AI Bench Studio Usage Guide

V 1.4

Table of Contents

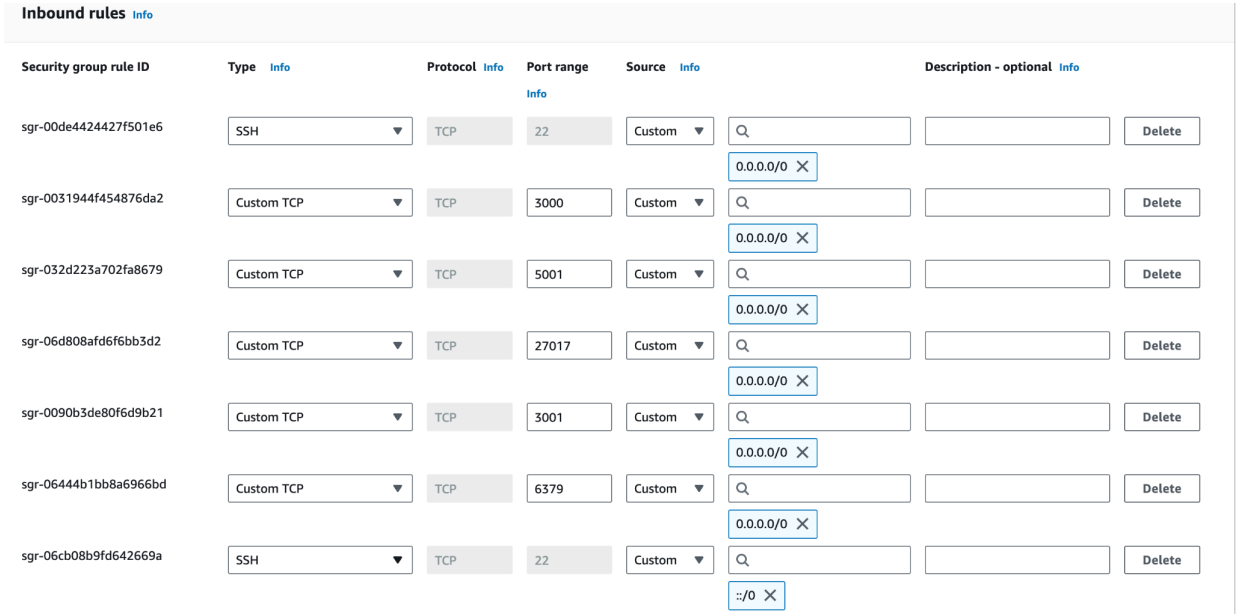
This guide provides a step-by-step process for setting up AI Bench Studio. The instructions include configuring the application and setting up benchmarking using YCSB. Additionally, detailed information about the configuration and functionalities of the system is provided to ensure a seamless experience.

Open Ports Through AWS Security Group [Optional]

This port-open step only follow if you're using AWS EC2

Log into your AWS Console and navigate to the Security Group Settings. Make sure to open the necessary ports for the application (this will be up to the customer's discretion, but at minimum the ports must be available to the IP of your AWS machine):

- ☐ 3000: Frontend GUI Interface for AIBS
- ☐ 3001: Grafana Server built into AIBS
- ☐ 5001: Backend Executable
- ☐ 6379: Local Redis DB Server for Testing
- ☐ 27017: MongoDB for application storage



Application Configuration Container

Supported Platforms

This guide provides instructions for running AWS Elastic Container Registry (ECR) commands on the following platforms:

- Windows
- macOS
- Ubuntu

Pre-requisites

Before running the commands, ensure the following tools are installed and configured:

1. Docker

- Install Docker following the official instructions:
    - Docker for Windows
    - Docker for macOS
    - Docker for Ubuntu

Verify Docker installation:

docker --version

2. AWS CLI

- Install AWS CLI using the official documentation:
    - AWS CLI Installation

Verify AWS CLI installation:

aws --version

AWS Configuration

Before using AWS ECR, you need to configure AWS CLI with your credentials.

Run the following command to configure AWS CLI:
aws configure

1. Provide the following information when prompted:
    - **AWS Access Key ID**: Your AWS access key.
    - **AWS Secret Access Key**: Your AWS secret key.
    - **Default region**: `us-east-1` (or your desired region).
    - **Default output format**: `json` (or your preferred format).

Pulling Docker Images from AWS ECR

Step 1: Authenticate Docker with AWS ECR

Run the following command to authenticate Docker to an AWS ECR registry:

aws ecr get-login-password \

--region us-east-1 | docker login \

--username AWS \

--password-stdin 709825985650.dkr.ecr.us-east-1.amazonaws.com

On Windows OS, use the below command:

aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 709825985650.dkr.ecr.us-east-1.amazonaws.com

Step 2: Define the Container Images

Specify the container images you want to pull:

CONTAINER_IMAGES="709825985650.dkr.ecr.us-east-1.amazonaws.com/bhojr/ai-bench-studio:1.4.4"

On Windows OS, use the below command:

$CONTAINER_IMAGES = "709825985650.dkr.ecr.us-east-1.amazonaws.com/bhojr/ai-bench-studio:1.4.4"

Step 3: Pull the Docker Images

Run the following loop to pull each image:

for i in $(echo $CONTAINER_IMAGES | sed "s/,/ /g"); do docker pull $i; done

On Windows OS, use the below command:

$CONTAINER_IMAGES -split ',' | ForEach-Object { docker pull $_ }"

Step 4: Run the Docker Images

Create config.json:

touch config.json

Run the following command to run the AIBS container:

docker run -d \

-v ./config.json:/config.json -v ./config.json:/app/Ai-Bench-Studio/config.json -v ./config.json:/app/config.json \

-p 6379:6379 -p 3000:3000 -p 3001:3001 -p 5001:5001 \

--name aibs-app \

709825985650.dkr.ecr.us-east-1.amazonaws.com/bhojr/ai-bench-studio:1.4.4

To collect values about config.json, have to follow these steps: CONFIG.JSON steps

Once config.json setup is completed, you can run below command

docker restart aibs-app

Troubleshooting

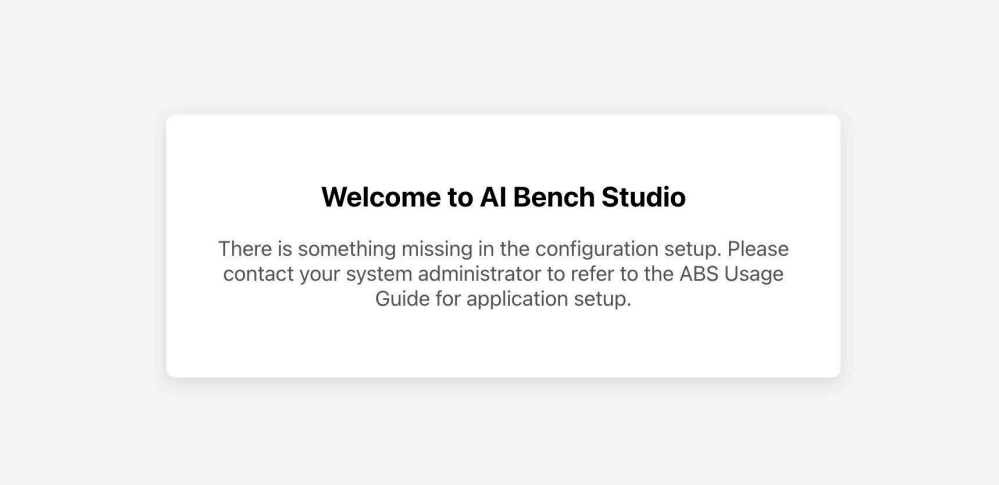
- **Authentication Errors**: Ensure your AWS credentials are correctly configured and have the necessary permissions to access the ECR registry.
- **Docker Not Found**: Verify Docker is installed and running.
- **Region Mismatch**: Make sure the specified region (`us-east-1`) matches the region of your ECR repository.

Navigate to the Frontend UI

Once ports are opened, you can access the frontend by visiting:

```
http://<EC2_IP>:3000
```

On the first visit, you will be directed to the admin configuration warning screen. During a first-time setup this is expected.



Once you see this screen, you can continue to the next section to configure the application. If you see this screen after application setup, please follow the below steps again.

CONFIG.JSON Setup

The config.json file will store all relevant configuration settings for the application to run. Open the `config.json` file using your preferred text editor (Please refrain from making changes in the config directory, only the config.json):

```
vi config.json
```

Here is the base schema of config.json.

```
{
  "RBAC_IPV4": "<rbac_ipv4>:<backend_port>",
  "ABS_AWS_EC2_IP": "<aibs_ipv4>",
  "NavBarOrientation": "vertical",
  "GRAFANA_API_KEY": "glsa_5ecret",
  "TEST_REDIS_SERVER": "start",
  "MONGO_URI": "mongodb://localhost:27017/admin",
  "LICENSE_KEY": "<Licence key>"
}
```

After opening, we will need to set the following configuration values:

a. "RBAC_IPV4"-> This will come from the AWS Instance page of your RBAC subscription. Please include the full name with port (default for RBAC is :3000).
Example:

```
"RBAC_IPV4":"ec2-1-2-3-4.compute-1.amazonaws.com:3000"
```

b. "ABS_AWS_EC2_IP" -> This will come from the AWS Instance page and is the the IPv4 address of your AIBS Machine
Example:

```
"ABS_AWS_EC2_IP": "1.2.3.4"
```

c. "LICENSE_KEY" -> You can get your license to use the platform from https://www.baideac.com/licensing

1. Create account
2. Request Trial Key
   - Select “AIBS”
   - Click on Request.
3. In the listed table, you can get the licence key.

Example:

```
"LICENSE_KEY": "ortaread:prtywfgswr"
```

After this is completed, save the config file. And restart the docker container.

Configure Grafana API Key

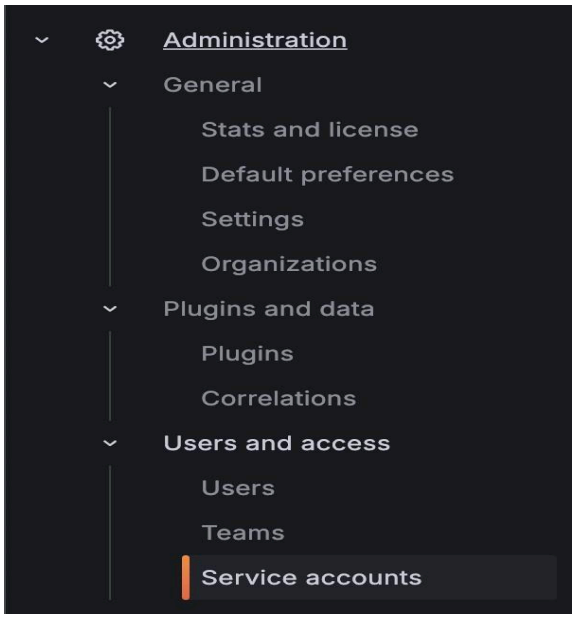
When starting the application for the first time, a one time setup process must be done in order for the Grafana integration to work. This involves generating an API key and setting the datasource for Grafana. This section will outline the steps for generating an API key for Grafana.

1. Navigate to Grafana Admin UI at `http://<EC2_IP>:3000/settings`
   Make sure this port is open for this operation, after the one time setup, you may restrict it back to the minimum security requirements in the first section.

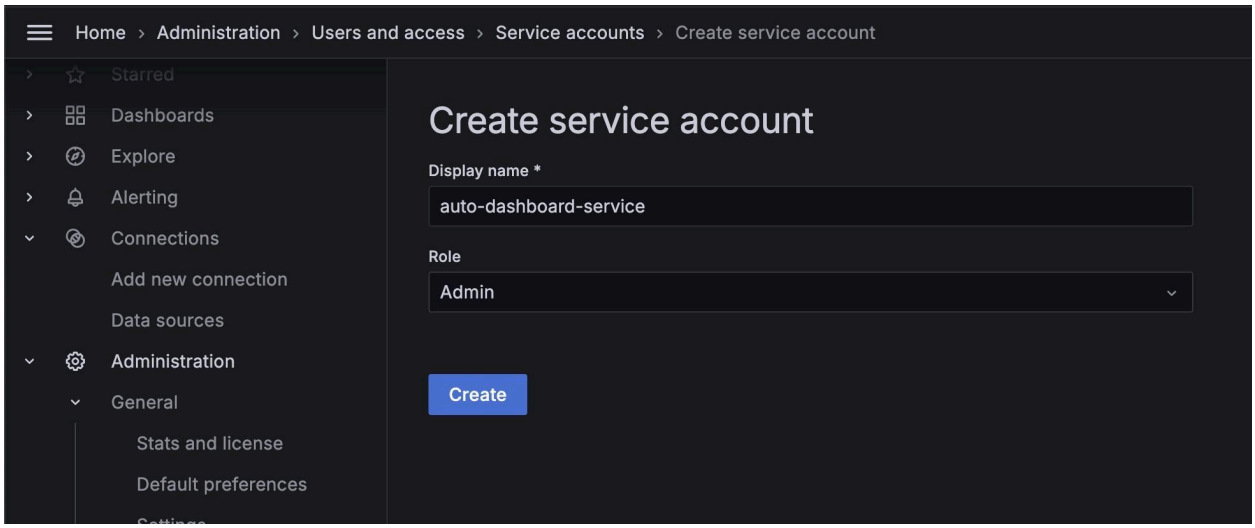
2. Log in with the default credential. These credentials can be changed as there will be a prompt to do so. While this is not necessary, it is recommended to change and store this password for the system admin.

   Username: admin
   Password: admin

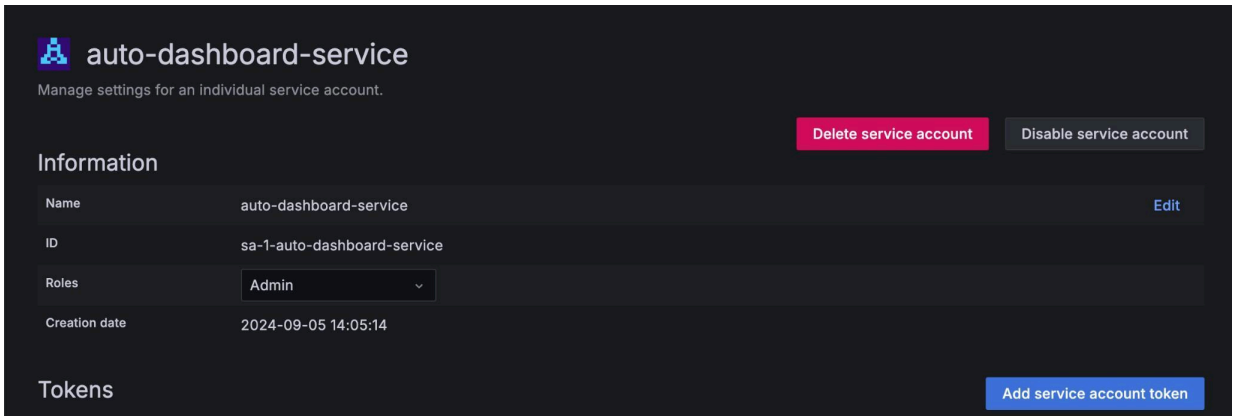
3. Once logged in as admin, using the sidebar navigate to Administration > Users and Access > Service Accounts



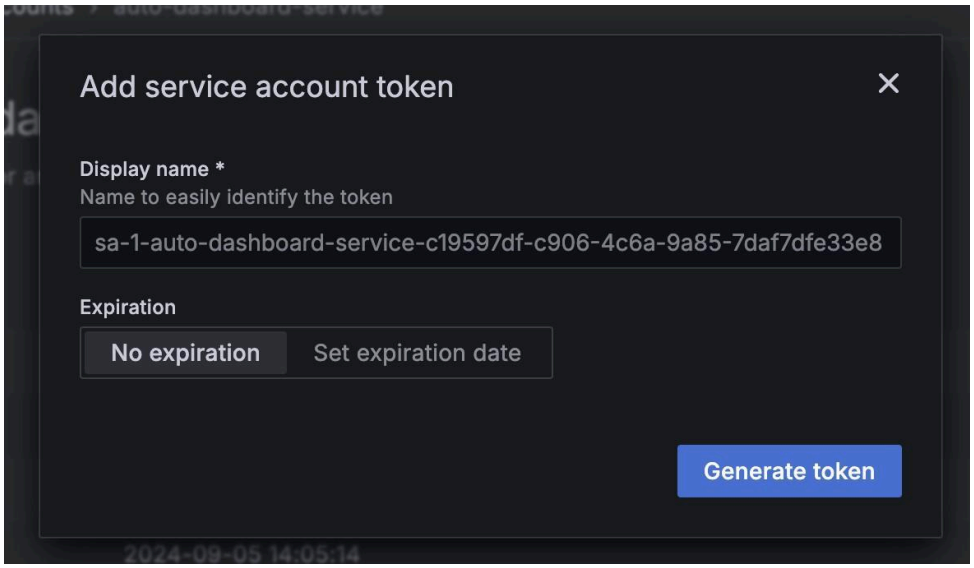
4. Click Add Service Account button, name it `auto-dashboard-service`, and assign the role Admin. Then click the Create Button.



5. Once created, click the “Add service account token” Button.



6. In the following screen, ensure the display name is left as default and the expiration is set to “No expiration”. Once this is done, click the Generate token button.



7. Copy the token and add it to `config.json`:
   Example: `"GRAFANA_API_KEY": "glsa_1234567890"`

Configure Grafana Datasource

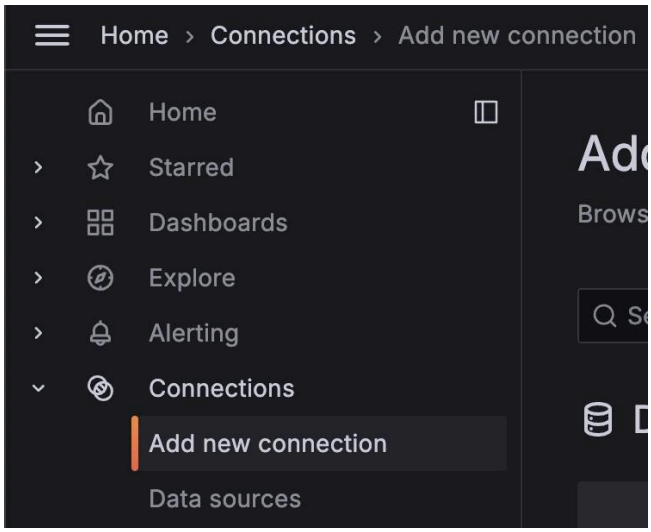
When starting the application for the first time, a one time setup process must be done in order for the Grafana integration to work. This involves generating an API key and setting the datasource for Grafana. This section will outline the steps for connecting the datasource for Grafana.

1. Navigate to Grafana UI at `http://<EC2_IP>:3000/settings`
   Make sure this port is open for this operation, after the one time setup, you may restrict it back to the minimum security requirements in the first section.

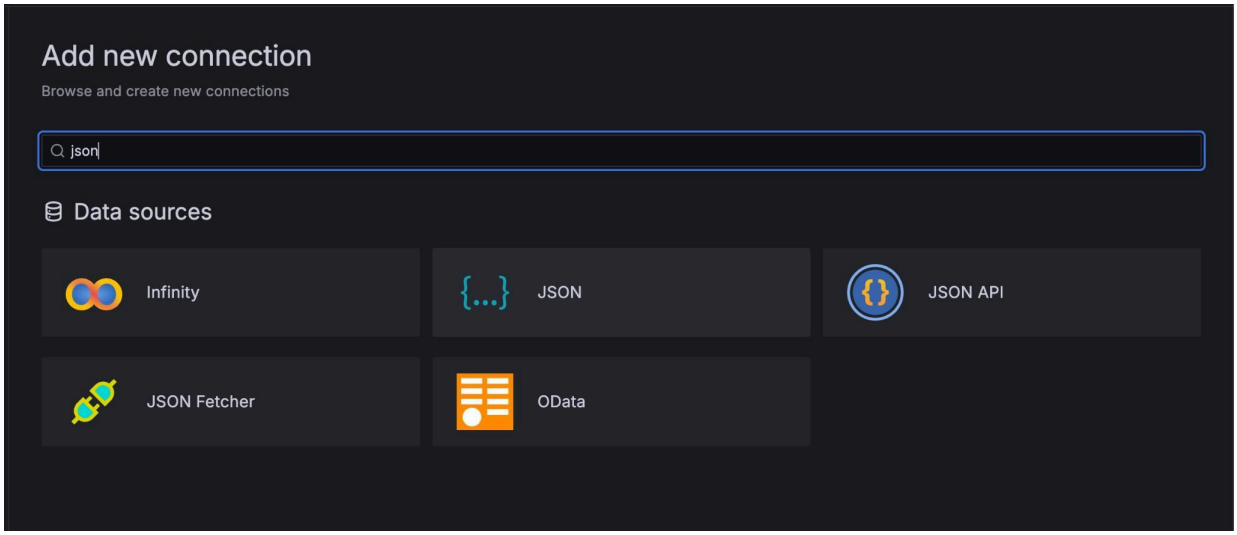
2. Log in with the default credential. These credentials can be changed as there will be a prompt to do so. While this is not necessary, it is recommended to change and store this password for the system admin.

   Username: admin
   Password: admin

3. Once logged in as admin, using the sidebar navigate to Administration > Connections > Add new connection

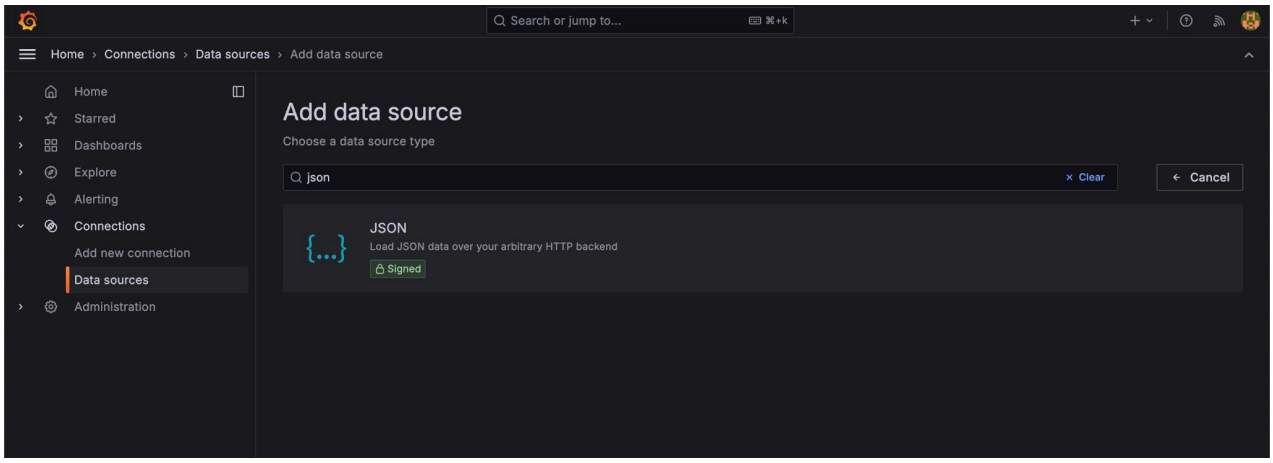


4. Search for “JSON” and select the data source. When selected, there will be a button to install. Click Install.

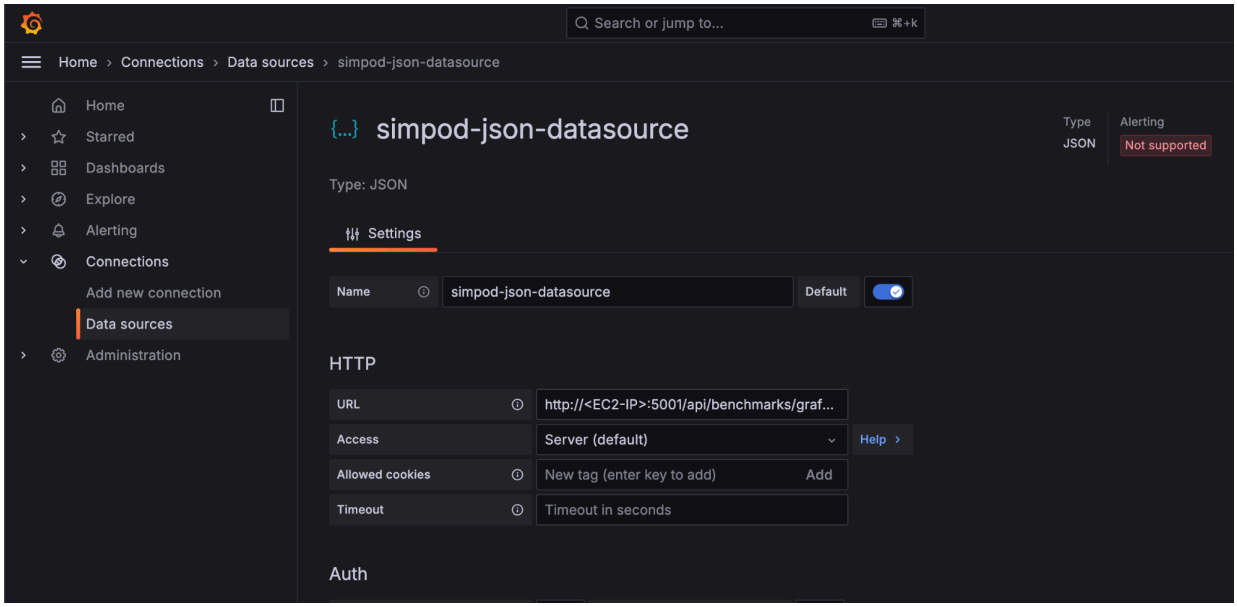


5. Using the sidebar navigate to Administration > Connections > Data Sources and click on the “Add New Data Source” Button.

6. Search for the JSON data source and click on it.



7. Update the following inputs:
    - Name: `simpod-json-datasource`
    - URL: `http://<EC2_IP>:5001/api/benchmarks/grafana-data`



8. Click Save & Test at the bottom of the page (ignore the "Not Found" error message, as it is expected).

After completion of the above steps, all the configuration for Grafana is complete.

Benchmarking with YCSB

YCSB (Yahoo! Cloud Serving Benchmark) is a framework used for evaluating the performance of various database systems under different workloads. It allows users to simulate different operations such as reads, writes, updates, and scans to benchmark the database and application performance.

Running YCSB Benchmarks in AI Bench Studio

In AI Bench Studio, you can run YCSB benchmarks using custom configurations:

- ☐ Benchmark Name: Specify any custom name for the benchmark.
- ☐ Workload Type: Define the type of workload (predefined workloads from YCSB or custom data sets).
- ☐ Operation Count: The total number of operations to be performed during the benchmark (e.g., 1000 operations).
- ☐ Read, Update, Insert, Scan Proportions: These are the proportions of the operation count for the CRUD operations. These must add up to 1. For example, if the Read Proportion is 0.5 and the Operation Count is 1000, it means 500 read operations will be performed.
- ☐ DB Configuration: Select the DB configuration to run the benchmark.

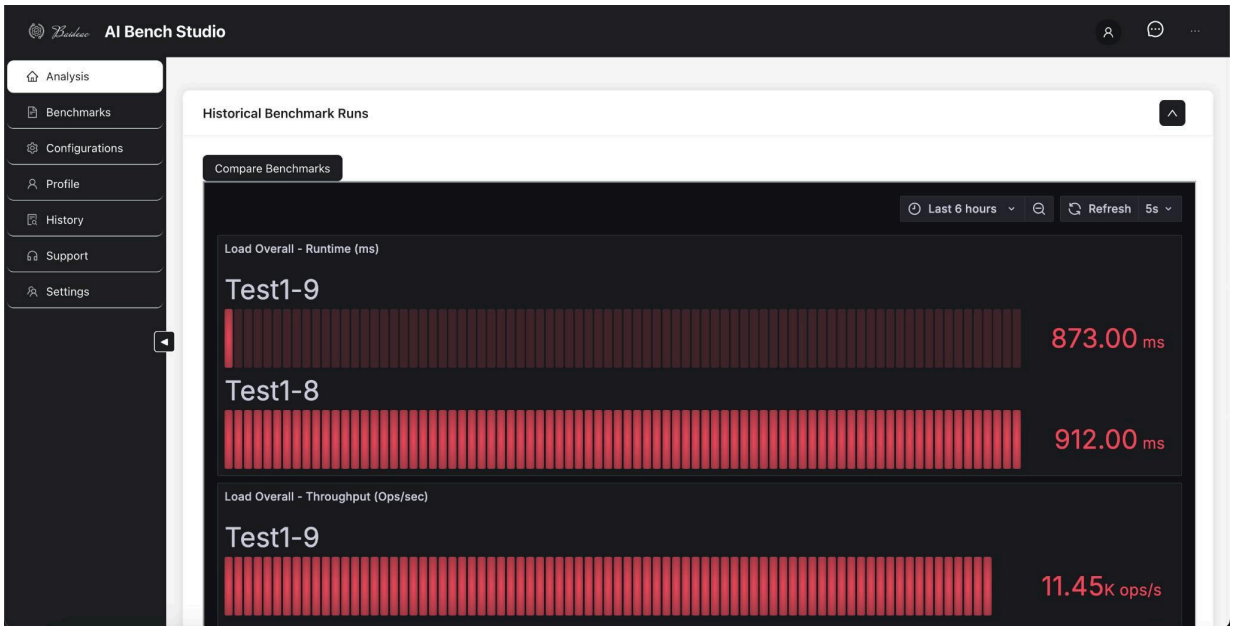
After inputting the parameters, you can view live logs to monitor the progress. Common errors, such as database connection issues, can be resolved by checking these logs for connection issues.

Application Usage Guide

Once both the RBAC and AIBS setups are completed, the application will be ready for use. Below is a Guide to navigate the different features offered in AI Bench Studio.

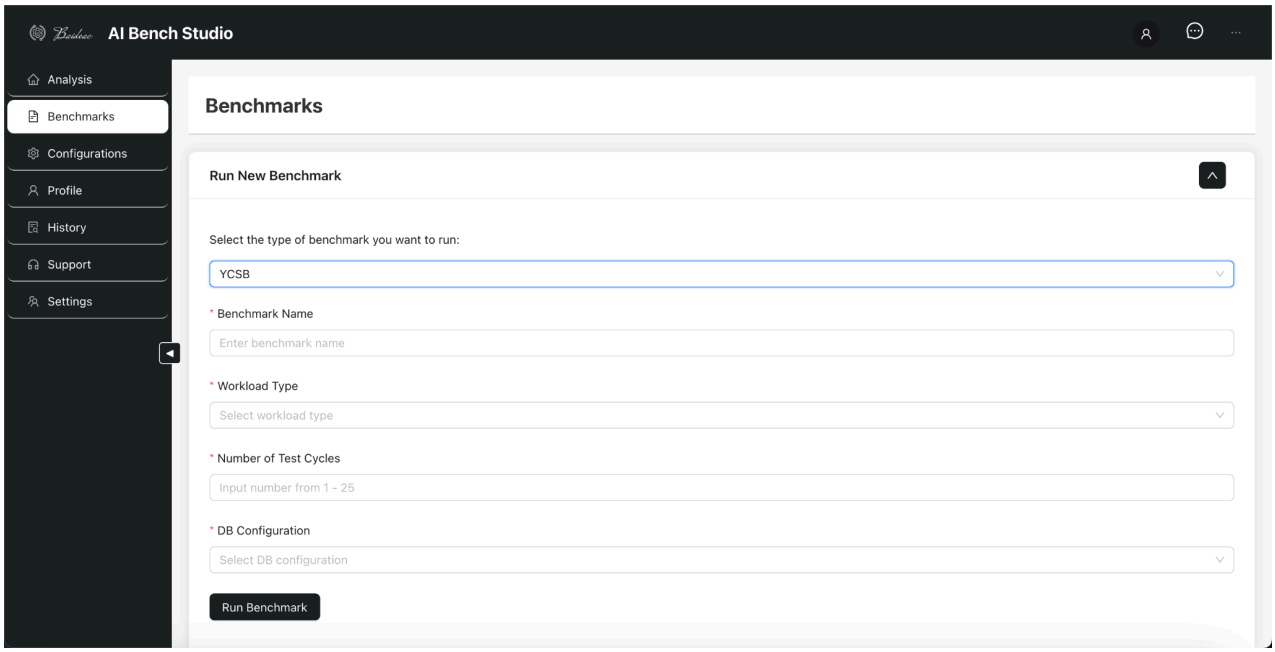
Analysis Page

The analysis page hosts the Grafana Dashboard and allows you to compare benchmark runs. The dashboard becomes available after running at least two benchmarks. After at least 2 benchmarks are complete, you will be able to generate a comparison dashboard to compare across multiple different benchmarks. You can also delete any benchmarks in this page.



Benchmarks Page

The benchmarks page allows you to run AI model and database configuration benchmarks. In this version, YCSB benchmarking is available, and you use predefined/generated data from YCSB or upload custom CSV, S3, or NFS storage data for testing.



Key parameters include:

- Operation Count: Defines how many operations to run.
- Proportions (Read, Update, Insert, Scan): Must total 1.
- DB Configuration: Select the database configuration to use (e.g., Redis).

To run a benchmark, you must have a DB configuration loaded. The screen should direct you there if there are no available DB Configurations. Once the parameters are set, you can start the benchmark and track progress in the live logs.

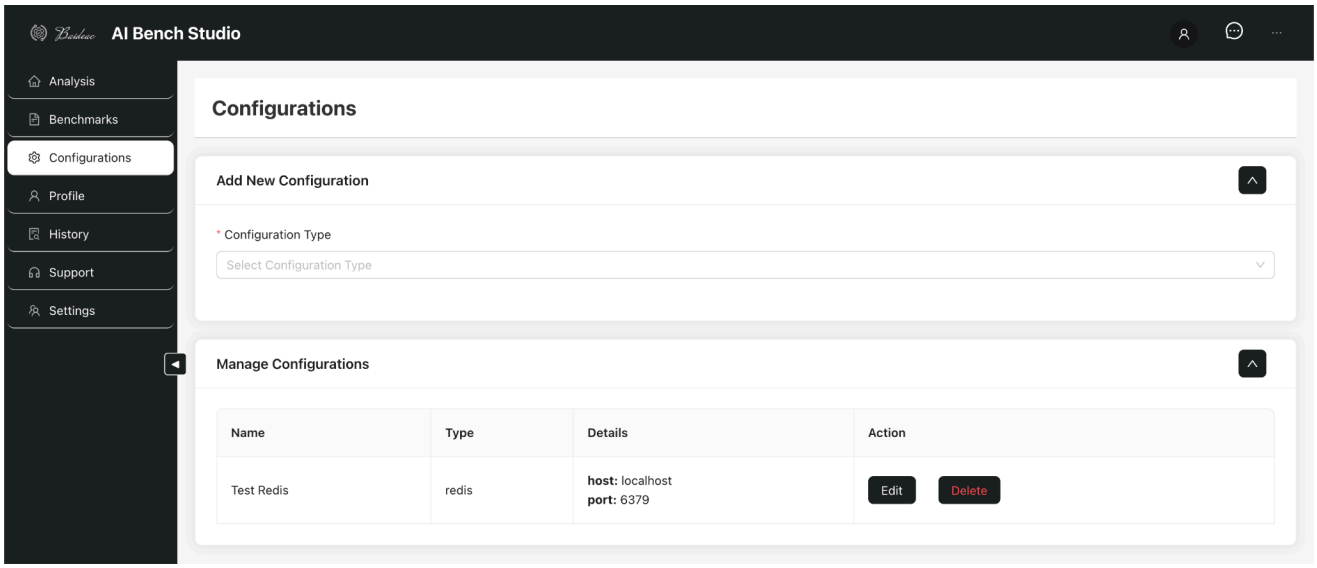
Workload Types:

- ☐ Predefined: This is data generated directly from the YCSB library itself. Using this option, the user can input the number of records to generate and the operation count.
- ☐ Custom
  - ☐ CSV Upload: This method allows for the user to upload a CSV for custom data insertion, up-to 100 MB.
  - ☐ S3: Using AWS S3 or any S3 compatible cloud storage solution, the user can provide the location and the keys to use data directly from a CSV in S3
  - ☐ NFS Storage: Using a NFS storage mount, mounted to the EC2 if the subscribed module, the user can provide the mount path for a CSV located in NFS storage to use for the benchmark.

Configurations

In the configurations page, you can create configurations for an AI model or a database. DB Configurations are used for running benchmarks in this version. You can input the DB type, host, port, and other necessary details. Once at least one configuration is added, you're ready to run benchmarks. Additionally, the AIBS setup out of the box comes with a redis server ready for testing. To use this, just add a configuration with the following:

Hostname: EC2 IP or localhost
Public IP Port: 6379

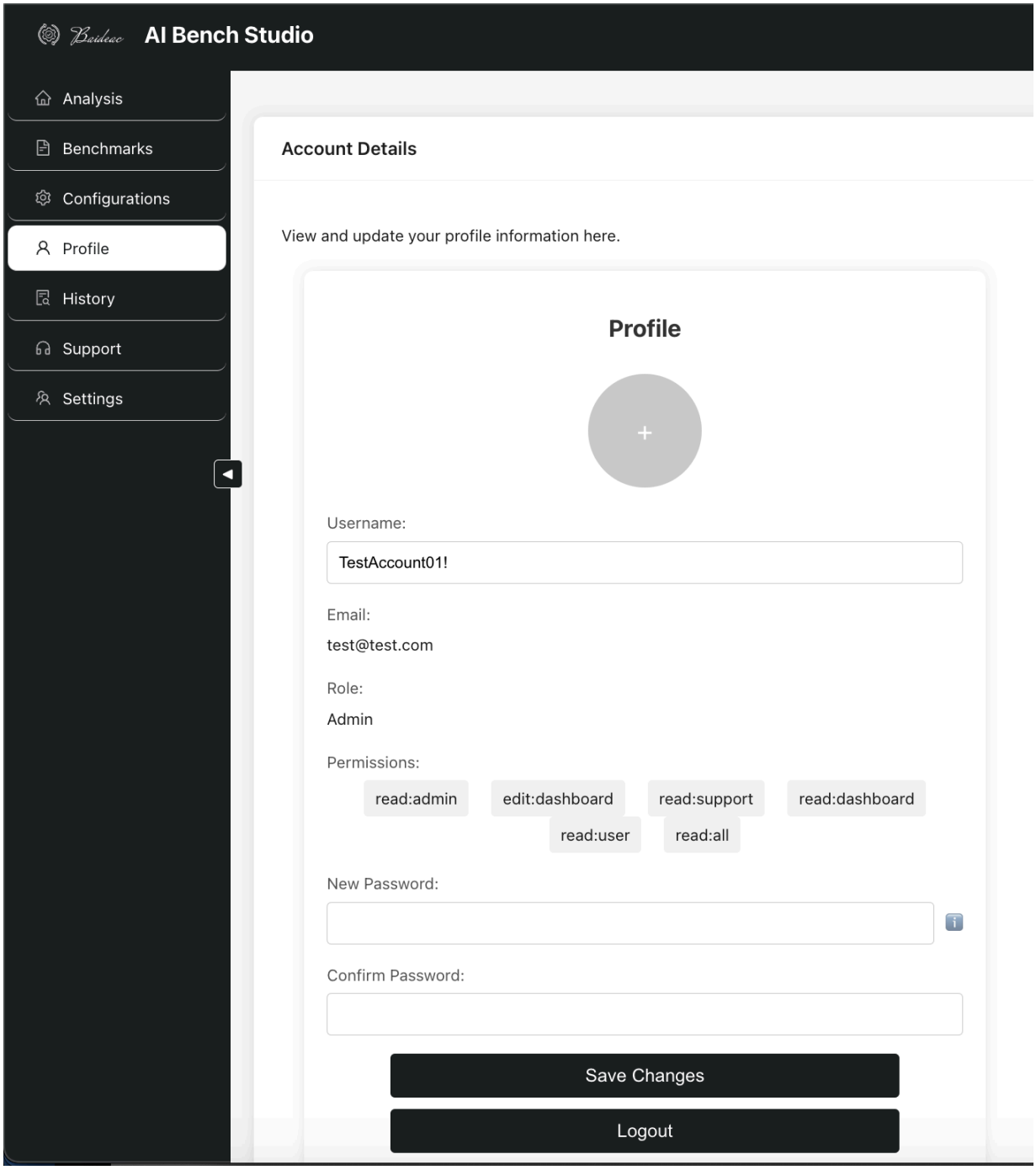


Supported Database/Servers:

- Redis
- Apache Geode

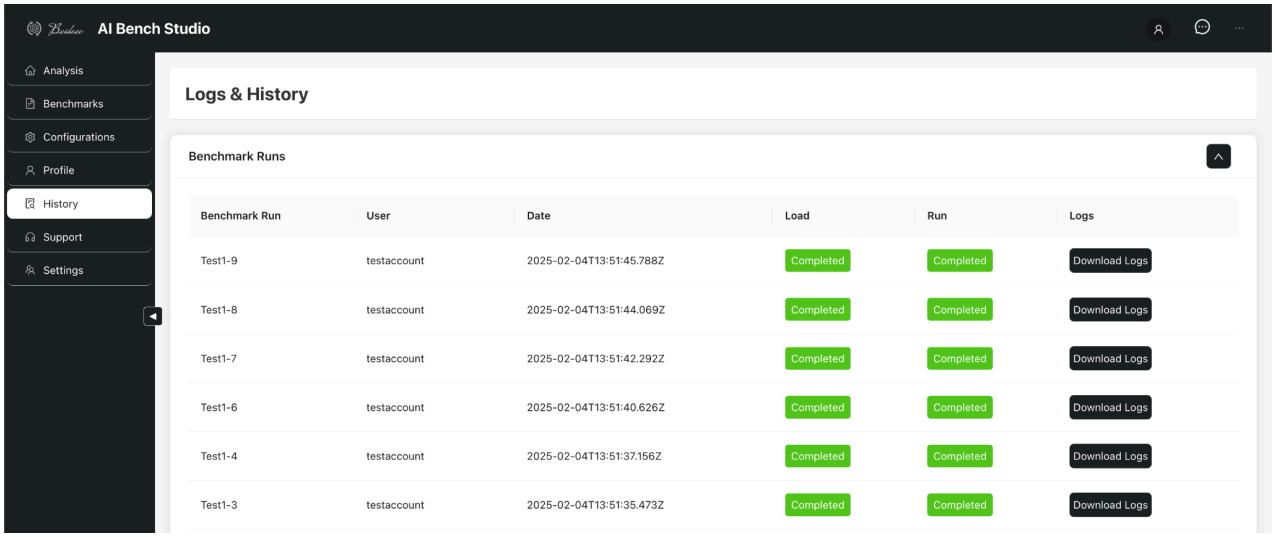
Profile

Powered by RBAC, the profile section allows users to manage profile settings, including changing the profile image, username, and password.



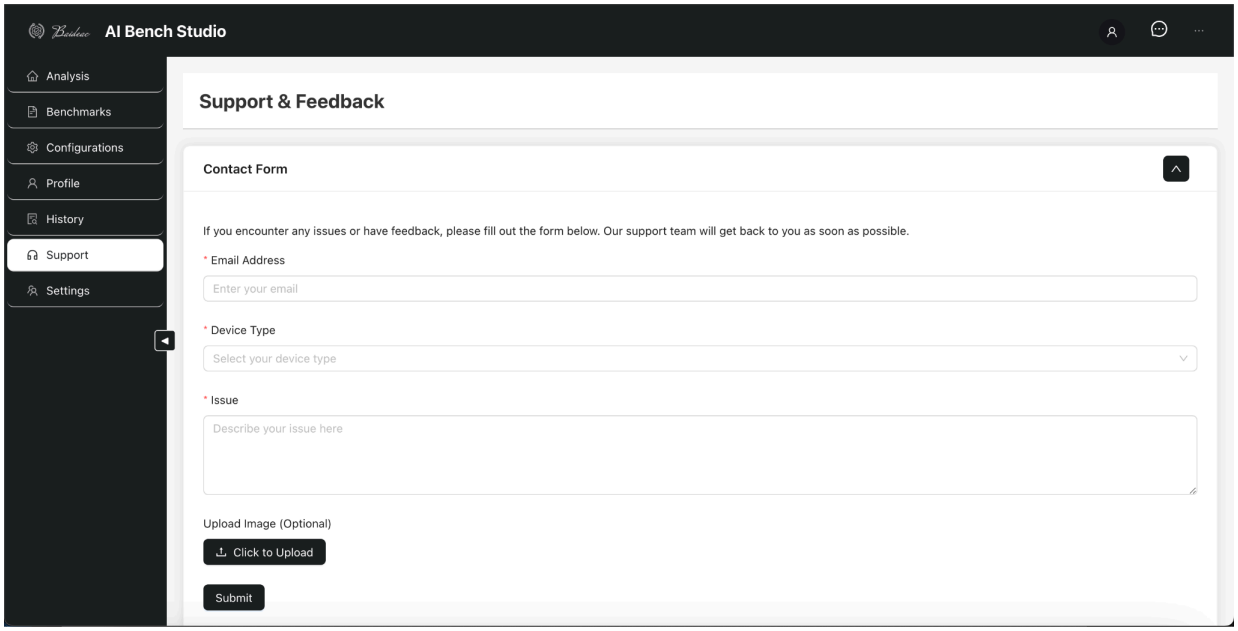
History

In the History page, you can download logs for previously run benchmarks, which include output and error details. This section is useful for debugging.



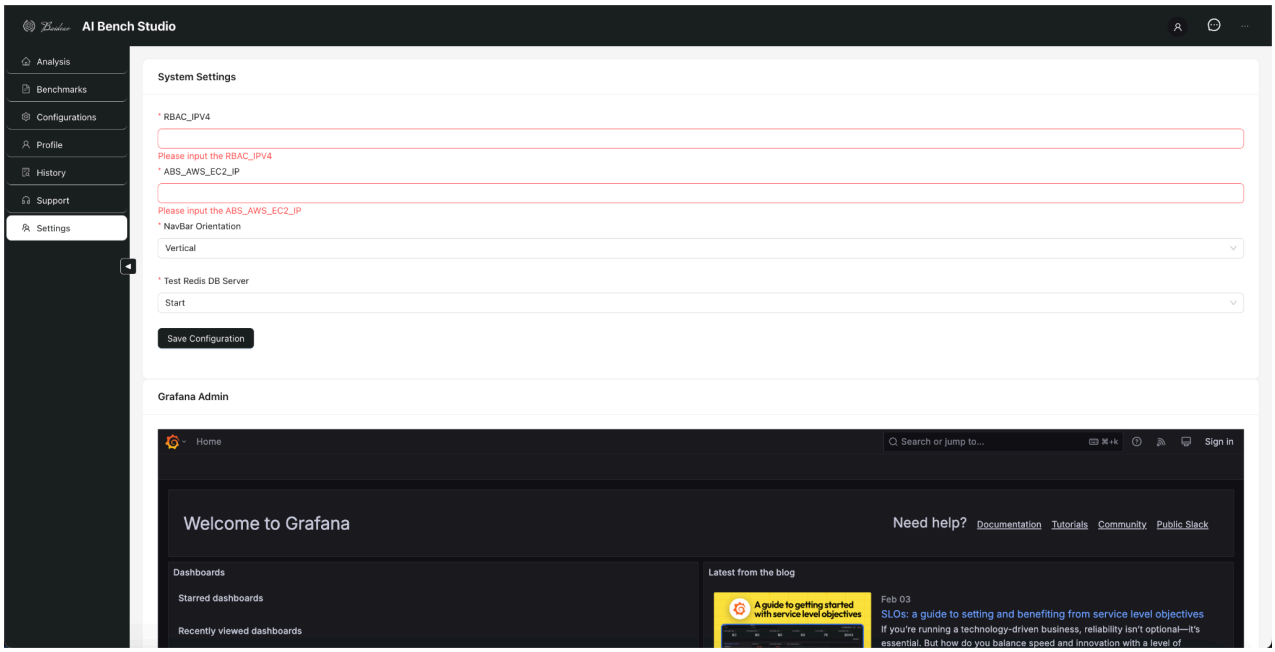
Support & Feedback

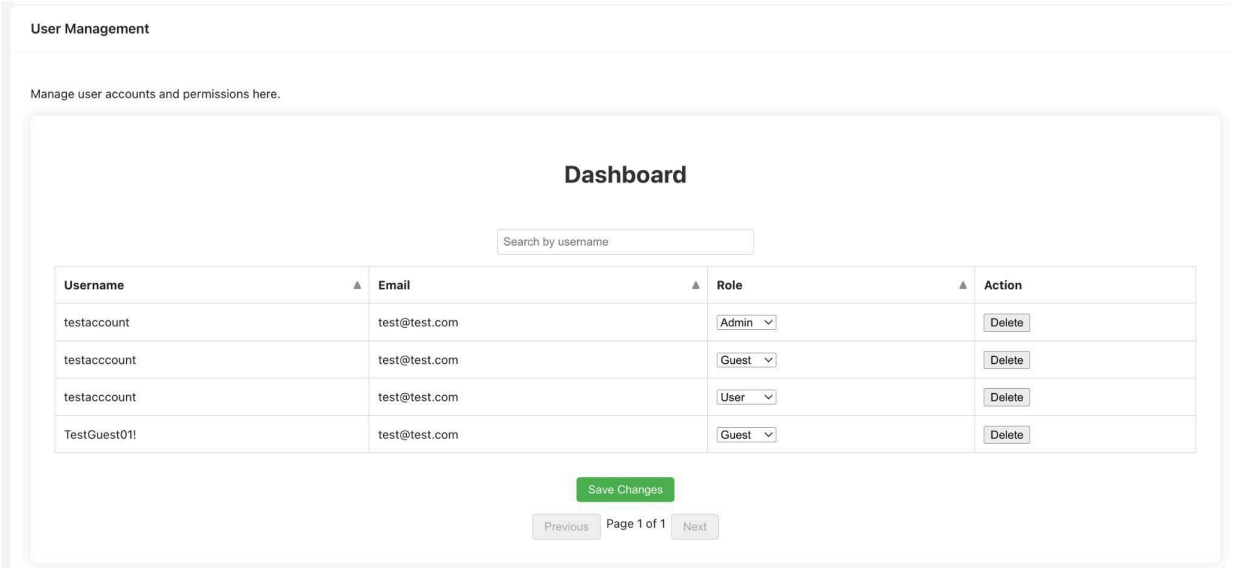
A feedback form is available for users to send issues and feedback to the AI Bench Studio support team.



Settings

Depending on your role (Admin, Viewer, etc.), the Settings page lets you adjust system settings via a form. Admins can also start/stop the local Redis server and customize the NavBar orientation for the application. The Grafana Admin UI is also available here if the admin needs to make any changes from the usage guide to the Grafana settings. The User Dashboard, powered by RBAC, allows Admins to modify, delete, and manage user permissions.





Conclusion

This guide outlines everything you need to set up, configure, and use AI Bench Studio, providing a comprehensive framework for managing AI model benchmarking and database performance analysis using YCSB. By following the steps above, your system will be ready for real-time AI model/DB Configuration testing and database benchmarking, enabling rich insights and performance analysis.If there are any further questions, please contact support.