Setup Guide - HyperAccel LLM Chatbot

Prerequisite

Before launching, ensure you meet the following requirements:

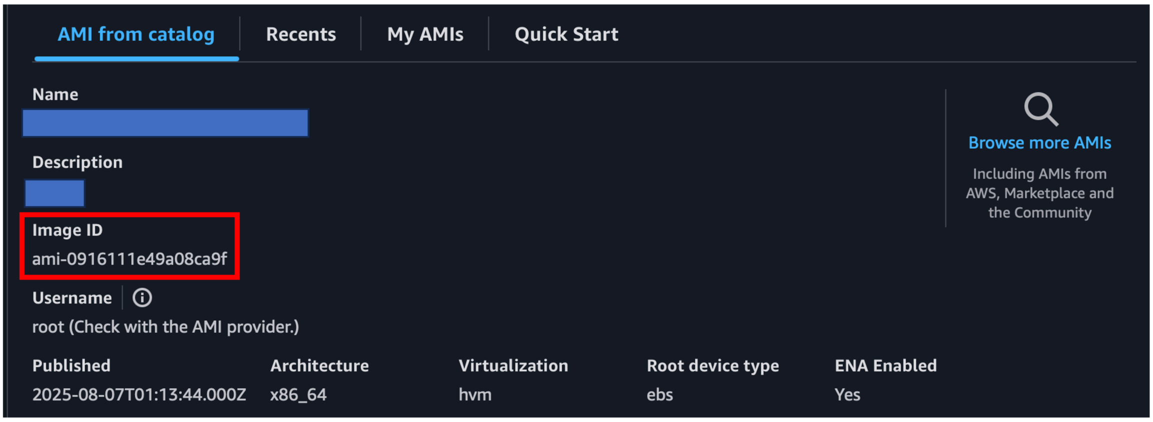
1. **Hugging Face Account**
    - You must have an active Hugging Face account. You can create one at https://huggingface.co/join.
2. **Hugging Face Access Token**
    - Obtain a personal access token from Hugging Face. This token is required to download and use models from the Hugging Face Hub.
    - Generate a token at https://huggingface.co/settings/tokens
3. **Model Access Permissions (for gated models)**
    - If you plan to use gated models such as LLaMA, HyperCLOVA X, ensure that your Hugging Face account has been granted permission to access them.
    - Visit the model's page on Hugging Face and request access if needed.

Quick Start

1. Launch an F2 Instance

Use the following configuration to launch your EC2 instance:

- **AMI ID**: ami-0916111e49a08ca9f



Name
[redacted]

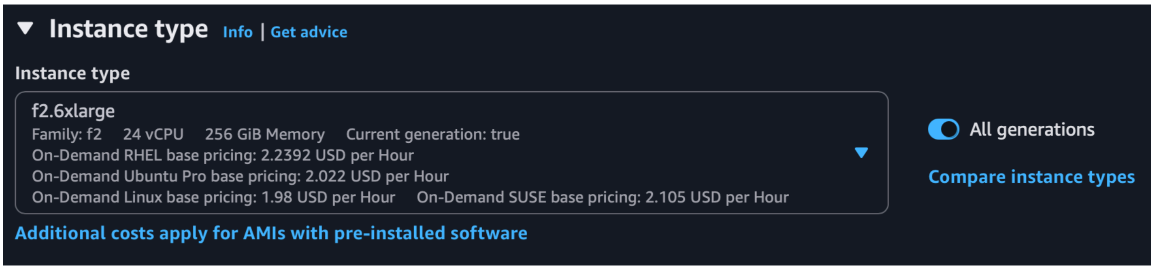
Description
[redacted]

Image ID
ami-0916111e49a08ca9f

Username
root (Check with the AMI provider.)

Published	Architecture	Virtualization	Root device type	ENA Enabled
2025-08-07T01:13:44.000Z	x86_64	hvm	ebs	Yes

- **Instance Type**: f2.6xlarge


Instance type Info | Get advice
All generations
Compare instance types
Additional costs apply for AMIs with pre-installed software

Instance type
f2.6xlarge
Family: f2 24 vCPU 256 GiB Memory Current generation: true
On-Demand RHEL base pricing: 2.2392 USD per Hour
On-Demand Ubuntu Pro base pricing: 2.022 USD per Hour
On-Demand Linux base pricing: 1.98 USD per Hour On-Demand SUSE base pricing: 2.105 USD per Hour

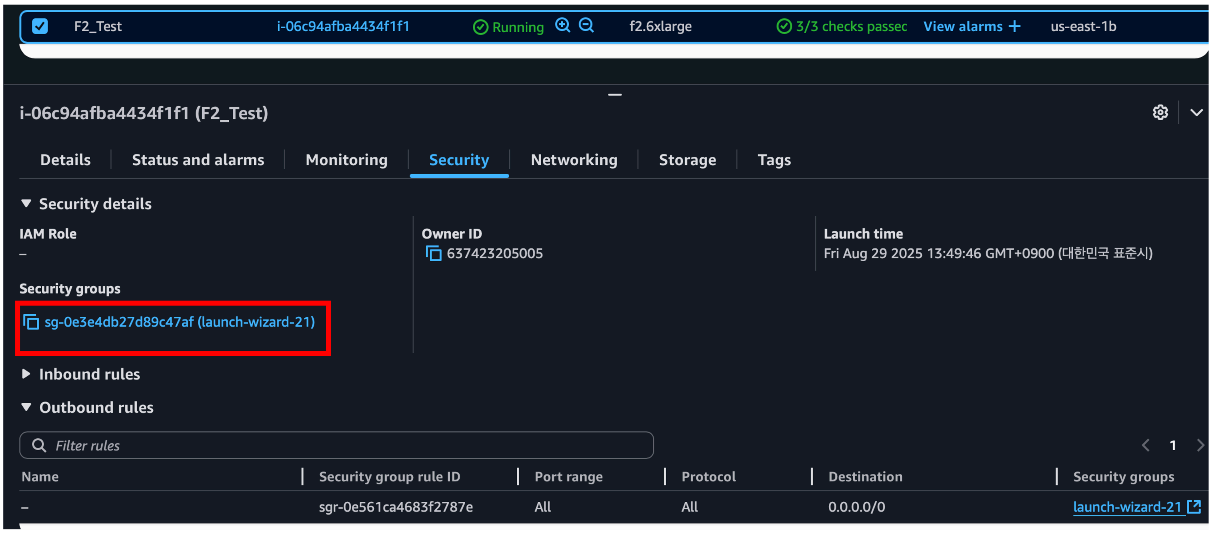
- **Region**: us-east-1[US East (N. Virginia)], us-west-2[US West (Oregon)], eu-west-2[Europe (London)], ap-southeast-2[Asia Pacific (Sydney)]

After launching the instance, add the following inbound rules in the security tab:

- **Port 22** (TCP protocol): SSH access
- *Port 5173 (**TCP protocol): Chat UI frontend server access
- **Source - 0.0.0.0/0** (applies to all ports, allowing access from any IP address)

Port 22 is open by default and we need to open port 5173

Go to instances and select Security



☑	F2_Test	i-06c94afba4434f1f1	Running	f2.6xlarge	3/3 checks passec	View alarms +	us-east-1b

i-06c94afba4434f1f1 (F2_Test)

▼ Security details

IAM Role
–

Owner ID
637423205005

Launch time
Fri Aug 29 2025 13:49:46 GMT+0900 (대한민국 표준시)

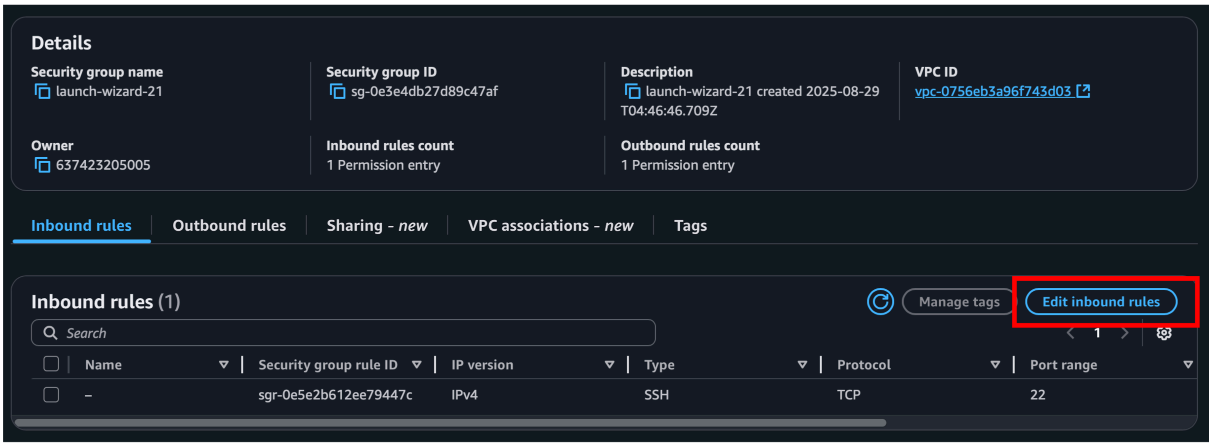
Security groups
sg-0e3e4db27d89c47af (launch-wizard-21)

▶ Inbound rules

▼ Outbound rules

Name	Security group rule ID	Port range	Protocol	Destination	Security groups
–	sgr-0e561ca4683f2787e	All	All	0.0.0.0/0	launch-wizard-21

Select Edit inbound rules



Details

Security group name
launch-wizard-21

Security group ID
sg-0e3e4db27d89c47af

Description
launch-wizard-21 created 2025-08-29 T04:46:46.709Z

VPC ID
vpc-0756eb3a96f743d03

Owner
637423205005

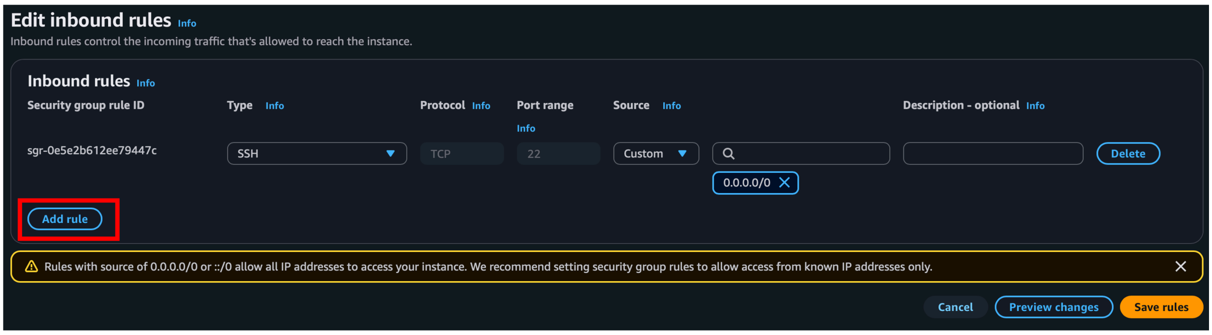
Inbound rules count
1 Permission entry

Outbound rules count
1 Permission entry

Inbound rules (1)

□	Name	Security group rule ID	IP version	Type	Protocol	Port range
□	–	sgr-0e5e2b612ee79447c	IPv4	SSH	TCP	22

Click Add rule



Edit inbound rules Info

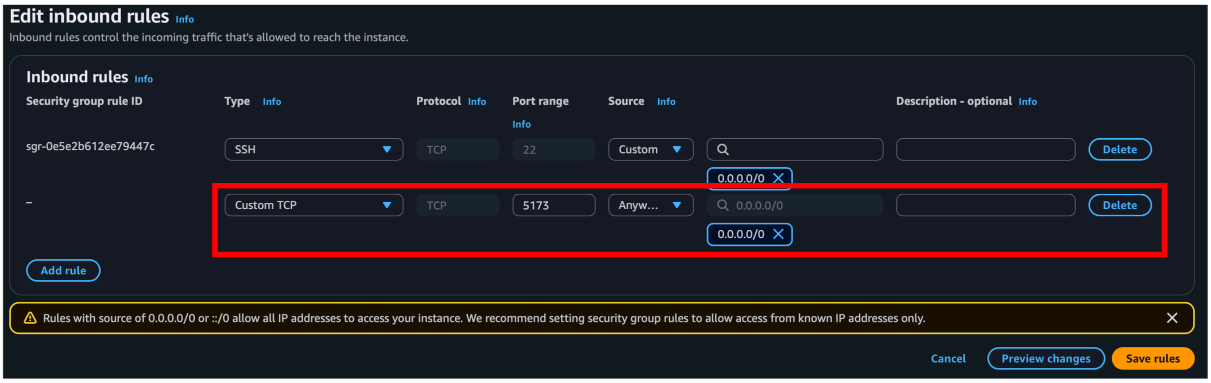
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-0e5e2b612ee79447c	SSH	TCP	22	Custom	0.0.0.0/0 ✕		Delete

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Set Port range as 5173 and source as 0.0.0.0/0



Edit inbound rules Info

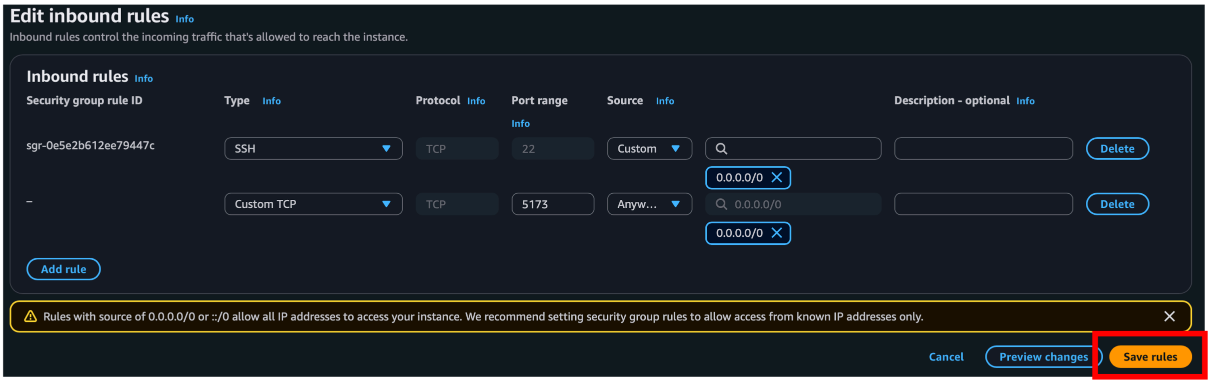
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source		Description - optional
sgr-0e5e2b612ee79447c	SSH	TCP	22	Custom	0.0.0.0/0	
–	Custom TCP	TCP	5173	Anyw...	0.0.0.0/0	

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Save rules to save added rule



Edit inbound rules Info

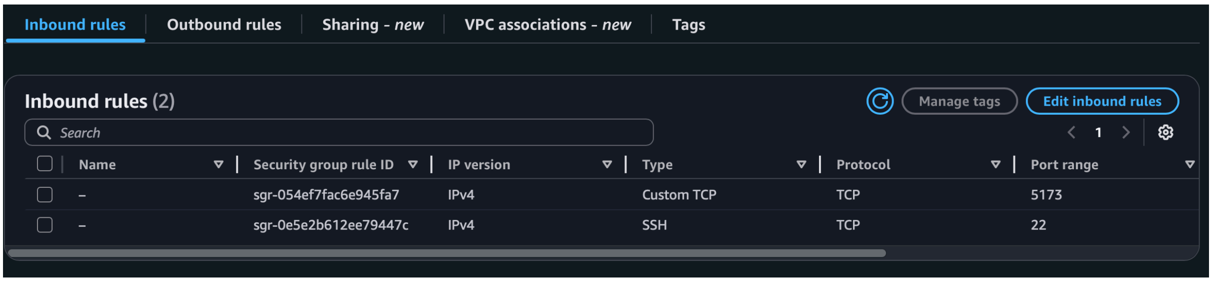
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source		Description - optional
sgr-0e5e2b612ee79447c	SSH	TCP	22	Custom	0.0.0.0/0	
–	Custom TCP	TCP	5173	Anyw...	0.0.0.0/0	

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Now we have opened port 5173



Inbound rules (2)

☐	Name	Security group rule ID	IP version	Type	Protocol	Port range
☐	–	sgr-054ef7fac6e945fa7	IPv4	Custom TCP	TCP	5173
☐	–	sgr-0e5e2b612ee79447c	IPv4	SSH	TCP	22

2. Run Backend Server (vLLM)

Notice : Please proceed following steps in a new terminal

run the following commands step by step:

Launch the Docker container

```
>>> set_container
```

Set Huggingface token

```
>>> export HF_TOKEN="<huggingface_token>"
```

Activate the virtual environment

```
>>> cd /workspace/dev/Projects/vllm-orion
>>> source .venv/bin/activate
```

Set environment variables for F2

```
>>> set-f2-env
```

Load the FPGA bitstream

```
>>> load-f2-bitstream
```

Start vllm server

(For example, use: vllm serve LGAI-EXAONE/EXAONE-3.5-2.4B-Instruct --config config/config.yaml)

Notice : model setup takes a few minutes

```
>>> vllm serve <huggingface_model_name> --config config/config.yaml
```

3. Run Frontend Server (Chat UI)

Notice : You need to proceed this step in a new terminal

```
# This step doesn't need to use docker container.
```

```
>>> cd ~/Projects/HyperDex-Container/hyperaccel-chat-ui
>>> npm run dev -- --host                  # Start the frontend server
```

Functional Test

1. Backend Server Only

Notice : You need to proceed this step in a new terminal

To verify the backend is working properly, run the following test command in your terminal:

- Test command:

```
>>> curl <http://localhost:8000/v1/completions> \\
  -H "Content-Type: application/json" \\
  -d '{"model": "LGAI-EXAONE/EXAONE-3.5-2.4B-Instruct", "prompt": "Who are you?", "max_tokens": 30, "temperature": 1  }'
```

- Expected output:

```
{"id":"cmpl-<id>","object":"text_completion","created":<num>,"model":"LGAI-EXAONE/EXAONE-3.5-2.4B-Instruct","choices":[{"index":0,"text":"\\n\\nOptions:\\n-  a scientist\\n-  a farmer\\n-  you\\n-  athlete\\n-  a schoolboy/","logprobs":null,"finish_reason":"length","stop_reason":null,"prompt_logprobs":null}],"usage":{"prompt_tokens":4,"total_tokens":34,"completion_tokens":30,"prompt_tokens_details":null},"kv_transfer_params":null}
```

2. Frontend Server Connection Test

To verify frontend webpage is connected to server, check if we can reach to port 5173

- Test Command

  ```
  nz -zv <F2_instance_public_IPv4_address> 5173
  ```

- Expected output

```
Connection to <F2_instance_public_IPv4_address> 5173 port [tcp/*] succeeded!
```

3. Backend + Frontend Server Test

To test both backend and frontend servers together:

1. Open the Chat UI in a web browser:

```
http://<F2_instance_public_IPv4_address>:5173/
```

2. In the Chat UI, enter a prompt into the “Ask anything” input box at the bottom of the screen, and press Enter.
   - Example Input:

```
Who are you?
```

   - Expected Output:

```
I am an artificial intelligence designed to assist and communicate with users like yourself. My goal is to provide helpful information, engage in conversations, and support tasks across a wide range of topics. How may I assist you today? If you have any specific questions or need information on particular subjects, feel free to ask!
```

How to Change Large Language Models?

To switch to a different Hugging Face model, follow these steps:

1. When starting the backend server, replace <huggingface_model_name> with the new model you want to use.

```
>>> vllm serve <huggingface_model_name> --config config/config.yaml
```

2. Edit the .env.local file in the Chat UI project directory and change the "name" field to match the new model.

```
# File: ~/Projects/HyperDex-Container/hyperaccel-chat-ui/.env.local

MODELS=[
  {
    "name": "<huggingface_model_name>",
    "parameters": {
      "temperature": 1.0,
      "truncate_prompt_tokens": 256
    },
    "endpoints": [
      {
        "baseURL": "<http://127.0.0.1:8000/v1>",
        "type": "openai"
      }
    ]
  }
]
```

Replace <huggingface_model_name> with the name of your new model.

3. After updating the configuration, restart the Chat UI server:

```
>>> cd ~/Projects/HyperDex-Container/hyperaccel-chat-ui
>>> npm run dev -- --host
```

Now your Chat UI is connected to the new model.

Supported Large Language Models

(Replace <huggingface_model_name> with one of the following models)

You can use any of the models listed below by replacing <huggingface_model_name> in the command or configuration files:

- meta-llama/Llama-3.1-8B-Instruct
- meta-llama/Llama-3.2-1B-Instruct
- meta-llama/Llama-3.2-3B-Instruct
- naver-hyperclovax/HyperCLOVAX-SEED-Text-Instruct-0.5B

- naver-hyperclovax/HyperCLOVAX-SEED-Text-Instruct-1.5B
- LGAI-EXAONE/EXAONE-3.5-2.4B-Instruct
- LGAI-EXAONE/EXAONE-3.5-7.8B-Instruct

Be sure to use the full model name exactly as shown above when configuring your backend and frontend settings.

TroubleShooting

- If the vLLM server fails to initialize, please re-upload the F2 bitstream and restart the vLLM server.

```
>>> load-f2-bitstream             # Load the FPGA bitstream
>>> vllm serve <huggingface_model_name> --config config/config.yaml
```

- If you don’t have access to the model on Hugging Face, please check whether your account (with the issued token) has been granted permission to access the model. Especially, LLaMA models and HyperCLOVA X models are gated models and require access permissions.