see organization-wide data.

isLocalAdminUser(account)

Checks if a user has local admin privileges, which includes both system admins and chat admins.

```
const canManagePanes = Setting.isLocalAdminUser(account);
```

Returns `true` for:

- System admins
- Chat admin users

isLocalAndStoreAdminUser (account)

Similar to `isLocalAdminUser()` but also ensures the user can manage store content (not restricted to "non-store-admin" homepage).

```
if (Setting.isLocalAndStoreAdminUser(account)) {  
  // Allow file operations  
}
```

Practical Examples

Restricting Menu Items

```
// App.js - Building navigation menu  
if (!Setting.isAdminUser(this.state.account)) {  
  if (!Setting.isChatAdminUser(this.state.account)) {  
    return res; // Return limited menu for regular users  
  }  
}
```

Conditional UI Elements

```
// UsagePage.js - User selection dropdown  
<option key="all" value="All"  
  disabled={!Setting.canViewAllUsers(this.props.account)}>  
  All  
</option>
```

Permission-Based Features

```
// ProviderEditPage.js - Sensitive field access
<Input.Password
  value={this.state.provider.providerKey}
  disabled={!Setting.isAdminUser(this.props.account)}
  onChange={e => this.updateProviderField("providerKey",
  e.target.value)}
/>
```

Benefits of Centralized Functions

Before these utilities existed, permission checks were scattered across components using inline conditions like `account.name === "admin" || account.type === "chat-admin"`. This made the code:

- **Fragile:** Changing role logic required updating multiple files
- **Inconsistent:** Different components might check permissions differently
- **Error-prone:** Easy to miss edge cases or make typos

Centralized utilities solve these issues by providing a single source of truth. Updates to permission logic happen in one place and automatically apply throughout the application.

Implementation Notes

All utility functions perform null/undefined checks on the account parameter, making them safe to call without prior validation. They return `false` for invalid or missing accounts, following a secure-by-default approach.

Backend Permissions

Casibase's Go backend centralizes permission checking through utility functions in `util/permission.go`. These functions provide consistent role validation across controllers, routers, and service layers.

Permission Functions

`IsAdminOrChatAdmin(user)`

Checks if a user is either a system admin or a chat admin.

```
if !util.IsAdminOrChatAdmin(user) {
    c.ResponseError("this operation requires admin privilege")
    return
}
```

Returns `true` for:

- Users with `IsAdmin` flag set
- Users with `Type == "chat-admin"`

This is the most commonly used permission check for operations that require elevated privileges.

`IsChatAdmin(user)`

Checks specifically for chat admin role.

```
if util.IsChatAdmin(user) {
    // Grant chat-admin specific permissions
}
```

Returns `true` only for users with `Type == "chat-admin"`.

IsVideoNormalUser(user)

Checks if a user has the video-normal-user role, which has restricted video editing permissions.

```
if util.IsVideoNormalUser(user) {
    if len(video.Remarks) > 0 || video.State != "Draft" {
        c.ResponseError("video can only be updated in Draft state")
        return
    }
}
```

Returns `true` for users with `Type == "video-normal-user"`.

Role Type Constants

The package defines constants for role types to eliminate hardcoded strings:

```
const (
    UserTypeChatAdmin      = "chat-admin"
    UserTypeVideoNormalUser = "video-normal-user"
)
```

These constants ensure type safety and prevent typos when checking user roles.

Usage in Controllers

Authorization Filter

The router middleware uses permission checks to protect admin endpoints:

```
// routers/authz_filter.go
func permissionFilter(ctx *context.Context) {
    user := GetSessionUser(ctx)

    if !util.IsAdminOrChatAdmin(user) {
        responseError(ctx, "this operation requires admin
privilege")
        return
    }
}
```

API Controllers

Controllers use these functions to enforce business logic permissions:

```
// controllers/account.go - User type assignment
if strings.Count(claims.Type, "-") <= 1 {
    if !util.IsAdminOrChatAdmin(&claims.User) {
        claims.Type = "chat-user"
    }
}

// controllers/video.go - Video update restrictions
if util.IsVideoNormalUser(user) {
    if len(video.Remarks) > 0 || video.State != "Draft" {
```

Service Layer

Object and service layers also benefit from centralized checks:

```
// object/util.go
func isAdmin(user *casdoorsdk.User) bool {
    return util.IsAdminOrChatAdmin(user)
}
```

Migration Impact

Before centralization, permission checks were duplicated across the codebase:

```
// Before: inline, repeated logic
isAdmin := user != nil && (user.IsAdmin || user.Type == "chat-
admin")
if !isAdmin {
    return error
}

// After: single, reusable function
if !util.IsAdminOrChatAdmin(user) {
    return error
}
```

The migration improved:

Maintainability: Permission logic changes happen in one place **Consistency:** All parts of the system use identical checks **Security:** Reduced risk of permission check errors

Safety Features

All permission functions include null safety checks:

```
func IsAdminOrChatAdmin(user *casdoorsdk.User) bool {
    if user == nil {
        return false
    }
    return user.IsAdmin || user.Type == UserTypeChatAdmin
}
```

This prevents nil pointer dereferences and ensures permission checks fail safely when user data is unavailable.

Developer Guide

Generating Swagger Files

Generating Swagger Files

Generating Swagger Files

Overview

As we know, the beego framework provides support for generating swagger files to clarify the API via the command line tool called "bee". Casibase is also built based on beego. However, we found that the swagger files generated by bee failed to categorize the APIs with the "@Tag" label. So, we modified the original bee to implement this function.

How to write the comment

Most rules are exactly identical to the original bee comment formats. The only discrepancy is that the API shall be divided into different groups according to the "@Tag" label. Therefore, developers are obliged to ensure that this tag is correctly added. Here is an example:

```
// @Title Login
// @Tag Login API
// @Description login
// @Param oAuthParams     query    string  true      "oAuth
parameters"
// @Param body    body    RequestForm  true      "Login
information"
// @Success 200 {object} controllers.api_controller.Response The
Response object
// @router /login [post]
func (c *ApiController) Login() {
```

APIs with the same "@Tag" labels will be put into the same group.

How to generate the swagger file

0. Write comments for the API in the correct format.
1. Fetch this repository: <https://github.com/casbin/bee>.
2. Build the modified bee. For example, in the root directory of casbin/bee, run the following command:

```
go build -o mybee .
```

3. Copy mybee to the base directory of Casibase.
4. In that directory, run the following command:

```
mybee generate docs
```

5. (Optional) If you want to generate swagger document for specific tags or apis, here are some example commands:

```
mybee generate docs --tags "Adapter API"  
mybee generate docs --tags "Adapter API,Login API"  
mybee generate docs --apis "add-adapter"  
mybee generate docs --apis "add-adapter,delete-adapter"
```

Notably: We only accept a comma  as the separator when multiple tags/apis provided.

Then you will find that the new swagger files are generated.

Deployment

Deploy Casdoor and Casibase

Discover how to deploy Casdoor and Casibase.

Deploy Casdoor and Casibase

Introduction



What is Casdoor?

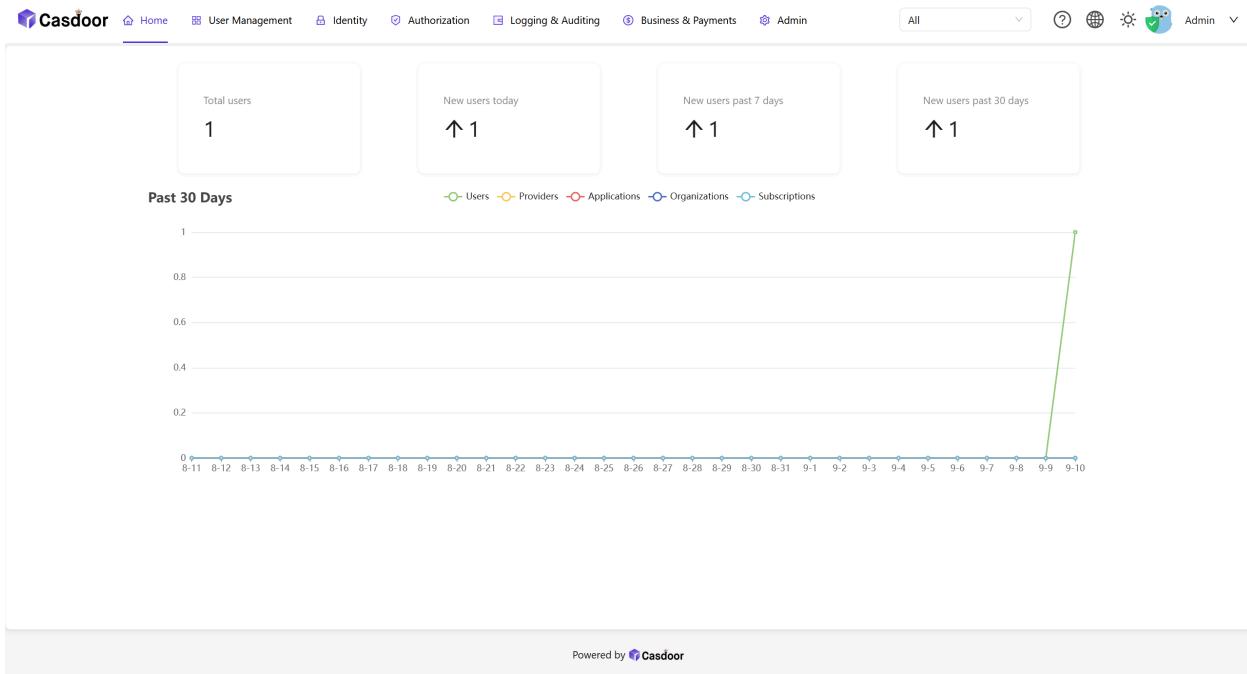
Casdoor is a powerful authentication system that provides a secure and reliable login experience. It's a prerequisite for Casibase, so be sure to deploy it first.

Refer to the [Casdoor](#) website for more information.

Step 1: Deploy Casdoor

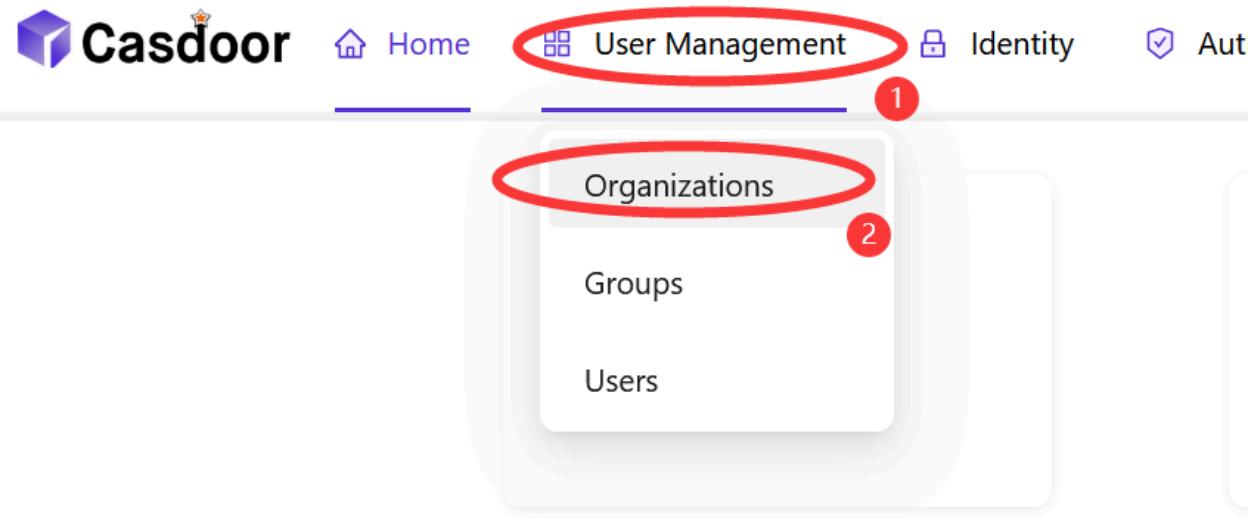
In [Casdoor Deployment Guide](#), you can find the detailed steps to deploy Casdoor.

Once you've deployed Casdoor, you'll look like this:



Step 2: Create an organization in Casdoor

In Casdoor, you can create an organization to manage your users and applications. You can create an organization by clicking the [User Management - Organizations](#) button on the home page.



Past 30 Days

Step 2.1: Add an organization

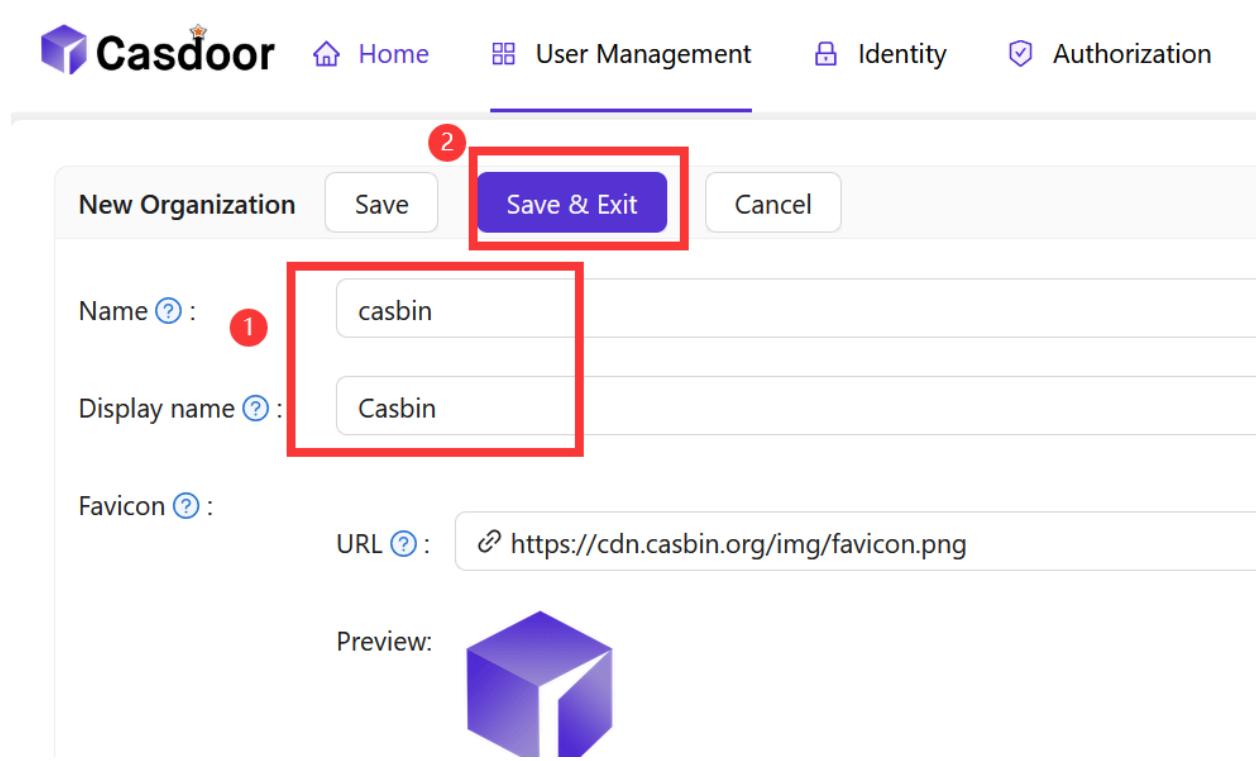
Click the **Add** button to add an organization.

The screenshot shows the 'Organizations' list page in the Casdoor application. At the top, there is a header with the Casdoor logo, 'Home', 'User Management', and 'Identity' menu items. Below the header, there is a search bar and a table with three columns: 'Name', 'Created time', and 'Display name'. The table contains one row with the data: 'built-in', '2023-09-10 19:31:50', and 'Built-in Organization'. A red circle highlights the 'Add' button in the top right corner of the table header area.

Name	Created time	Display name
built-in	2023-09-10 19:31:50	Built-in Organization

Step 2.2: Fill in the organization information

Fill in the organization information and click the **Save & Exit** button.



New Organization

Name ? : casbin

Display name ? : Casbin

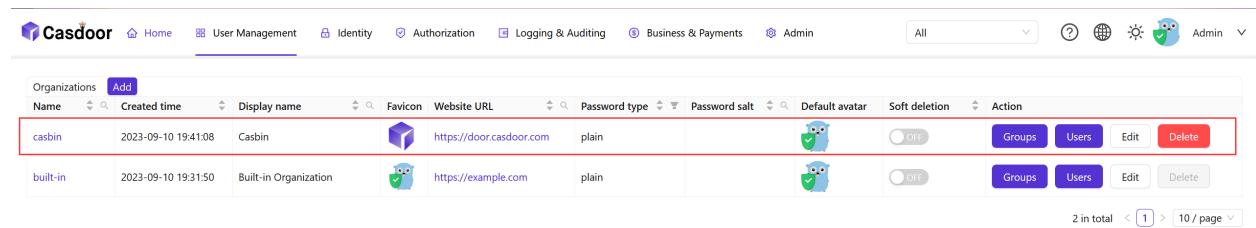
Favicon ? :

URL ? : <https://cdn.casbin.org/img/favicon.png>

Preview: 

Step 2.3: View the organization

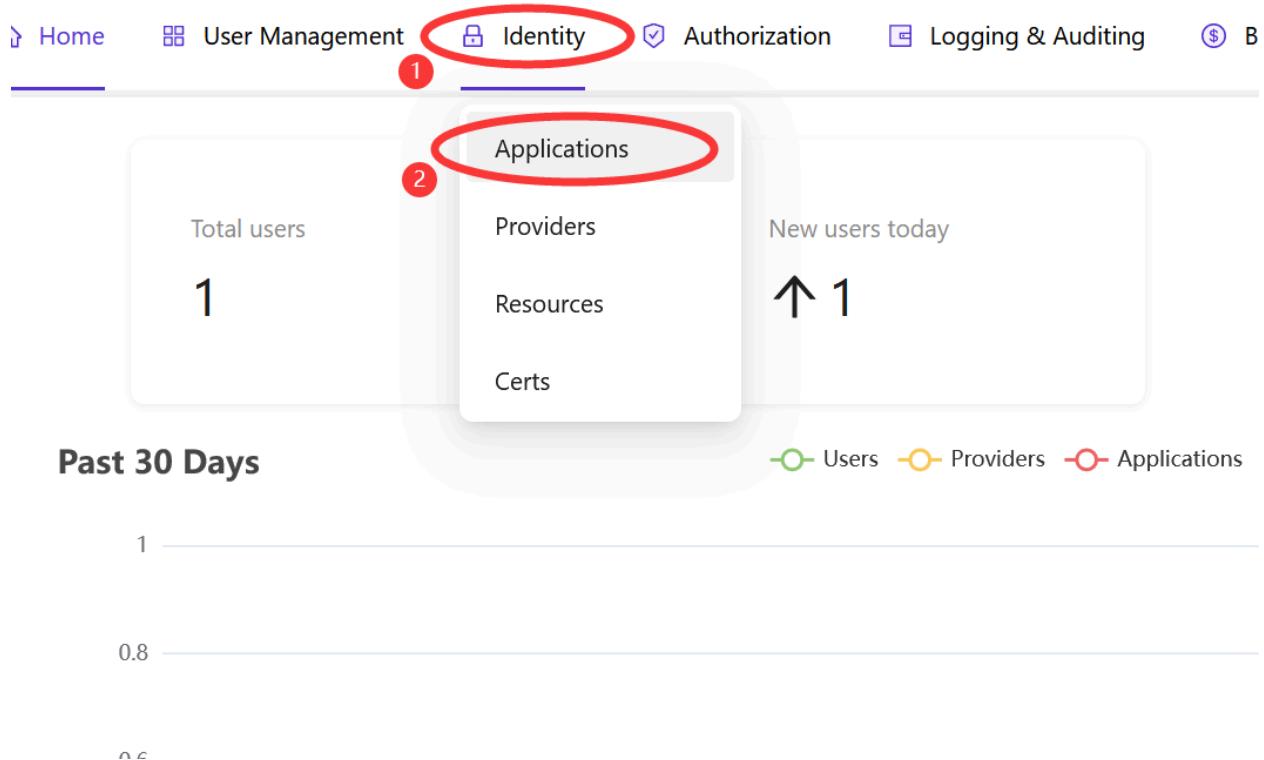
After adding the organization, you can view the organization information.



Name	Created time	Display name	Favicon	Website URL	Password type	Password salt	Default avatar	Soft deletion	Action
casbin	2023-09-10 19:41:08	Casbin		https://door.casdoor.com	plain			<input type="radio"/>	Groups Users Edit Delete
built-in	2023-09-10 19:31:50	Built-in Organization		https://example.com	plain			<input type="radio"/>	Groups Users Edit Delete

Step 3: Create an application in Casdoor

In Casdoor, you can create an application to manage your users and organizations. You can create an application by clicking the `Identity - Applications` button on the home page.



Step 3.1: Add an application

Click the `Add` button to add an application.

Applications		
Name	Created time	Display name
app-built-in	2023-09-10 19:31:50	Casdoor

Step 3.2: Fill in the application information

Fill in the application information and click the [Save & Exit](#) button.

Casdoor Home User Management Identity Authorization Logging & Auditing Business & Payments Ad

New Application Save Save & Exit Cancel

Name ?: app-casibase 5

Display name ?: Casibase 1

Logo ?: URL ?: https://cdn.casbin.org/img/casdoor-logo_1185x256.png

Preview: 

Home ?: [Edit](#)

Description ?: 2

Organization ?: casbin 2

Tags ?: 3

Client ID ?: 2786e0cbadfb56287a9a 3

Client secret ?: 4f9957d3e679efdb3391eb42b38d274d46fa1232 3

Cert ?: cert-built-in

Redirect URLs ?: [Add](#)

Redirect URI 4

<http://localhost:14000/callback>

Step 3.3: View the application

After adding the application, you can view the application information.

Casdoor Home User Management Identity Authorization Logging & Auditing Business & Payments Admin

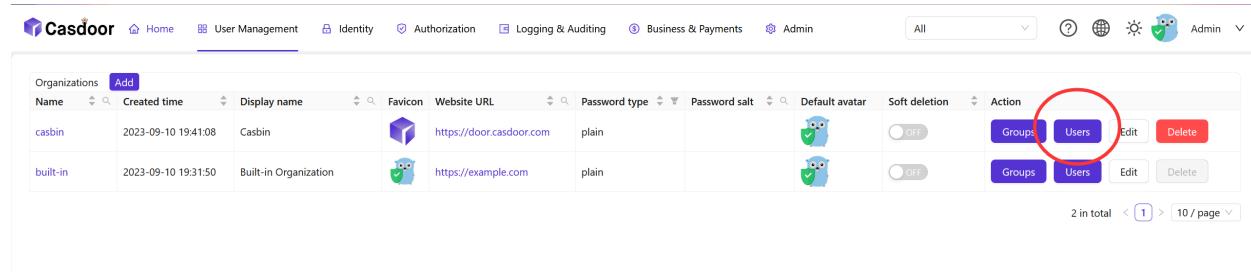
Applications [Add](#)

Name	Created time	Display name	Logo	Organization	Providers	Action
app-casibase	2023-09-10 19:44:08	Casibase		casbin	provider_captcha_default	Edit Delete
app-built-in	2023-09-10 19:31:50	Casdoor		built-in	provider_captcha_default	Edit Delete

2 in total < 1 > 10 / page

Step 4: Create a user in Casdoor for Casibase

In Casdoor, you can create a user to login Casibase. You can create a user by clicking the `User Management - Organizations - Users` button from the home page.



The screenshot shows the Casdoor User Management interface. The top navigation bar includes Home, User Management (which is active and highlighted in blue), Identity, Authorization, Logging & Auditing, Business & Payments, Admin, and various system icons. The main content area is titled 'Organizations' and shows a table with two rows. The first row represents the 'casbin' organization, which was created on 2023-09-10 19:41:08. The second row represents the 'built-in' organization, created on 2023-09-10 19:31:50. Each organization row has columns for Name, Created time, Display name, Favicon, Website URL, Password type, Password salt, Default avatar, Soft deletion, and Action. The 'Action' column for each row contains three buttons: 'Groups' (purple), 'Users' (blue, which is circled in red), and 'Edit' (grey). Below the table, a message indicates '2 in total' with navigation arrows and a page size of '10 / page'.

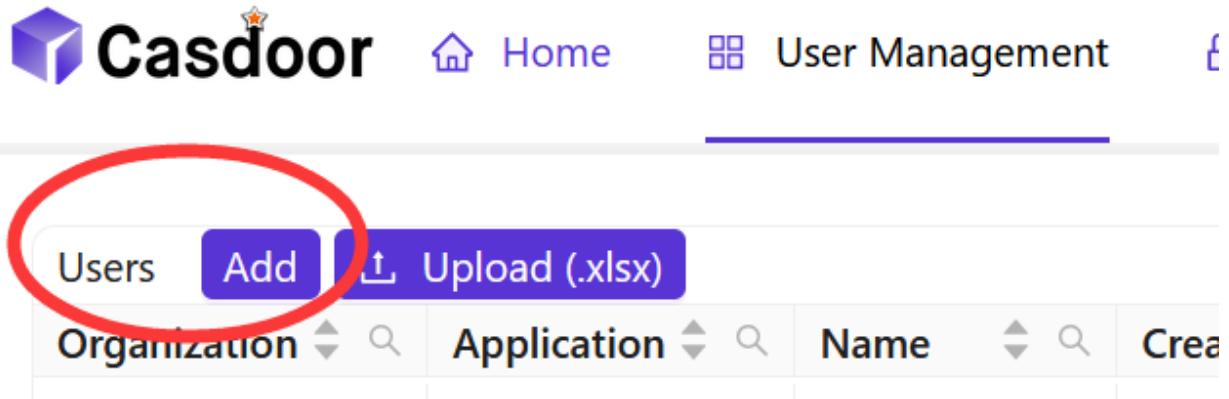


A user is a member of an organization who can login to applications in the organization.

Refer to the [Casdoor](#) website for more information.

Step 4.1: Add a user

Click the `Add` button to add a user.



Step 4.2: Fill in the user information

Fill in the user information and click the **Save & Exit** button.

Casdoor [Home](#) [User Management](#) [Identity](#) [Authorization](#) [Logging & Auditing](#) [Business & Payments](#) [⚙️](#)

New User [Save](#) [Save & Exit](#) [Cancel](#)

Organization [?](#) : casbin 1

ID [?](#) : d5bc730c-312c-406e-ae03-e6580d7590f4

Name [?](#) : jimmy

Display name [?](#) : Jimmy 2

Avatar [?](#) : Preview:  Upload a photo...

User type [?](#) : normal-user

Password [?](#) : [Modify password...](#) 3

Email [?](#) : t414w5@example.com

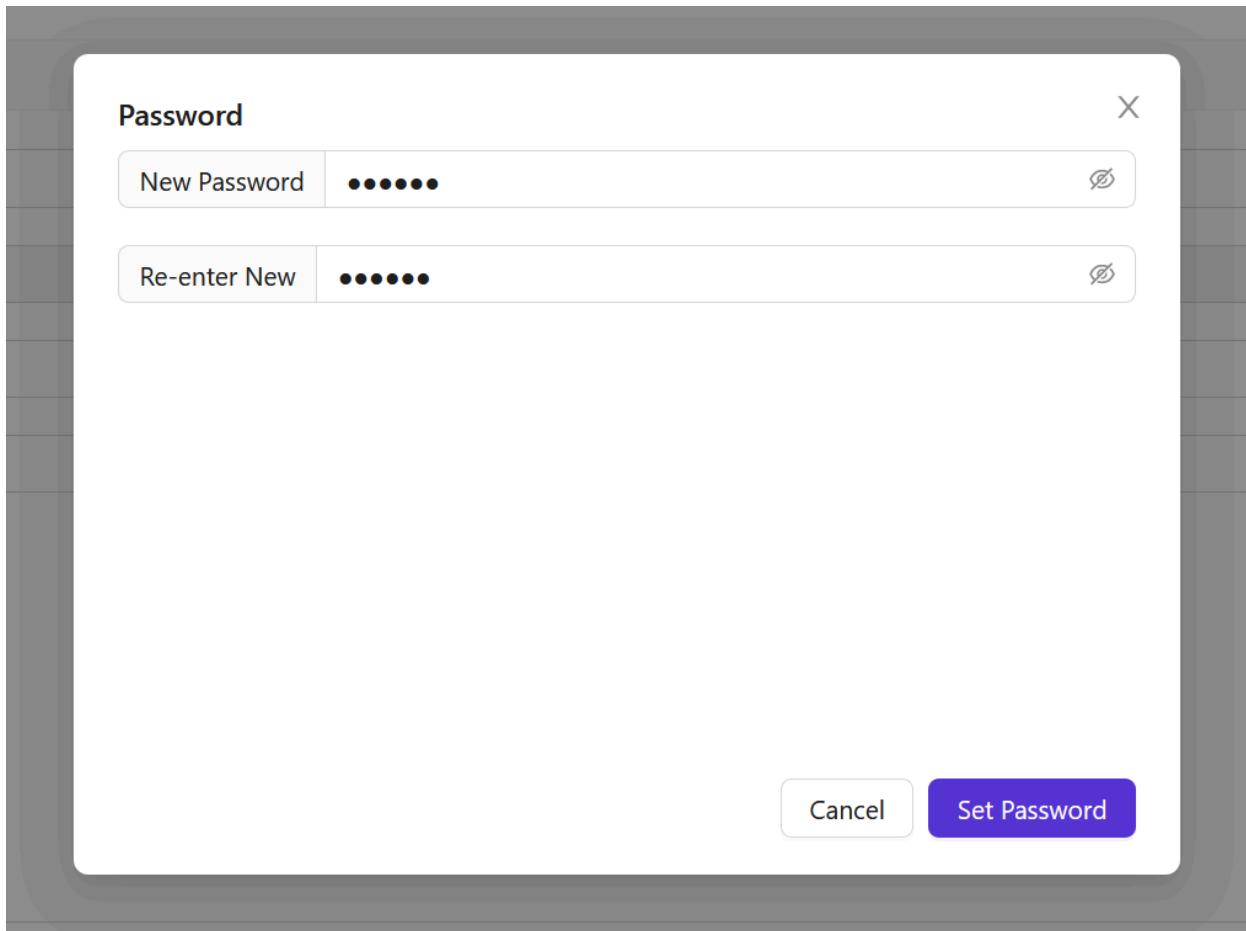
Phone [?](#) : +1 71700415009

Country/Region [?](#) : Please select country/region

Location [?](#) :

- **Password**

You can set the user's password by clicking the [Modify password](#) button.



- Admin

You can set the user's admin permission by clicking the `Is admin` button.

Permissions [?](#) :

Groups [?](#) :

3rd-party logins [?](#) :

Is admin [?](#) :

Is forbidden [?](#) :

Is deleted [?](#) :

Multi-factor authentication [?](#) : Multi-factor methods

Step 4.3: View the user

After adding the user, you can view the user information.

Organization	Application	Name	Created time	Display name	Avatar	Email	Phone	Affiliation	Country/Region	Tag	Is ac	Action	
casbin	app-built-in	jimmy	2023-09-10 20:51:18	Jimmy		t414w5@example.com	71700415009	Example Inc.			Off	Edit	Delete

Step 5: Deploy Casibase

Like Casdoor, you can deploy Casibase by following the [Casibase Deployment Guide](#).

Once you've deployed Casibase, you'll look like this:

How to Connect to Casibase

Overview

Learn about different ways to connect to and integrate with Casibase.

Casibase SDKs

Learn how to integrate and use Casibase SDKs with your applications.

Using Casibase OpenAI API Compatible Interface

Learn how to connect external chat UIs to Casibase using OpenAI API compatibility.

Overview

Overview

In this section, we will show you how to connect your application to Casibase.

Casibase provides two main methods for integrating with your applications:

- [Casibase SDK](#) - For direct integration with Casibase's API
- [OpenAI API Compatibility](#) - For connecting existing OpenAI-compatible UIs and clients

Casibase SDK

What is Casibase SDK?

Casibase SDK provides a programmatic way to interact with Casibase services. It offers a convenient set of APIs that allow developers to manage tasks, knowledge bases, and other features of Casibase directly from their applications.

We recommend using the Casibase SDK for the following reasons:

1. It provides direct access to Casibase-specific functionality
2. It simplifies authentication and configuration
3. It handles error cases and provides a more developer-friendly experience

Currently, Casibase offers a Java SDK, with more language support planned for the future.

OpenAI API Compatibility

What is OpenAI API Compatibility?

Casibase supports the OpenAI API format, allowing you to connect any OpenAI-compatible chat UI or client application to Casibase. This makes it easy to use popular open-source chat interfaces with Casibase's backend.

We recommend using the OpenAI API compatibility for the following reasons:

1. It allows you to use your preferred chat UI with Casibase
2. It simplifies integration if you're already using OpenAI-compatible tools
3. It provides a standardized way to interact with Casibase's AI capabilities

This approach is particularly useful if you want to quickly integrate Casibase with existing applications that already support the OpenAI API format.

Casibase SDKs

Introduction

Casibase provides SDKs to help developers integrate with Casibase's APIs more easily. The SDKs offer a convenient way to interact with Casibase's services for tasks like managing AI conversations, knowledge bases, and more.

Currently, Casibase offers a Java SDK, with more language support planned for the future.

Backend SDK	Description	SDK code	Example code
Java SDK	For Java backends	casibase-java-sdk	-

How to use Casibase SDK?

1. Backend SDK configuration

When your application starts up, you need to initialize the Casibase SDK config by providing the required parameters.

Take casibase-java-sdk as an example:

```
CasibaseConfig config = new CasibaseConfig(  
    "https://demo-admin.casibase.com", // endpoint  
    "your-client-id", // clientId  
    "your-client-secret", // clientSecret
```

All the parameters for initialization are explained as follows:

Parameter	Required	Description
endpoint	Yes	Casibase Server URL, like https://demo-admin.casibase.com or http://localhost:14000
clientId	Yes	Client ID for the Casibase application
clientSecret	Yes	Client secret for the Casibase application
organizationName	Yes	The name for the Casibase organization, e.g., casbin
applicationName	No	The name for the Casibase application, e.g., app-casibase

2. Available Services

Once you have initialized the configuration, you can create and use the available services. Currently, the only available service is `TaskService`.

```
TaskService taskService = new TaskService(config);
```

TaskService

`TaskService` supports basic task operations, such as:

- `getTask(String name)`: Get a single task by task name.
- `getTasks()`: Get all tasks under the `organizationName`.

- `addTask(Task task)`: Add a new task to the database.
- `updateTask(Task task)`: Update an existing task in the database.
- `deleteTask(Task task)`: Delete a task from the database.

Using Casibase OpenAI API Compatible Interface

This document is a step-by-step tutorial designed for beginners. It will guide you through the process of connecting external chat UIs to Casibase using its OpenAI API compatibility feature.

Introduction

Casibase now supports integration with external chat UIs through OpenAI API compatibility. This feature allows you to use popular open-source chat interfaces with Casibase's backend, giving you more flexibility in how you interact with your knowledge base system.

If you're looking to use your preferred chat UI with Casibase, this guide will walk you through the simple setup process.

Step 1: Set Up Casibase with a Model Provider

Before connecting an external UI, ensure you have Casibase properly set up with a model provider. If you haven't done this yet, please refer to the [Add an AI Model Provider](#) tutorial.

Step 2: Get Your OpenAI-compatible API Key

When you create a model provider in Casibase, an API key is automatically generated. This key allows external applications to communicate with Casibase using the OpenAI API format.

Step 2.1: Access Your API Key

Navigate to the **Providers** section and select your model provider. Only administrators can view and modify API keys.

Name:	provider_prm93r
Display name:	New Provider - prm93r
Category:	Model
Type:	OpenAI
Sub type:	text-davinci-003
Client secret	
Temperature:	1.00
Top P:	1.00
Presence penalty:	0.00
Frequency penalty:	0.00
API key:	sk-UflKsbiYjzBvjeUFJbDpxuKg
Provider URL:	https://platform.openai.com/account/api-keys
State :	Active

TIP

If the API key field is empty, Casibase will automatically generate a new key when you save the provider.

Step 3: Configure Your External Chat UI

Once you have your API key, you can configure your external chat UI to connect to Casibase.

Step 3.1: Configure with chatgpt-web

For this example, we'll use [chatgpt-web](#), a popular open-source ChatGPT interface.

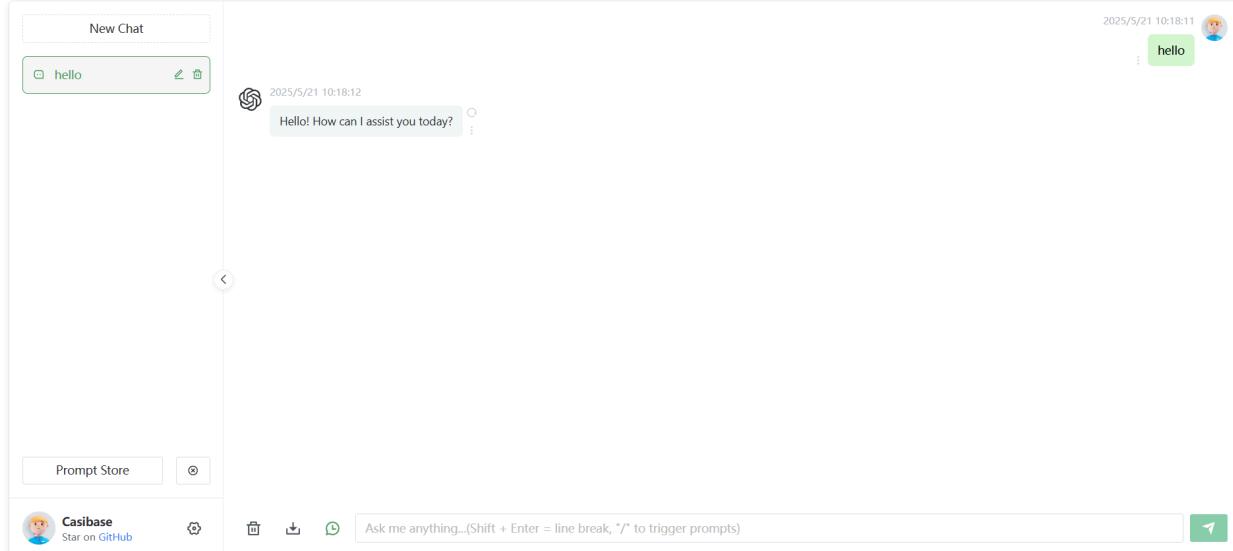
1. Locate the `service/.env` file in your chatgpt-web installation
2. Modify the following variables:
 - `OPENAI_API_KEY`: Set this to your Casibase-generated API key
 - `OPENAI_API_BASE_URL`: Set this to `http://your-casibase-backend:port/api`

```
# Example configuration
OPENAI_API_KEY=sk-UflKsbiYjzBvjeUFJbDpxuKg
OPENAI_API_BASE_URL=http://localhost:14000/api
```

Make sure your Casibase backend is accessible from the machine running your chat UI. Check firewall settings if you encounter connection issues.

Step 4: Test Your Integration

Start your chat UI application and test the connection. You should now be able to interact with Casibase through your preferred interface. If everything is set up correctly, you should see responses from Casibase in your chat UI.



Compatible Chat UIs

Casibase's OpenAI API compatibility has been tested with these popular chat interfaces:

- [chatgpt-web](#)

Other chat UIs that use the standard OpenAI API format should also work with Casibase.

Providers



Overview

Providers Overview



Model Providers

Introduction



Embedding Providers

Introduction



Storage Providers

Introduction



Text-to-Speech Providers

Introduction



Speech-to-Text Providers

Introduction



Bot Providers

Introduction



Model Providers

1 items



Scan Providers

6 items



Blockchain Providers

2 items



Private Cloud Providers

1 items



Public Cloud Providers

1 items

Overview

Casibase is an open source AI knowledge base system designed to provide efficient and flexible knowledge management and dialogue solutions for enterprises. One of its core features is Providers, which allow users to integrate multiple AI models, storage services, and infrastructure management capabilities. These integrations enhance the functionality and performance of the system.

Providers span several categories including AI Providers (Model and Embedding), Storage Providers, Cloud Providers, and Scan Providers, each serving distinct roles in the system. The provider list interface includes filtering capabilities, allowing you to quickly locate providers by category or type through dedicated filter controls.

1. Model Providers

Model Providers is a component in Casibase for integrating and managing AI models. It allows users to integrate various pre-trained AI models into the system for smarter knowledge processing and dialogue generation. With Model Provider, users can easily switch between different AI models, choosing the most appropriate model according to specific needs.

Casibase integrates with popular AI models through its provider system. Choose from cloud services like OpenAI and Alibaba Cloud, or connect to open-source models through Hugging Face and local deployments.

Model Provider Types

OpenAI — GPT-3.5-turbo, GPT-4, and reasoning-focused o1 models

Azure OpenAI — Access OpenAI models through Azure infrastructure

Alibaba Cloud — Qwen models plus DeepSeek v3, v3.1, v3.2, and R1

DeepSeek — DeepSeek V3.2 series with chat and reasoner variants optimized for reasoning tasks

Claude — Anthropic's Claude models including Opus 4.5, Sonnet 4, and Haiku for different use cases

Moonshot (Kimi) — Moonshot v1 series and Kimi K2 models with advanced thinking capabilities

Ernie — Baidu's ERNIE-Bot series

Hugging Face — Open-source models including Llama-2, ChatGLM2, and community contributions

2. Embedding Providers

Data vectorisation

The main role of Embedding Providers is to transform various types of data (e.g., text, images, etc.) into dense vector representations. This transformation is a key step in data processing and analysis in Casibase, enabling data to be stored, retrieved and analysed in a more efficient manner.

Knowledge Retrieval

By converting both the data in the knowledge base and the user's query into vectors, Embedding Providers enables the system to perform fast knowledge retrieval based on vector similarity. This greatly improves the efficiency and

accuracy of knowledge base retrieval.

Flexible model support

Embedding Providers support a variety of embedding models, users can choose the most suitable model according to their needs.

3. Storage Providers

We can configure the storage providers in Casdoor. and use it in Casibase, which is the component used to manage Casibase data storage and retrieval. It allows users to store data in different storage services and access the data through a unified interface. With Storage Providers, users can flexibly choose storage services to ensure data security and efficient access. supports two types of storage: Local and Cloud.

4. Text-to-Speech Providers

Text-to-Speech (TTS) Providers is a component in Casibase that enables the conversion of text responses into natural-sounding speech. It allows the system to communicate with users through voice synthesis, enhancing the interactive experience of the knowledge base system.

Provider Support

Currently, Casibase supports Alibaba Cloud's Text-to-Speech service, with various voice options available through the cosyvoice-v1 interface. The system is designed to be extensible, allowing for the integration of additional TTS providers in the future.

5. Speech-to-Text Providers

Speech-to-Text (STT) Providers is a component in Casibase that enables the conversion of spoken language into written text. It allows the system to understand and process voice queries, enhancing the interactive experience of the knowledge base system.

Local

We support uploading files to the local system.

Cloud

We support AWS S3, Azure Blob Storage, MinIO, Alibaba Cloud OSS, Tencent Cloud COS, and we are constantly adding more Cloud storage services.

6. Cloud Providers

Cloud Providers enable Casibase to connect to and manage infrastructure across private and public cloud platforms. These providers allow centralized visibility and control of cloud resources.

Private Cloud Providers connect to containerized environments like Docker and Kubernetes, enabling container lifecycle management and orchestration directly from Casibase.

Public Cloud Providers scan and catalog cloud infrastructure resources across major platforms. They automatically discover virtual machines, storage volumes, network components, and other cloud assets, enriching each with detailed

configuration information through multi-level API discovery.

7. Bot Providers

Bot Providers enable integration between Casibase and enterprise messaging platforms. They allow Casibase to respond to messages in real-time through bot interfaces, bringing AI capabilities directly into workplace communication tools.

WeCom Bot Provider connects Casibase to WeCom (WeChat Work) through the bot interface. It processes incoming messages, generates AI responses using configured stores, and supports streaming responses for better user experience. The provider handles message encryption, decryption, and URL verification required by WeCom's security protocol.

8. Scan Providers

Scan Providers perform network discovery, security auditing, and system assessment operations. Casibase integrates specialized scanning tools that analyze infrastructure assets, detect vulnerabilities, and assess system configurations.

Nmap Scan Provider conducts network reconnaissance by scanning IP addresses and port ranges. It identifies open ports, running services, service versions, and operating system details. The provider returns structured JSON results showing discovered hosts, accessible services, and potential security concerns.

OS Patch Provider evaluates system patch status and identifies missing security updates. It checks installed software versions against known vulnerabilities and recommends necessary patches to maintain system security.

Scan providers work with the Scan object to execute operations against target assets. Results are captured in both raw and structured formats, enabling detailed

analysis through the web interface.

Model Providers

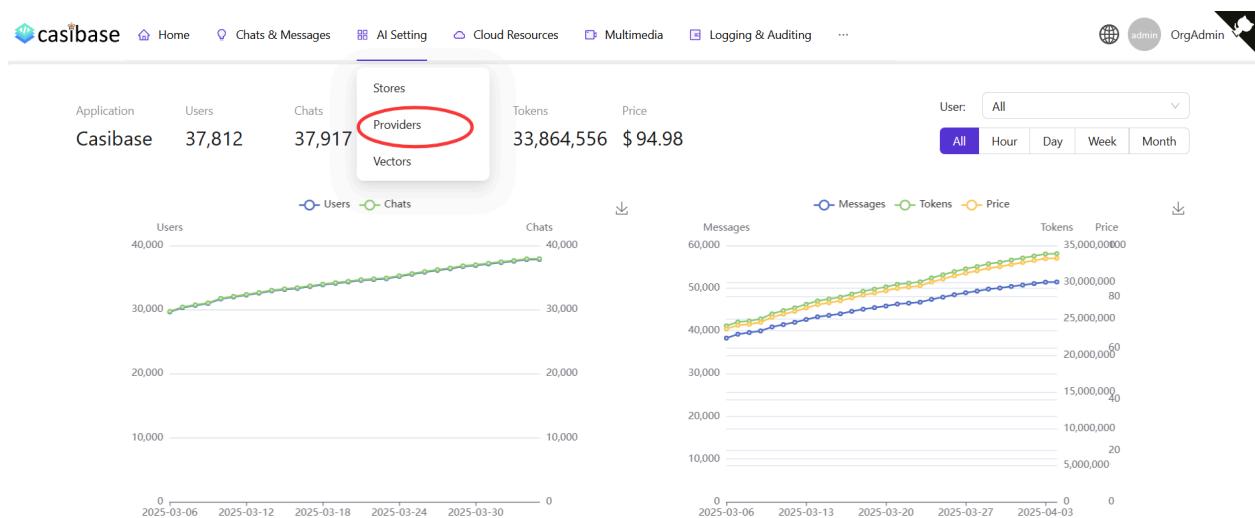
Introduction

Model providers connect AI capabilities to your Casibase instance. They power everything from chatbots to document analysis by integrating large language models (LLMs) directly into your knowledge base workflows.

Add a Model Provider

Adding a model provider takes just a few clicks:

Click the **Providers** button on the home page.



Click the **Add** button to add a model provider.

Providers	Add	Add Storage Provider						
Name		Display name	Category	Type	Sub type	API key	Secret key	Region
provider_tts_alibabacloud_cosyvoice		Provider TTS AlibabaCloud Cosyvoice	Text-to-Speech	Alibaba Cloud	cosyvoice-v1		***	
provider_blockchain_chainmaker		Provider Blockchain ChainMaker	Blockchain	Tencent ChainMaker (Demo Network)	text-davinci-003	AKIDObZHjqUrxnNltkKdfQvAWcS1JK4XidG	***	ap-beijing
provider_model_alibabacloud_deepseek_r1		Provider Model AlibabaCloud DeepSeek-R1	Model	Alibaba Cloud	deepseek-r1		***	
provider_cloud_alibabacloud		Provider Cloud AlibabaCloud	Public Cloud	Aliyun	text-davinci-003	LTA15thKUsopAioN6xi2LMg	***	cn-beijing
dall-e-3		dall-e-3	Model	OpenAI	dall-e-3		***	
provider_model_azure_gpt4_1		Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_model_azure_gpt4_2		Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_model_azure_gpt4_3		Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_embedding_openai_v3		Provider Embedding OpenAI V3	Embedding	OpenAI	text-embedding-ada-002		***	
provider_model_openai_gpt4_vision		Provider Model OpenAI GPT4 Vision	Model	OpenAI	gpt-4-vision-preview		***	

Fill in Model Provider Details

Fill in the model provider details and click the **Save & Exit** button.

[Home](#)[Chat](#)[Stores](#)[Providers](#)[Vectors](#)[Chats](#)[Me](#)[Edit Provider](#)[Save](#)

Name:

provider_openai_model

Display name:

OpenAI model

Category:

Model

Type:

OpenAI

Sub type:

text-davinci-003

Secret key:

Provider URL:

<https://platform.openai.com/account/api-keys>[Save](#)

Supported Providers

Provider	Models & Features
OpenAI	GPT-3.5-turbo, GPT-4, o1 series. Reasoning models show step-by-step thinking. Web search available for real-time information.
Azure OpenAI	Access OpenAI models through Azure infrastructure with official SDK support.
Alibaba Cloud	Qwen models and DeepSeek (v3, v3.1, v3.2, R1). Web search with inline results.
DeepSeek	DeepSeek V3.2 with chat and reasoner variants for advanced reasoning. Updated pricing for improved affordability.
Claude	Anthropic's Claude family including the new Opus 4.5 for complex tasks, plus Sonnet and Haiku variants for balanced performance and speed.
Moonshot (Kimi)	Moonshot v1 series (8k, 32k, 128k context) and Kimi K2 models with preview, turbo, and thinking variants. Auto-tier pricing with kimi-latest.
Hugging Face	Open-source models including Llama-2, ChatGLM2, Baichuan2, and thousands more.
OpenRouter	Unified API access to Claude, PaLM, GPT-4, and other

Provider	Models & Features
	providers.
Ollama	Run models like Llama, Mistral, and Phi locally on your machine. No API key needed.

⚠ CONFIGURATION NOTES

When adding a provider, configure these fields:

- **Category:** Choose **Model** for language models or **Embedding** for text vectorization
- **Type:** Select the specific provider (OpenAI, Hugging Face, Ollama, etc.)
- **API key:** Your API key or access token, required for cloud providers like OpenAI, Azure OpenAI, DeepSeek, not needed for Ollama (runs locally without authentication)

Example: Adding OpenAI

[Home](#)[Chat](#)[Stores](#)[Providers](#)[Vectors](#)[Chats](#)[Messages](#)[Edit Provider](#)[Save](#)

Name: provider_openai_model

Display name: OpenAI model

Category: Model

Type: OpenAI

Sub type: OpenAI

Secret key: Hugging Face

OpenRouter

Ernie

Provider URL: <https://platform.openai.com/account/api-keys>

[Save](#)

STREAMING SUPPORT

Not all models support streaming output (showing responses as they're generated). GPT-3.5-turbo and most modern models do, but if you're using older or custom models, check with your provider.

Once configured, your model provider powers Casibase's chat interface, document Q&A, and other AI features. You can switch between providers anytime to compare results or optimize for cost and performance.



casbin

Home	Chat	Stores	Providers	Vectors	Chats	Messages	Tasks	Resources	Permissions	Logs
Providers										
Name	Display name	Category	Type	Sub type	API key	Secret key	Provider URL	Action		
provider_openai_model	OpenAI model	Model	OpenAI	text-davinci-003	***		https://platform.openai.com/account/api-keys	Edit	Delete	

1 >

Embedding Providers

Introduction

Embedding is a technique used to represent words and documents as vectors.

Embedding providers allow you to analyze and process data within your knowledge base system, making it more intelligent and efficient.

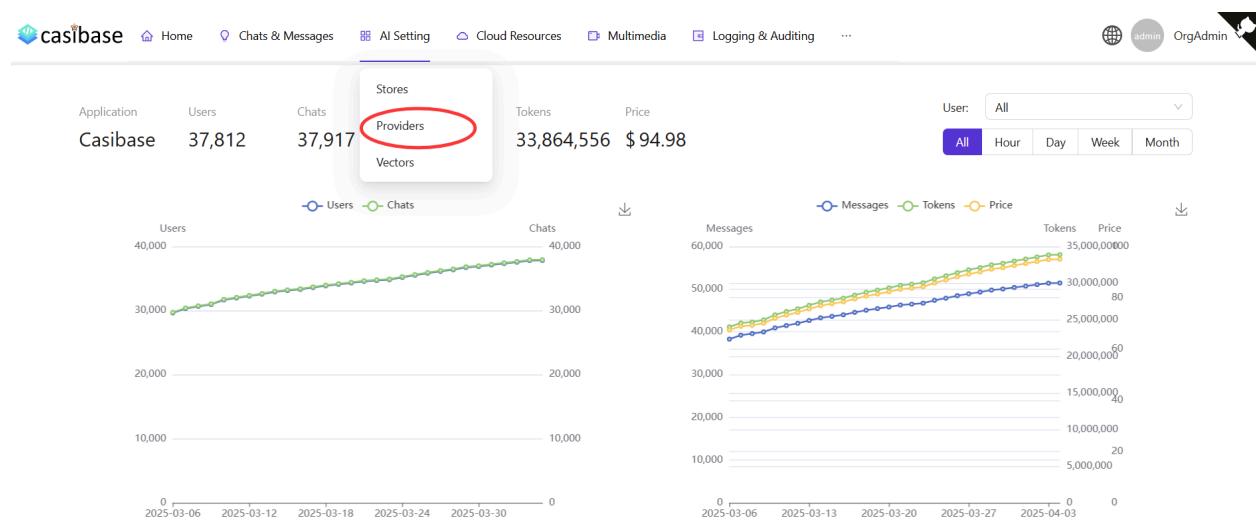
Refer to the [Core Concepts](#) section of our previous documentation for more information about embedding.

In Casibase, you can add an embedding provider by following these steps:

Add a New Embedding Provider

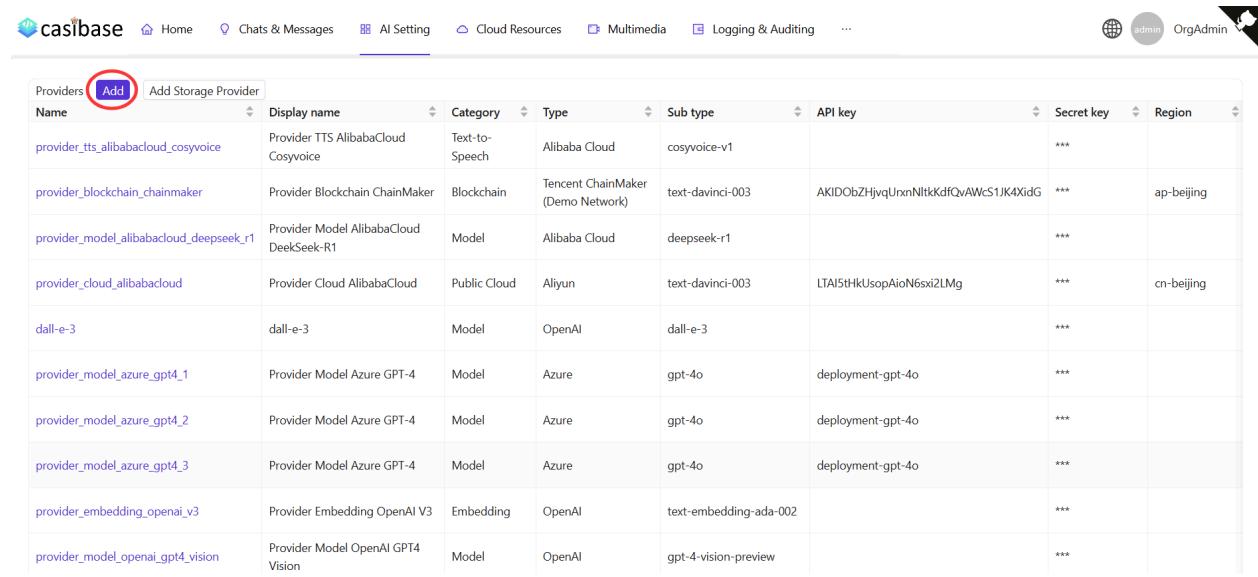
Embedding providers are used to integrate embedding into Casibase. You can add them by following these steps:

Click the **Providers** button on the page.



Add an Embedding Provider

Click the **Add** button to add an embedding provider.



The screenshot shows a table of embedding providers. The 'Add' button in the top-left corner of the table header is highlighted with a red circle. The table has columns for Name, Display name, Category, Type, Sub type, API key, Secret key, and Region. The data rows include various providers like AlibabaCloud Cosyvoice, Blockchain ChainMaker, Model AlibabaCloud Deepseek-R1, and several Azure and OpenAI models.

Providers	Add	Add Storage Provider	Name	Display name	Category	Type	Sub type	API key	Secret key	Region
provider_tts_alibabacloud_cosyvoice	Provider TTS AlibabaCloud Cosyvoice	Text-to-Speech	Alibaba Cloud	cosyvoice-v1				***		
provider_blockchain_chainmaker	Provider Blockchain ChainMaker	Blockchain	Tencent ChainMaker (Demo Network)	text-davinci-003			AKIDObZHjqUrxnNltkKdfQvAWcS1JK4XidG	***	ap-beijing	
provider_model_alibabacloud_deepseek_r1	Provider Model AlibabaCloud DeepSeek-R1	Model	Alibaba Cloud	deepseek-r1				***		
provider_cloud_alibabacloud	Provider Cloud AlibabaCloud	Public Cloud	Aliyun	text-davinci-003			LTAI5tHkUsopAioN6xi2LMg	***	cn-beijing	
dall-e-3	dall-e-3	Model	OpenAI	dall-e-3				***		
provider_model_azure_gpt4_1	Provider Model Azure GPT-4	Model	Azure	gpt-4o			deployment-gpt-4o	***		
provider_model_azure_gpt4_2	Provider Model Azure GPT-4	Model	Azure	gpt-4o			deployment-gpt-4o	***		
provider_model_azure_gpt4_3	Provider Model Azure GPT-4	Model	Azure	gpt-4o			deployment-gpt-4o	***		
provider_embedding_openai_v3	Provider Embedding OpenAI V3	Embedding	OpenAI	text-embedding-ada-002				***		
provider_model_openai_gpt4_vision	Provider Model OpenAI GPT4 Vision	Model	OpenAI	gpt-4-vision-preview				***		

Fill in Embedding Provider Details

Fill in the embedding provider details and click the **Save & Exit** button.

[Home](#)[Chat](#)[Stores](#)[Providers](#)[Vectors](#)[CI](#)[Edit Provider](#)[Save](#)

Name:

embedding_openai_adasimilarity

Display name:

Embedding_OpenAI_AdaSimilarity

Category:

Embedding

Type:

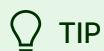
OpenAI

Sub type:

AdaSimilarity

Secret key:

Provider URL:

<https://platform.openai.com/account/api-keys>[Save](#)

Casibase supports many embedding providers, including:

- [OpenAI](#)

- AdaSimilarity
- DavinciSimilarity
- AdaEmbedding2
-
- Hugging Face
 - sentence-transformers/paraphrase-MiniLM-L6-v2
 -

Return providers list page:

Name	Display name	Category	Type	Sub type	API key	Secret key	Provider URL	Action
embedding_openai_adasimilarity	Embedding_OpenAI_AdaSimilarity	Embedding	OpenAI	1		***	https://platform.openai.com/account/api-keys	<button>Edit</button> <button>Delete</button>
model_openai_text_davinci_003	Model OpenAI text-davinci-003	Model	OpenAI	text-davinci-003		***	https://platform.openai.com/account/api-keys	<button>Edit</button> <button>Delete</button>

Now, you can use the embedding provider to convert text to vectors.

After adding an embedding provider, you can use it to retrieve similar documents in Casibase. For more information, please refer to the [Core Concepts](#) section of our previous documentation.

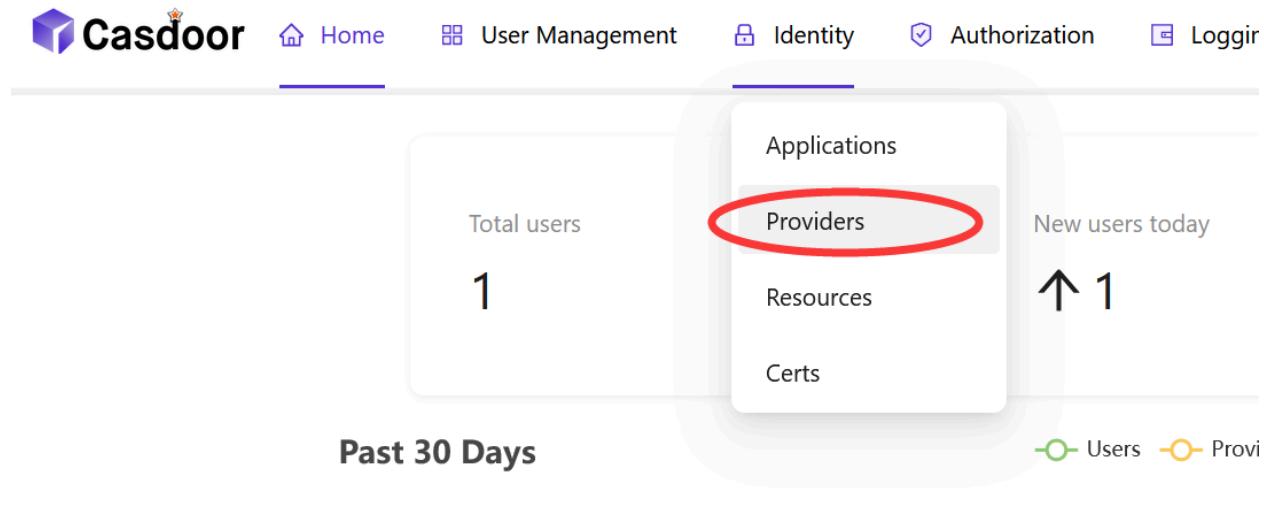
Storage Providers

Introduction

Adding a storage provider to Casibase enables you to efficiently manage and store data, making it an essential component for your knowledge base system.

Add a New Storage Provider

Storage providers are used to store data. They can be added in Casdoor by clicking the `Identity - Providers` button on the home page.



Click the `Add` button to add a storage provider.

[Home](#)[User Management](#)[Identity](#)

Providers

[Add](#)

Name



Organization



Created time



Di

provider_captcha_default

admin (Shared)

2023-09-10 19:31:50

Ca

Fill in the storage provider information

Fill in the storage provider information and click the [Save & Exit](#) button.

New Provider

Save

Save & Exit

Cancel

Name ②:

provider_storage_1

Display name ②:

Provider_storage_1

Organization ②:

admin (Shared)

Category ②:

Storage

Type ②:

aws AWS S3

Client ID ②:

Alibaba Cloud OSS

aws AWS S3

Client secret ②:

Azure Blob

Endpoint ②:

Google Cloud Storage

Endpoint (Intranet) ②:

Local File System

:

MinIO

Bucket ②:

Qiniu Cloud Kodo

 TIP

Casdoor supports many storage providers, including:

- [AWS S3](#)
- [Azure Blob](#)
- [Google Cloud Storage](#)
- [MinIO](#)

- [Qiniu Cloud Kodo](#)
- [Alibaba Cloud OSS ...](#)

Example

Add an Aliyun OSS storage provider

⚠ CAUTION

- Client ID: The AccessKey ID of your Aliyun OSS account.
- Client Secret: The AccessKey Secret of your Aliyun OSS account.

Category [?](#) : Storage

Type [?](#) :  Alibaba Cloud OSS

Client ID [?](#) : LTA***NLf

Client secret [?](#) : Vo6***pi8

Endpoint [?](#) :  oss-cn-beijing.aliyuncs.com

Endpoint (Intranet) [?](#) :

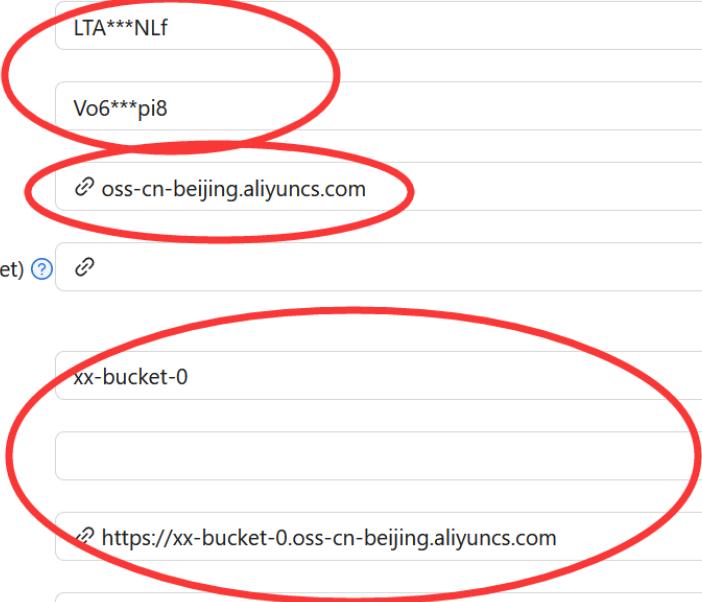
Bucket [?](#) : xx-bucket-0

Path prefix [?](#) :

Domain [?](#) :  https://xx-bucket-0.oss-cn-beijing.aliyuncs.com

Provider URL [?](#) :  https://github.com/organizations/xxx/settings/applications/1234567

[Save](#) [Save & Exit](#) [Cancel](#)



View the storage provider

After adding the storage provider, you can view the storage provider information.

Name	Organization	Created time	Display name	Category	Type	Client ID	Provider URL	Action
provider_storage_1	admin (Shared)	2023-09-10 21:23:02	Provider_storage_1	Storage			https://github.com/organizations/xx...	Edit Delete

Text-to-Speech Providers

Introduction

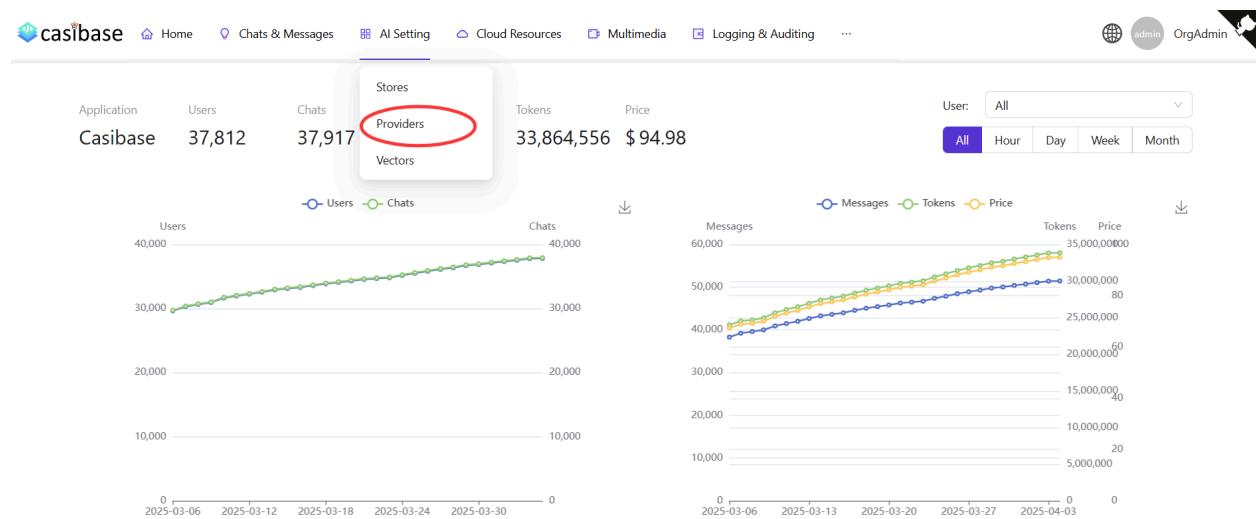
Text-to-Speech (TTS) is a technology that converts text into spoken voice output. TTS providers allow your Casibase applications to communicate with users through synthesized speech, enhancing the user experience and accessibility of your knowledge base system.

In Casibase, integrating a TTS provider enables your AI applications to verbally respond to queries, creating more interactive and engaging user experiences.

Add a New Text-to-Speech Provider

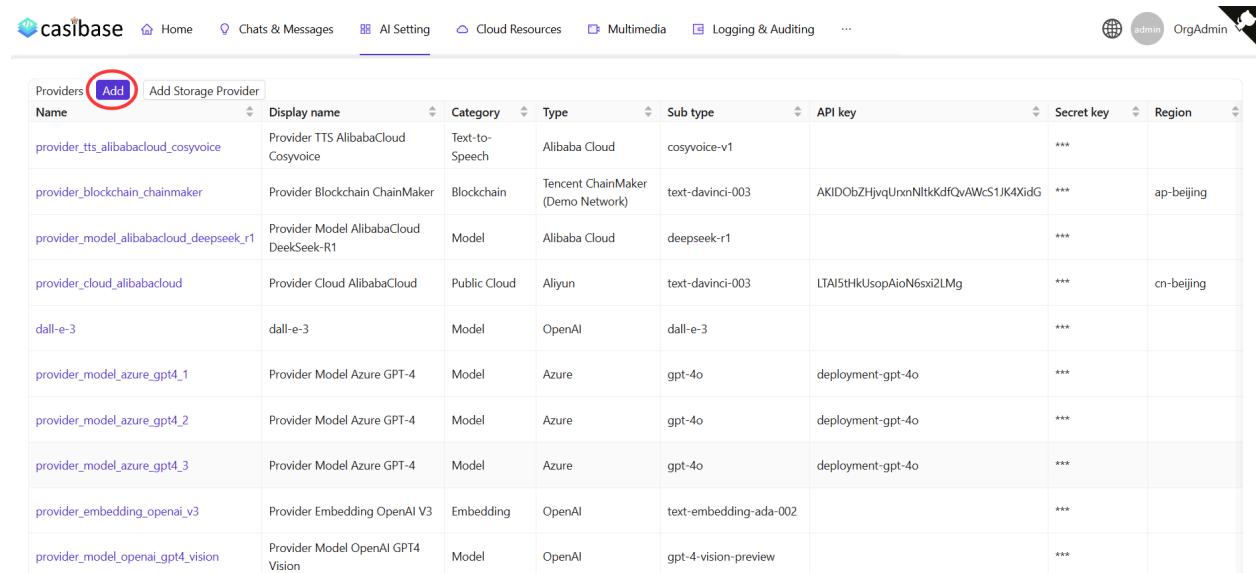
Text-to-Speech providers are used to integrate voice synthesis capabilities into Casibase. You can add them by following these steps:

Click the **Providers** button on the page.



Add a Text-to-Speech Provider

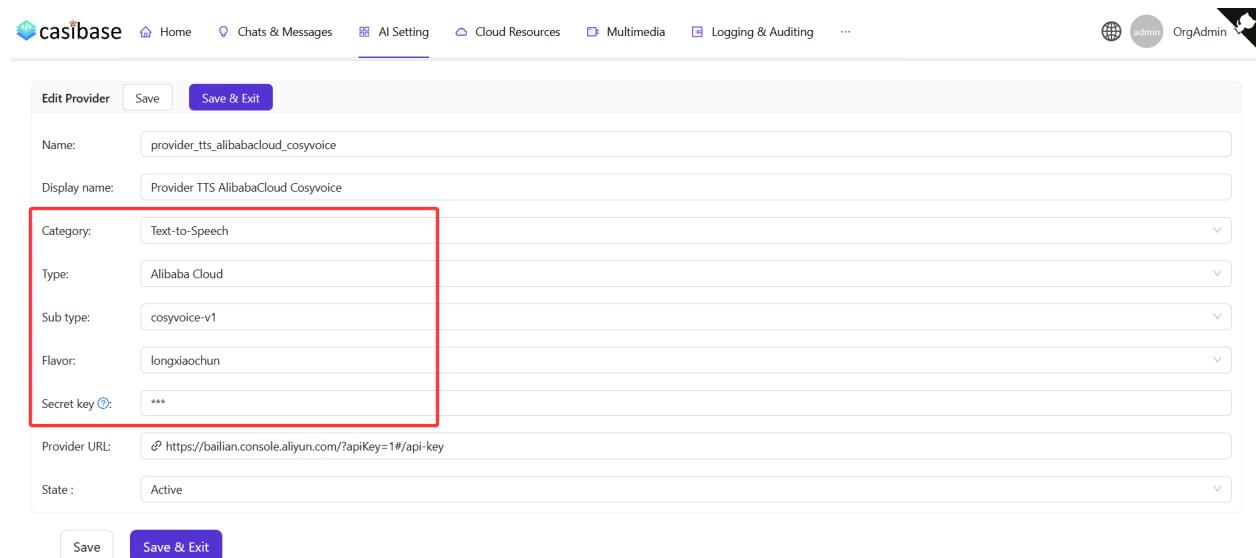
Click the **Add** button to add a Text-to-Speech provider.



Name	Display name	Category	Type	Sub type	API key	Secret key	Region
provider tts alibabacloud cosyvoice	Provider TTS AlibabaCloud Cosyvoice	Text-to-Speech	Alibaba Cloud	cosyvoice-v1		***	
provider blockchain chainmaker	Provider Blockchain ChainMaker	Blockchain	Tencent ChainMaker (Demo Network)	text-davinci-003	AKIDObZHjqUrxnNltkKdfQvAWcS1JK4XidG	***	ap-beijing
provider model alibabacloud deepseek_r1	Provider Model AlibabaCloud DeepSeek-R1	Model	Alibaba Cloud	deepseek-r1		***	
provider cloud alibabacloud	Provider Cloud AlibabaCloud	Public Cloud	Aliyun	text-davinci-003	LTAI5tHkUsopAioN6xi2LMg	***	cn-beijing
dall-e-3	dall-e-3	Model	OpenAI	dall-e-3		***	
provider model azure gpt4_1	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider model azure gpt4_2	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider model azure gpt4_3	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider embedding openai v3	Provider Embedding OpenAI V3	Embedding	OpenAI	text-embedding-ada-002		***	
provider model openai gpt4 vision	Provider Model OpenAI GPT4 Vision	Model	OpenAI	gpt-4-vision-preview		***	

Fill in Text-to-Speech Provider Details

Fill in the embedding provider details and click the **Save & Exit** button.



Edit Provider Save Save & Exit

Name:	provider tts alibabacloud cosyvoice
Display name:	Provider TTS AlibabaCloud Cosyvoice
Category:	Text-to-Speech
Type:	Alibaba Cloud
Sub type:	cosyvoice-v1
Flavor:	longxiaochun
Secret key <small>②</small> :	***
Provider URL:	https://bailian.console.aliyun.com/?apiKey=1#api-key
State :	Active

Save Save & Exit



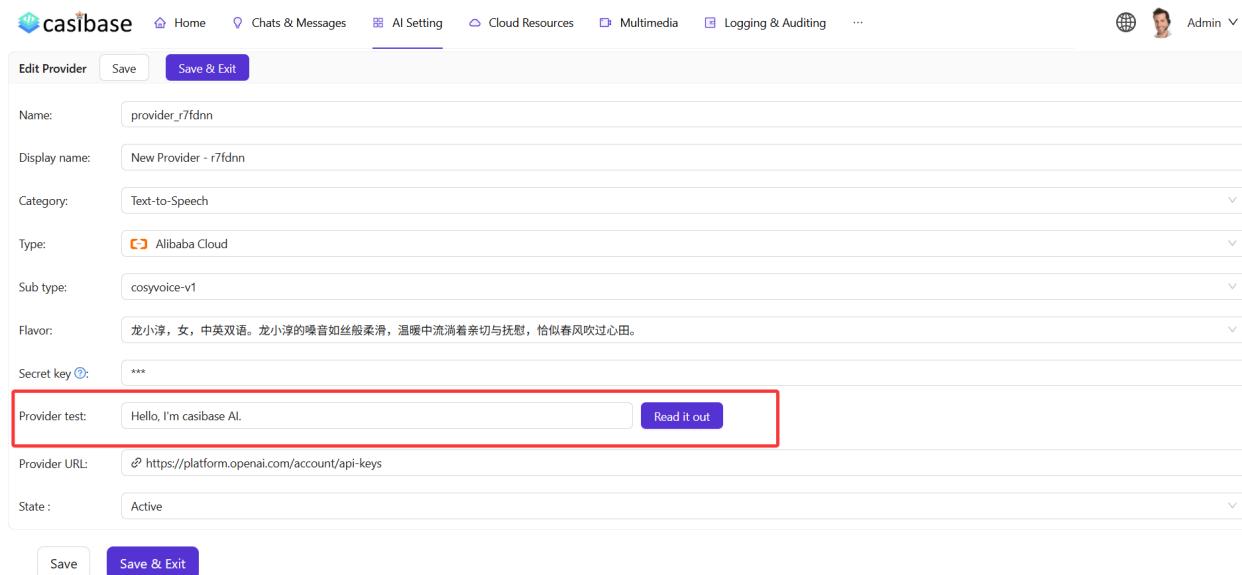
TIP

Casibase currently supports the following Text-to-Speech provider:

- Alibaba Cloud
 - cosyvoice-v1 (with multiple voice options)

Testing Your Text-to-Speech Provider

You can test your TTS provider by clicking the `Read it out` button. This will allow you to enter text and hear the synthesized speech output.



The screenshot shows the Casibase 'Edit Provider' interface. The provider is configured as follows:

- Name: provider_r7fdnn
- Display name: New Provider - r7fdnn
- Category: Text-to-Speech
- Type: Alibaba Cloud
- Sub type: cosyvoice-v1
- Flavor: 龙小淳, 女, 中英双语. 龙小淳的嗓音如丝般柔滑, 温暖中流淌着亲切与抚慰, 怡似春风吹过心田.
- Secret key: ***
- Provider test: Hello, I'm casibase AI. (This field is highlighted with a red box.)
- Read it out (A button next to the provider test input field.)
- Provider URL: <https://platform.openai.com/account/api-keys>
- State: Active

At the bottom, there are 'Save' and 'Save & Exit' buttons.

This testing feature allows you to verify your TTS configuration before implementing it in your applications, ensuring the voice quality and settings meet your requirements.

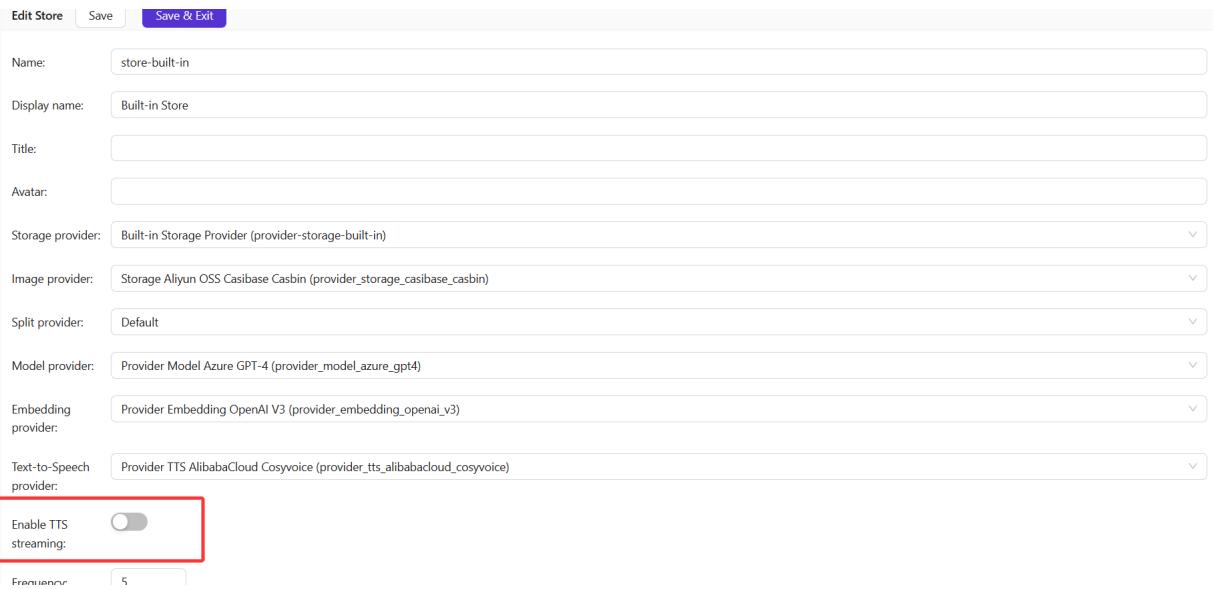
Voice Options for Alibaba Cloud

When using Alibaba Cloud's `cosyvoice-v1`, you can choose from various voice options:

- longwan
- longcheng
-

Using Text-to-Speech in Stores

After adding a Text-to-Speech provider, you can select this provider in your store settings and choose whether to enable TTS streaming.



Edit Store Save Save & Exit

Name: store-built-in

Display name: Built-in Store

Title:

Avatar:

Storage provider: Built-in Storage Provider (provider-storage-built-in)

Image provider: Storage Aliyun OSS Casibase Casbin (provider_storage_casibase_casbin)

Split provider: Default

Model provider: Provider Model Azure GPT-4 (provider_model_azure_gpt4)

Embedding provider: Provider Embedding OpenAI V3 (provider_embedding_openai_v3)

Text-to-Speech provider: Provider TTS AlibabaCloud Cosyvoice (provider_tts_alibabacloud_cosyvoice)

Enable TTS streaming:

Frontend: 5

Now, your store can convert text responses to speech, providing a more interactive experience for users.

Speech-to-Text Providers

Introduction

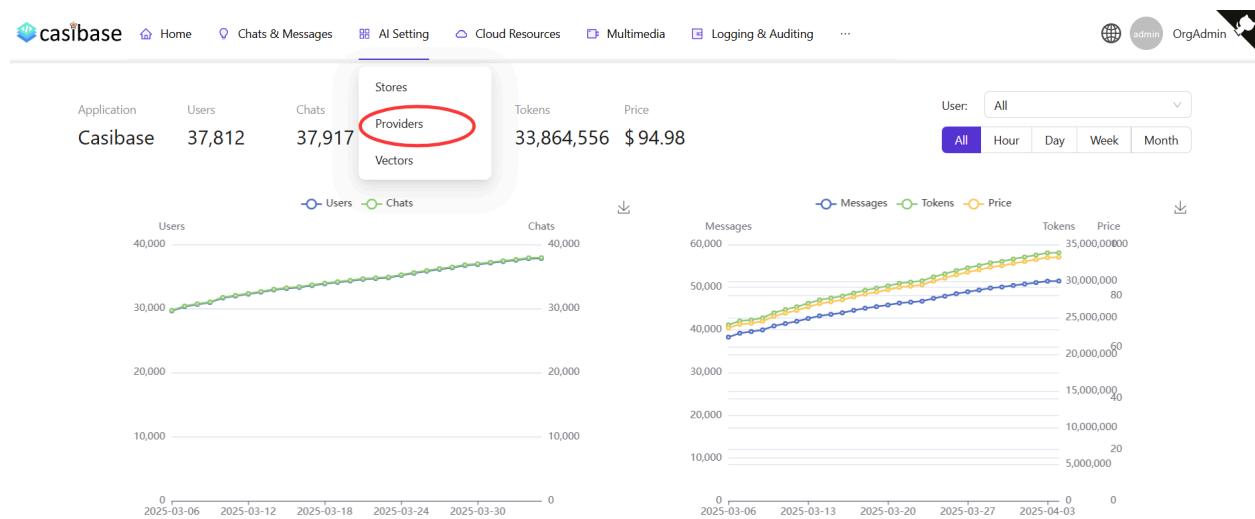
Speech-to-Text (STT) is a technology that converts spoken language into written text. STT providers allow your Casibase applications to understand and process spoken user input, enhancing the user experience and accessibility of your knowledge base system.

In Casibase, integrating an STT provider enables your AI applications to receive and process voice queries, creating more interactive and natural user interactions.

Add a New Speech-to-Text Provider

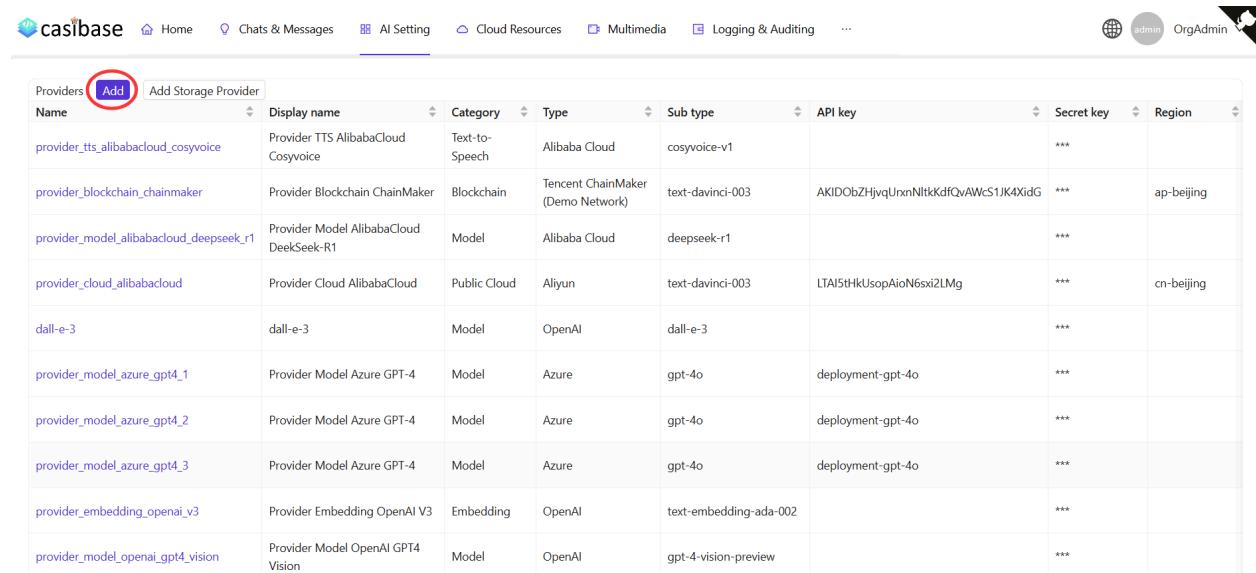
Speech-to-Text providers are used to integrate voice recognition capabilities into Casibase. You can add them by following these steps:

Click the **Providers** button on the page.



Add a Speech-to-Text Provider

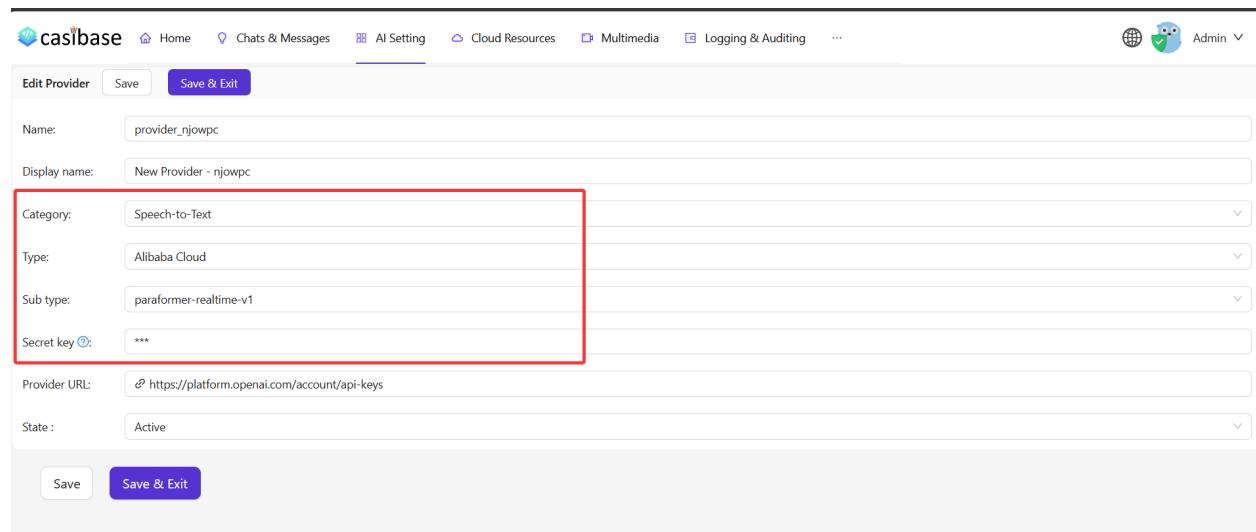
Click the **Add** button to add a Speech-to-Text provider.



Providers	Add	Add Storage Provider					
Name	Display name	Category	Type	Sub type	API key	Secret key	Region
provider_tts_alibabacloud_cosyvoice	Provider TTS AlibabaCloud Cosyvoice	Text-to-Speech	Alibaba Cloud	cosyvoice-v1		***	
provider_blockchain_chainmaker	Provider Blockchain ChainMaker	Blockchain	Tencent ChainMaker (Demo Network)	text-davinci-003	AKIDObZHjqUrxnNltkKdfQvAWcS1JK4XidG	***	ap-beijing
provider_model_alibabacloud_deepseek_r1	Provider Model AlibabaCloud DeepSeek-R1	Model	Alibaba Cloud	deepseek-r1		***	
provider_cloud_alibabacloud	Provider Cloud AlibabaCloud	Public Cloud	Aliyun	text-davinci-003	LTAI5tHkUsopAioN6xi2LMg	***	cn-beijing
dall-e-3	dall-e-3	Model	OpenAI	dall-e-3		***	
provider_model_azure_gpt4_1	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_model_azure_gpt4_2	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_model_azure_gpt4_3	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_embedding_openai_v3	Provider Embedding OpenAI V3	Embedding	OpenAI	text-embedding-ada-002		***	
provider_model_openai_gpt4_vision	Provider Model OpenAI GPT4 Vision	Model	OpenAI	gpt-4-vision-preview		***	

Fill in Speech-to-Text Provider Details

Fill in the speech-to-text provider details and click the **Save & Exit** button.



casibase Home Chats & Messages AI Setting Cloud Resources Multimedia Logging & Auditing ... Admin

Edit Provider

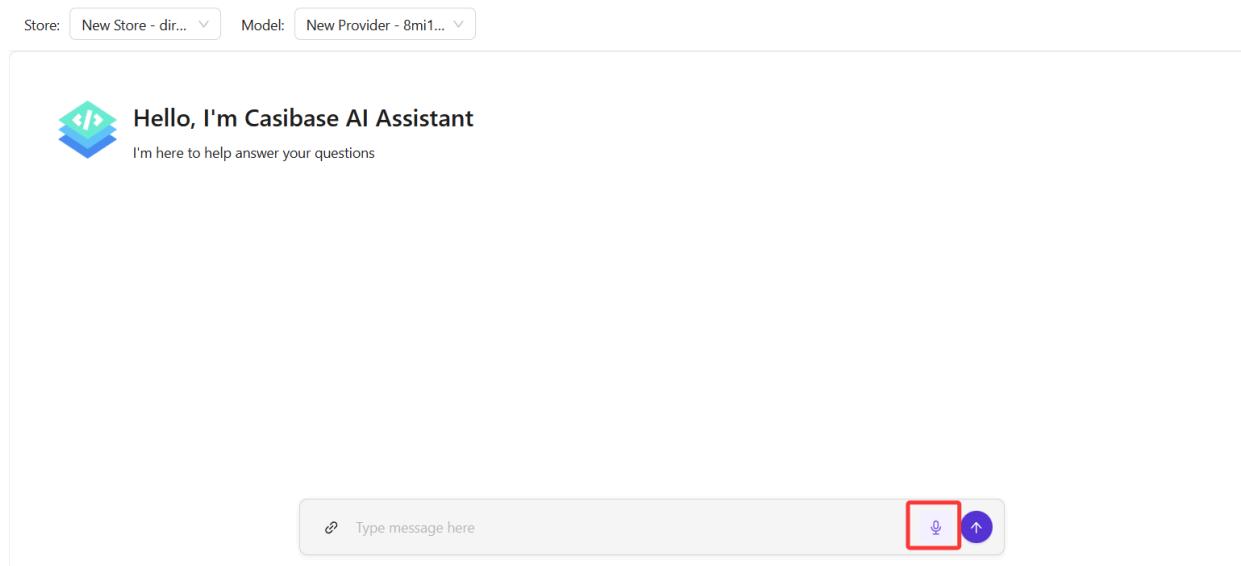
Name:	provider_njowpc
Display name:	New Provider - njowpc
Category:	Speech-to-Text
Type:	Alibaba Cloud
Sub type:	parafomer-realtime-v1
Secret key <small>(?)</small> :	***
Provider URL:	<input type="text" value="https://platform.openai.com/account/api-keys"/>
State :	Active

Using Voice Recognition

When you click the voice recognition button in your Casibase application, the following process occurs:

1. The browser will request permission to access your microphone
2. Once granted, the system will begin listening and automatically convert your speech to text
3. After you finish speaking, the recognized text will be automatically sent as a message

This feature enables hands-free interaction with your Casibase applications, making them more accessible and convenient to use.



Casibase currently supports the following Speech-to-Text provider:

- [Alibaba Cloud](#)
 - [paraformer-realtime-v1](#)

Bot Providers

Introduction

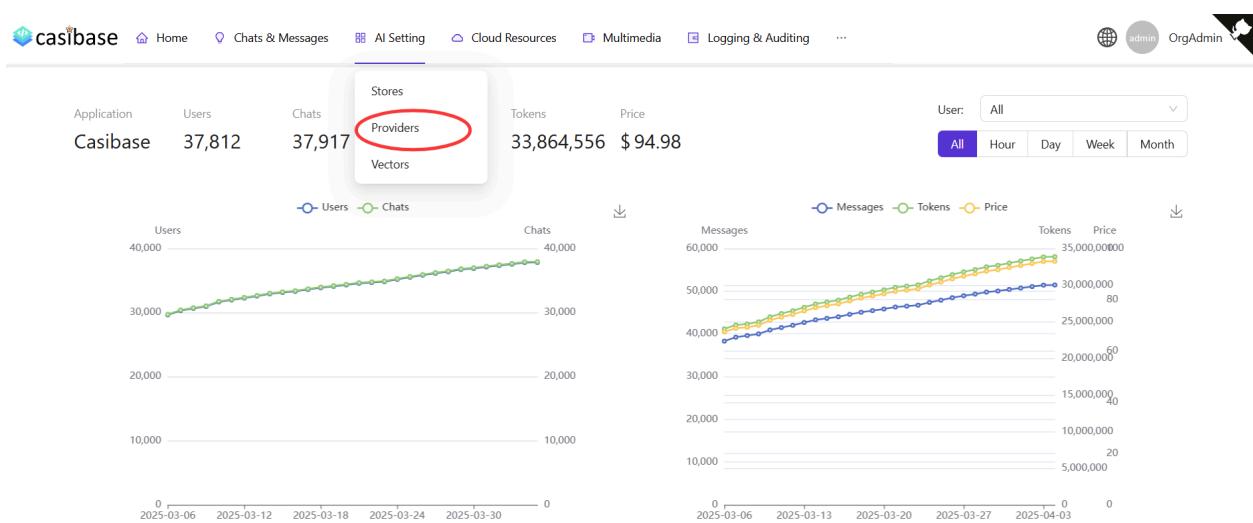
Bot Providers enable Casibase to integrate with enterprise messaging platforms. Through bot interfaces, Casibase can receive messages, process them with AI models, and respond in real-time within workplace communication tools.

These integrations allow teams to access Casibase's AI knowledge base capabilities directly from their familiar messaging applications without switching contexts or opening separate interfaces.

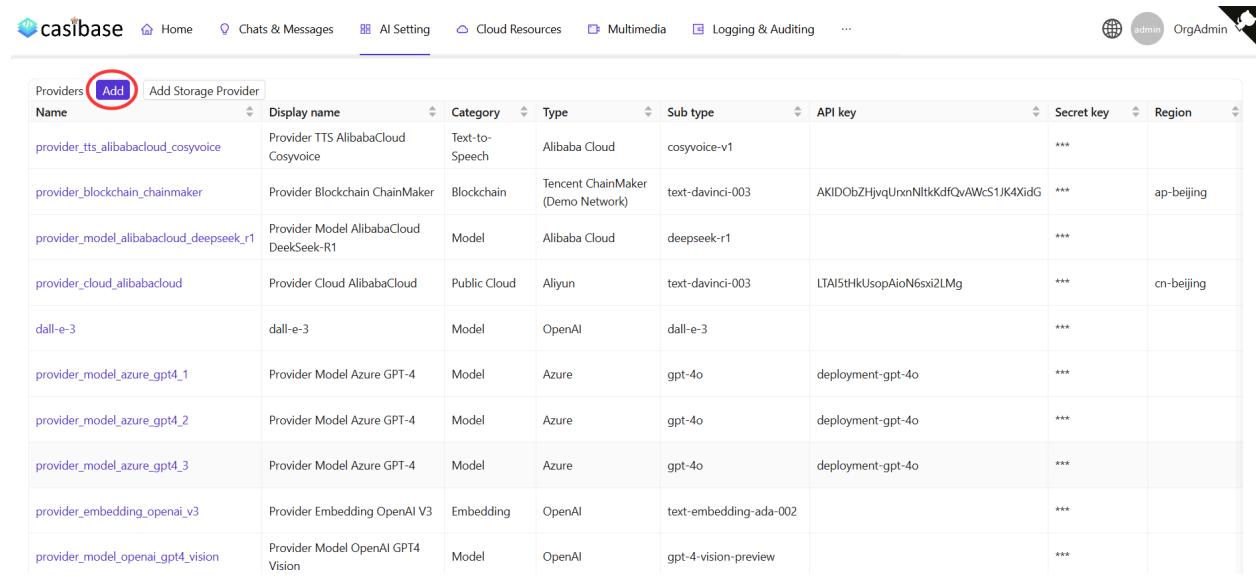
Adding a Bot Provider

Bot providers connect Casibase to enterprise messaging platforms. Follow these steps to configure one:

Click the **Providers** button on the page.



Click the **Add** button to add a new provider.



Providers									Add Storage Provider
Name	Display name	Category	Type	Sub type	API key	Secret key	Region		
provider_tts_alibabacloud_cosyvoice	Provider TTS AlibabaCloud Cosyvoice	Text-to-Speech	Alibaba Cloud	cosyvoice-v1		***			
provider_blockchain_chainmaker	Provider Blockchain ChainMaker	Blockchain	Tencent ChainMaker (Demo Network)	text-davinci-003	AKIDObZHjqUxnNlkKdfQyAWcS1JK4XidG	***	ap-beijing		
provider_model_alibabacloud_deepseek_r1	Provider Model AlibabaCloud DeepSeek-R1	Model	Alibaba Cloud	deepseek-r1		***			
provider_cloud_alibabacloud	Provider Cloud AlibabaCloud	Public Cloud	Aliyun	text-davinci-003	LTAI5thkUsopAioN6xi2LMg	***	cn-beijing		
dall-e-3	dall-e-3	Model	OpenAI	dall-e-3		***			
provider_model_azure_gpt4_1	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***			
provider_model_azure_gpt4_2	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***			
provider_model_azure_gpt4_3	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***			
provider_embedding_openai_v3	Provider Embedding OpenAI V3	Embedding	OpenAI	text-embedding-ada-002		***			
provider_model_openai_gpt4_vision	Provider Model OpenAI GPT4 Vision	Model	OpenAI	gpt-4-vision-preview		***			

Select "Bot" as the category and fill in the provider details, then click **Save & Exit**.

WeCom Bot Provider

The WeCom Bot Provider integrates Casibase with WeCom (WeChat Work), enabling AI-powered responses within WeCom conversations.

Configuration

To configure a WeCom Bot Provider, you need three credentials from the WeCom Admin Console:

- **Bot ID:** The unique identifier for your WeCom bot (also called `aibotid`). Found in the bot application details page.
- **Token:** The verification token you set when configuring the bot's callback settings in WeCom.

- **AES Key:** The encoding AES key (EncodingAESKey) you set for message encryption/decryption in the callback configuration.

How It Works

The WeCom Bot Provider processes messages through the following flow:

1. WeCom sends encrypted messages to the callback URL
2. Casibase decrypts the message using the configured AES key and token
3. The text content is processed by the default store's AI model
4. Casibase generates a response and encrypts it
5. The encrypted response is sent back to WeCom

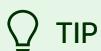
The provider supports streaming responses, which provide better user experience by showing incremental progress as the AI generates the answer.

Callback URL

After creating the provider, configure the callback URL in WeCom's bot settings:

```
https://your-casibase-domain/api/wecom-bot/callback/{botId}
```

Replace `{botId}` with your actual Bot ID. WeCom will verify this URL before activating the bot.



Casibase currently supports the following Bot provider:

- WeCom (WeChat Work)
 - WeCom Bot

Model Providers

Introduction

Model Providers enable AI capabilities in Casibase by integrating various large language models (LLMs) and AI services. These providers allow you to chat with AI, analyze documents, generate embeddings, and perform other intelligent tasks.

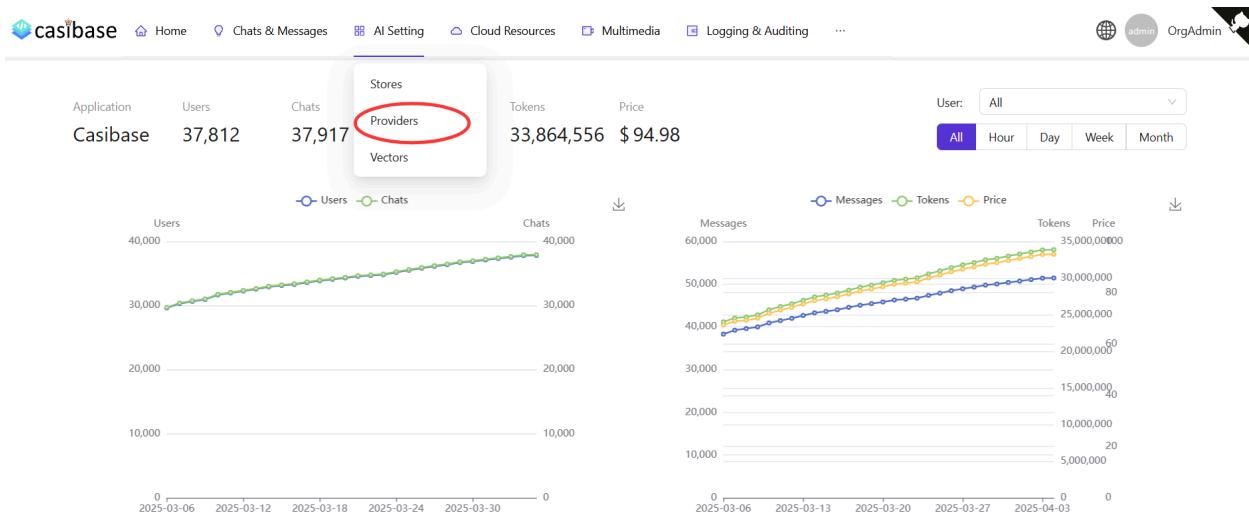
Casibase supports a wide range of model providers, from major cloud services like OpenAI and Azure OpenAI to local and custom model deployments. This flexibility lets you choose the right model for your use case, whether you prioritize performance, cost, privacy, or specific capabilities.

Refer to the [Core Concepts](#) section for more information about providers in general.

Add a New Model Provider

Model providers are used to integrate LLM capabilities into Casibase. You can add them by following these steps:

Click the **Providers** button on the page.



Add a Model Provider

Click the **Add** button to add a new model provider.

The table lists existing model providers and includes an **Add** button in the header.

Name	Display name	Category	Type	Sub type	API key	Secret key	Region
provider tts_alibabacloud_cosyvoice	Provider TTS AlibabaCloud Cosyvoice	Text-to-Speech	Alibaba Cloud	cosyvoice-v1		***	
provider_blockchain_chainmaker	Provider Blockchain ChainMaker	Blockchain	Tencent ChainMaker (Demo Network)	text-davinci-003	AKIDObZHjqUrxnNltkKdIQvAWcS1JK4XidG	***	ap-beijing
provider_model_alibabacloud_deepseek_r1	Provider Model AlibabaCloud DeepSeek-R1	Model	Alibaba Cloud	deepseek-r1		***	
provider_cloud_alibabacloud	Provider Cloud AlibabaCloud	Public Cloud	Aliyun	text-davinci-003	LTAI5tHkUsopAioN6xi2LMg	***	cn-beijing
dall-e-3	dall-e-3	Model	OpenAI	dall-e-3		***	
provider_model_azure_gpt4_1	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_model_azure_gpt4_2	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_model_azure_gpt4_3	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_embedding_openai_v3	Provider Embedding OpenAI V3	Embedding	OpenAI	text-embedding-ada-002		***	
provider_model_openai_gpt4_vision	Provider Model OpenAI GPT4 Vision	Model	OpenAI	gpt-4-vision-preview		***	

Fill in Model Provider Details

Fill in the required configuration, setting the Category to "Model" and selecting the appropriate Type. Then click the **Save & Exit** button to save.

More information about specific model provider types can be found below:

Local Model Provider

Configure local and custom model providers with OpenAI-compatible APIs

Supported Providers

Provider	Description
OpenAI	GPT-3.5, GPT-4, o1 series with web search and step-by-step reasoning displays
Azure OpenAI	OpenAI models through Azure infrastructure with official SDK integration
Alibaba Cloud	Qwen models and DeepSeek (v3, v3.1, v3.2, R1) with web search support
DeepSeek	DeepSeek V3.2 with chat and reasoner models featuring updated pricing
Claude	Anthropic's Claude models including Opus 4.5, Sonnet 4, and Haiku variants
Moonshot (Kimi)	Moonshot v1 (8k/32k/128k) and Kimi K2 models with thinking capabilities and auto-tier pricing
Hugging	Open-source models hosted on Hugging Face

Provider	Description
Face	
OpenRouter	Unified access to multiple AI models through one API
Ollama	Run models like Llama, Mistral, and Phi locally without API keys
Local	Self-hosted models or custom providers with OpenAI-compatible APIs

After adding a model provider, you can use it for chatting, document analysis, question answering, and other AI-powered features in Casibase.

Local Model Provider

The Local model provider in Casibase enables you to connect to self-hosted/local models or any custom model provider that implements an OpenAI Chat Completion-style interface. This gives you maximum flexibility to use models not directly supported by Casibase while maintaining full control over your infrastructure and data.

When to Use Local Provider

The Local provider is recommended in the following scenarios:

Custom Provider Integration:

Use the Local provider when you need to connect to a model provider that is not listed in the Casibase provider dropdown, but exposes an OpenAI-compatible Chat Completion API.

Many modern LLM services implement this standard interface for compatibility, making them easy to integrate via the Local provider.

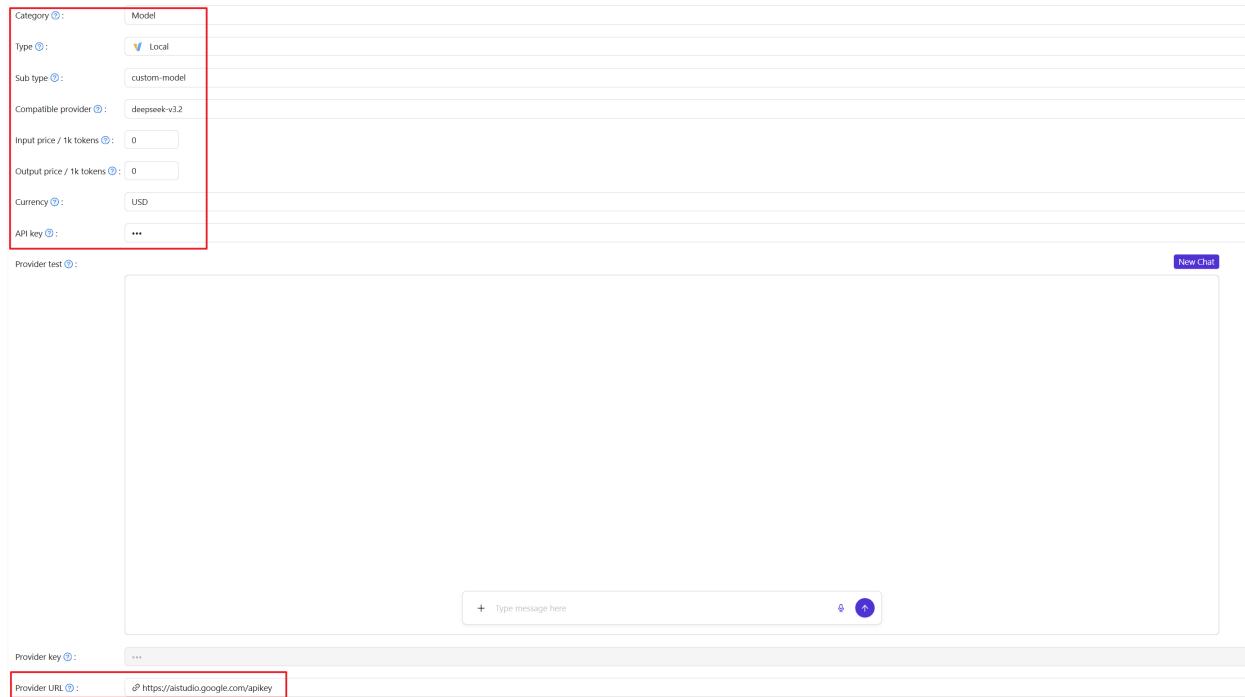
Self-hosted or Local Models:

The Local provider is also suitable when running your own models in local or private environments. This includes popular frameworks such as vLLM, LocalAI, LM Studio, llama.cpp, or any custom deployment that supports the OpenAI Chat Completion format.

By using the Local provider, all data remains within your own infrastructure, making it ideal for sensitive workloads, regulatory compliance, and development or testing environments.

Configuration

The following image shows the provider edit page when configured as a local type:



Required Fields

When adding a Local model provider, configure these essential fields:

Category: Set to `Model` to indicate this is a model provider rather than embedding or other service types.

Type: Select `Local` from the dropdown.

Subtype: This field is automatically set to `custom-model` and cannot be changed. It identifies the provider as a custom implementation.

Compatible provider: Enter the model name you want to use. You can either select from the dropdown suggestions or type any model name directly.

Provider URL: The HTTP(S) endpoint where your model service is running. This is the base URL that Casibase will use to make requests. For example:

- `http://localhost:8000/v1` for a local OpenAI-compatible server
- `http://192.168.1.100:8000/v1` for vLLM
- `https://my-model-service.company.com/api/v1` for a custom deployment
- `https://cloud.infini-ai.com/maas/v1`

The endpoint should implement the `/chat/completions` path that accepts OpenAI format requests.

API key: If your model service requires authentication, provide the API key or token here. Leave empty if your service doesn't require authentication (common for local deployments). The key is sent in the Authorization header as `Bearer <key>`.

Pricing Configuration

These fields help track usage costs when using paid services or for internal billing:

Input price / 1k tokens: Cost per 1,000 input tokens. Enter the numeric value (e.g., `0.001` for \$0.001 per 1k tokens). Set to `0` for free models.

Output price / 1k tokens: Cost per 1,000 output tokens.

Currency: The currency for pricing. This is used for cost tracking and reporting.

Configuration Example

OpenAI-compatible service:

- Compatible provider: Any model you prefer
- Provider URL: Service endpoint URL
- API key: Your service API key
- Input/Output price: According to the service pricing

Using the Provider

After saving your Local model provider, you can use it just like any other provider in Casibase. Select it when creating chats, configuring stores for RAG, or any other feature that requires a model provider.

When the provider is in use, Casibase sends requests to your configured Provider URL using the OpenAI Chat Completion format. Your service should respond with compatible JSON responses.

Troubleshooting

Connection refused: Verify the Provider URL is correct and the service is running. Check firewalls and network connectivity.

Authentication errors: Ensure the API key is correct if your service requires authentication. Some services use different authentication methods - verify your service supports Bearer token authentication.

Unexpected responses: Confirm your service implements the OpenAI Chat

Completion API format correctly. Check the service logs for details about request/response formats.

Model Not Found: If the service returns a “Model Not Found” error, verify that the Compatible provider field in Casibase is configured correctly and that the model has been loaded and is available in the deployment environment.

Scan Providers

Introduction

Scan Providers enable automated network discovery, security auditing, and system assessment in Casibase. These providers integrate specialized scanning tools to analyze infrastructure assets and identify security issues.

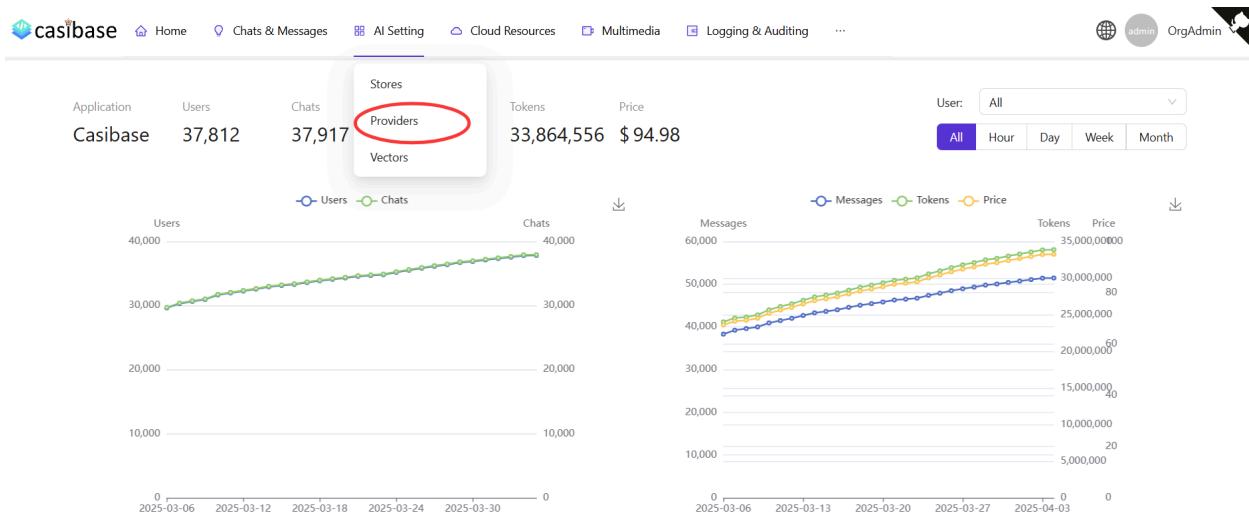
By configuring scan providers, you can perform network reconnaissance, detect open ports and running services, assess patch status, and identify missing security updates across your infrastructure. The results are captured in both raw and structured formats for detailed analysis.

Refer to the [Core Concepts](#) section of our previous documentation for more information about providers.

Add a New Scan Provider

Scan providers are used to integrate network and security scanning capabilities into Casibase. You can add them by following these steps:

Click the **Providers** button on the page.



Add a Scan Provider

Click the **Add** button to add a new scan provider.

The table lists existing providers and has an 'Add' button highlighted with a red circle.

Name	Display name	Category	Type	Sub type	API key	Secret key	Region
provider tts_alibabacloud_cosyvoice	Provider TTS AlibabaCloud Cosyvoice	Text-to-Speech	Alibaba Cloud	cosyvoice-v1		***	
provider_blockchain_chainmaker	Provider Blockchain ChainMaker	Blockchain	Tencent ChainMaker (Demo Network)	text-davinci-003	AKIDObZHjqUxnNltkKdIQvAWcS1JK4XidG	***	ap-beijing
provider_model_alibabacloud_deepseek_r1	Provider Model AlibabaCloud DeepSeek-R1	Model	Alibaba Cloud	deepseek-r1		***	
provider_cloud_alibabacloud	Provider Cloud AlibabaCloud	Public Cloud	Aliyun	text-davinci-003	LTAI5tHkUsopAioN6xi2LMg	***	cn-beijing
dall-e-3	dall-e-3	Model	OpenAI	dall-e-3		***	
provider_model_azure_gpt4_1	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_model_azure_gpt4_2	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_model_azure_gpt4_3	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***	
provider_embedding_openai_v3	Provider Embedding OpenAI V3	Embedding	OpenAI	text-embedding-ada-002		***	
provider_model_openai_gpt4_vision	Provider Model OpenAI GPT4 Vision	Model	OpenAI	gpt-4-vision-preview		***	

Fill in Scan Provider Details

Fill in the required configuration, setting the Category to "Scan" and selecting the appropriate Type. Then click the **Save & Exit** button to save.

More information about each scan provider type can be found below:

Nmap Scan Provider

Network discovery and security auditing with Nmap

OS Patch Scan Provider

System patch assessment and security update detection

Nuclei Scan Provider

Vulnerability scanning with Nuclei

ZAP Scan Provider

Web application security testing with OWASP ZAP

Subfinder Scan Provider

Subdomain discovery with Subfinder

httpx Scan Provider

HTTP service probing with httpx



TIP

Casibase supports multiple scan provider types:

- **Nmap:** Network discovery and security auditing through port scanning, service detection, and OS fingerprinting
- **OS Patch:** System security assessment by checking for missing patches and available updates
- **Nuclei:** Template-based vulnerability scanning for detecting security issues and misconfigurations
- **ZAP:** Web application security testing using OWASP Zed Attack Proxy
- **Subfinder:** Passive subdomain discovery for attack surface mapping
- **httpx:** HTTP service probing for web server enumeration and technology detection

Now you can use scan providers to assess your infrastructure security and maintain visibility into your network assets.

After adding a scan provider, you can create scans to execute network discovery or security assessments against your infrastructure assets, with results stored for historical analysis.

Nmap Scan Provider

The Nmap Scan Provider integrates Network Mapper into Casibase for network discovery and security auditing. It scans target systems to identify open ports, running services, and operating system details.

Configuration

Create an Nmap Scan Provider by navigating to Providers and adding a new provider with Category set to "Scan" and Type set to "Nmap". Configure the display name and owner as needed. The provider operates locally within Casibase and requires no external credentials or API keys.

Network Scanning

Nmap performs comprehensive network analysis by probing target systems. Port discovery identifies which ports are open, closed, or filtered. Service detection determines what software is running on each port and attempts to identify version numbers. OS fingerprinting analyzes network responses to determine the target's operating system and version.

Using the Provider

From the Scans page, create a new scan and select the Nmap provider. Choose your target by selecting an asset from your inventory or entering an IP address manually for ad-hoc scanning.

Command Templates

The provider edit page includes command templates for common scan types.

Select a template from the dropdown to quickly configure your scan:

- Ping Scan: Quick host discovery without port scanning
- Quick Scan: Fast scan of common ports with service detection
- Intense Scan: Comprehensive scan with OS detection and script scanning
- Port Scan: Scan specific port ranges
- OS Detection: Identify target operating system
- Version Detection: Detect service versions on open ports
- Stealth Scan: Lower visibility SYN scan
- UDP Scan: Scan UDP ports instead of TCP

Custom Commands

You can customize the Nmap command to meet specific requirements. The command field supports the `%s` placeholder for the target IP address. For example, `-sV -p 1-1000 %s` scans ports 1-1000 with version detection. The command input validates input to prevent command injection by rejecting shell metacharacters such as `;`, `&`, `|`, `$`, and backticks.

The test widget allows you to verify functionality before running scans. Enter a target, optionally modify the command, click "Scan", and view results immediately.

Scan Results

Nmap returns results as structured JSON containing host information and discovered services. A typical response includes the target IP, host state, and an

array of port details:

```
{  
  "hosts": [  
    {  
      "ip": "192.168.1.1",  
      "state": "up",  
      "ports": [  
        {  
          "port": 22,  
          "state": "open",  
          "service": "ssh",  
          "version": "OpenSSH 8.2"  
        }  
      ]  
    }  
  ]  
}
```

The web interface renders this data in organized tables showing port numbers, states, service names, and detected versions. This structured presentation makes it easy to quickly assess what services are exposed on your network.

Network Considerations

Ensure Casibase has network connectivity to your target systems. Firewalls may need configuration to permit scanning traffic from the Casibase server. Network scanning can trigger intrusion detection systems, so coordinate with your security team before running scans against production infrastructure.

Schedule scans during maintenance windows when possible to minimize any impact on running systems. While Nmap is designed to be non-intrusive, scanning can still generate significant network traffic depending on the scope and scan type configured.

OS Patch Scan Provider

The OS Patch Scan Provider assesses system security by checking for missing patches and available updates. It analyzes installed software versions and compares them against known security vulnerabilities to identify systems requiring updates.

Configuration

Create an OS Patch Scan Provider by navigating to Providers and adding a new provider with Category set to "Scan" and Type set to "OS Patch". Configure the display name and owner. Like other scan providers, it requires no external credentials or API keys.

Patch Assessment

The provider evaluates system patch status by connecting to target systems and querying package managers for update information. It identifies the current patch level, finds missing security updates, and compares installed software versions against available updates. The assessment generates prioritized recommendations for patches that should be installed.

The scan results include three views: **All Patches** (default), showing both available and installed patches in one unified view, **Available Patches** for updates ready to install, and **Installed Patches** for currently applied updates. The All Patches view displays available patches first, making it easy to identify pending updates at a glance.

Running Scans

Execute OS Patch scans against Linux-based virtual machines and servers. Target systems must be accessible from Casibase and have package management tools properly configured.

From the Scans page, create a new scan and select the OS Patch provider. Choose your target asset from the inventory or enter system details manually. The provider edit page also includes a test widget for verifying connectivity and functionality before running production scans.

Remote Scanning

For distributed deployments, Casibase supports remote patch scanning across multiple machines. Deploy Casibase instances on each asset machine with a shared database. When a scan is created for a specific asset (matched by hostname to asset.displayName), the Casibase instance on that machine automatically detects and executes the scan. Results are saved to the shared database, allowing all instances to view scan progress and results in real-time.

Installing Patches

The scan interface includes patch installation capabilities directly from the scan results. Click the install button next to any available patch to initiate installation on the target system. The interface displays installation progress and status, updating in real-time as the patch is applied. Installation requires appropriate permissions on the target system.

Scan Results

Results provide patch assessment data formatted for review. The structured output includes available security updates, identifies package version discrepancies, and provides update priority recommendations. Installation impact analysis helps you understand what changes each update will make to the system.

This information appears in both structured format for quick scanning and raw output for detailed analysis. The web interface organizes the data to highlight critical security patches that require immediate attention versus routine updates.

System Requirements

Target systems should have network connectivity allowing Casibase to reach them. Package manager tools must be installed and functional on the target. The scan queries the package database to retrieve information about installed software and available updates.

Regular scanning helps maintain visibility into your security posture across infrastructure. Compare results over time to track how patch levels change and ensure critical updates are applied promptly. The historical record preserved in Casibase makes it easy to demonstrate compliance with patch management policies.

Nuclei Scan Provider

The Nuclei Scan Provider integrates the Nuclei vulnerability scanner into Casibase for automated security testing. Nuclei uses template-based scanning to detect vulnerabilities, misconfigurations, and exposed services across web applications and infrastructure.

Configuration

Create a Nuclei Scan Provider by navigating to Providers and adding a new provider with Category set to "Scan" and Type set to "Nuclei". The provider operates locally and requires the Nuclei binary to be installed on the Casibase server.

Vulnerability Scanning

Nuclei executes template-based vulnerability checks against target systems. Templates define specific vulnerability signatures, allowing for accurate detection of security issues ranging from common misconfigurations to critical CVEs. The scanner can test web applications, APIs, and network services using its extensive template library.

Using the Provider

From the Scans page, create a new scan and select the Nuclei provider. Enter your target URL or IP address for scanning.

Command Templates

The provider includes command templates for common scanning scenarios:

- Quick Scan: Fast scan with high-severity templates only
- CVE Scan: Focus on known CVE vulnerabilities
- Web Scan: Target web applications with HTTP templates
- All Templates: Comprehensive scan using all available templates

Custom Commands

Customize the Nuclei command to match your requirements. The command field supports the `%s` placeholder for the target. For example, `-u %s -severity critical,high` scans only for critical and high-severity issues. Input validation prevents command injection by blocking dangerous characters.

Use the test widget to verify scanning before running production scans. Enter a target, adjust the command if needed, and review the results.

Scan Results

Nuclei returns structured JSON output containing vulnerability findings. Each result includes:

```
{  
  "template-id": "cve-2021-12345",  
  "info": {  
    "name": "Vulnerability Name",  
    "severity": "high",  
    "description": "Detailed description"  
}
```

The web interface displays findings in organized tables showing template IDs, severity levels, matched locations, and vulnerability details. Color coding by severity helps prioritize remediation efforts.

Security Considerations

Run Nuclei scans only against systems you have authorization to test.

Vulnerability scanning can trigger security alerts and may impact system performance. Coordinate with your security team before scanning production systems.

Keep Nuclei templates updated regularly to detect the latest vulnerabilities. The scanner's effectiveness depends on having current vulnerability signatures in its template database.

ZAP Scan Provider

The ZAP Scan Provider integrates OWASP Zed Attack Proxy into Casibase for automated web application security testing. ZAP actively tests web applications to identify security vulnerabilities including injection flaws, broken authentication, and cross-site scripting.

Configuration

Create a ZAP Scan Provider by navigating to Providers and adding a new provider with Category set to "Scan" and Type set to "ZAP". The provider requires the ZAP daemon to be installed and running on the Casibase server.

Web Application Security Testing

ZAP performs active security testing by sending crafted requests to web applications and analyzing responses for vulnerability indicators. The scanner tests for OWASP Top 10 vulnerabilities and other common security issues. It can spider web applications to discover pages, then systematically test each endpoint for security flaws.

Using the Provider

From the Scans page, create a new scan and select the ZAP provider. Enter the target web application URL for testing.

Command Templates

The provider includes templates for different scanning modes:

- **Quick Scan:** Fast baseline scan with minimal configuration
- **Baseline Scan:** Standard security baseline assessment
- **Full Scan:** Comprehensive active scanning
- **AJAX Spider:** Advanced crawling for JavaScript-heavy applications

Custom Commands

Customize ZAP commands to target specific testing scenarios. The command field accepts the `%s` placeholder for the target URL. For example, `-quickurl %s -j` runs a quick scan with JSON output. Security validation prevents command injection attacks.

Test your configuration using the test widget before running production scans. Enter a target URL, modify the command as needed, and review results immediately.

Scan Results

ZAP returns detailed JSON output containing identified vulnerabilities:

```
{  
  "site": "https://example.com",  
  "alerts": [  
    {  
      "pluginid": "10021",  
      "alert": "X-Content-Type-Options Header Missing",  
      "severity": "INFO",  
      "uri": "/index.html",  
      "type": "X-Content-Type-Options",  
      "value": "none",  
      "confidence": 100  
    }  
  ]  
}
```

The web interface organizes findings by risk level (High, Medium, Low, Informational) with color-coded indicators. Each alert includes the affected URL, vulnerability description, solution recommendations, and CWE/WASC references for further research.

Security Considerations

Active security testing can modify application data and trigger security controls. Always obtain proper authorization before scanning web applications. Run scans against test environments when possible, and coordinate with application owners for production testing.

ZAP generates significant HTTP traffic during scans. Schedule testing during maintenance windows to avoid impacting users. Configure scan policies appropriately to balance thoroughness with testing time and resource consumption.

Subfinder Scan Provider

The Subfinder Scan Provider integrates subdomain discovery capabilities into Casibase through the Subfinder tool. Subfinder passively discovers subdomains using certificate transparency logs, search engines, and other public sources to map an organization's attack surface.

Configuration

Create a Subfinder Scan Provider by navigating to Providers and adding a new provider with Category set to "Scan" and Type set to "Subfinder". The provider operates locally and requires the Subfinder binary on the Casibase server.

Subdomain Discovery

Subfinder performs passive subdomain enumeration by querying various data sources without directly probing target infrastructure. This approach discovers subdomains quietly through certificate transparency logs, DNS databases, and search engine results. The tool aggregates results from multiple sources to provide comprehensive subdomain mapping.

Using the Provider

From the Scans page, create a new scan and select the Subfinder provider. Enter the target domain name for subdomain discovery.

Command Templates

The provider includes templates for different discovery modes:

- **Basic Scan:** Standard subdomain discovery
- **Silent Mode:** Minimal output for scripting
- **Recursive Scan:** Discover subdomains of found subdomains
- **All Sources:** Query all available data sources
- **Passive Only:** Strictly passive discovery

Custom Commands

Customize Subfinder commands for specific requirements. The command field supports the `%s` placeholder for the domain. For example, `-d %s -all -json` queries all sources with JSON output. Input validation prevents command injection by blocking shell metacharacters.

Use the test widget to verify discovery before running production scans. Enter a domain name, adjust the command if needed, and review discovered subdomains.

Scan Results

Subfinder returns JSON output containing discovered subdomains:

```
{  
  "host": "mail.example.com",  
  "source": "crtsh"  
}
```

The web interface displays results in sortable tables showing each discovered

subdomain and its data source. Color-coded source tags help identify which services provided each finding. Summary statistics show total subdomains discovered and source breakdown.

Reconnaissance Considerations

Subdomain discovery is typically legal since it uses only publicly available information. However, check local regulations and organizational policies before conducting reconnaissance. Discovered subdomains may reveal infrastructure details useful for security assessments or unauthorized access attempts.

Use discovery results responsibly. Subdomain mapping helps organizations understand their external attack surface, but the same information could assist malicious actors. Protect subdomain lists appropriately and use findings to improve security posture.

httpx Scan Provider

The httpx Scan Provider integrates HTTP service probing capabilities into Casibase through the httpx tool. httpx efficiently probes HTTP and HTTPS services to gather detailed information about web servers, technologies, and configurations.

Configuration

Create an httpx Scan Provider by navigating to Providers and adding a new provider with Category set to "Scan" and Type set to "httpx". The provider operates locally and requires the httpx binary on the Casibase server.

HTTP Service Probing

httpx probes HTTP and HTTPS services to extract detailed service information. The tool detects web servers, identifies technologies through fingerprinting, extracts titles and response headers, and determines TLS certificate details. It efficiently handles large-scale probing while respecting rate limits and connection constraints.

Using the Provider

From the Scans page, create a new scan and select the httpx provider. Enter target URLs or hosts for HTTP probing.

Command Templates

The provider includes templates for common probing scenarios:

- Basic Probe: Standard HTTP/HTTPS detection
- Full Scan: Comprehensive information gathering
- Technology Detection: Identify web technologies
- Title Extraction: Extract page titles
- Status Only: Quick service availability check

Custom Commands

Customize `httpx` commands for specific requirements. The command field supports the `%s` placeholder for targets. For example, `-u %s -tech-detect -json` enables technology detection with JSON output. Input validation prevents command injection attacks.

Use the test widget to verify probing before running production scans. Enter target URLs or hosts, adjust the command if needed, and review service information.

Scan Results

`httpx` returns structured JSON output containing service details:

```
{  
  "url": "https://example.com",  
  "status-code": 200,  
  "content-length": 1234,  
  "content-type": "text/html",  
  "encoding": "utf-8",  
  "body": "Hello, world!"}
```

The web interface displays results in organized tables showing URLs, status codes, response sizes, page titles, web servers, and detected technologies. Response time metrics help identify performance characteristics. TLS information displays certificate details for HTTPS services.

Network Considerations

Ensure Casibase has network connectivity to target services. Firewalls may need configuration to permit HTTP/HTTPS probing from the Casibase server. Rate limiting prevents overwhelming target services, but scanning can still generate significant traffic depending on scope.

HTTP probing is generally non-intrusive but may trigger monitoring alerts. Coordinate with service owners before scanning production infrastructure. Schedule scans appropriately to minimize any impact on service performance.

BlockChain Providers

Introduction

Blockchain technology provides an immutable and transparent ledger for data integrity verification. In Casibase, blockchain providers serve as a crucial security layer by uploading data to blockchain networks, ensuring that your knowledge base data cannot be tampered with or altered maliciously.

By leveraging blockchain's decentralized and cryptographic properties, Casibase can guarantee data authenticity and provide audit trails for all data modifications. This is particularly important for organizations that need to maintain data integrity compliance or require verifiable proof of data authenticity.

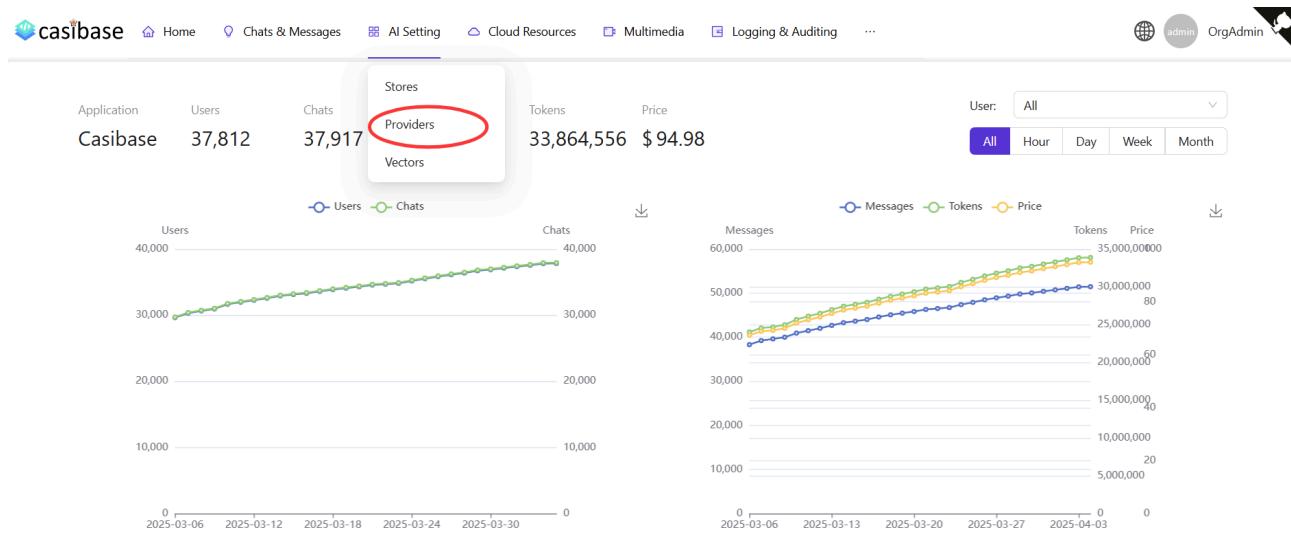
Refer to the [Core Concepts](#) section of our previous documentation for more information about providers.

In Casibase, you can add a blockchain provider by following these steps:

Add a New Blockchain Provider

Blockchain providers are used to integrate blockchain data integrity features into Casibase. You can add them by following these steps:

Click the [Providers](#) button on the page.



Add a Blockchain Provider

Click the [Add](#) button to add a blockchain provider.

Providers	Add	Add Storage Provider						
Name	Display name	Category	Type	Sub type	API key	Secret key	Region	
provider_tts_alibabacloud_cosyvoice	Provider TTS AlibabaCloud Cosyvoice	Text-to-Speech	Alibaba Cloud	cosyvoice-v1		***		
provider_blockchain_chainmaker	Provider Blockchain ChainMaker	Blockchain	Tencent ChainMaker (Demo Network)	text-davinci-003	AKIDObZHjvqUrxnNtkKdfQvAwcS1JK4XidG	***	ap-beijing	
provider_model_alibabacloud_deepseek_r1	Provider Model AlibabaCloud DeepSeek-R1	Model	Alibaba Cloud	deepseek-r1		***		
provider_cloud_alibabacloud	Provider Cloud AlibabaCloud	Public Cloud	Aliyun	text-davinci-003	LTAI5tHkUsopAioN6sx2LMg	***	cn-beijing	
dall-e-3	dall-e-3	Model	OpenAI	dall-e-3		***		
provider_model_azure_gpt4_1	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***		
provider_model_azure_gpt4_2	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***		
provider_model_azure_gpt4_3	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***		
provider_embedding_openai_v3	Provider Embedding OpenAI V3	Embedding	OpenAI	text-embedding-ada-002		***		
provider_model_openai_gpt4_vision	Provider Model OpenAI GPT4 Vision	Model	OpenAI	gpt-4-vision-preview		***		

Fill in Blockchain Provider Details

Fill in the required configuration according to the blockchain you use, then click the **Save & Exit** button to save.

More information about the configuration can be found below:

Chainmaker Configuration

ChainMaker is a high-performance, enterprise-grade blockchain platform developed under the leadership of China. It is designed to provide secure, controllable, and sc...

Ethereum Configuration

Ethereum is a decentralized blockchain platform that enables smart contracts and decentralized applications (dApps). It is one of the most popular blockchain platforms...

Edit Provider		Save	Save & Exit
Name	eth_win_test		
Display name	eth_win		
Category	Blockchain		
Type	Ethereum		
Private key	***		
Contract Address	D4600b1804b4FD07194476C35825175B30F0f9Ec		
Invoke method	save		
Browser URL	http://127.0.0.1:5051/txpage?blocknumber={bh}		
Provider URL	http://192.168.31.234:8545		
Is default	<input checked="" type="checkbox"/>		
State	Active		



TIP

Casibase supports various blockchain networks for data integrity verification, including:

- [ChainMaker](#)
 - [ChainMaker](#)
 - [Tencent Chainmaker](#)
- [Ethereum](#)
 - Private networks (compatible with Ethereum JSON-RPC)
 - [Geth](#)
 - [Ganache](#)
 - Other Ethereum JSON-RPC compatible chains

Now, you can use the blockchain provider to ensure data integrity and prevent tampering.

After adding a blockchain provider, you can use it in Casibase to create immutable data records. The data records will be committed to the blockchain, providing proof of authenticity and preventing any unauthorized tampering.

Chainmaker Configuration

ChainMaker is a high-performance, enterprise-grade blockchain platform developed under the leadership of China. It is designed to provide secure, controllable, and scalable blockchain infrastructure for industries such as finance, government, and supply chain. ChainMaker supports multiple consensus mechanisms, smart contracts, privacy protection, and other features, meeting the needs of large-scale commercial applications. Its open-source, modular architecture allows developers to flexibly customize and extend functionalities according to actual business scenarios.

In this chapter, you will learn how to configure and use ChainMaker, including setting up storage providers and other operations, to help you quickly get started and apply the features of the ChainMaker platform.

1. Configuration field description

When configuring a ChainMaker provider in Casibase, you need to fill in several key fields. Each field has a specific meaning and is required for the correct integration with the ChainMaker blockchain. The following list explains the purpose of each field:

- `Name`: The unique identifier for this blockchain provider.
- `Display name`: The display name shown in the UI for this provider.
- `Category`: The type of service, here it should be `Blockchain`.
- `Type`: The blockchain type, here it should be `ChainMaker`.
- `orgId`: The organization ID in the ChainMaker network.
- `ChainId`: The chain ID of the ChainMaker blockchain.
- `AuthType`: The account mode. Currently, only `permissionedwithcert` is

supported.

- `User key`: The user's private key for authentication in the ChainMaker server.
- `User cert`: The user's certificate for authentication in the ChainMaker server.
- `Sign key`: The user's private key for signing transactions in the ChainMaker server.
- `Sign cert`: The user's certificate for signing transactions in the ChainMaker server.
- `Node address`: The address of the ChainMaker node to connect to.
- `Contract name`: The name of the smart contract to interact with.
- `Invoke method`: The method name to invoke on the contract.
- `Browser URL`: The URL for viewing the blockchain in a browser.
- `Chainmaker endpoint`: The API endpoint for the ChainMaker service. See: <https://github.com/casibase/chainmaker-server>

Please make sure to fill in each field accurately according to your ChainMaker deployment information. This will ensure that Casibase can successfully connect and interact with your ChainMaker blockchain.

Next, we will use "Deploying ChainMaker via the Management Console" as an example for our introduction. If you have not yet deployed ChainMaker, please follow the [ChainMaker documentation](#) for deployment.

2. Configure ChainMaker

2.1 chainId, orgId and authType configuration

Obtaining Blockchain Information from the Web Panel

To retrieve blockchain information such as `chainId`, `orgId`, and `authType`, log in

to the **ChainMaker Management Console** (Web panel). Navigate to the relevant blockchain management section, where these configuration details are displayed. Copy the required values and use them when configuring the ChainMaker provider in Casibase.

ChainMaker Management Console

Blockchain management

Blockchain overview

Contract management

On-chain management

Vote management

Organizational Information

Node information

Blockchain Explorer

Key metrics

Cumulative number of transactions	The latest block height	On-chain nodes	Number of organizations on the chain
23	22	1 pcs	1 pcs

Blockchain information

Blockchain ID	Connect the plugin wallet	Connect to a browser	Chain permission management	Modify the chain configuration	Download the chain configuration
CasibaseChainMaker	ChainId				
CasibaseChainMaker					
v2.3.5(2030500)					
0					
permissionedWithCert	authType				
ONLY					
100 transactions					
600s					
10ms					

Support the creation of v2.3.5 version bottom chain, Subscribe to the v2.0+ version of the bottom chain

Blockchain management

Blockchain overview

Contract management

On-chain management

Vote management

Organizational information

Node information

Blockchain Explorer

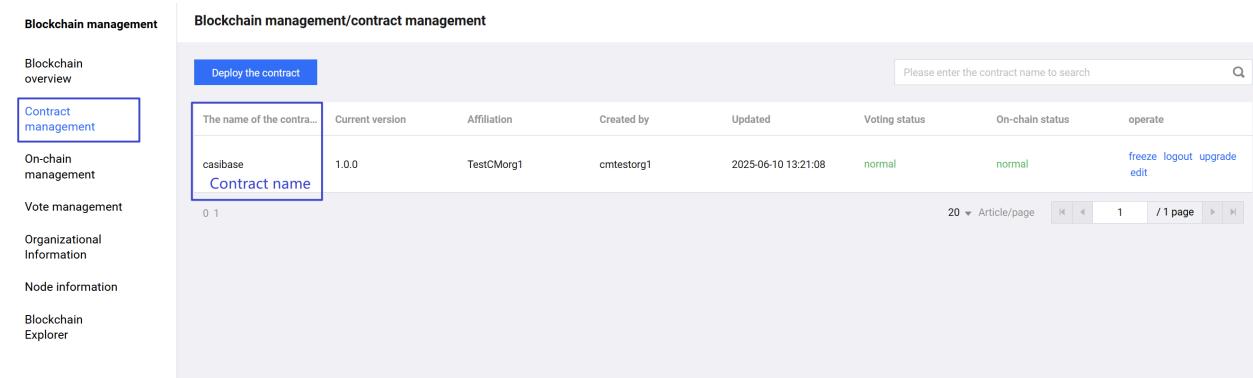
Blockchain management/organization information

Organization ID	The name of the organization	Creation time	Number of nodes
TestCMorg1	cmtestorg1	2025-06-10 10:18:33	1

0 1

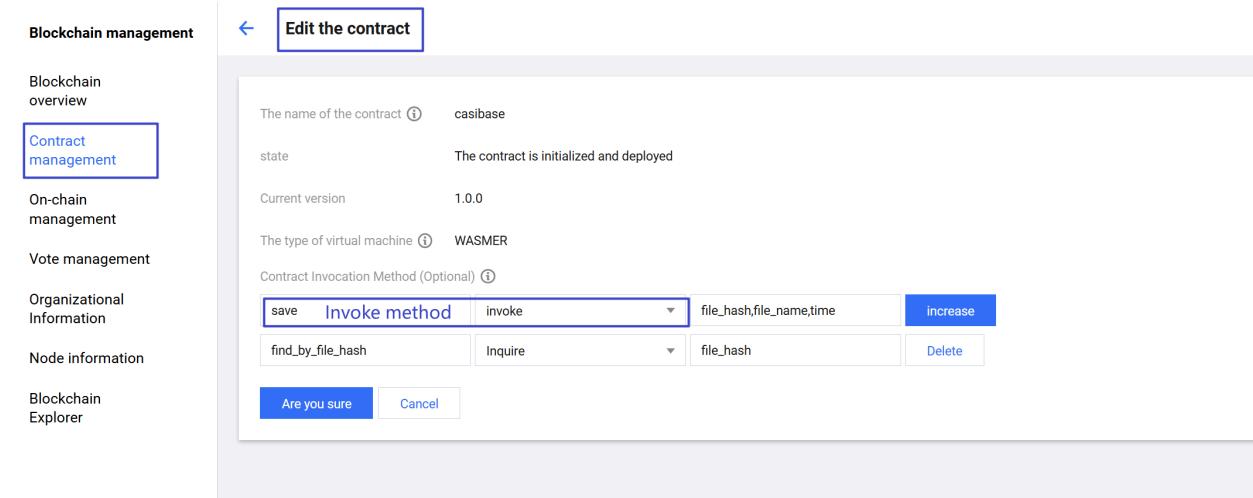
10 Article/page / 1 page

2.2 Contract name and Invoke method configuration



The screenshot shows the 'Blockchain management/contract management' interface. The left sidebar has a 'Contract management' menu item highlighted. The main area displays a table of contracts. One contract, 'casibase', is selected and its details are shown in the table: Current version 1.0.0, Affiliation TestCMorg1, Created by cmtestorg1, Updated 2025-06-10 13:21:08, Voting status normal, On-chain status normal. The 'operate' column for this row contains buttons for 'freeze', 'logout', 'upgrade', and 'edit'. The table has a search bar at the top right and a pagination bar at the bottom right.

Click the edit button in the "operate" column:



The screenshot shows the 'Edit the contract' dialog box. The left sidebar has a 'Contract management' menu item highlighted. The dialog box contains fields for 'The name of the contract' (casibase), 'state' (The contract is initialized and deployed), 'Current version' (1.0.0), and 'The type of virtual machine' (WASMER). Below these, there is a section for 'Contract Invocation Method (Optional)'. It includes a dropdown for 'Invoke method' (set to 'invoke'), a dropdown for 'Contract Invocation Method' (set to 'Inquire'), and a 'Delete' button. At the bottom of the dialog are 'Are you sure' and 'Cancel' buttons.

2.3 key, certificate, and nodeAddr configuration

Blockchain information	
Blockchain ID	CasibaseChainMaker
The name of the blockchain	CasibaseChainMaker
Blockchain version	v2.3.5(2030500)
Configure the version	0
Account Mode	permissionedWithCert
Consensus strategy	ONLY
The maximum size of a block	100transactions
How long the transaction expires	600s
Block generation interval	10ms

Click the "download the chain configuration" button and extract the archive:

2.3.1 node addr

You can find the node_addr configuration in `~\CasibaseChainMaker\ sdk_configs\ sdk_config.yml`.

Alternatively, you can check nodeAddr elsewhere, but note that the displayed port is the p2p port. The one we use should be the rpc port, which is the p2p port + 1000 by default.

Blockchain management

Blockchain overview

Contract management

On-chain management

Vote management

Organizational Information

Node information

Blockchain Explorer

Blockchain management/node information

Please enter the node name to search

Node name	Affiliation	Node type	Node ID	Node address	Ledger synchronization type	operate
cmtestnode1	cmtestorg1	Consensus nodes	Qmdcq5NhATkgqEi7q3Tvx...	39.107.236.48:11301	FULL	View
0 1						

The correct port is the current value plus 1000; i.e. 12301

2.3.2 user certificate and key

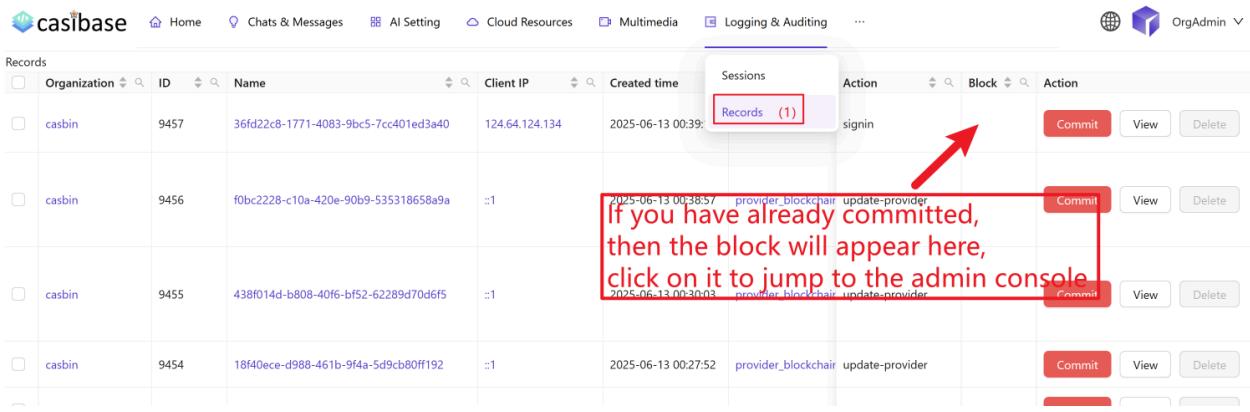
You can find the corresponding user certificate and key in `~\CasibaseChainMaker\ sdk_configs\crypto-config\TestCMorg1\user\cmtestuser1`. Fill in the provider fields accordingly.

cmtestuser1.sign.crt	2025/6/12 23:49	安全证书	1 K
cmtestuser1.sign.key	2025/6/12 23:49	KEY 文件	1 K
cmtestuser1.tls.crt	2025/6/12 23:49	安全证书	1 K
cmtestuser1.tls.key	2025/6/12 23:49	KEY 文件	1 K

2.4 Browser URL and ChainMaker endpoint

The Browser URL refers to the URL of the ChainMaker management console, which allows for quick access from Casibase in the future:

`https://manage.casvisor.com/chains/CasibaseChainMaker/nodes?chainMode=permissionedWithCert` [Browser URL](https://manage.casvisor.com/chains/CasibaseChainMaker/nodes?chainMode=permissionedWithCert)



The screenshot shows a table with columns: Organization, ID, Name, Client IP, Created time, Sessions, Action, Block, and Action. The 'Sessions' and 'Action' columns are highlighted with red boxes. A red arrow points from the 'Sessions' box to the 'Block' column of the first row. A red callout box contains the text: 'If you have already committed, then the block will appear here, click on it to jump to the admin console'.

Organization	ID	Name	Client IP	Created time	Sessions	Action	Block	Action
casbin	9457	36fd22c8-1771-4083-9bc5-7cc401ed3a40	124.64.124.134	2025-06-13 00:39:	signin	Commit	View	Delete
casbin	9456	f0bc2228-c10a-420e-90b9-535318658a9a	⋮1	2025-06-13 00:38:57	provider_blockchain update-provider	Commit	View	Delete
casbin	9455	438f014d-b808-40f6-bf52-62289d70d6f5	⋮1	2025-06-13 00:30:03	provider_blockchain update-provider	Commit	View	Delete
casbin	9454	18f40ece-d988-461b-9f4a-5d9cb80ff192	⋮1	2025-06-13 00:27:52	provider_blockchain update-provider	Commit	View	Delete

The ChainMaker endpoint refers to the IP and port of the ChainMaker server that Casibase needs to connect to, for example: 127.0.0.1:13900

Ethereum Configuration

Ethereum is a decentralized blockchain platform that enables smart contracts and decentralized applications (dApps). It is one of the most popular blockchain platforms, supporting a wide range of applications from DeFi to NFTs. Ethereum uses a proof-of-stake consensus mechanism and provides a robust ecosystem for developers to build and deploy smart contracts.

 INFO

In this chapter, you will learn how to configure and use Ethereum, including setting up blockchain providers and other operations, to help you quickly get started and apply the features of the Ethereum platform.

1. Configuration field description

When configuring an Ethereum provider in Casibase, you need to fill in several key fields. Each field has a specific meaning and is required for the correct integration with the Ethereum blockchain.

Field Descriptions:

- **Name**: The unique identifier for this blockchain provider.
- **Display name**: The display name shown in the UI for this provider.
- **Category**: The type of service, here it should be **Blockchain**.
- **Type**: The blockchain type, here it should be **Ethereum**.
- **Private key**: The private key of the Ethereum account used for signing transactions.
- **Contract Address**: The address of the smart contract to interact with on the

Ethereum blockchain.

- **Invoke method**: The method name to invoke on the smart contract.
- **Browser URL**: The URL for viewing the blockchain in a browser, with block number template support.
 - Format: `http://127.0.0.1:5051/txpage?blocknumber={bh}` where `{bh}` will be replaced with the actual block number when visiting the block.
- **Provider URL**: The JSON-RPC endpoint URL for connecting to the Ethereum network (e.g., Geth, Ganache, or other node).

Please make sure to fill in each field accurately according to your Ethereum deployment information. This will ensure that Casibase can successfully connect and interact with your Ethereum blockchain.

2. Configure Ethereum

Example

Edit Provider		Save	Save & Exit
Name <small>?</small> :	eth_win_test		
Display name <small>?</small> :	eth_win		
Category <small>?</small> :	Blockchain		
Type <small>?</small> :	 Ethereum		
Private key <small>?</small> :	***		
Contract Address <small>?</small> :	D4600b1B04b4FD07194476C35825175B30F0f9Ec		
Invoke method <small>?</small> :	save		
Browser URL <small>?</small> :	 http://127.0.0.1:5051/txpage?blocknumber={bh}		
Provider URL <small>?</small> :	 http://192.168.31.234:8545		
Is default <small>?</small> :	<input checked="" type="checkbox"/>		
State <small>?</small> :	Active		

2.1 Provider URL Configuration

The Provider URL is the JSON-RPC endpoint that Casibase will use to communicate with the Ethereum network. **This is the first and most important configuration** as it establishes the connection to your Ethereum network.

Example Provider URL

```
http://127.0.0.1:8545
```

You can use:

- [geth](#): A popular Ethereum client that provides a JSON-RPC interface.
- [ganache](#): A personal blockchain for Ethereum development that can be used for testing and development purposes.
- [other](#): Ethereum JSON-RPC compatible chains. Any other Ethereum-compatible chain that supports the JSON-RPC interface.

Example: Geth Dev Mode

To quickly experiment with Ethereum using Geth, you can start Geth in developer mode. This mode launches a local Ethereum node with instant mining and pre-funded accounts, making it ideal for testing and development.

```
geth --dev --http --http.api eth,web3,net --http.corsdomain  
"https://remix.ethereum.org"
```

This command starts a local Ethereum node with HTTP JSON-RPC enabled and sets the CORS domain to allow cross-origin requests from <https://remix.ethereum.org>. This configuration is suitable for online contract deployment and interaction using Remix web-based tools. You can use the default account (private key can be found in the `geth` console at first launch) and the endpoint (`http://127.0.0.1:8545`) as your Provider URL in Casibase for immediate testing and development.

```

WARN [07-19|02:42:46.376]      stored on a ramdisk, and will be lost if your machine is restarted.
WARN [07-19|02:42:46.376] 4. Mining is enabled by default. However, the client will only seal blocks if trans-
actions
WARN [07-19|02:42:46.376]      are pending in the mempool. The miner's minimum accepted gas price is 1.
WARN [07-19|02:42:46.376] 5. Networking is disabled; there is no listen-address, the maximum number of peers
is set
WARN [07-19|02:42:46.376]      to 0, and discovery is disabled.
WARN [07-19|02:42:46.376]
WARN [07-19|02:42:46.376]
WARN [07-19|02:42:46.376] Running in ephemeral mode. The following account has been prefunded in the genesis
:
WARN [07-19|02:42:46.376]
WARN [07-19|02:42:46.376]      Account
WARN [07-19|02:42:46.376]      -----
WARN [07-19|02:42:46.376]      0x71562b71999873db5b286df957af199ec94617f7 (10^49 ETH)
WARN [07-19|02:42:46.376]
WARN [07-19|02:42:46.376]      Private Key
WARN [07-19|02:42:46.376]      -----
WARN [07-19|02:42:46.376]      0xb71c71a67e1177ad4e901695e1b4b9ee17ae16c6668d313eac2f96[REDACTED]
WARN [07-19|02:42:46.376]
INFO [07-19|02:42:46.376] Starting peer-to-peer node           instance=Geth/v1.16.1-stable-12b4131f/linux
-amd64/gol.24.4
WARN [07-19|02:42:46.376] P2P server will be useless, neither dialing nor listening
INFO [07-19|02:42:46.381] IPC endpoint opened           url=/tmp/geth.ipc
INFO [07-19|02:42:46.381] HTTP server started           endpoint=127.0.0.1:8545 auth=false prefix=
cors=https://remix.ethereum.org vhosts=localhost
INFO [07-19|02:42:46.382] New local node record           seq=1,752,864,166,381 id=5a498da1b5df4f0c i
p=127.0.0.1 udp=0 tcp=0
INFO [07-19|02:42:46.382] Started P2P networking           self=enode://2947b9f976fea97f00cf1be7e58b88
995a40f02daacb1eb6052fd298e7acb9e52e7481686d3f6101762a7a48e5b639e1540db8d958baff182b2bfdafb8a79e04@127.0.0.1:0
INFO [07-19|02:42:46.382] Started log indexer

```

Understanding the Console Output:

When you run Geth in dev mode, the console will display important information as shown in the image above:

- **Private Key:** The console shows the private key of the pre-funded account that you can use for testing. This key is automatically generated and displayed in the console output.
- **HTTP Endpoint:** The console confirms that the HTTP JSON-RPC server is running on `http://127.0.0.1:8545`. This is the endpoint address you should use as your Provider URL in Casibase.
- **Account Address:** The corresponding Ethereum address for the generated private key is also displayed.

Important: Copy and save these values immediately as they are essential for configuring your Casibase provider. The private key will be needed for the `Private key` field, and the HTTP endpoint will be your `Provider URL`.

For more details, see the [Geth Dev Mode documentation](#).

2.2 Private key Configuration

The private key is essential for signing transactions on the Ethereum blockchain.

 NOTE

The private key should be provided without the hexadecimal prefix `0x`.

Example Private Key

```
# Example private key in geth dev mode (without 0x prefix)
b71c71a67e1177ad4e901695e1b4b9ee17ae16c6668d313eac2f96dbcda3f291
```

 DANGER

This is just an example private key for demonstration purposes. Never use this key in production or for real funds!

You can obtain your private key from various sources:

- **Initially generated by Geth:** When you start Geth in `dev` mode, it generates a pre-funded account with a private key displayed in the console.
- **Ethereum Clients:** Generate a new account using Ethereum clients like Geth.

 INFO

The private key field is automatically masked as a password field.

2.3 Invoke Method and Contract Address

Configuration

Invoke Method Configuration

The invoke method is the specific function name in the smart contract that you want to call.

In Casibase, your smart contract should implement specific methods to ensure compatibility:

- `save`: This method is used to store data in the contract. It should accept parameters as a tuple (struct).

```
struct DataItem {  
    string key;  
    string field;  
    string value;  
}  
  
// Define event, returns key, field, value in order  
event DataSaved(string key, string field, string value);  
  
// Save struct data and emit an event for tracking  
function save(DataItem memory _data) public {  
    emit DataSaved(_data.key, _data.field, _data.value);  
}
```

Method name `save` can be customized, but it should accept a struct as an argument.

You can refer to the [Example](#) to see how to implement the `save` method in your smart contract.

Make sure your contract includes these methods to enable seamless integration with Casibase.

Contract Address Configuration

```
# Example contract address (without 0x prefix)
c36fED2CE2E1Bb14b330465f4498D4892C8ee194
```

The contract address is the deployed smart contract's address on the Ethereum blockchain. You can obtain the contract address after deploying a smart contract.

Example for Contract Deployment Reference

To deploy a smart contract on Ethereum, you can refer to the [Casibase/contract-storage-eth](#). This repository provides sample Solidity contracts and deployment scripts using Go and Remix.

Getting Started with the Repository:

1. **Get Example Code:** Clone or download the repository to access sample Solidity contracts and deployment scripts.
2. **Contract Compilation:** Pre-compiled contract artifacts (ABI and bytecode) are available in the [releases](#) section of the repository.
3. **Setup for Go Script Deployment:** If using the Go deployment script, download the contract artifacts from releases and place them in the `build/` folder within the Go script's working directory.
4. **Deployment Options:** You can deploy the contract using either:
 - **Go Script:** Use the provided Go deployment script in the repository for programmatic deployment (requires contract artifacts in `build/` folder)
 - **Remix IDE:** Deploy contracts online using [Remix](#) with the contract source code

After deployment using either method, you can obtain the contract address from the deployment output.

Use the go script in the reference to deploy the contract

```
$ go run deploy.go
Starting contract deployment...
Connected to Ethereum node: http://192.168.31.234:8545
Deploying from address: 0x71562b71999873DB5b286dF957af199Ec94617F7
Loaded bytecode from: build/SaveContract.bin
Loaded ABI from: build/SaveContract.abi
Gas price: 9 wei
Gas limit: 0
Deploying contract...
Transaction sent: 0xf377a667d3216a1a45b3c3d0944745ea1cbe8ab17745f6e95c44d4f7a5a3fd8f
Contract address: 0xc36fED2CE2E1Bb14b330465f4498D4892C8ee194
Waiting for transaction confirmation...
Contract deployed successfully!
Gas used: 611787
Block number: 198

Running contract test...
Calling save function with: key=test_key_123, field=test_field, value=test_value_456
Save transaction: 0xb060530b6de01bd0537595b86dd0ebcac9007ebab3f24c096c42857ac6fdb3f2
Save function called successfully!
Retrieved data - Key: test_key_123, Field: test_field, Value: test_value_456
Log data - Key: test_key_123, Field: test_field, Value: test_value_456
```

Or you can use the block explorer to find the contract address.

Transaction Dashboard

TOTAL NO. OF TRANSACTIONS 1	BLOCK NUMBER 198	BLOCK HASH 0x58cdd0285c82aca 9676f47e96b2f6f08 7ddc83c4155f5b88 6373a115f466cb5	Transaction Status SUCCESSFUL
--------------------------------	---------------------	---	----------------------------------

Transaction Overview

Transaction Hash	0xf377a667d3216a1a45b3c3d0944745ea1cbe8ab17745f6e95c44d4f7a5a3fd8f
Transaction Gas	617810
Transaction Gas Price	9
Transaction Nonce	197
Transaction To	0xc36fED2CE2E1Bb14b330465f4498D4892C8ee194 [CONTRACT CREATION]
Transaction From	0x71562b71999873DB5b286dF957af199Ec94617F7
Transaction Value [wei]	0
Transaction Status	SUCCESSFUL

2.4 Browser URL Configuration

The Browser URL lets you view specific blockchain blocks and transactions in a web browser. By using a template with the `{bh}` placeholder, Casibase can automatically redirect you to the corresponding block details in your chosen blockchain explorer.

`http://127.0.0.1:5051/txpage?blocknumber={bh}`

Organization	ID	Name	Client IP	Created time	Provider	User	Method	Request URI	Action	Block	Action
built-in	115	0af1c434-a708-4238-a55a-5ae322b2f3f2	127.0.0.1	2025-07-19 03:08:20	eth_win_test	admin	POST	/api/signin?code=f333196946c	signin	200	<button>Query</button> <button>View</button> <button>Delete</button>
built-in	111	aea40549-4bfb-41a4-a9cd-51fb4004fe49	::1	2025-07-14 00:20:36	eth_win_test	admin	POST	/api/update-provider	update-provider		<button>Commit</button> <button>View</button> <button>Delete</button>
built-in	110	88cfe21e-b4e3-4c8b-8b37-8ca5941fed55	::1	2025-07-14 00:19:24	eth_win_test	admin	POST	/api/delete-provider	delete-provider		<button>Commit</button> <button>View</button> <button>Delete</button>
built-in	109	cf93e75c-501d-4aa7-a350-70509a90bae4	::1	2025-07-14 00:19:21	eth_win_test	admin	POST	/api/delete-provider	delete-provider	25	<button>Query</button> <button>View</button> <button>Delete</button>
built-in	108	357977eb-4d44-44d5-8714-13f7a1f5a2fe	::1	2025-07-14 00:19:02	eth_win_test	admin	POST	/api/update-provider	update-provider	24	<button>Query</button> <button>View</button> <button>Delete</button>
built-in	107	66b29d6d-2b16-4d6b-9349-6be11748bea7	::1	2025-07-14 00:03:25	eth_win_test	admin	POST	/api/signin?code=f7a6534407f	signin	23	<button>Query</button> <button>View</button> <button>Delete</button>

💡 TEMPLATE FOR BROWSER URL

When you use the `{bh}` placeholder in the Browser URL template, Casibase will replace it with the actual block number and allow you to jump directly to the relevant block information in your blockchain explorer.

Example: Ganache CLI Block Explorer

To quickly view Ethereum blocks and transactions, you can use the open-source blockchain explorer [casibase/ganache-cli-block-explorer](https://github.com/casibase/ganache-cli-block-explorer). This tool provides a simple web interface for browsing blocks, transactions, and contract events on your local Ethereum node.

The screenshot shows the Ganache Block-Explorer interface. The left sidebar has a blue header with the logo and the text 'GANACHE BLOCK-EXPLORER'. It includes a 'Dashboard' button and a 'Menu' button. The main content area is titled 'Transaction Dashboard' and contains four cards: 'TOTAL NO. OF TRANSACTIONS' (1), 'BLOCK NUMBER' (200), 'BLOCK HASH' (0x65594c64b90b5805e0c876786b8319f5838179d6d0c314e53e3272b588fc83be), and 'Transaction Status' (green). Below this is a 'Transaction Overview' table with the following data:

Transaction Hash	0xd05c3fb8aa26d168ae6d3cdb88948a6487282813afb738f2ec7773d86ae63a6a
Transaction Gas	336909
Transaction Gas Price	9
Transaction Nonce	199
Transaction To	0xD460001B04b4FD07194476C35825175B30F09Ec
Transaction From	0x71562b71999873DB5b286dF957af199Ec94617F7
Transaction Value [wei]	0
Transaction Status	SUCCESSFUL

Note: This explorer is based on [vivekganesan01/ganache-cli-block-explorer](#) and includes additional features contributed by Casibase.

After installation and startup, you can access block details directly from the above address as the Browser URL in Casibase.

Private Cloud Providers

Introduction

In Casibase, Private Cloud Providers act as a bridge, allowing you to connect to and manage various cloud-native resources, such as Docker and Kubernetes (K8s), directly from the Casibase interface. Their core objective is to provide a centralized dashboard for monitoring and operating your containerized services, integrating their management seamlessly into your Casibase workflow.

By configuring a provider, you enable Casibase to communicate with your private cloud or on-premises data center. This provides an ideal solution for organizations that want a unified interface to manage both their knowledge base and the infrastructure it runs on, enhancing operational efficiency and control.

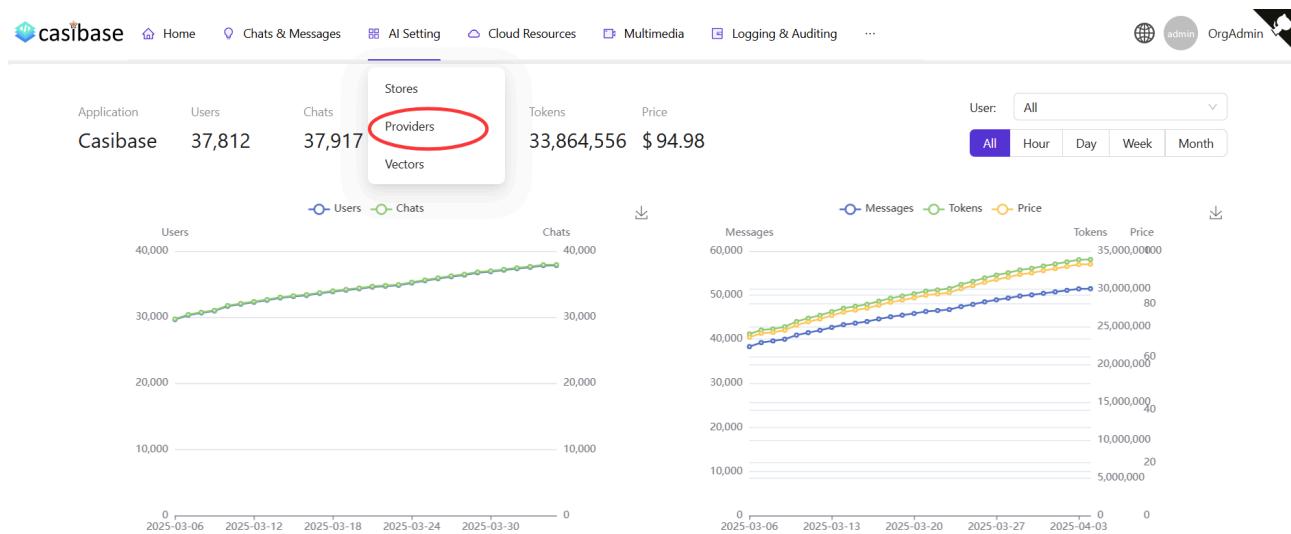
Refer to the [Core Concepts](#) section of our previous documentation for more information about providers.

In Casibase, you can add a private cloud provider by following these steps:

Add a New Private Cloud Provider

Private cloud providers are used to integrate cloud-native management features into Casibase. You can add them by following these steps:

Click the [Providers](#) button on the page.



Add a Private Cloud Provider

Click the [Add](#) button to add a new private cloud provider.

Providers	Add	Add Storage Provider						
Name	Display name	Category	Type	Sub type	API key	Secret key	Region	
provider_tts_alibabacloud_cosyvoice	Provider TTS AlibabaCloud Cosyvoice	Text-to-Speech	Alibaba Cloud	cosyvoice-v1		***		
provider_blockchain_chainmaker	Provider Blockchain ChainMaker	Blockchain	Tencent ChainMaker (Demo Network)	text-davinci-003	AKIDObZHjvqUrxnNtkKdfQvAWcS1JK4XidG	***	ap-beijing	
provider_model_alibabacloud_deepseek_r1	Provider Model AlibabaCloud DeepSeek-R1	Model	Alibaba Cloud	deepseek-r1		***		
provider_cloud_alibabacloud	Provider Cloud AlibabaCloud	Public Cloud	Aliyun	text-davinci-003	LTAI5tHkUsopAioN6sx2LMg	***	cn-beijing	
dall-e-3	dall-e-3	Model	OpenAI	dall-e-3		***		
provider_model_azure_gpt4_1	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***		
provider_model_azure_gpt4_2	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***		
provider_model_azure_gpt4_3	Provider Model Azure GPT-4	Model	Azure	gpt-4o	deployment-gpt-4o	***		
provider_embedding_openai_v3	Provider Embedding OpenAI V3	Embedding	OpenAI	text-embedding-ada-002		***		
provider_model_openai_gpt4_vision	Provider Model OpenAI GPT4 Vision	Model	OpenAI	gpt-4-vision-preview		***		

Fill in Private Cloud Provider Details

Fill in the required configuration according to the cloud-native platform you use, then click the **Save & Exit** button to save.

More information about the configuration can be found below:

Kubernetes Configuration

Kubernetes (K8s) is an open-source container orchestration platform for automating the deployment, scaling, and management of containerized applications. It has bec...

TIP

Casibase supports several mainstream cloud-native technologies and platforms, including:

- **Docker:** To connect to a Docker host and manage the lifecycle of its containers (e.g., start, stop, view status) directly within Casibase.
- **Kubernetes (K8s):** To connect to a Kubernetes cluster and manage its resources, such as Pods and Deployments, providing a high-level orchestration view within Casibase.

Now, you can use this private cloud provider to monitor and manage services in your cloud-native environment.

After adding a private cloud provider, you can use Casibase as a control panel to oversee your containerized applications, simplifying management and providing a unified view of your services alongside your knowledge base.

Kubernetes Configuration

Kubernetes (K8s) is an open-source container orchestration platform for automating the deployment, scaling, and management of containerized applications. It has become the de facto standard for managing applications in modern, cloud-native environments. By providing a robust framework for running distributed systems resiliently, Kubernetes simplifies complex operational tasks.

In this chapter, you will learn how to configure and use a Kubernetes provider in Casibase. This will allow you to connect Casibase to your Kubernetes cluster, enabling you to monitor and manage your cloud resources directly from the Casibase interface.

1. Configuration Field Description

When configuring a Kubernetes provider in Casibase, you need to fill in several key fields. Each field has a specific meaning and is required for the correct integration with your Kubernetes cluster. The following list explains the purpose of each field:

- `Name`: The unique identifier for this private cloud provider.
- `Display name`: The display name shown in the UI for this provider.
- `Category`: The type of service; here it should be `Private Cloud`.
- `Type`: The cloud-native platform type; here it should be `Kubernetes`.
- `Config text`: The raw text content of your `kubeconfig` file, which contains the credentials and endpoint information needed to connect to your Kubernetes cluster.

Please make sure to fill in each field accurately. The `Config text` is crucial for establishing a successful connection.

2. Configure Kubernetes

The primary method for connecting Casibase to your Kubernetes cluster is by using your `kubeconfig` file.

2.1 Obtain Your Kubeconfig File

Before proceeding, you must ensure that the `kubeconfig` file you intend to use can successfully connect to your Kubernetes cluster. A reliable way to get the raw configuration is to run the following command in your terminal:

```
kubectl config view --raw > kubeconfig.yaml
```

This command will save the complete, flattened configuration into a file named `kubeconfig.yaml` in your current directory. You can then open this file to copy its contents.

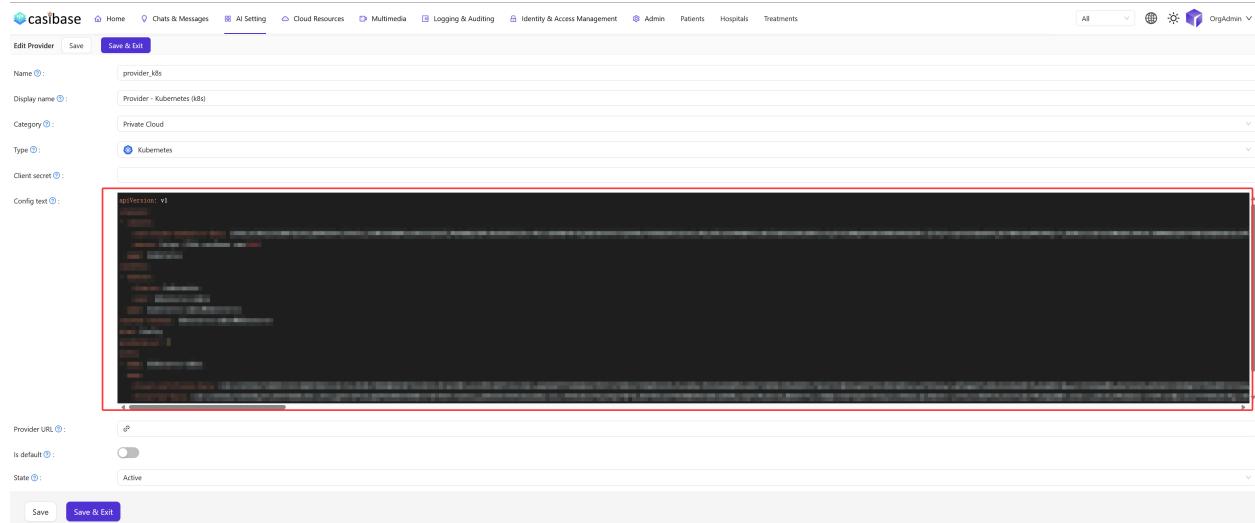
You can test your configuration file with a command that checks for pods across all namespaces. This is a more reliable test to confirm connectivity.

```
kubectl --kubeconfig=./kubeconfig.yaml get pods -A
```

If this command runs successfully (even if it just lists pods from system namespaces), you are ready to proceed.

2.2 Fill in the Provider Details

Copy the entire content of your valid `kubeconfig.yaml` file and paste it into the `Config text` field in the provider configuration form.



3. Verify the Connection

After you have filled in the details and saved the provider, you can verify if the connection was successful.

Navigate to the Cloud Resources > Applications section within Casibase. Here, you will see a list of your configured providers. Check the status of the Kubernetes provider you just added.

- **Active:** If the status is `Active`, Casibase has successfully connected to your Kubernetes cluster.
- **Inactive:** If the status is `Inactive`, there was an issue with the connection. Please double-check the content of your `Config text` and ensure that there is network connectivity between Casibase and your Kubernetes cluster's API server.

Casibase Applications								
Actions	Name	Display name	Created time	Description	Template	Status	Namespace	Action
Edit	application_grekm	New Application - grekm	2025-08-02 21:59:58		template_2	Running	casibase-application_grekm	Edit Undeploy Delete
Edit	application_jijkl	New Application - jijkl	2025-08-02 09:43:55		template_2	Running	casibase-application_jijkl	Edit Undeploy Delete
Edit	application_x04b7i	New Application - x04b7i	2025-08-01 23:44:27		template_2	Not Deployed	casibase-application-x04b7i	Edit Deploy Delete

Once the connection is active, you can begin to monitor and manage your Kubernetes resources through Casibase.

Public Cloud Providers

Introduction

Public Cloud Providers enable Casibase to connect to and scan cloud infrastructure across major cloud platforms. The system automatically discovers and catalogs cloud resources, providing centralized visibility and management of your cloud assets.

When configured, Casibase can scan resources like virtual machines, storage volumes, and network components, enriching each asset with detailed configuration information through multi-level API calls.

Supported Cloud Platform

Casibase currently supports Alibaba Cloud for complete asset discovery including ECS instances, disks, VPCs, and other resources. Additional cloud platforms are planned for future releases.

Add a Public Cloud Provider

Navigate to the Providers page and click  to create a new provider.

Providers	Add	Add Storage Provider	Name	Display name	Category	Type	Sub type	API key	Secret key	Region
provider_tts_alibabacloud_cosyvoice	Provider TTS AlibabaCloud Cosyvoice	Text-to-Speech	Alibaba Cloud	cosyvoice-v1					***	
provider_blockchain_chainmaker	Provider Blockchain ChainMaker	Blockchain	Tencent ChainMaker (Demo Network)	text-davinci-003				AKIDObZHjqUrxnNltkKdfQvAWcS1JK4XidG	***	ap-beijing
provider_model_alibabacloud_deepseek_r1	Provider Model AlibabaCloud DeepSeek-R1	Model	Alibaba Cloud	deepseek-r1					***	
provider_cloud_alibabacloud	Provider Cloud AlibabaCloud	Public Cloud	Aliyun	text-davinci-003				LTA15tHKUsopAioN6xi2LMg	***	cn-beijing
dall-e-3	dall-e-3	Model	OpenAI	dall-e-3					***	
provider_model_azure_gpt4_1	Provider Model Azure GPT-4	Model	Azure	gpt-4o				deployment-gpt-4o	***	
provider_model_azure_gpt4_2	Provider Model Azure GPT-4	Model	Azure	gpt-4o				deployment-gpt-4o	***	
provider_model_azure_gpt4_3	Provider Model Azure GPT-4	Model	Azure	gpt-4o				deployment-gpt-4o	***	
provider_embedding_openai_v3	Provider Embedding OpenAI V3	Embedding	OpenAI	text-embedding-ada-002					***	
provider_model_openai_gpt4_vision	Provider Model OpenAI GPT4 Vision	Model	OpenAI	gpt-4-vision-preview					***	

Configuration

Select "Aliyun" (Alibaba Cloud) as the provider type and fill in the required credentials:

- **Type:** Select "Aliyun" (Alibaba Cloud)
- **Client ID:** Your Alibaba Cloud Access Key ID
- **Client Secret:** Your Alibaba Cloud Access Key Secret
- **Region:** The cloud region to scan (e.g., cn-hangzhou)

Click **Save & Exit** to complete the configuration.

Asset Scanning

After adding a public cloud provider, Casibase can scan and catalog your cloud resources. The scanning process:

1. Connects to the cloud platform using provided credentials
2. Discovers resources using the SearchResources API

3. Enriches resource details through secondary API calls
4. Stores asset information with full configuration details

Scanned Resources

For Alibaba Cloud, the following resource types are automatically discovered:

- ECS instances with CPU, memory, IP addresses, and OS details
- Disk volumes with size, type, and attachment information
- VPC networks with CIDR blocks and routing configuration
- Security groups and network interfaces

Viewing Assets

After scanning, view discovered assets in the Assets page. Each asset displays:

- Resource ID and display name
- Resource type and region
- Current state
- Tags and metadata
- Detailed properties specific to the resource type

More configuration details:

Alibaba Cloud Configuration

Configure Alibaba Cloud asset scanning

Alibaba Cloud Configuration

Overview

The Alibaba Cloud provider serves dual purposes in Casibase. For model integration, it connects you to Qwen's language models and DeepSeek's reasoning-focused models (v3, v3.1, v3.2, and R1) through Alibaba's Model Studio platform. When chatting with these models, you can enable web search to have the AI fetch current information from the internet—results appear inline with citations.

For infrastructure management, the same provider scans your Alibaba Cloud resources, discovering ECS instances, disks, VPCs, and other assets across your deployment.

Prerequisites

Before configuring the Alibaba Cloud provider, you need:

- An Alibaba Cloud account
- Access Key ID and Access Key Secret with appropriate permissions
- Access to at least one region where resources are deployed

Required Permissions

The Access Key should have permissions to call:

- `resourcecenter:SearchResources` - For resource discovery
- `ecs:DescribeInstances` - For ECS instance details
- `ecs:DescribeDisks` - For disk details
- `vpc:DescribeVpcs` - For VPC details

Configuration Steps

Create Access Key

1. Log in to Alibaba Cloud console
2. Navigate to AccessKey Management
3. Create a new Access Key pair
4. Save both the Access Key ID and Access Key Secret securely

Add Provider in Casibase

1. Open Casibase and navigate to Providers
2. Click the `Add` button
3. Fill in the configuration:
 - **Name:** A descriptive name for this provider
 - **Type:** Select "Aliyun" (this is the internal identifier for Alibaba Cloud)
 - **Category:** Select "Public Cloud"
 - **Client ID:** Paste your Access Key ID
 - **API key:** Paste your Access Key Secret
 - **Region:** Enter the target region (e.g., cn-hangzhou, cn-beijing)
4. Click `Save & Exit`

Asset Discovery

Once configured, Casibase performs two-level asset discovery:

Level 1: Resource Discovery

Uses the SearchResources API to scan all resource types in the specified region. This provides basic information about each resource including ID, type, region, and tags.

Level 2: Detailed Enrichment

For specific resource types, Casibase makes additional API calls to gather comprehensive details:

ECS Instances

- Instance specifications (CPU cores, memory size)
- Operating system information
- Public and private IP addresses
- Billing method and instance status

Disks

- Disk category (cloud, cloud_efficiency, cloud_ssd)
- Size and attachment status
- Associated instance information

VPCs

- CIDR block configuration
- Associated VSwitches
- Routing table information

Resource Properties

Each discovered asset includes standard fields plus resource-specific properties. The properties are stored as key-value pairs and displayed in the asset detail view.

Common properties include instance types, network configurations, storage specifications, and billing information.

Scanning Multiple Regions

To scan resources in multiple regions, create separate providers for each region. This approach allows granular control over which regions are scanned and monitored.

Troubleshooting

No resources found: Verify the Access Key has correct permissions and the specified region contains resources.

Authentication errors: Check that the Access Key ID and Secret are correctly entered and the key is active.

Missing details: Ensure the Access Key has permissions for the detailed API calls (DescribeInstances, DescribeDisks, DescribeVpcs).

Stores

Overview

Stores Overview

Store Configuration

After adding storage providers, model providers and embedding providers, we can configure the stores

Overview

1. Overview of the Stores Function

In Casibase, the Stores function is one of its core modules, which allows users to integrate storage, modelling, and embedding service providers for knowledge base data storage, text vector conversion, and interaction with chatbots. With the Stores feature, users can build an efficient, flexible and powerful AI knowledge management system.

2. Advantages of Stores

2.1 Multi-model integration

Casibase's Stores feature supports multiple mainstream AI language models, including OpenAI (e.g., GPT-3.5, GPT-4), Azure OpenAI, HuggingFace, Google Gemini, and so on. This multi-model support allows users to choose the most suitable AI model for their specific needs and find a balance between performance, cost and features.

2.2 Multiple storage and embedding options

Users are free to choose storage and embedding service providers to meet different data storage and processing needs. This flexibility enables users to configure the most appropriate storage and embedding solution based on their technology stack and business requirements.

2.3 Multi-Store Mode

Casibase supports a multi-Store model that allows users to use different models, storage and embedding services in different Stores to provide customised services for different scenarios and users. This feature enables users to flexibly configure and switch Stores according to different business requirements.

2.4 Cross-Store Vector Sharing

Stores in Casibase can be configured to use vectors from other stores through the **Vector stores** field. This allows you to create a main store that searches across multiple specialized knowledge bases, or let different stores share their knowledge with each other. Instead of duplicating content, stores can dynamically access relevant information from other stores while maintaining their own separate vector databases.

2.5 Streamlined Management

The interface adapts to how you work. File-focused workflows can use the "Hide chat" toggle to clear away AI provider columns from the store list. Each store can also include example questions that appear when users start chatting, helping them understand what to ask without reading documentation.

2.6 Direct File Navigation

When reviewing knowledge sources in chat responses, you can click on any referenced file to open the store's file viewer with that specific document automatically selected and displayed. This seamless navigation lets you verify AI responses against source materials without manually searching through your document tree.

3. Summary

Casibase's Stores feature provides users with a powerful knowledge management tool that enables them to flexibly build and manage knowledge bases by integrating multiple AI models, stores and embedded services. Its multi-Store model and enterprise-level features further enhance the flexibility and security of the system, which is suitable for a variety of application scenarios.

Casibase is an open source AI knowledge base system designed to provide efficient and flexible knowledge management and dialogue solutions for enterprises. One of its core features is Providers, which allows users to integrate multiple AI models and storage services to enhance the functionality and performance of the system. Providers are divided into three main categories: Model Providers, Embedding Providers and Storage Providers, which are responsible for handling AI models and data storage, respectively.

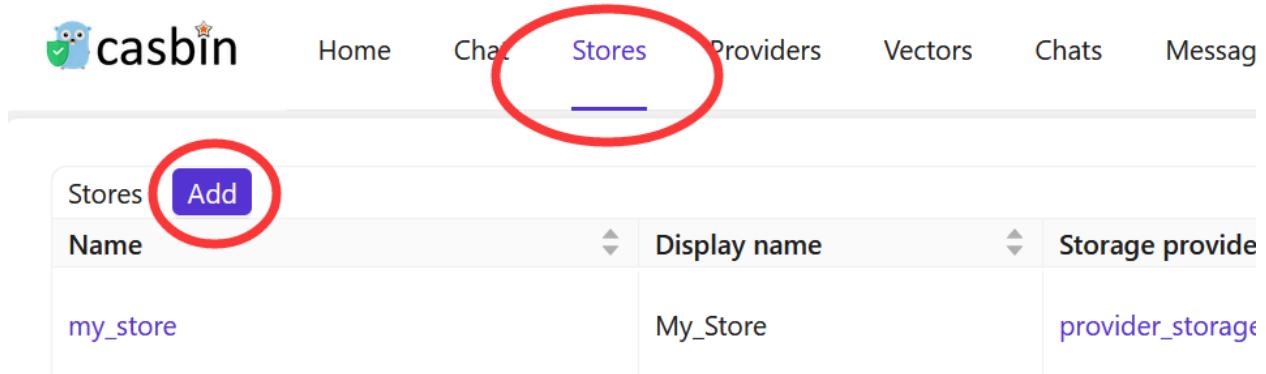
Store Configuration

After adding storage providers, model providers and embedding providers, we can configure the stores

1. Add a New Store

Stores are used to integrate storage, model, and embedding providers into Casibase. You can add them by following these steps:

Click the `Stores` button on the home page and then click the `Add` button to add a store.



The screenshot shows the Casibase interface. At the top, there is a navigation bar with the Casibase logo, followed by links for Home, Chat, Stores (which is highlighted in blue and circled in red), Providers, Vectors, Chats, and Messages. Below the navigation bar is a table titled 'Stores'. The table has columns for 'Name', 'Display name', and 'Storage provider'. A single row is visible, showing 'my_store' in the 'Name' column, 'My_Store' in the 'Display name' column, and 'provider_storage' in the 'Storage provider' column. The 'Add' button in the table header is also circled in red.

Name	Display name	Storage provider
my_store	My_Store	provider_storage

Fill in Store Details

Fill in the store details and click the `Save & Exit` button.

casbin Home Chat Stores Providers Vectors Chats Messages Tasks Resources Permissions Logs Jimmy

Edit Store Save

Name: store_v6c22m

Display name: New Store - v6c22m

Storage provider: ▼

Model provider: ▼

Embedding provider: ▼

File tree:

X

storage provider is empty

Go to Store

Save

Powered by Casibase

Select the storage provider, model provider, embedding provider, text-to-speech provider and speech-to-text provider you added before.

casbin

Home Chat Stores Providers Vectors Chats Messages Tasks Resources  P

Edit Store Save

Name: my_store

Display name: My_Store

Storage provider: Provider_storage_1 (provider_storage_1)

Model provider: Model OpenAI text-davinci-003 (model_openai_text_davinci_003)

Embedding provider:

File tree:

Embedding_OpenAI_Adasimilarity (embedding_openai_adasimilarity)

My_Store

- alibaba_oss
 - audio
 - AC / DC - Highway To Hell.mp3 (8.34 MB)
 - document
 - casdoor-knowledge.doc (18.0 KB)
 - casdoor-knowledge.docx (10.9 KB)
 - casdoor-knowledge.html (23.5 KB)
 - casdoor-knowledge.md (2.12 KB)
 - casdoor-knowledge.pdf (107 KB)
 - image
 - lena.jpg (105 KB)
 - lena.tiff (768 KB)
 - video
 - my_video.mkv (456 KB)

Click the Save & Exit button and return to the stores list page:

casbin

Home Chat Stores Providers Vectors Chats Messages Tasks Resources  Permissions  Logs 

Jimmy

Stores Add

Name	Display name	Storage provider	Model provider	Embedding provider	Action
my_store	My_Store	provider_storage_1 	model_openai_text_davinci_003	embedding_openai_adasimilarity	View Refresh Vectors Edit Delete

< 1 >

The store list shows many columns for AI configurations like model providers, embedding providers, and voice settings. If you're mainly using stores for file management, toggle "Hide chat" at the top of the list to simplify the view and show only the essentials: name, storage provider, and status.

Now, you can use the store to store knowledge base data, convert text to vectors, and chat with the chatbot.

When configuring your store, consider adding example questions that appear when users start a new chat. These suggestions help users discover what they can ask and get them started quickly. You can also configure which other stores this one should search through using the **Vector stores** field - useful when you want one store to pull knowledge from multiple sources.

Vector Stores

Sometimes you need one store to search through knowledge from multiple other stores. For example, you might have separate stores for different topics or departments, but want a main store that can answer questions by searching across all of them. The **Vector stores** field makes this possible by letting a store use vectors from other stores in addition to its own.

To configure this, navigate to the store edit page and find the **Vector stores** field. You can select one or more stores from the dropdown list. When you chat with this store, it will automatically search through both its own vectors and the vectors from all the stores you selected. This way, you can create a centralized knowledge hub without duplicating content across multiple stores.



The store always uses its own vectors automatically. You only need to specify additional stores in the **Vector stores** field when you want to search

across multiple knowledge bases.

In the next section, we will learn how to chat with the chatbot in Casibase.

2. File Management

Automatic Vector Generation

When you upload files to a store, Casibase automatically generates vector embeddings in the background. Each file goes through several states that you can track in the file list:

- **Pending:** File uploaded, waiting for embedding to start
- **Processing:** Currently generating vectors from the file content
- **Finished:** Vector generation completed successfully
- **Error:** Something went wrong during embedding (check the error text for details)

This happens without any manual intervention - just upload your files and let Casibase handle the rest. The system splits your documents into chunks, generates embeddings for each chunk, and stores them for fast retrieval during chat.

File Status Tracking

Navigate to the Files page to see all uploaded files across your stores. Each file shows its current status and token count. Files in the "Error" state display detailed error messages so you can troubleshoot issues like unsupported file formats or provider configuration problems.

When you delete a file from storage, Casibase automatically removes its associated vectors from the database. This keeps your vector database clean and prevents outdated information from appearing in chat responses.

Re-embedding Individual Files

Sometimes you need to regenerate vectors for a specific file - perhaps you've changed the embedding model or fixed a previous error. The "Refresh Vectors" button in the file list lets you re-embed individual files without affecting other content in the store.

Click the button next to any file to delete its existing vectors and generate fresh ones. The file status updates in real-time as processing completes, and the token count reflects the newly generated embeddings. This targeted approach saves time compared to refreshing the entire store when you only need to update a few files.

3. Store Isolation for Users

Casibase lets you restrict users to specific stores through Casdoor's Homepage field. When a user's Homepage matches a store name, they become bound to that store and work within it exclusively - perfect for multi-tenant setups where teams need separate knowledge bases.

Bound users see their assigned store locked in the top bar selector. They can view and query only that store's data, and API calls to other stores fail with an error. They also cannot add, delete, or rename stores, keeping their workspace stable and isolated.

To bind a user, edit their Casdoor profile and set the Homepage field to the exact store name (case-sensitive). The binding activates on their next login. Users with

an empty Homepage or one that doesn't match any store keep full access to all stores, so you can mix restricted and unrestricted users easily.

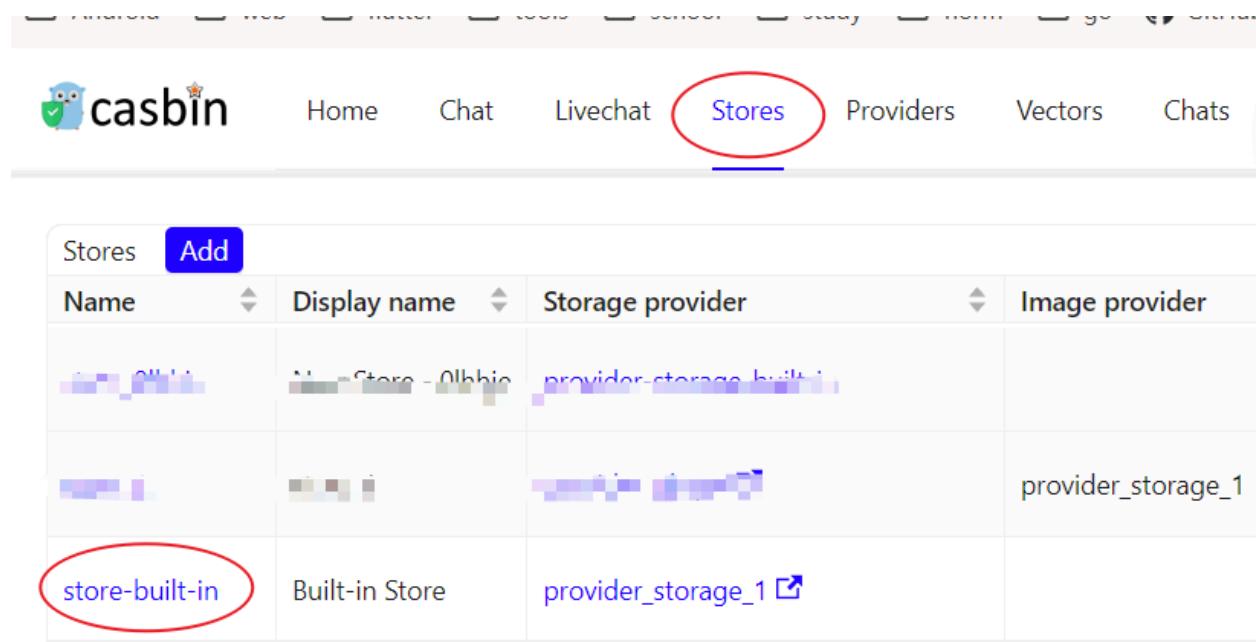
4. Support Multi-store

The multi-store mode provides users with different models, suggestions, and more within each distinct store.

Enable Multi-store

First, you should enable multi-store mode in the built-in store.

Click the `Stores` button on the home page and then click the `store-built-in` button to enter the store-built-in store.



Name	Display name	Storage provider	Image provider
store-built-in	My Store - Olbbie	provider_storage_builtin	
store	My Store	provider_storage_1	
store-built-in	Built-in Store	provider_storage_1	

Scroll down and find the `Can Select Store` field, tick it.

The screenshot shows the Casibase interface. At the top, there is a message box with a close button and the text 'No data'. Below it, a 'Suggestion count: 3' box is shown. Under 'Theme color:', a blue square is selected. A red oval highlights the 'Can Select Store: ' option. The 'File tree:' section shows a search bar with 'Please input your search term' and a magnifying glass icon. The tree structure includes 'Built-in Store' and 'casibase' (selected), which contains 'user_jaycd9' (selected), 'chat_0edmwu' (selected), 'chat_1gs4l9' (selected), and 'chat_1ikor4' (selected). The 'chat_0edmwu' folder contains several files: '20240508_070749-1.png' (40.4 KB), '20240508_070749.png' (40.4 KB), '20240508_070937-1.txt' (0 B), '20240508_070937.txt' (40.4 KB), '20240508_071008-1.txt' (18 B), '20240508_071008.txt' (40.4 KB), and '20240508_071037.md' (6.21 KB). The 'chat_1gs4l9' folder contains '20240509_202051.jpg' (639 KB).

Add Usable Stores

The multi-store mode only provides usable stores. To make a store usable, you need to configure its storage provider, model provider, and embedding provider.

Select For Conversation

Casibase provides a very convenient method for selecting a store.



[+ New Chat](#)

New Chat - 7

store_1

store-built-in

New Chat - 8

New Chat - 5

New Chat - 8

New Chat - 7

Chat

C

L

T

D

...

You are an expert in your field

Thank you for recognizing my expertise. Whether it's related to my specific area of knowledge or expertise to provide insightful answers and solutions to any problems you may have.

Just hover your mouse over "New Chat" and then you can select the Store you wish to use from the list that appears below.

If you click the "New Chat" button, the system will assign you a default Store.

Forms

Forms Overview

Introduction to Forms in Casibase

Form Configuration

How to configure and customize forms in Casibase

Forms Overview

Introduction

Forms in Casibase provide a powerful way to customize the display of list pages throughout the application. By configuring forms, you can control which columns are visible, their order, and their width in various list views such as Records, Providers, Stores, and more.

The Forms feature was ported from Casdoor and allows administrators to tailor the user interface to their specific needs, improving usability and focusing on the most relevant information.

Key Features

Customizable Column Visibility

Control which columns appear in list pages by toggling the visibility setting for each form item. This helps reduce clutter and focus on the most important data.

Adjustable Column Width

Set custom widths for each column to optimize the display based on the content type and your screen size preferences.

Column Ordering

Arrange columns in the order that makes the most sense for your workflow by reordering form items.

Multiple Form Categories

Forms support different categories to serve various purposes:

- **Table:** Traditional table-based forms for displaying structured data
- **iFrame:** Forms that can embed external content
- **List Page:** Forms specifically designed to customize list page columns

Form Structure

Each form in Casibase consists of:

- **Organization:** The organization that owns the form
- **Name:** Unique identifier for the form
- **Display Name:** Human-readable name shown in the interface
- **Position:** Placement or order of the form
- **Category:** Type of form (Table, iFrame, or List Page)
- **Type:** Specific form type (e.g., "records" for the Records list page)
- **URL:** Associated URL or endpoint (for Table and iFrame categories)
- **Form Items:** Collection of columns or fields to display

Form Items

Form items define the individual columns in a list page. Each form item includes:

- **Name:** Internal column identifier (e.g., "organization", "name", "createdTime")
- **Label:** Display label shown in the column header
- **Type:** Data type (currently "Text" for list pages)

- **Visible:** Whether the column is displayed
- **Width:** Column width in pixels

Use Cases

Forms are particularly useful for:

1. **Customizing Record Views:** Tailor the Records list page to show only relevant columns for your use case
2. **Simplifying Complex Tables:** Hide technical columns that aren't needed by all users
3. **Optimizing Screen Space:** Adjust column widths to fit more information on screen
4. **Role-Based Views:** Create different forms for different user roles or workflows

Getting Started

To start using Forms in Casibase:

1. Navigate to the Forms section in the Casibase admin interface
2. Create a new form or edit an existing one
3. Configure the form category, type, and items
4. Preview your changes in real-time
5. Save and apply the form to the corresponding list page

For detailed instructions on configuring forms, see the [Form Configuration](#) guide.

Form Configuration

Accessing Forms

Forms can be accessed through the Casibase admin interface:

1. Log in to your Casibase admin dashboard
2. Navigate to the Forms section from the main menu
3. You'll see a list of existing forms organized by category and type

Creating a New Form

Step 1: Basic Information

To create a new form:

1. Click the Add button on the Forms list page
2. Fill in the basic form information:
 - Organization: Select the organization (typically your organization name)
 - Name: Enter a unique identifier for the form
 - Display Name: Provide a human-readable name
 - Position: Set the position or order (optional)

Step 2: Select Form Category

Choose the appropriate category for your form:

- Table: For traditional table-based forms
- iFrame: For embedding external content

- **List Page:** For customizing list page columns (recommended for most use cases)

Step 3: Configure Form Type

If you selected **List Page** as the category:

1. Choose the **Type** from the dropdown menu
 - Currently supported: **Records**
 - More types will be added in future versions
2. The form will automatically populate with default columns for the selected type

Step 4: Customize Form Items

For each form item (column), you can configure:

Name

The internal identifier for the column. This corresponds to the data field being displayed.

Available columns for **Records** include:

- `organization`: Organization name
- `id`: Record ID
- `name`: Record name
- `clientIp`: Client IP address
- `createdTime`: Creation timestamp
- `provider`: AI provider name

- `provider2`: Secondary provider
- `user`: Associated user
- `method`: HTTP method
- `requestUri`: Request URI
- `language`: Language
- `query`: Query parameters
- `region`: Geographic region
- `city`: City
- `unit`: Unit information
- `section`: Section
- `response`: Response data
- `object`: Related object
- `errorText`: Error messages
- `isTriggered`: Trigger status
- `action`: Action column
- `block`: Block information
- `block2`: Secondary block information

Visible

Toggle to show or hide the column in the list page. Hidden columns are still available in the form configuration but won't appear in the UI.

Width

Set the column width in pixels. This helps optimize the display based on the content length and your screen size.

TIP

- Use narrower widths (90-120px) for short fields like IDs, dates, and status indicators
- Use wider widths (200-300px) for longer text fields like names, descriptions, and URLs
- Adjust widths based on your typical content length to avoid truncation

Step 5: Reorder Columns

Arrange columns in your preferred order:

1. Use the Up arrow button to move a column up
2. Use the Down arrow button to move a column down
3. The leftmost columns will appear first in the list page

Step 6: Add or Remove Columns

- **Add Column:** Click the **Add** button to create a new custom column
- **Remove Column:** Click the **Delete** button next to a column to remove it
- **Reset to Default:** Click **Reset to Default** to restore the original column configuration

Step 7: Preview and Save

1. View the **Preview** section at the bottom of the form editor
2. The preview shows how your form will appear in the actual list page
3. Click on the preview to open the full list page in a new window
4. Once satisfied, click **Save & Exit** to apply your changes

Editing Existing Forms

To modify an existing form:

1. Navigate to the Forms list page
2. Click on the form name or the edit button
3. Make your desired changes
4. Click **Save & Exit** to apply the updates

Changes take effect immediately for all users viewing the corresponding list page.

Form Categories in Detail

List Page Forms

List Page forms are the most commonly used type in Casibase. They allow you to:

- Customize which columns appear in list views
- Control column order and width
- Show/hide columns based on user needs
- Create optimized views for different workflows

CAUTION

When configuring List Page forms:

- The **Action** column is always displayed at the end, regardless of form item configuration
- At least one column should be visible for the list page to be functional

- Column names must match the actual data fields available in the backend

Table Forms

Table forms are used for structured data display in table format. Configuration is similar to List Page forms but may have different available fields based on the data source.

iFrame Forms

iFrame forms allow you to embed external content or applications within the Casibase interface:

1. Set the URL field to the external content address
2. Configure display settings as needed
3. The content will be displayed in an embedded frame

Vectors

Overview

Vectors Overview

Vectors Generation

The generation of vectors needs to be used in conjunction with stores, which means that you need to configure stores before you can understand vectors.

Overview

In Casibase, vectors are one of its core strengths. Vector technology plays a key role in knowledge representation and retrieval, and by pairing it with the stores feature, which converts data such as text and images into dense vectors, Casibase enables efficient similarity search and data analysis.

For information on the definition of vectors, see the [core-concepts](#) section in our previous documentation.

Application of vector technology in Casibase

Knowledge Embedding

Users can upload files in various formats (e.g. TXT, Markdown, Docx, PDF, etc.) and select embedding methods (e.g. Word2Vec, GloVe, BERT, etc.) to generate knowledge and corresponding vectors. These vectors are stored in a vector database for quick retrieval and query.

Similarity Search

Casibase converts the knowledge into vectors and stores them in a vector database. This vector representation supports a powerful similarity search function, which allows users to quickly find relevant information based on context or content.

When you chat with a knowledge base, the system automatically finds and ranks the most relevant document fragments based on semantic similarity. Each chat response includes a knowledge sources button that shows exactly which parts of

your documents were used to generate the answer, along with relevance scores displayed as percentages. This transparency helps you understand and verify the AI's reasoning while providing quick access to source materials.

Vectors Generation

The generation of vectors needs to be used in conjunction with stores, which means that you need to configure stores before you can understand vectors.

Vectors are actually the result of embedding, which is the process of converting various types of data, such as text and images, into dense vector representations. This step is essential to facilitate efficient data processing and analysis within Casibase. With embedding, questions in chat and knowledge files in storage will be converted into vectors that will be used in the next step of knowledge search.

1. Automatic Embedding

Casibase automatically generates vectors when you upload files to a store. After configuring a store with storage, model, and embedding providers, simply upload your files through the file tree interface. The system immediately queues them for embedding and processes them in the background.

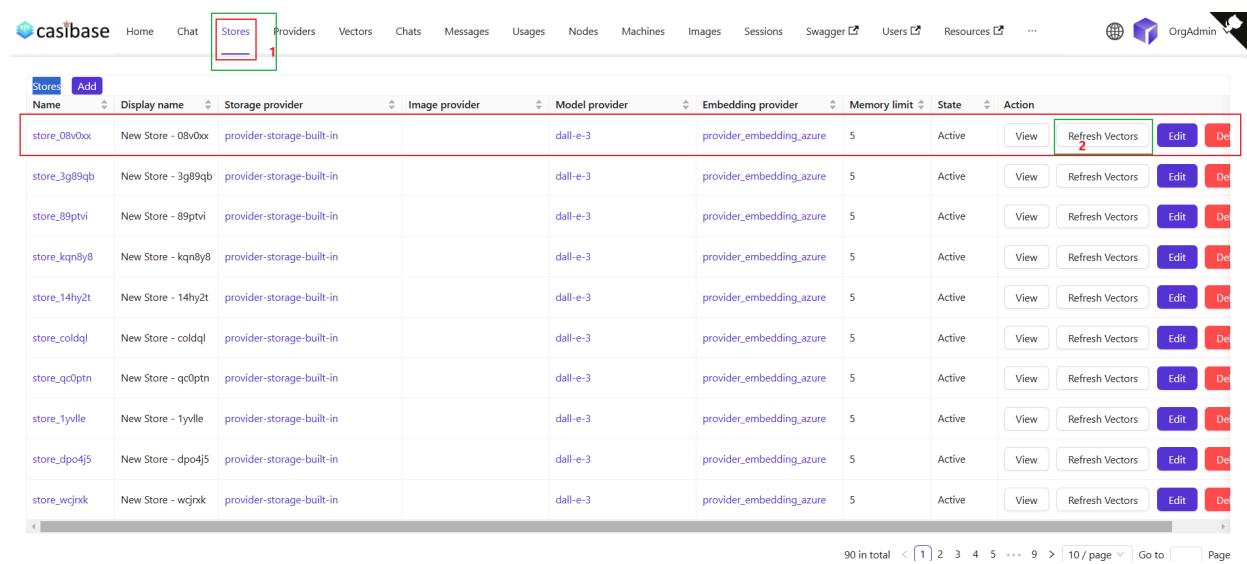
Each file progresses through states you can monitor in the Files page: Pending, Processing, Finished, or Error. The system calculates token counts during embedding and displays them alongside each file. If embedding fails for any file, the error details appear in the file list so you can quickly identify and fix issues like missing provider configurations or unsupported file formats.

When you delete files from storage, their vectors are automatically removed from the database. This keeps your knowledge base synchronized - you won't get outdated responses from files that no longer exist.

2. Refresh Vectors

The Refresh Vectors action is set as a button on each store data under the stores menu. After configuring stores with storage providers, navigate to the [Stores](#) page to view the file tree for the storage providers.

By clicking on the Refresh Vectors button for a particular store, it will generate the corresponding vectors for all the files in the file tree for that store by embedding them. The following figure shows the page and the operation.



The screenshot shows the Casibase Stores page. The 'Stores' tab is selected, indicated by a red box and the number '1'. The page displays a table of store configurations. The columns include: Name, Display name, Storage provider, Image provider, Model provider, Embedding provider, Memory limit, State, and Action. The 'Action' column contains three buttons: 'View', 'Refresh Vectors' (highlighted with a green box and the number '2'), and 'Edit/Delete'. A red box highlights the 'Refresh Vectors' button for the first store in the list. The table lists ten stores, each with a unique name and display name, and all are configured with 'provider-storage-built-in' as the storage provider. The 'Action' column for each row contains the 'View', 'Refresh Vectors', 'Edit', and 'Delete' buttons. The bottom of the page shows a navigation bar with '90 in total' and a page number '1'.

When you refresh vectors, the system automatically removes all existing vectors for that store before generating new ones. This ensures your vector database stays synchronized with your current files - if you've deleted files from storage, their vectors will be removed as well. The refresh always creates a clean, up-to-date vector set based on your current file tree.

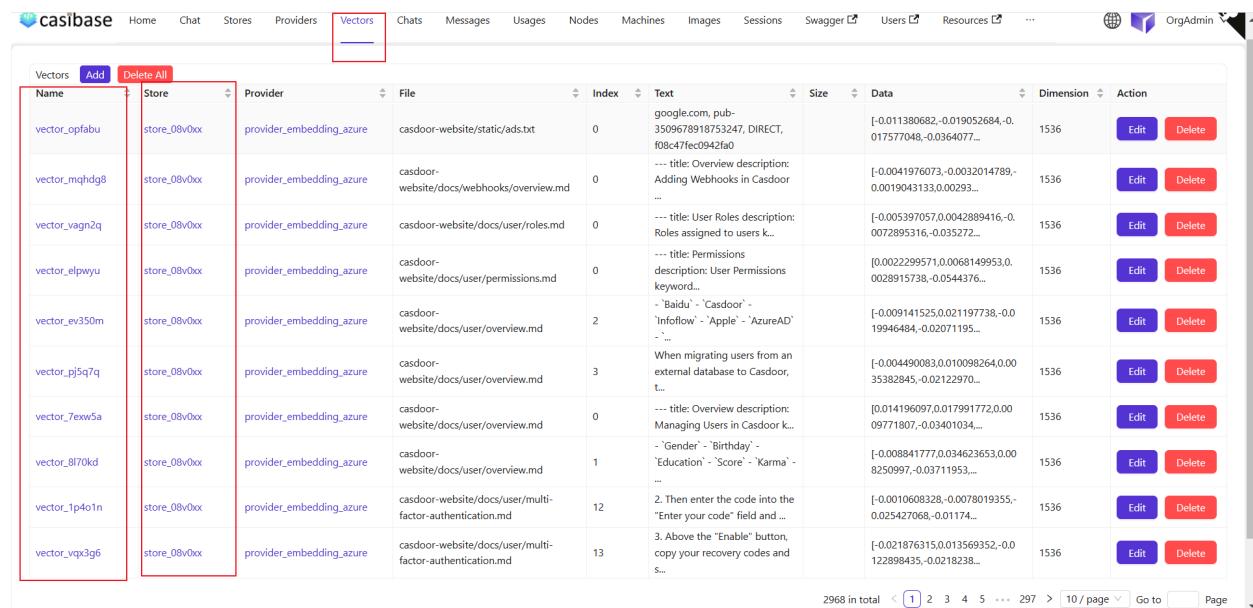
Re-embedding Individual Files

For targeted updates, use the "Refresh Vectors" button in the Files page next to

any specific file. This regenerates vectors for just that file while leaving others untouched - useful when you've changed embedding settings or need to fix errors on specific documents without reprocessing your entire knowledge base.

3. View vectors

After that, we can view the specific vectors generated by that store in the vector menu.



Vectors		Add	Delete All	Provider	File	Index	Text	Size	Data	Dimension	Action
vector_opfabu	store_08v0xx	provider_embedding_azure	casdoor-website/static/ads.txt	0	google.com, pub-3509678918753247, DIRECT, f08c47fe0c0942fa0		[-0.011380682,-0.019052684,-0.017577048,-0.0364077...]	1536			<button>Edit</button> <button>Delete</button>
vector_mqhdg8	store_08v0xx	provider_embedding_azure	casdoor-website/docs/webhooks/overview.md	0	--- title: Overview description: Adding Webhooks in Casdoor		[-0.0041976073,-0.0032014789,-0.00190431330.00293...]	1536			<button>Edit</button> <button>Delete</button>
vector_vagn2q	store_08v0xx	provider_embedding_azure	casdoor-website/docs/user/roles.md	0	--- title: User Roles description: Roles assigned to users k...		[-0.005397057,0.0042889416,-0.0072895316,-0.035272...]	1536			<button>Edit</button> <button>Delete</button>
vector_elpwu	store_08v0xx	provider_embedding_azure	casdoor-website/docs/user/permissions.md	0	--- title: Permissions description: User Permissions keyword...		[-0.002229951,0.0068149953,-0.0028915738,-0.0544376...]	1536			<button>Edit</button> <button>Delete</button>
vector_ev350m	store_08v0xx	provider_embedding_azure	casdoor-website/docs/user/overview.md	2	- 'Baidu' - 'Casdoor' - 'Infoflow' - 'Apple' - 'AzureAD' - '...		[-0.009141525,0.021197738,-0.19946484,-0.02071195...]	1536			<button>Edit</button> <button>Delete</button>
vector_pj5q7q	store_08v0xx	provider_embedding_azure	casdoor-website/docs/user/overview.md	3	When migrating users from an external database to Casdoor, t...		[-0.004490083,0.010098264,0.0035382845,-0.02122970...]	1536			<button>Edit</button> <button>Delete</button>
vector_7exw5a	store_08v0xx	provider_embedding_azure	casdoor-website/docs/user/overview.md	0	--- title: Overview description: Managing Users in Casdoor k...		[0.014196097,0.017991772,0.09771807,-0.03401034,...]	1536			<button>Edit</button> <button>Delete</button>
vector_ll70kd	store_08v0xx	provider_embedding_azure	casdoor-website/docs/user/overview.md	1	- 'Gender' - 'Birthday' - 'Education' - 'Score' - 'Karma' - '...		[-0.008841777,0.034623653,0.008250997,-0.03711953...]	1536			<button>Edit</button> <button>Delete</button>
vector_1p401n	store_08v0xx	provider_embedding_azure	casdoor-website/docs/user/multi-factor-authentication.md	12	2. Then enter the code into the "Enter your code" field and ...		[-0.0010608328,-0.0078019355,-0.025427068,-0.01174...]	1536			<button>Edit</button> <button>Delete</button>
vector_vqx3g6	store_08v0xx	provider_embedding_azure	casdoor-website/docs/user/multi-factor-authentication.md	13	3. Above the "Enable" button, copy your recovery codes and s...		[-0.021876315,0.013569352,-0.022898435,-0.0218238...]	1536			<button>Edit</button> <button>Delete</button>

We can see that the files in the stores from the previous step of refreshing vectors have been converted into vectors to display here.

The screenshot shows the casibase web interface with the 'Edit Vector' page open. The page includes fields for 'Name' (vector_opfabu), 'Display name' (google.com, pub-350967919175241), 'Store' (store_0bv0xx), 'Provider' (provider_embedding_azure), 'File' (casdoor-website/static/ads.txt), 'Text' (a large redacted block), 'Size' (1536), and 'Dimension' (a large redacted block). The 'Data' field contains a massive list of numerical values representing the vector data, which is also heavily redacted with a black bar.

The edit page of my vectors shows specific information such as the name of the store, the name of the embedding model, the name of the file in which the embedding was performed, the file size, the dimension, the vectors data, and so on.

TextSplitters

Overview

Text Splitters Overview

Overview

Text Splitters are a crucial component in building large language model (LLM) applications. Their primary role is to break long texts into multiple shorter segments, which facilitates subsequent tasks such as text embeddings, retrieval-augmented generation (RAG), and question-answering systems.

In LLMs, text splitting is performed for several main reasons:

- **Optimizing Efficiency and Accuracy:** By decomposing large blocks of text into smaller segments, the relevance and accuracy of the embeddings produced by the LLM can be optimized. Chunking helps ensure that the embedded content contains minimal noise while retaining semantic relevance. For instance, in semantic search, when indexing a document corpus, each document contains valuable information on specific topics. Applying an effective chunking strategy ensures that search results accurately capture the essence of a user's query.
- **Limiting the Context Window Size:** When using models like GPT-4, there is a limit to the number of tokens that can be processed. For example, GPT-4 has a context window size limit of 32K tokens. While this limit is generally not an issue, it is important to consider chunk size from the beginning. If the text chunks are too large, information might be lost or not all content may be embedded in the context, which can affect the model's performance and output.
- **Handling Long Documents:** While embedding vectors for long documents can capture the overall context, they might overlook important details pertaining to specific topics, leading to outputs that are either imprecise or incomplete. Chunking enables better control over the extraction and embedding of information, thereby reducing the risk of information loss.

Casibase currently offers multiple splitting methods, allowing users to apply

different processing strategies for various text scenarios.

Default Text Splitter

The default text splitter is designed to efficiently segment text based on token count and textual structure. Its splitting strategy includes:

- Line Reading and Paragraph Recognition: The text is read line by line, with consecutive blank lines used to accurately determine paragraph breaks. It also sensitively identifies natural breakpoints through markers, ensuring logical and precise text segmentation.
- Special Handling for Code Blocks: Code blocks enclosed by ``` symbols are treated separately. The number of lines within a code block determines whether it can stand alone as a segment. This mechanism preserves the integrity of code blocks while effectively preventing any single text segment from exceeding the token limit.
- Maintaining Sentence Integrity: Throughout the splitting process, strict adherence to sentence integrity is maintained, ensuring that sentences are never divided. This feature guarantees that each text segment contains a complete unit of information. Regardless of the complexity of the text, segmentation is accurately performed at sentence boundaries, effectively avoiding ambiguity and information loss due to broken sentences.

Q&A Splitter

The Q&A splitter focuses on the precise segmentation of question-and-answer formatted texts and offers the following core advantages:

- Accurate Splitting of Q&A Units: It uses a line-by-line scanning mechanism to intelligently identify the structure of Q&A texts. By determining whether each line begins with "Q:" or "A:", it precisely locates the boundaries between

questions and answers, ensuring that each Q&A pair is completely segmented. This guarantees the independence and completeness of each Q&A unit, providing clean data for subsequent Q&A processing and analysis.

- Clear and Logical Implementation: The code is simple and intuitive, making it easy to understand and maintain. By managing the state of the current Q&A pair and a flag indicating whether an answer is being collected, the process of text segmentation is clearly controlled, ensuring the correct pairing of each Q&A unit.

Chats



Overview

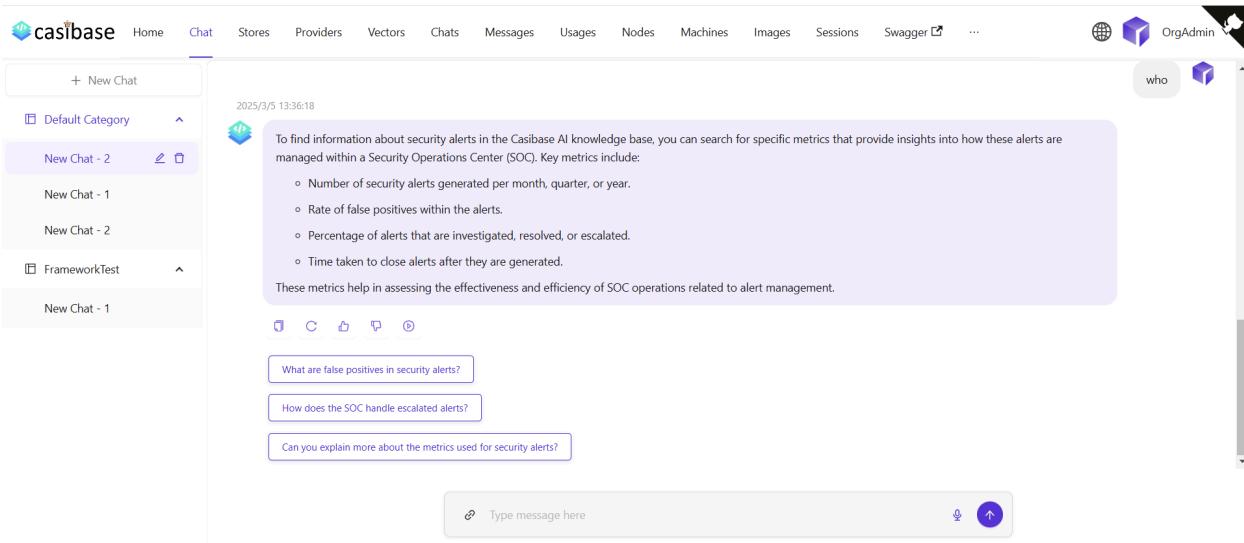
Chats Overview

Overview

In this section, we introduce the most central part of Casibase: chat and its management.

1. Chat

When you log in to Casibase, the chat interface is your starting point. This makes it easy to immediately begin conversations with AI without navigating through menus. Once we have configured the store, we can have a dialogue with the AI. This is shown in the image below:



The chat interface adapts to different AI capabilities. When using OpenAI or Alibaba Cloud providers, you'll see a web search toggle that lets the AI fetch current information from the internet. Search results appear inline with the response, complete with source URLs and timestamps for verification. For OpenAI's reasoning models (o1 and o1-mini), the chat displays the model's step-by-step thinking process in expandable sections, letting you follow along as it

works through complex problems.

Source Attribution

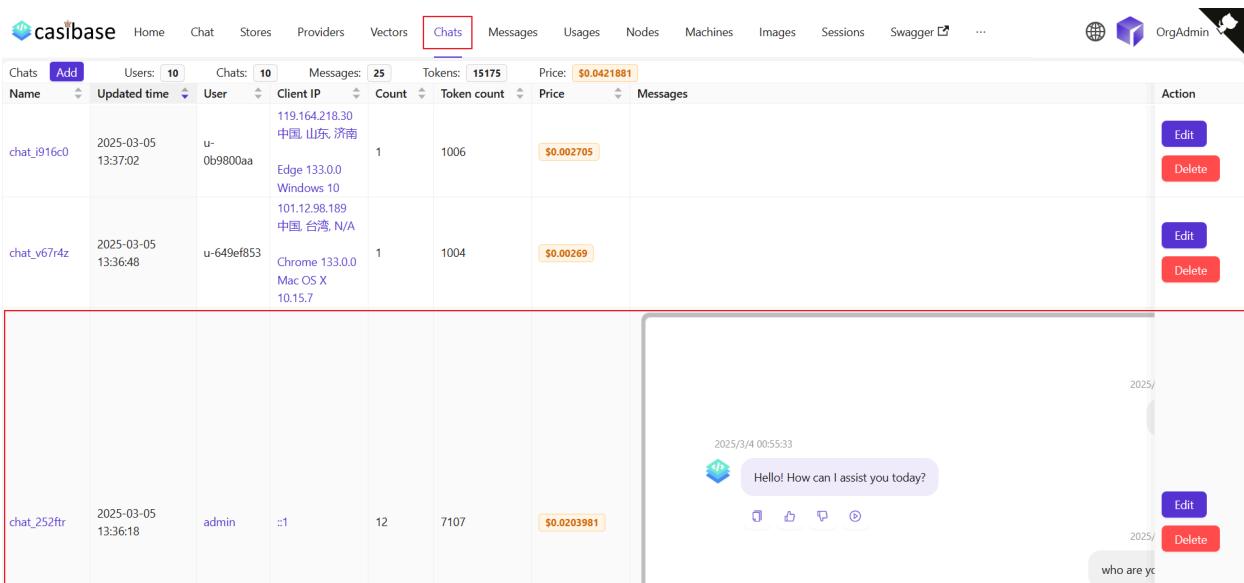
When the AI provides answers, you can trace where the information came from through two types of sources:

Knowledge sources appear when chatting with a store that has embedded documents. Click the button showing the number of knowledge sources to open a drawer that displays each relevant document fragment with its relevance score. You can click on any source to view the complete vector entry, or click on the file name/icon to jump directly to that file in the store's file viewer with the specific document pre-selected.

Web sources appear when web search is enabled. These show the websites the AI referenced when answering your question, complete with titles, URLs, and brief descriptions. The drawer displays search results with timestamps and allows you to visit the original sources.

2. Chats (Chat management)

We can manage our chat sessions from the Chats menu.



The screenshot shows the casibase Chats page. At the top, there are navigation links: Home, Chat, Stores, Providers, Vectors, Chats (highlighted in red), Messages, Usages, Nodes, Machines, Images, Sessions, Swagger, and ...

Below the navigation, there is a search bar with the text 'Users: 10 Chats: 10 Messages: 25 Tokens: 15175 Price: \$0.0421881'. The 'Chats' tab is selected.

The main table has columns: Name, Updated time, User, Client IP, Count, Token count, Price, and Messages. The table contains three rows of data:

- chat_j916c0 (2025-03-05 13:37:02, u-0b9800aa, 119.164.218.30, 中国 山东 济南, Edge 133.0.0, Windows 10, 1, 1006, \$0.002705)
- chat_v67r4z (2025-03-05 13:36:48, u-649ef853, 101.129.8.189, 中国 台湾, N/A, Chrome 133.0.0, Mac OS X 10.15.7, 1, 1004, \$0.00269)
- chat_252ftr (2025-03-05 13:36:18, admin, ::1, 12, 7107, \$0.0203981)

On the right side of the table, there is a detailed view of the third chat session (chat_252ftr). It shows a message from the AI: "Hello! How can I assist you today?". Below the message are icons for reply, like, dislike, and share. The message is timestamped "2025/3/4 00:55:33".

This page allows the user to view the information of the created chats, and the user can also click on Edit to view or edit them. Regular users can access and view their own chat sessions, while administrators have full access to all chats in the system for management purposes. They display the following information:

- **Name**: The name of the created chat.
- **Updated time**: The time when the chat is Updated.
- **User**: The user to whom the chat belongs.
- **Client IP**: Client IP of the chat.
- **Count**: Number of inputs and outputs for this chat.
- **Token count**: The total number of tokens used for this chat.
- **Price**: Total price spent on this chat.
- **Messages**: Showing the content of the chat's message.
- **Store**: Display the Store to which the chat belongs.
- **Model Provider**: The AI provider used for this chat (e.g., OpenAI, Claude).
- **Category**: Display the Category to which the chat belongs.

Token usage and costs are automatically tracked for billing purposes. See the [Billing & Usage](#) section for more details on how transactions are created and

managed.

Messages

Overview

Messages Overview

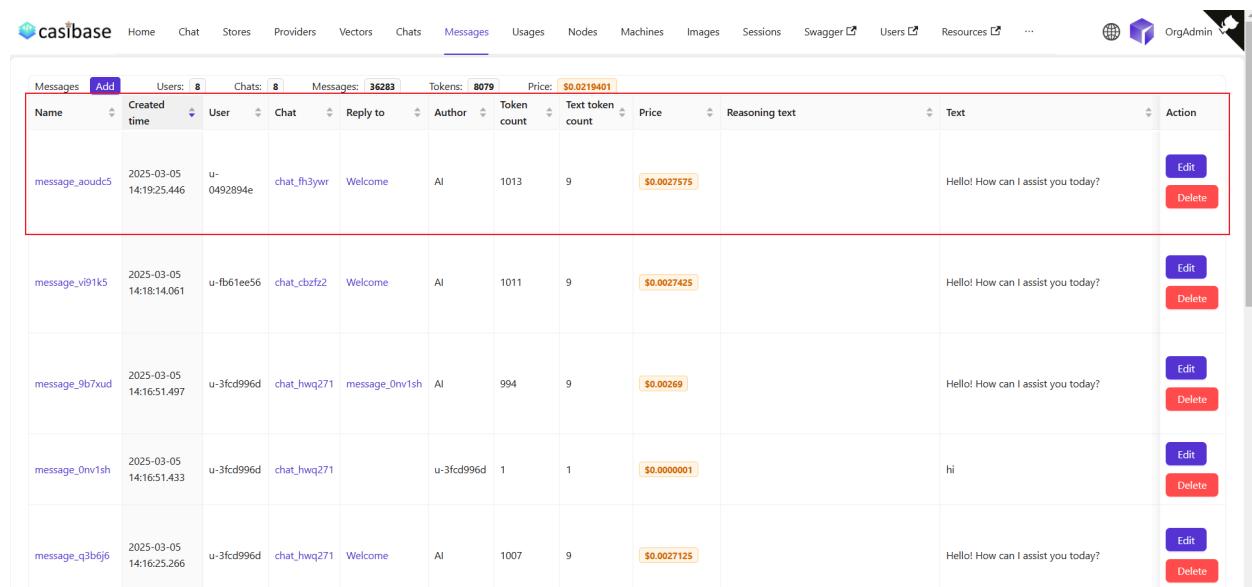
Overview

In this section, we introduce the functionality of message in Casibase.

Messages

The messages module manages all the messages in our sessions. Users can view and track their own message history, while administrators can access all messages across the system. Each message displays detailed metadata including creation time, associated chat, parent message references, token usage, pricing information, and the AI model provider that generated the response.

The model provider field identifies which AI service (OpenAI, Claude, etc.) processed each message. Token costs are automatically calculated and recorded as transactions for billing. Learn more about cost tracking in the [Billing & Usage](#) section.



Name	Created time	User	Chat	Reply to	Author	Token count	Text token count	Price	Reasoning text	Text	Action
message_aoudc5	2025-03-05 14:19:25.446	u-0492894e	chat_fh3ywr	Welcome	AI	1013	9	\$0.0027575		Hello! How can I assist you today?	<button>Edit</button> <button>Delete</button>
message_v191k5	2025-03-05 14:18:14.061	u-fb61ee56	chat_cbzfz2	Welcome	AI	1011	9	\$0.0027425		Hello! How can I assist you today?	<button>Edit</button> <button>Delete</button>
message_9b7xud	2025-03-05 14:16:51.497	u-3fcfd996d	chat_hwq271	message_0nv1sh	AI	994	9	\$0.00269		Hello! How can I assist you today?	<button>Edit</button> <button>Delete</button>
message_0nv1sh	2025-03-05 14:16:51.433	u-3fcfd996d	chat_hwq271		u-3fcfd996d	1	1	\$0.0000001		hi	<button>Edit</button> <button>Delete</button>
message_q3b6j6	2025-03-05 14:16:25.266	u-3fcfd996d	chat_hwq271	Welcome	AI	1007	9	\$0.0027125		Hello! How can I assist you today?	<button>Edit</button> <button>Delete</button>

Billing & Usage

Overview

Billing and Usage Tracking

Transactions

Transaction tracking and billing integration

Overview

Casibase tracks AI token usage and costs automatically through its integration with Casdoor. Every conversation with an AI model generates transaction records that capture consumption details for billing and analytics.

How It Works

When you send a message to an AI model, Casibase validates your balance before generating the response. The system estimates the token cost, checks if you have sufficient funds, and only proceeds if validation succeeds. This prevents situations where you receive an AI response but can't pay for it.

Chat → Balance Check → Message → Transaction

The validation happens in two steps. First, Casibase estimates the cost by asking the model provider how many tokens your question will likely consume. It then validates this estimated cost against your balance using a dry run transaction. If you have insufficient funds, you'll see an error immediately—no AI processing occurs. If validation succeeds, the AI generates your answer and the actual transaction records the real cost.

Each chat session maintains a running total of tokens used and costs incurred. Individual messages record their own token consumption, and transactions are created in Casdoor after the AI completes its response. This ensures accurate billing records and enables usage monitoring across your organization.

Token Tracking

Token usage is tracked at multiple levels:

- Chats display total tokens used and cumulative price for the entire conversation
- Messages show tokens consumed for each AI response
- Transactions record the billing details in Casdoor for centralized accounting

The model provider field identifies which AI service (OpenAI, Claude, etc.) handled each interaction, making it easy to understand usage patterns across different providers.

Price Calculation

Costs are calculated using proper decimal handling to prevent floating-point precision loss. The system automatically applies the correct pricing model based on the provider and model being used, ensuring accurate billing for all token usage.

Transactions

Transactions connect Casibase usage to Casdoor's billing system. Each AI-generated message automatically creates a transaction record that captures the associated costs and metadata.

Transaction Structure

Transactions use a structured format to track chat interactions:

- **Type:** Chat identifier - links the transaction to a specific conversation
- **Subtype:** Message identifier - tracks which message generated the cost
- **Provider:** Model provider name (e.g., "OpenAI", "Claude")
- **Amount:** Token cost calculated for the message

This structure provides complete traceability from billing records back to the original conversations and messages. Organizations can use this data for cost analysis, usage reporting, and budget tracking across teams.

Balance Validation

Before any AI processing begins, Casibase validates that you have sufficient balance to cover the estimated cost. The system asks the model provider to calculate how many tokens your question will likely consume, then checks this amount against your balance using a dry run transaction—a temporary validation that doesn't actually deduct funds.

If the validation fails, you'll receive an error message right away and the AI won't process your request. This saves you from waiting for an AI response only to

discover you can't afford it. Once validation succeeds, the AI generates your answer and a real transaction records the actual cost, which may differ slightly from the estimate based on the final response length.

Automatic Creation

Transactions are created after the AI completes your answer, ensuring that billing records stay synchronized with actual usage. You don't need to manually track or create transaction records - the system handles this automatically for every AI interaction.

Integration with Casdoor

Transaction records are stored in Casdoor, enabling centralized billing management across all your applications. This integration allows organizations to:

- Track AI usage costs alongside other application expenses
- Generate unified billing reports
- Set up usage alerts and budgets
- Analyze spending patterns across different AI providers

The transaction data flows seamlessly from Casibase to Casdoor, providing real-time visibility into AI usage costs without requiring manual data entry or reconciliation.

Records

Records

Data records and aggregation

Records

Records in Casibase support efficient data logging and analytics through a built-in aggregation mechanism. Instead of storing thousands of individual entries, you can create consolidated records that represent multiple data points.

Count Field

Records include a `Count` field that defaults to `1` for all new records. This field indicates how many actual data points the record represents.

When uploading consolidated data, set the count to match the number of underlying records. For example, if you have 100 similar log entries, create one record with `count: 100` rather than inserting 100 separate records. This approach significantly reduces storage requirements while maintaining accurate statistics.

API Behavior

The count field works consistently across all record operations:

- **Default value:** `1` for all new records created via API or UI
- **Backward compatibility:** Zero values display as `1` in the frontend
- **Type:** Integer field supporting positive values

When using the `add-record` or `add-records` API endpoints, include the count value to specify how many data points the record aggregates.

Use Cases

Records with aggregation are particularly useful for:

- **Log consolidation:** Summarize multiple similar log entries into a single record
- **Analytics:** Maintain accurate counts for reporting without storing raw data
- **System monitoring:** Track event frequencies without detailed individual records
- **Data archival:** Compress historical data while preserving statistical accuracy

The count field enables efficient data management by separating the logical representation (one record) from the actual count (potentially many occurrences).

Scans

Scans

Network and security scanning in Casibase

Scans

Scans in Casibase enable network discovery, security auditing, and system assessment across your infrastructure. The scan feature provides automated scanning capabilities for assets like virtual machines and network hosts, delivering structured results for analysis.

What is a Scan?

A Scan represents a scanning operation executed against a target asset. Each scan instance records the target, provider used, execution time, and results. Scans can be triggered manually or configured to run on demand against assets in your infrastructure.

Scan Workflow

Start by navigating to the Scans page and creating a new scan instance. You'll need to configure the target selection, choose a scan provider like Nmap or OS Patch, and set any necessary scan parameters.

When selecting your target, you can use **Asset Mode** to pick an existing asset from your inventory, or **Manual Mode** to enter an IP address or hostname directly. For Virtual Machine assets, the system automatically uses the public IP address from the asset properties.

Click the "Start" button to initiate the scan. The scan executes asynchronously, transitioning through states from Pending to Running to Completed. The Runner field tracks which system instance is actively executing the scan, useful in distributed deployments. If errors occur, the ErrorText field captures diagnostic

information to help troubleshoot issues.

The scan provider performs the configured operation against your target and returns results. These results appear in three formats: a **Structured View** with organized tables and formatted data, the **Raw JSON** showing parsed data structures, and the **Raw Text** showing the complete unprocessed output from the scan tool. All formats are stored in the database for historical reference, allowing you to review previous scans and track changes over time.

To reset a scan and clear its results, use the **Clear** button available on scan pages. This removes the scan state and results while preserving the scan configuration, letting you run a fresh scan with the same settings.

From the Scans list page, you can quickly review scan results without opening the full editor. The **Result** column provides a popup view displaying the same structured, JSON, and raw text formats available in the scan detail page.

Scan Providers

Casibase supports multiple scan provider types, each optimized for specific scanning tasks.

The **Nmap Scan Provider** performs network discovery and security auditing by scanning ports, detecting services, and identifying system information. It handles port scanning with customizable ranges, service version detection, operating system fingerprinting, and network topology mapping. Results come back as structured JSON containing host information, open ports, detected services, and system details, which the web interface renders in organized tables.

The **OS Patch Provider** checks system patch status and identifies missing security updates. The provider uses efficient local cache queries to list installed patches, ensuring fast scan performance without querying online update services. When

listing available patches for installation, the system does query Windows Update online to identify new updates.

This hybrid approach balances speed and accuracy, providing quick status checks while enabling comprehensive update discovery when needed. Results include patch status information, available updates, and security recommendations in a structured format for quick review.

Scan Configuration

When configuring a scan in the provider edit page, you can test the provider functionality directly. The scan configuration widget automatically selects the first available provider to streamline setup, and lets you switch between Asset and Manual Input modes for target selection, execute test scans to verify provider configuration, and view scan output before saving configurations. The system saves both the scan configuration and provider settings to the database before initiating scan execution, ensuring all settings are persisted.

For Manual Input mode, the target matching logic intelligently routes scans to the appropriate runner instance. When you specify a hostname as the target, the system matches it against each runner instance's hostname to ensure the scan executes on the correct machine. For IP address targets (excluding localhost), the scan routes to the instance whose network interfaces include that IP address, checking both private and public IPs.

localhost and loopback addresses like 127.0.0.1 can be claimed by any instance, allowing flexible local scanning. This ensures distributed scan deployments work correctly, with each instance claiming scans intended for its specific machine.

API Integration

Scans support programmatic access through REST APIs. Use `GET /get-scans` to retrieve all scans with pagination, or `GET /get-scan` to fetch a specific scan by ID. The `POST /add-scan` endpoint creates a new scan, while `POST /update-scan` modifies scan configuration and `POST /delete-scan` removes a scan.

The `POST /scan-asset` API performs on-demand scanning and returns results without creating a persistent scan record, making it useful for quick assessments.

Working with Scans

For cloud-based virtual machines, Asset Mode automatically resolves the correct public IP address, ensuring scans reach the intended target even as infrastructure changes. Schedule regular scans to maintain visibility into your security posture, but consider the impact on target systems when determining frequency.

When analyzing results, review both the structured and raw formats. The structured view provides quick insights, while raw output offers complete details for in-depth analysis. Choose Nmap for network discovery and port scanning, or OS Patch for security update assessment depending on what you need to learn about your systems.

Container Cloud

Overview

Container Cloud Overview

Template

In Casibase, a Template is a reusable base configuration for an application. It contains the core Kubernetes manifest files, typically structured for use with Kustomize. Y...

Applications

1 items

Overview

Once you have successfully connected Casibase to your private cloud providers (like Kubernetes), this section will guide you on how to manage cloud-native resources directly through the Casibase interface.

Casibase provides a powerful system based on Docker and Kubernetes, designed for individuals and organizations to build their own dedicated container cloud environment. Built on the Casbin permission management engine, it implements fine-grained access control policies for secure and controllable private cloud operations.

Core Concepts

Casibase utilizes a streamlined two-part system for managing container deployments:

1. Templates: Reusable Application Blueprints

Templates are pre-configured Kubernetes manifests that serve as blueprints for your applications. Each template contains:

- **Base Configuration:** Complete Kubernetes resources (Deployments, Services, ConfigMaps, etc.) required to run an application
- **Customizable Parameters:** Configurable fields that can be modified during deployment
- **Version Management:** Template versioning for consistent deployments across environments

2. Applications: Live Application Instances

Applications are running instances created from templates. They represent actual workloads deployed to your Kubernetes cluster:

- **Parameter Customization:** Override template defaults with specific configurations (replicas, image versions, resource limits)
- **Namespace Isolation:** Each application runs in its own dedicated namespace for security and organization
- **Lifecycle Management:** Complete application lifecycle control from deployment to termination

Key Features

Declarative Application Orchestration: Transform from resource-level management to application-level management, simplifying complex multi-resource deployments into simple "select template → configure → deploy" workflows.

Service Governance Integration: Built-in support for service mesh and gateway templates (Istio, Linkerd, Nginx Ingress) enabling one-click deployment of microservice governance capabilities including service discovery, circuit breaking, and rate limiting.

Enhanced Platform Visualization:

- **Application Dashboard:** Monitor and manage all your deployed applications with real-time status updates
- **Resource Insights:** Deep visibility into underlying Kubernetes resources, logs, and events for each application

Kustomize-Powered Flexibility: Leverages Kubernetes-native Kustomize for configuration management, ensuring consistency across development, testing, and production environments while maintaining the ability to customize deployments per environment.

This approach helps you standardize your infrastructure, ensure deployment consistency, and streamline the process of launching and managing containerized services. It eliminates the complexity of manual Kubernetes resource orchestration while maintaining full control over your applications.

Please proceed to the following sections to learn more about managing templates and applications:

- [Kubernetes Templates](#)
- [Kubernetes Applications](#)

Template

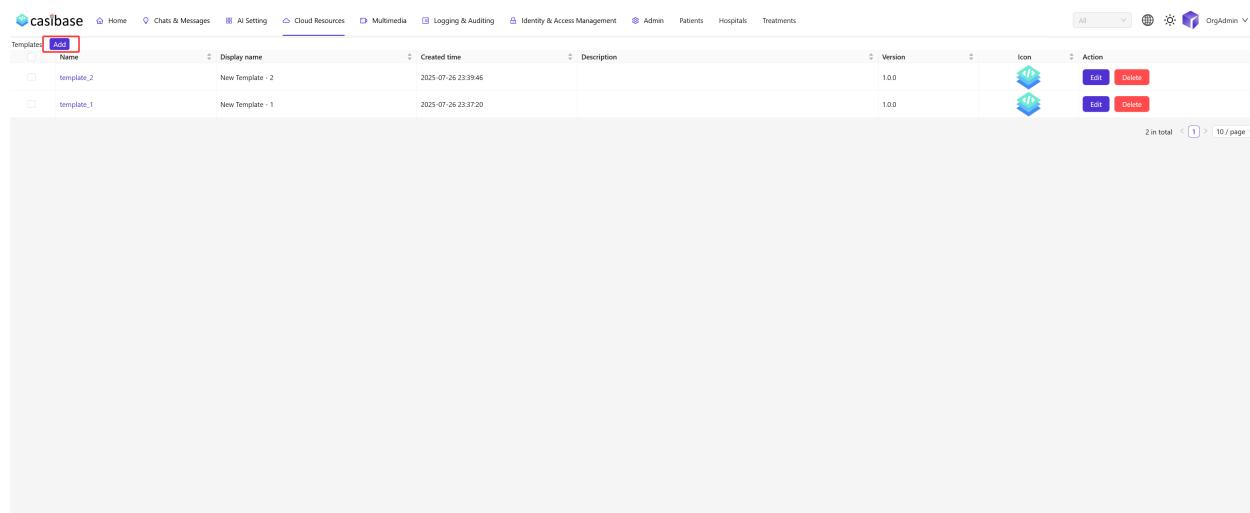
In Casibase, a **Template** is a reusable base configuration for an application. It contains the core Kubernetes manifest files, typically structured for use with Kustomize. You define a template once, and it can then be used as a blueprint to create multiple, customized application instances.

This model allows you to standardize your deployment patterns, ensuring consistency and simplifying the process of launching new services.

This chapter will guide you through creating and managing templates in Casibase.

Create a New Template

First, navigate to the **Cloud Resources > Templates** section and click the **Add** button to open the creation page.

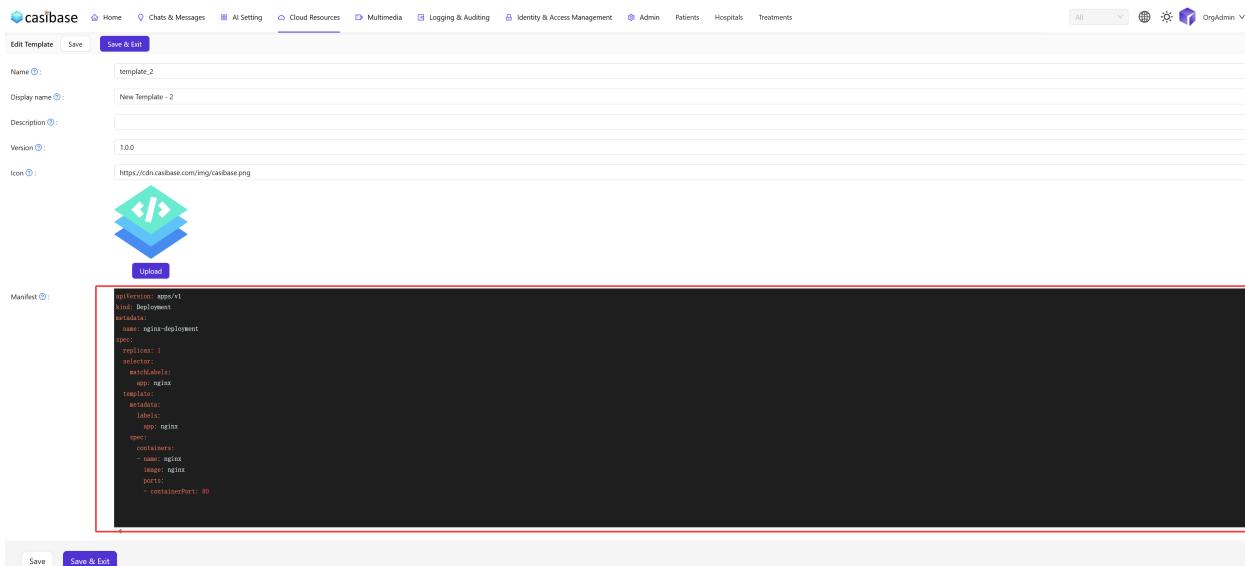


The screenshot shows the Casibase interface with the 'Cloud Resources' and 'Templates' sections selected. A red box highlights the 'Add' button in the top right corner of the list table. The table displays two existing templates: 'template_2' and 'template_1'. Each row includes columns for 'Name', 'Display name', 'Created time', 'Description', 'Version', 'Icon', and 'Action' (with 'Edit' and 'Delete' buttons). The bottom right corner of the page shows pagination information: '2 in total' and '10 / page'.

Name	Display name	Created time	Description	Version	Icon	Action
template_2	New Template - 2	2025-07-26 23:39:46		1.0.0		<button>Edit</button> <button>Delete</button>
template_1	New Template - 1	2025-07-26 23:37:20		1.0.0		<button>Edit</button> <button>Delete</button>

You will need to fill in the following fields, which correspond to the template's properties:

- **Name**: A unique identifier for the template (e.g., `my-app-template`). This is a required field.
- **Display name**: A user-friendly name that will be shown in the UI (e.g., `My App Template`).
- **Description**: A brief description of what this template is for.
- **Version**: The version of the template (e.g., `1.0.0`).
- **Icon**: A URL to an icon image that represents the template in the UI.
- **Manifest**: The raw YAML text of your Kubernetes manifests. This content serves as the base for Kustomize deployments.



```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        name: nginx
        image: nginx
        ports:
          - containerPort: 80

```

After saving, your template will be available in the selection list when you create a new application.

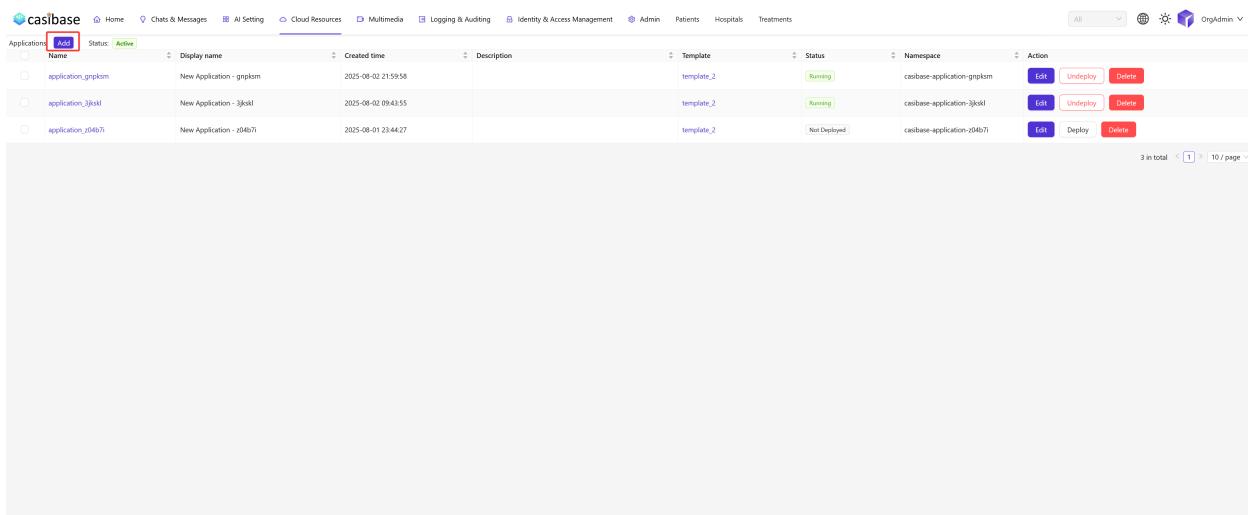
Application

An Application in Casibase is a specific, deployable instance created from a Template. Before you can create an application, you must first have at least one template defined. When you create an application, you select a base template and can then apply specific customizations before deploying it to your Kubernetes cluster.

This chapter will guide you through creating, deploying, and managing applications in Casibase.

Create a New Application

Navigate to the Cloud Resources > Applications section and click the **Add** button to open the creation page.



Name	Display name	Created time	Description	Template	Status	Namespace	Action
application_grpkbm	New Application - grpkbm	2025-08-02 21:59:58		template_2	Running	casibase-application-grpkbm	Edit Undeploy Delete
application_3jokl	New Application - 3jokl	2025-08-02 09:43:55		template_2	Running	casibase-application-3jokl	Edit Undeploy Delete
application_z0467i	New Application - z0467i	2025-08-01 23:44:27		template_2	Not Deployed	casibase-application-z0467i	Edit Deploy Delete

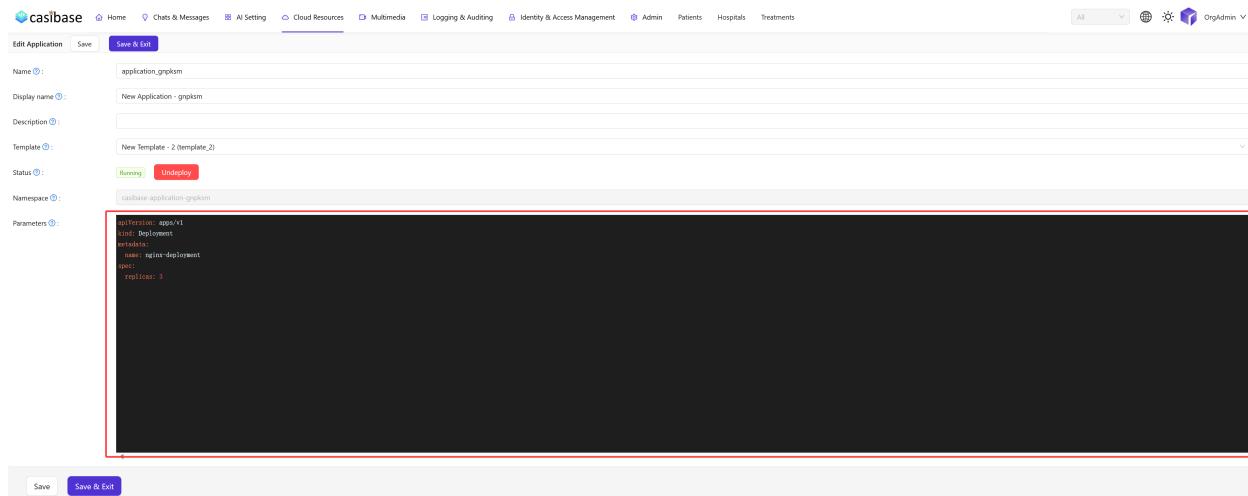
The key fields for an application are:

- **Name**: A unique name for your application instance (e.g., `my-app-prod`). This

is a required field.

- **Display name**: A user-friendly name that will be shown in the UI (e.g., **My App (Production)**).
- **Description**: A brief description of this specific application instance.
- **Template**: Select a pre-existing template from the dropdown list. This will be the base for your application.
- **Parameters**: This field is used for customization. Here you can provide specific Kustomize patches or other variable substitutions in YAML format to override or extend the base **Manifest** from the selected template.

Note: Fields like **Status** and **Namespace** are managed by the system. The **Namespace** is automatically generated based on the application name upon creation and cannot be modified by the user. The **Status** is updated based on its deployment state (e.g., **Not Deployed**, **Running**, **Pending**).



Deploying and Monitoring an Application

After creating an application, it will appear in the applications list. From here, you can manage its lifecycle.

- **Deploy:** Click the `Deploy` button to apply the application's configuration to your Kubernetes cluster. Casibase will use Kustomize to merge the base template's `Manifest` with your application's `Parameters` and run `kubectl apply`.
- **Undeploy:** The `Undeploy` button will remove the application's resources from your Kubernetes cluster.

Applications	Name	Display name	Created time	Description	Template	Status	Namespace	Action
	application_gnpkm	New Application - gnpkm	2025-08-02 21:59:58		template_2	Running	casibase-application-gnpkm	<button>Edit</button> <button>Undeploy</button> <button>Delete</button>
	application_3jkdl	New Application - 3jkdl	2025-08-02 09:43:55		template_2	Running	casibase-application-3jkdl	<button>Edit</button> <button>Undeploy</button> <button>Delete</button>
	application_z04b7i	New Application - z04b7i	2025-08-01 23:44:27		template_2	Not Deployed	casibase-application-z04b7i	<button>Edit</button> <button>Deploy</button> <button>Delete</button>

By using this template-and-application model, you can effectively standardize and scale your Kubernetes deployments through the Casibase interface.

Creating Databases with KubeBlocks

Overview

[KubeBlocks](#) is an open-source Kubernetes operator and toolset designed to simplify the complexity of running and managing data infrastructure, such as databases, message queues, and streaming systems, on Kubernetes. It provides a declarative approach to deploying and managing stateful applications, allowing you to manage them as easily as stateless ones.

This guide will walk you through the process of creating a database cluster using KubeBlocks.

Why KubeBlocks?

- **Simplified Management:** Automates the database lifecycle, including deployment, upgrades, scaling, and monitoring.
- **Production-Ready:** Supports high availability, backup and restore, and robust monitoring.
- **Versatile:** Supports a wide range of databases, including MySQL, PostgreSQL, MongoDB, Redis, and more.

Installing KubeBlocks

You can install KubeBlocks using [Helm](#). For more installation options, refer to the [KubeBlocks Installation Guide](#).

This guide will use Helm for the installation.

Step 1: Deploy Snapshot Controller

KubeBlocks requires the Snapshot Controller to manage volume snapshots. First, check if it is already installed in your cluster.

```
kubectl get crd volumesnapshotclasses.snapshot.storage.k8s.io
kubectl get crd volumesnapshots.snapshot.storage.k8s.io
kubectl get crd volumesnapshotcontents.snapshot.storage.k8s.io
```

If it is not installed, you can deploy it using the following commands:

```
helm repo add piraeus-charts https://piraeus.io/helm-charts/
helm repo update
# Update the namespace to an appropriate value for your
environment (e.g. kb-system)
helm install snapshot-controller piraeus-charts/snapshot-
controller -n kb-system --create-namespace
```

Then, verify the installation:

```
kubectl get pods -n kb-system | grep snapshot-controller
```

The Snapshot Controller should be in the `Running` state.

Step 2: Get the Latest KubeBlocks Version

Get the latest stable version tag (e.g. v1.0.1):

```
curl -s "https://api.github.com/repos/apecloud/kubeblocks/releases?per_page=100&page=1" | jq -r '.[] | select(.prerelease == false) | .tag_name' | sort -v -r | head -n 1

# Example output:
# v1.0.1
```

Step 3: Create KubeBlocks CRDs

Create the Custom Resource Definitions (CRDs) required by KubeBlocks.

```
# Replace <VERSION> with the version you selected
kubectl create -f https://github.com/apecloud/kubeblocks/releases/download/<VERSION>/kubeblocks_crds.yaml

# Example: If the version is v1.0.1
kubectl create -f https://github.com/apecloud/kubeblocks/releases/download/v1.0.1/kubeblocks_crds.yaml
```

Step 4: Install KubeBlocks with Helm

1. Add the KubeBlocks Helm repository:

```
helm repo add kubeblocks https://apecloud.github.io/helm-charts
helm repo update
```

2. Install KubeBlocks: This command installs the KubeBlocks chart into the `kb-system` namespace.

```
helm install kubeblocks kubeblocks/kubeblocks --namespace kb-system --create-namespace
```

Creating a Database

Once KubeBlocks is installed, you can create a database cluster using `kubectl`.

Currently supported databases include:

- MySQL
- PostgreSQL
- MongoDB
- Redis
- Kafka
- Milvus
- Qdrant
- RabbitMQ
- Elasticsearch

Example: Creating a Demo MySQL Cluster

1. Create a file named `my-mysql-cluster.yaml`:

```
apiVersion: apps.kubeblocks.io/v1
kind: Cluster
metadata:
  name: mycluster
  namespace: demo
spec:
  # Deletes all resources when the cluster is deleted
  terminationPolicy: Delete
  componentSpecs:
    - name: mysql
```

2. Apply the manifest to create the cluster:

```
kubectl apply -f my-mysql-cluster.yaml
```

More Information

For more detailed information, advanced configurations, and troubleshooting with KubeBlocks, please refer to the [official KubeBlocks documentation](#).

Nodes



Overview

Casibase nodes Overview



RDP

Casibase nodes RDP



VNC

Casibase nodes VNC

Overview

Casibase helps you to manage nodes, and connect to your nodes remotely, including remote desktop via RDP, VNC, SSH, and Telnet.

Protocol:

- SSH
- RDP
- VNC
- Telnet

Every node has the following basic properties:

- **Organization**: The organization that the node belongs to.
- **Name**: The unique node name.
- **Description**: The Description of the node.
- **IP**: Domain name or IP address.
- **Protocol**: The port number of the Protocol.
- **Port**: The port number of the node.
- **Username**: The username to connect to the node, such as `root`, `administrator`, `sa`, etc.
- **Password**: The password to connect to the node.
- **OS**: The operating system of the node, including `Windows` and `Linux`, used to classify the node.
- **Tag**: The tag of the node, used to classify the node.

In this chapter, you will learn how to start connecting to your nodes.

Let's explore together!

RDP

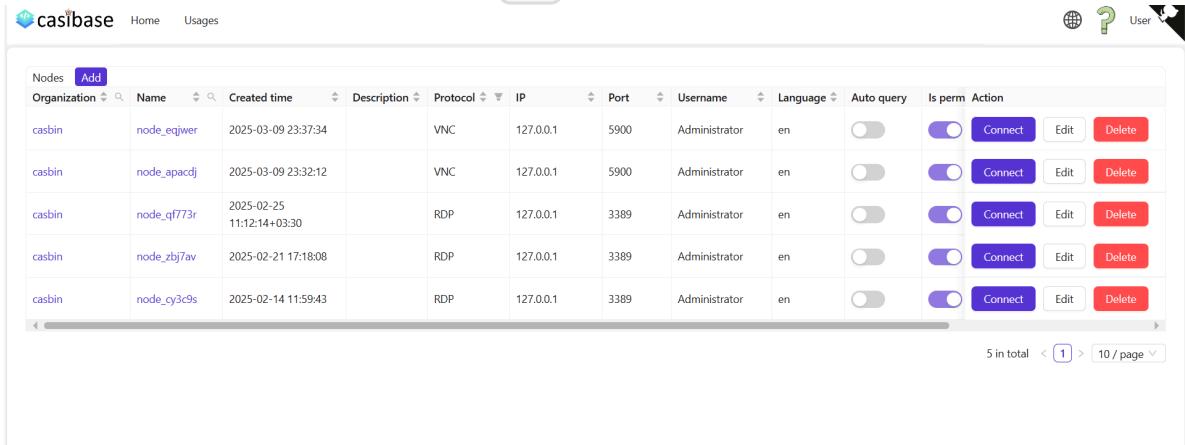
Casibase Support Connect to your nodes via RDP protocol:

Rdp connection

1. Start Guacamole Server

```
docker run --name guacd -d -p 4822:4822 guacamole/guacd
```

2. Add a new node, set protocol to `rdp`



Organization	Name	Created time	Description	Protocol	IP	Port	Username	Language	Auto query	Is perm	Action		
casbin	node_eqjwer	2025-03-09 23:37:34		VNC	127.0.0.1	5900	Administrator	en	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<button>Connect</button>	<button>Edit</button>	<button>Delete</button>
casbin	node_apacdj	2025-03-09 23:32:12		VNC	127.0.0.1	5900	Administrator	en	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<button>Connect</button>	<button>Edit</button>	<button>Delete</button>
casbin	node_qf773r	2025-02-25 11:12:14+03:30		RDP	127.0.0.1	3389	Administrator	en	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<button>Connect</button>	<button>Edit</button>	<button>Delete</button>
casbin	node_zbj7av	2025-02-21 17:18:08		RDP	127.0.0.1	3389	Administrator	en	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<button>Connect</button>	<button>Edit</button>	<button>Delete</button>
casbin	node_cy3c9s	2025-02-14 11:59:43		RDP	127.0.0.1	3389	Administrator	en	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<button>Connect</button>	<button>Edit</button>	<button>Delete</button>

The screenshot shows the 'Edit Node' page in the casibase interface. The node is named 'host-base' and is configured for RDP. The configuration fields include:

- Organization: casbin
- Name: host-base
- Description: 21212
- Protocol: RDP
- IP: 47.93.49.234
- Port: 3389
- Username: administrator
- Password: ***
- OS: Windows
- Tag: (empty)
- Language: en
- Auto query: On
- Is permanent: On

Services table:

No.	Name	Path	Port	Process ID	Expected status	Status	Message	Action

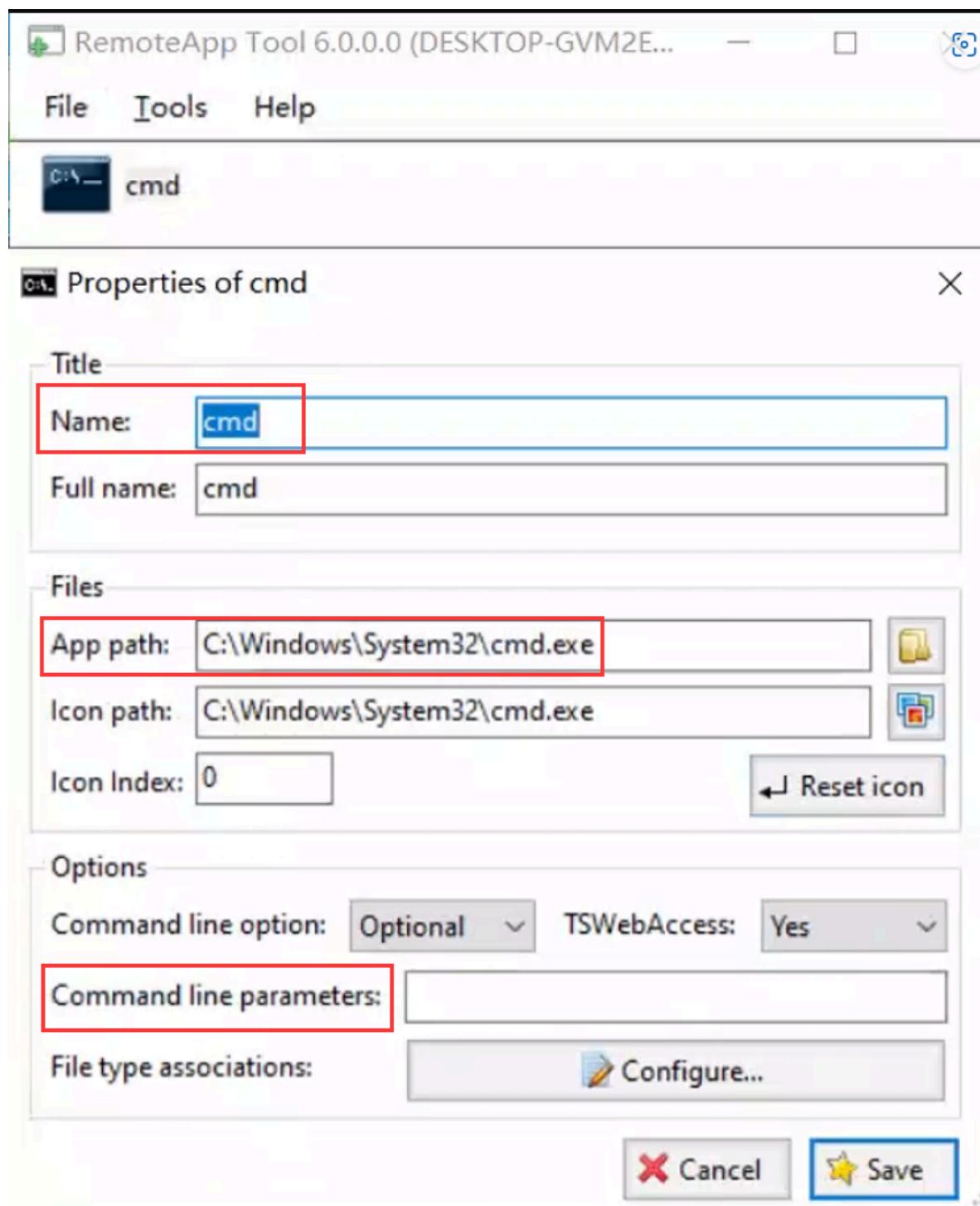
3. Connect to your node by clicking the **connect** button

Remote App

We support remote app on Windows nodes, you can add remote apps on **node Edit** page, and then you can connect to your remote app by clicking the **connect** button.

1. Configure your remote app on the server end.

You can use [RemoteApp Tool](#) to register apps.



2. Configure the remote app information in the node edit page according to the server-end configuration. 'remoteAppName', 'remoteAppDir', and 'remoteAppArgs' are required.



refer to [Configuring Guacamole — Apache Guacamole Manual v1.5.3](#)

3. Connect to your remote app.

VNC

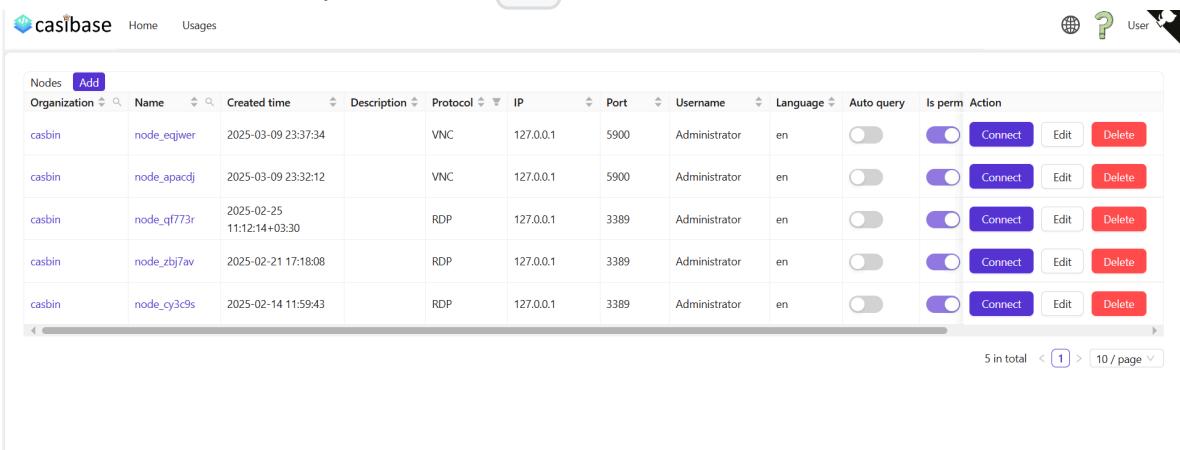
VCN Connect

VCN connection is similar to RDP connections.

1. Start Guacamole Server

```
docker run --name guacd -d -p 4822:4822 guacamole/guacd
```

2. Add a new node, set protocol to vnc



Nodes	Add	Organization	Name	Created time	Description	Protocol	IP	Port	Username	Language	Auto query	Is perm	Action
casbin			node_eqiwer	2025-03-09 23:37:34		VNC	127.0.0.1	5900	Administrator	en	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Connect
casbin			node_apacdj	2025-03-09 23:32:12		VNC	127.0.0.1	5900	Administrator	en	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Connect
casbin			node_qf773r	2025-02-25 11:12:14+03:30		RDP	127.0.0.1	3389	Administrator	en	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Connect
casbin			node_zbj7av	2025-02-21 17:18:08		RDP	127.0.0.1	3389	Administrator	en	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Connect
casbin			node_cy3c9s	2025-02-14 11:59:43		RDP	127.0.0.1	3389	Administrator	en	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Connect

3. Connect to your node by clicking the **connect** button.

Graphs

Graphs

Graph Visualization in Casibase

Graphs

Overview

Graphs in Casibase provide visual representation of relationships and connections between entities. The interactive graph interface allows you to explore complex data structures through an intuitive node-and-link visualization.

Asset Graph Transformation

Casibase can automatically generate graphs from cloud infrastructure assets. When you have assets defined (such as virtual machines, containers, or network resources), use the asset-to-graph transformation feature to visualize their relationships. The system analyzes asset properties to create nodes and establishes links based on resource dependencies and connections. This provides instant visibility into your infrastructure topology without manual graph definition.

Creating a Graph

Navigate to the Graphs section and click **Add** to create a new graph. Configure the graph with:

- **Name:** Unique identifier for the graph
- **Display Name:** Human-readable name shown in the UI
- **Category:** Select the graph category (Assets, Chats, etc.)
- **Data:** JSON structure defining nodes and links

Graph Categories

Casibase supports different graph categories for various data types:

Assets: Visualize cloud infrastructure and network topology. Define nodes and links manually or generate them automatically from asset data.

Chats: Analyze chat conversations with word cloud visualization. Select a store and time range to filter conversations, then generate word clouds from message content. The system automatically processes chat messages, removes stop words, and creates visualizations showing frequently used terms.

Graph Data Structure

Graphs use a JSON format with two main components:

Nodes: Individual entities in the graph

```
{  
  "id": "node1",  
  "name": "Node Name",  
  "label": "Display Label",  
  "icon": "icon-url"  
}
```

Links: Connections between nodes

```
{  
  "source": "node1",  
  "target": "node2",  
  "label": "Connection Type"  
}
```

Interactive Features

The graph visualization provides several interactive capabilities:

Node Selection: Click any node to view its details in a floating panel. Selected nodes are highlighted with a distinct visual style. The detail panel displays node metadata including all custom properties defined in the asset or node configuration, and shows related scan results when available. For assets with associated scans, you can review the scan history and results directly from the node detail view.

Canvas Navigation: Drag anywhere on the canvas to pan the view. The graph supports full directional movement for exploring large networks.

Visual Feedback: Nodes display custom icons and labels. Links connect at node edges rather than centers for cleaner visualization. When assets have associated scan results, the node displays a notification badge showing the scan count. The badge appears as a red pill-shaped indicator with white text, similar to iOS app notifications, making it immediately obvious which nodes have scan data available without needing to click through each one.

Layout Controls

Graphs support multiple layout algorithms to organize nodes:

- **Force-directed layout:** Nodes automatically arrange based on their connections, creating organic-looking graphs where connected nodes pull together while maintaining spacing.
- **Word Cloud layout:** Available for Chats category graphs, displays words sized by frequency for visual analysis of conversation topics.

- **None layout:** Positions nodes based on provided coordinates, useful when you want manual control over the exact placement.

Chat Analysis with Word Clouds

For Chats category graphs, Casibase provides specialized visualization through word clouds. Configure these graphs by:

1. **Selecting a Store:** Choose which chat store to analyze
2. **Setting Time Range:** Define start and end times to focus on specific periods
3. **Viewing Chat Data:** Review the filtered chats and messages in table format
4. **Generating Visualizations:** The system automatically creates word clouds from message content

The word cloud generation process extracts text from all messages in the selected time range, removes common stop words (in English and Chinese), and calculates word frequencies. Words appear larger based on how often they occur in the conversations, making it easy to identify key topics and themes.

Density Control

For graphs with many nodes, use the density slider to adjust node spacing. Higher density brings nodes closer together for a compact view, while lower density spreads them out for better readability. This is particularly useful when working with auto-generated graphs from cloud infrastructure.

Graph Editing

In edit mode, you can modify the graph structure and preview changes:

- Edit the JSON data to add, remove, or modify nodes and links

- Preview shows real-time visualization with a bordered canvas
- Changes are saved when you click `Save & Exit`

Node Properties

Each node can have custom properties:

- **Icon:** URL to an icon image displayed on the node
- **Label:** Text displayed below the node
- **Properties:** Additional metadata shown in the detail panel

When viewing node details, the panel shows comprehensive information about the selected node. For asset-derived graphs, this includes all metadata from the original asset such as IP addresses, resource IDs, locations, and other cloud-specific properties. Property values are displayed in an organized format with proper label formatting. Long text values are automatically truncated with ellipsis for readability.

For assets with scan results, the detail panel includes a dedicated section showing related scans. You can view scan execution history, status information, and access the full scan results without leaving the graph view. This integration provides immediate visibility into the security and operational status of your infrastructure assets as you explore the topology.

Error Handling

Graphs include built-in error handling. If data cannot be loaded or visualization fails, an error message displays in place of the graph, allowing you to identify and fix configuration issues.

Customization

The graph visualization adapts to the Casibase theme, using consistent colors and styling throughout the interface. Selected nodes use theme colors for visual emphasis while maintaining clarity.

Best Practices

Keep graph structures focused and organized. For complex networks, consider creating multiple smaller graphs rather than one large visualization. This improves performance and makes relationships easier to understand.

Use descriptive node labels and meaningful link labels to make the graph self-explanatory. Clear naming helps users quickly grasp the relationships without additional documentation.