Usage instructions:

1. Launch the product via 1-click from AWS Marketplace. **Wait** until the instance status changes to 'Running' and passes all health checks. Then, connect to your instance using your Amazon private key and the '**ubuntu**' user."

   To update software, use: **sudo apt update && sudo apt upgrade -y**

1. What you get: This instance comes preinstalled with:

- **Dify** – a visual, low-code AI app / workflow / agent builder
- **Langfuse** – an observability and analytics dashboard for LLM apps

Both run as Docker services on **Ubuntu 24.04** and start automatically when the server boots.

You connect to everything **from your browser**. SSH is only needed for advanced administration.

---

2. Prerequisites

To build and run AI apps with this studio, you will need one or more **model API keys**, for example:

- OpenAI (ChatGPT / GPT-4 / GPT-4.1, etc.)
- Anthropic (Claude)
- AWS Bedrock
- Azure OpenAI
- Other LLM / embedding providers you prefer

⚠ **Important:** *This AMI does **not** include any LLM API keys. You bring your own keys and configure them inside Dify.*

---

3. Launching the instance

1. **Choose instance type**
   - ○ Recommended minimum: **4 vCPU, 16 GB RAM** (e.g., t3.xlarge or better)
   - ○ Root volume: **100 GB gp3** (or more if you expect large workloads)
   - ○
2. Launch the instance as normal from the AWS console. When the instance starts, it will **automatically:**

- Start Docker
- Start Dify
- Start Langfuse

This can take **1–3 minutes** on first boot while containers come up.

---

4. First login – Dify (AI App Studio)

1. Find the **public IP** of the instance in the EC2 console.
2. In your browser, go to:
3. http://YOUR_PUBLIC_IP/install
4. You will see Dify's **initial setup page**:
   - Enter an **admin email**
   - Set an **admin password**
   - Click **Create** / **Continue**
5. After completing setup, open:
6. http://YOUR_PUBLIC_IP/
   - and log in with the admin account you just created.

---

5. Configure your AI providers (API Keys)

Inside Dify:

1. Log into the Dify UI at http://YOUR_PUBLIC_IP/
2. Go to **Settings** → **Model Providers** / **LLM Providers** (naming may differ slightly):
3. Add your keys for the providers you use, for example:
   - **OpenAI** – paste your OpenAI API key
   - **Anthropic** – paste your Claude key
   - **AWS Bedrock** – configure region & credentials
   - Any other supported provider you prefer
4. Save the settings.

Now you can:

- Create **chatbots**
- Build **agents and workflows**
- Add **RAG / knowledge base** apps
- Test everything directly in the Dify UI.

---

5. **First login – Langfuse (Observability)**

Note: Be patient: This can take 1–3 minutes on first boot while containers come up.

1. In your browser, open:
2. http://YOUR_PUBLIC_IP:3000
3. You'll see the **Langfuse** sign-up / login screen.
4. Create your first user account (email + password)
5. Go back to "Sign in" with your new account credentials

Langfuse will show:

- Traces of LLM calls
- Metrics and dashboards
- Cost and latency insights (once you integrate it with your apps)

---

7. Connecting Dify apps to Langfuse (optional but recommended)

This AMI ships Dify and Langfuse on the **same server**, so you can easily wire them together.

General idea (high-level):

1. In Langfuse:
   - Go to **Project settings / API keys** and create a **public** and **secret** key.
2. In your AI apps (or in intermediate services that call the models from Dify), use the **Langfuse SDK** or HTTP API and point them to:
3. http://YOUR_PUBLIC_IP:3000
4. Include the Langfuse keys in those calls so traces are recorded.

Note: Exact integration steps depend on your language / framework (Python, Node.js, etc.).
Refer to Langfuse's docs for the SDK you're using.

---

8. Optional: SSH access for administrators

SSH is **not required** for using Dify or Langfuse, but you can use it for admin tasks.

1. Connect via SSH:
2. ssh -i /path/to/your-key.pem ubuntu@YOUR_PUBLIC_IP

Useful commands:

Check running containers:

```
docker ps
```

Restart Dify+Langfuse:

```
cd ~/dify/docker
docker compose restart

cd ~/langfuse
docker compose restart
```

Stop stacks (for maintenance):

```
cd ~/dify/docker
docker compose down

cd ~/langfuse
 docker compose down
```

AWS Data

- Data Encryption Configuration: This solution does not encrypt data within the running instance.
- User Credentials are stored: /root/.ssh/authorized_keys & /home/ubuntu/.ssh/authorized_keys
- Monitor the health:
  - Navigate to your Amazon EC2 console and verify that you're in the correct region.
  - Choose Instance and select your launched instance.
  - Select the server to display your metadata page and choose the Status checks tab at the bottom of the page to review if your status checks passed or failed.

Extra Information: (Optional)

Allocate Elastic IP

To ensure that your instance **keeps its IP during restarts** that might happen, configure an Elastic IP. From the EC2 console:

1. Select ELASTIC IPs.
2. Click on the ALLOCATE ELASTIC IP ADDRESS.
3. Select the default (Amazon pool of IPv4 addresses) and click on ALLOCATE.
4. From the ACTIONS pull down, select ASSOCIATE ELASTIC IP ADDRESS.
5. In the box that comes up, note down the Elastic IP Address, which will be needed when you configure your DNS.
6. In the search box under INSTANCE, click and find your INSTANCE ID and then click ASSOCIATE.
7. Your instance now has an elastic IP associated with it.
8. For additional help: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html