AWS Marketplace Offering (HAI Build V2) - User Guide

Overview of HAI Build

HAI Build is an innovative platform transforming the software development lifecycle (SDLC) by seamlessly integrating AI-driven solutions into every stage. Designed to cater to both new and ongoing projects, HAI Build accelerates development while maintaining precision and adaptability through two key components:

1. Specifai

An intelligent application that redefines how teams gather, refine, and manage requirements. Powered by AI, it organises inputs into detailed Business, Product, Non-Functional, and User Interface specifications, ensuring clarity and actionable outputs. This smart solution bridges the gap between stakeholders and development teams with context-aware suggestions that enhance collaboration and decision-making.

2. Code Generator

A generative AI-powered tool that empowers developers to translate detailed tasks into deployable code. This plugin supports a wide range of programming languages and frameworks, ensuring adherence to industry standards and best practices. Leveraging a continuous feedback loop fosters collaboration between developers and AI, enabling the creation of adaptable, high-quality code.

3. Factifai

An innovative application revolutionizes test automation by directly controlling your computer through AI. Using built-in vision capabilities of Claude, OpenAI, and Gemini along with computer use, it can navigate any application naturally - clicking, typing, and verifying results just like a human would.

Introduction

This guide provides a detailed walkthrough for deploying HAI Build ecosystem including the Langfuse application server, using an AWS CloudFormation Stack.

Download and Setup

Before proceeding with deployment, ensure to have downloaded and prepared the client applications:

- **Specifai (Frontend Electron App):** The Desktop Electron App for intuitive requirement management
  Github / Download (Latest Release)
- **Code Generator (VS Code Plugin):** For AI-assisted code generation
  Github / VSCode Marketplace
- **Factifai Browser:** Robust Explore Mode for mapping web app structures and make humal like interactions.
  https://github.com/presidio-oss/factif-ai
- **Factifai Agent:** Transforms Natural Language to Test Cases. https://factifai.io/getting-started/installation.html

Deployment Process

- Automates resource provisioning for the Langfuse App Instance using a pre-configured CloudFormation template

By following this guide, users can seamlessly integrate HAI Build with Langfuse into their workflow, enabling precision-driven requirement management and adaptive, high-quality code generation.

Prerequisites

Before initiating the HAI Build deployment process, ensure the following prerequisites are met:

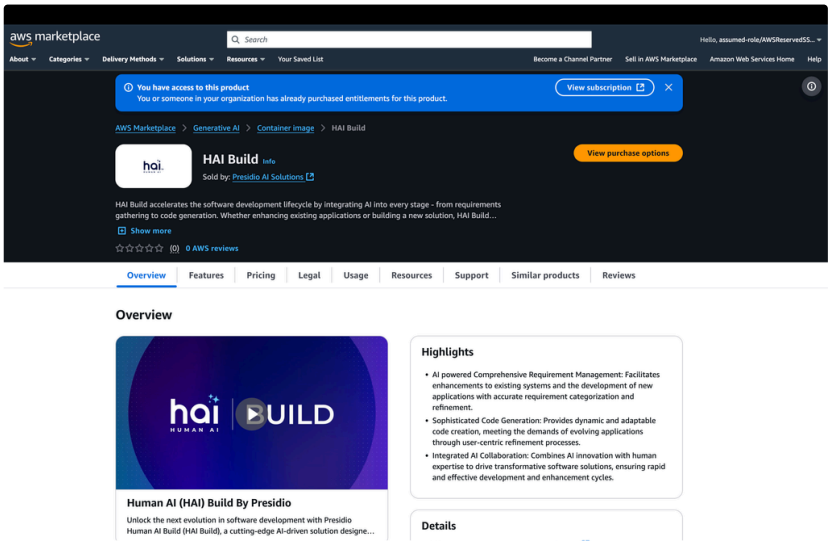
- **LLM Keys**:
  - Essential for Specifai, Code Generator and Factifai applications. All support multiple models, including AWS Bedrock (Documented Below), Anthropic, Azure OpenAI, OpenAI Native.
  - **Requirement:** You must obtain an LLM key from at least one LLM provider corresponding to your chosen model to enable the applications to function.
- **AWS Account**:
  - Active AWS account with sufficient permissions to deploy CloudFormation stacks and provision EC2, ECS, and related AWS services.
- **Langfuse Hosting Domain** (optional):
  - A custom domain for hosting the Langfuse Application for Observability (e.g., `langfuse.yourcompany.com`).
  - If using a custom domain, register it using Amazon Route 53 or your preferred domain registrar.
- **SSL Certificate** (optional):
  - An SSL/TLS certificate for your custom domain, managed within AWS Certificate Manager (ACM).
  - Refer to the ACM Documentation for detailed setup instructions. This is required if using a custom domain for Langfuse.

Steps to Deploy HAI Build

The following steps guide you through deploying the HAI Build Langfuse application server, using an AWS CloudFormation template.

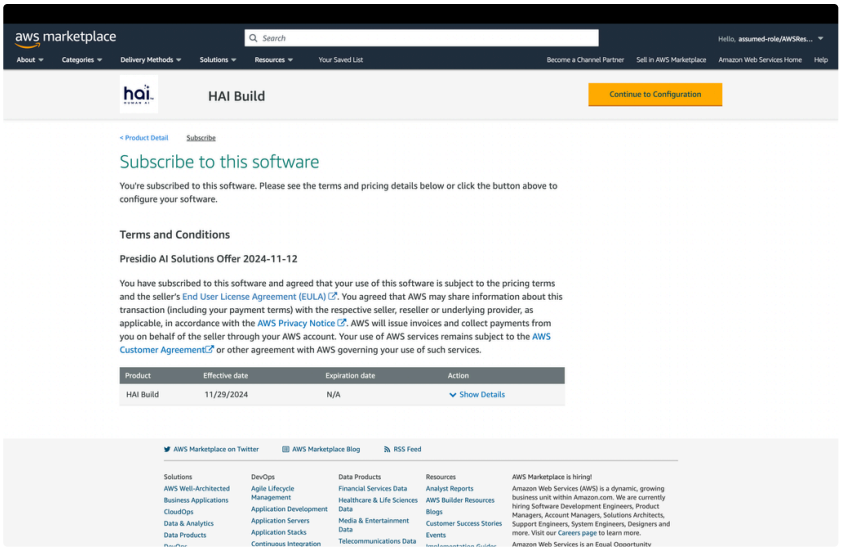
Step 1: Access AWS Marketplace

- Navigate to the HAI Build listing in the AWS Marketplace.
- Click "View Purchase Options" to review subscription options and pricing details.

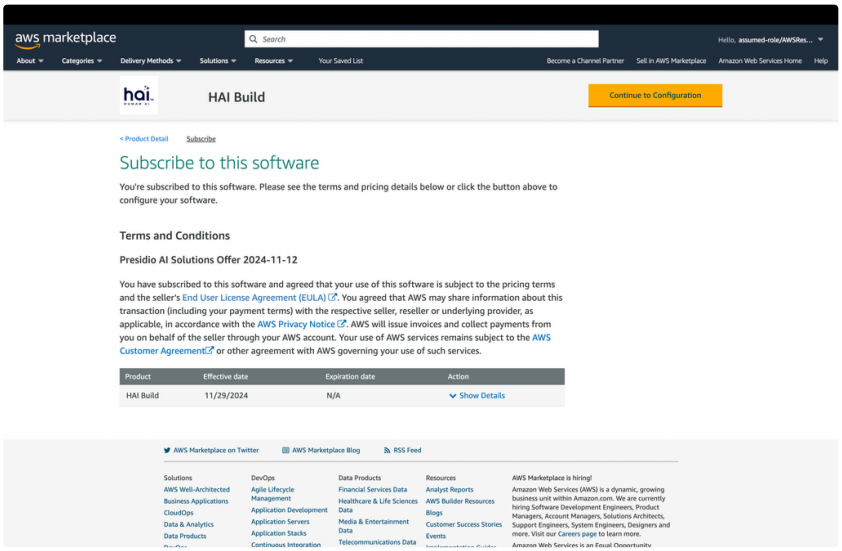


Step 2: Review Terms and Conditions

- Carefully review and accept the Terms and Conditions for the offering.

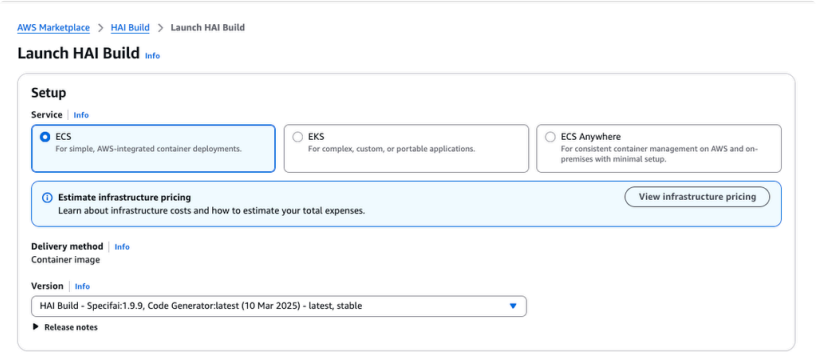


- Once accepted, click "Continue to Configure" to proceed to the version selection page.



Step 3: Launch CloudFormation Stack

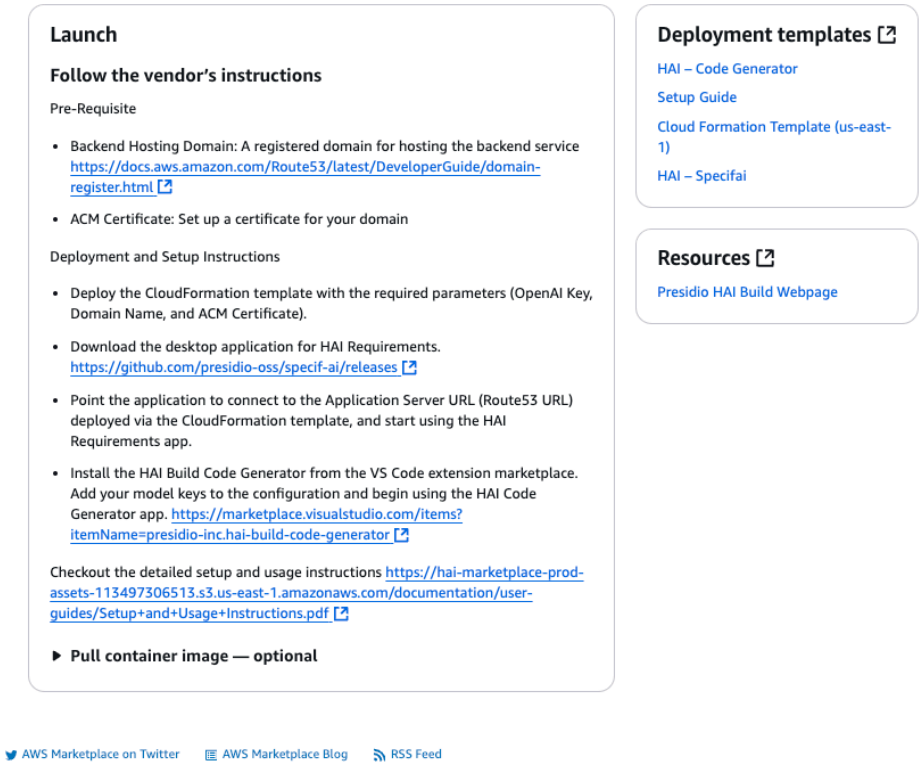
- Select **ECS Deployment Options**.



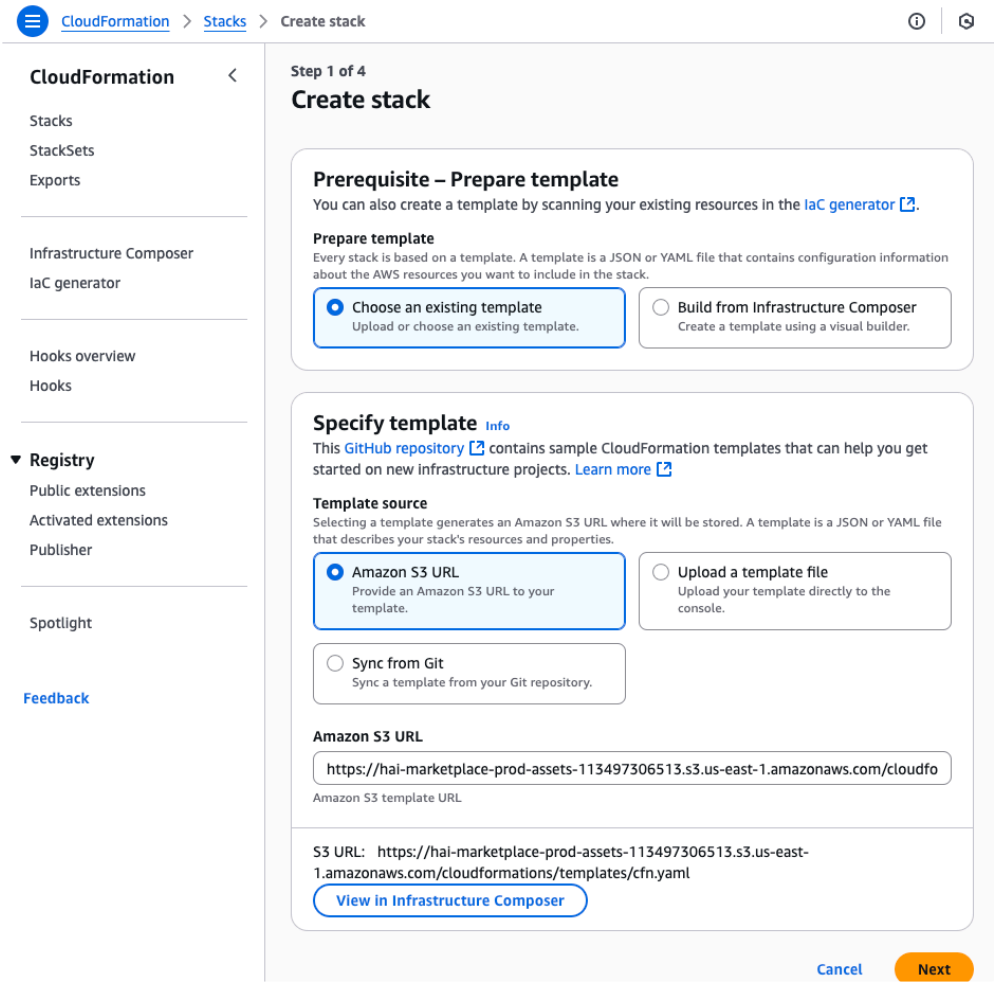
- **Review the configuration details:**
  - Review the configuration details presented on the page. The "Deployment template" section provides crucial links:
    - **Setup Guide:** Download this detailed guide for step-by-step instructions (referring to this document).

- **CloudFormation Template:** This link will launch the CloudFormation console with the Presidio HAI Build CFT template pre-filled. This template configures the Langfuse Server, which integrates with both Specifai and the HAI Code Generator for end-to-end traceability.
- **Specifai:** Link to the Releases page to download the latest version of the Electron Desktop app for macOS (Intel, ARM) or Windows.
- **Code Generator:** Link to the VS Code Plugin Marketplace compatible with macOS (Intel, ARM) or Windows.
- **Factifai Browser:** Link to Factifai Web Application.
- **Factifai Agent:** Link to install and setup Factifai Agent along with playwrite tool.

- Click on the **"Cloud Formation Template"** link provided in the AWS Marketplace.



2. This will navigate to the AWS CloudFormation page with the **Presidio HAI Build CFT template** pre-filled.



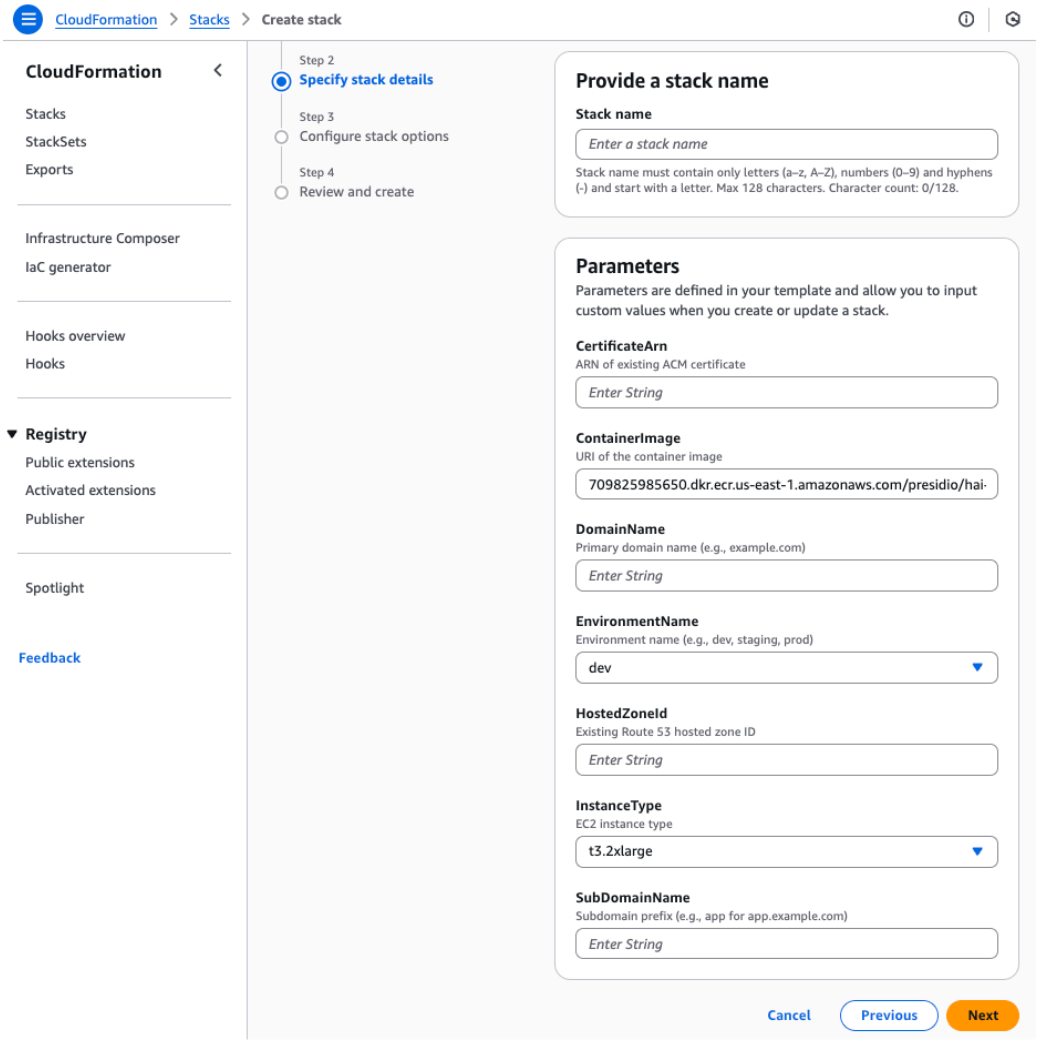
3. Click "**Next**" to proceed to the parameter configuration page.

Step 4: Provide Stack and Parameter Details

- **Enter Stack Name and Parameters**: In Step 2, fill out the required fields as outlined below:
  - **Stack name:** Enter a unique name for your CloudFormation stack.
    - *Constraint:* Must contain only letters (a-z, A-Z), numbers (0-9), and hyphens (-), and must start with a letter. Maximum 128 characters.
  - **Parameters:** Configure the custom values for your stack.

Parameter Name	Description	Notes
CertificateArn (optional)	ARN of an existing ACM certificate used for Langfuse hosting.	Required if using a custom domain.
ContainerImage	URI of the container image	709825985650.dkr.ecr.us-east-1.amazonaws.com/presidio/hai-build:3.63.1 (Pre-filled with the latest HAI Build image.)

DomainName (optional)	Primary domain name (e.g., http://example.com) (optional)	Used for Langfuse Hosting. So, it is required if using a custom domain for Langfuse. Example: https://example.com
EnvironmentName	Environment name (dev, staging, prod)	Default: dev
HostedZoneId (optional)	Existing Route 53 hosted zone ID (optional)	Used for Langfuse Hosting. So, it is required if using a custom domain for Langfuse.
InstanceType	EC2 instance type	Recommended: t3.2xlarge
SubDomainName (optional)	Subdomain prefix (e.g., app for app.example.com)	Example: app



Click "**Next**" to proceed.

Step 5: Configure Stack Options

1. Add tags for resource management (optional and best practices to track the resources).

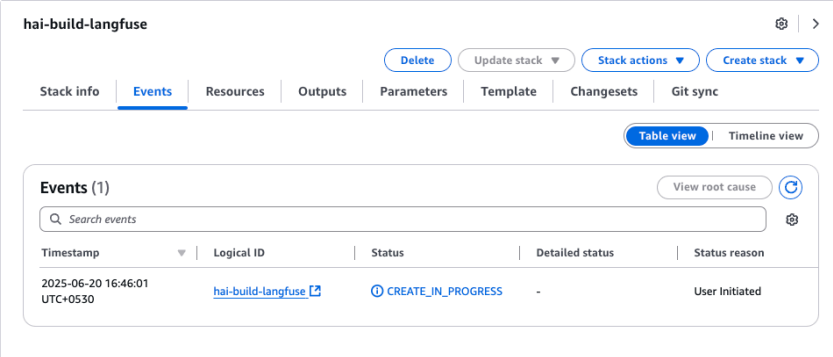
2. Configure stack options and advanced settings as needed.
3. Acknowledge and click "Next" to proceed to the review page.

Step 6: Review and Create Stack

1. Review all the configuration settings
2. Click "Submit" to deploy the infrastructure

Step 7: Monitor Stack Creation

1. Wait for the CloudFormation stack creation to complete (approximately 10-15 minutes.



2. Once complete, go to the "**Outputs**" tab of the CloudFormation stack

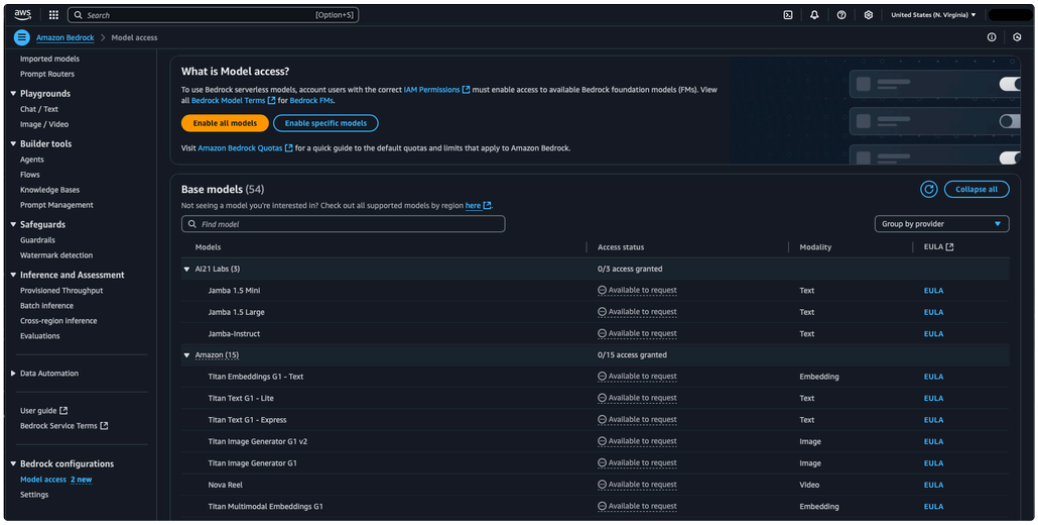
Step 9: Retrieve Output Values

From the "**Outputs**" tab, note the following values:

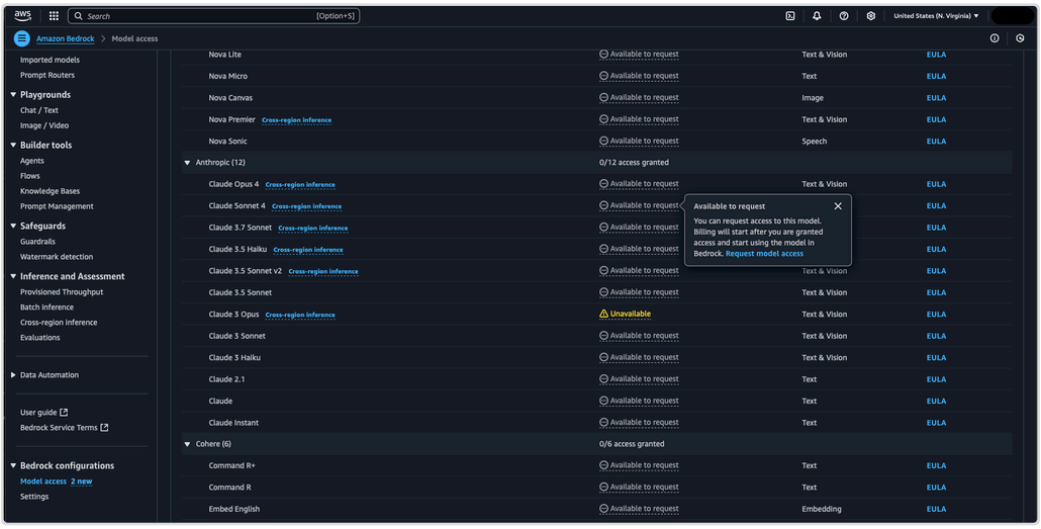
1. **ApplicationURL**: The Langfuse Application URL, which has to be integrated with Specifai and HAI Code Generator for traceability.
2. **BedrockUserAccessKeyId**: Access Key ID for Bedrock API access.
3. **BedrockUserSecretAccessKey**: Secret Access Key for Bedrock API access.

Request Foundation Model Access using AWS Bedrock

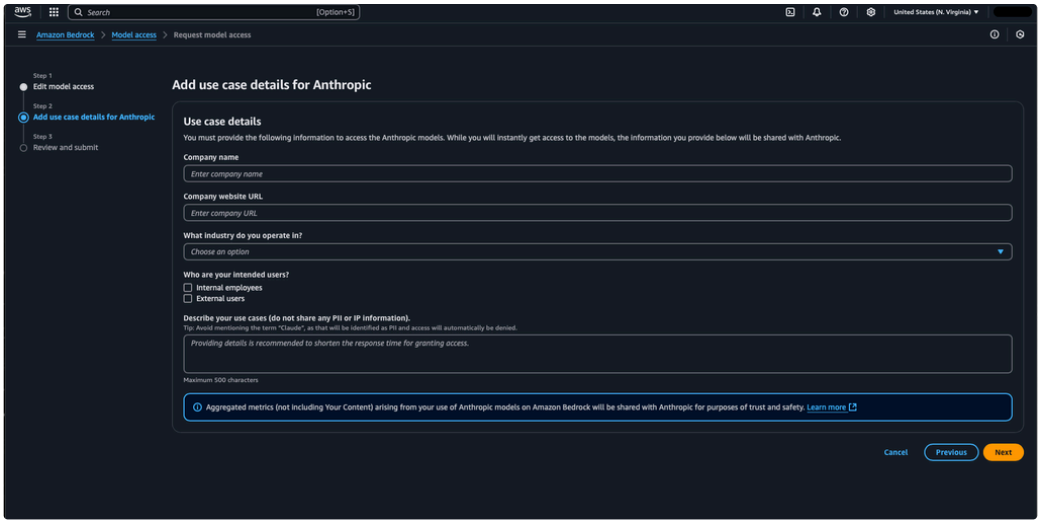
1. **Access the Model Access Page**:
   - In the left navigation pane, select **Model access**.
   - You'll see a list of available models, including Anthropic models (e.g., Claude 3 Sonnet, Claude 3 Haiku).



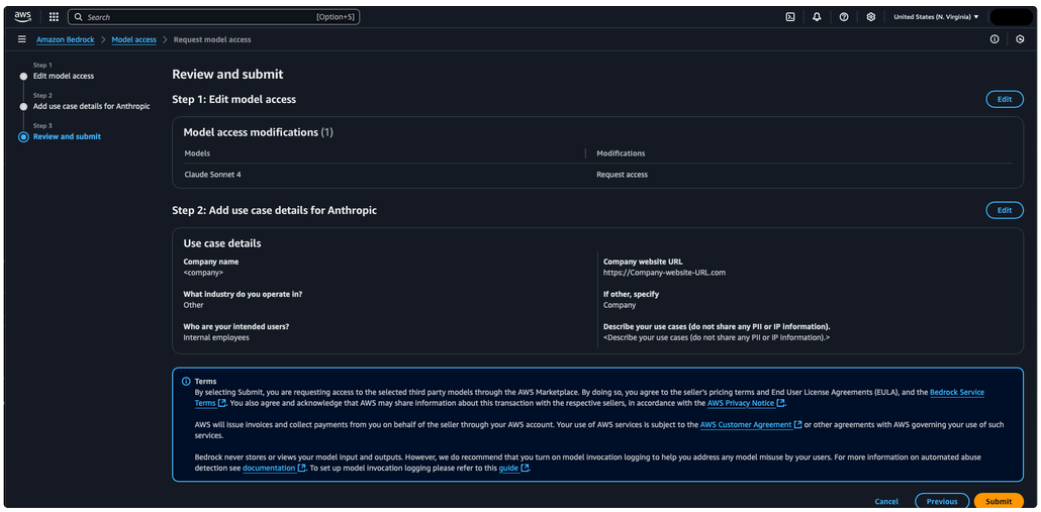
2. **Choose Models to Enable**:
   - Select **Modify model access**.
   - Choose **Enable specific models** and select the Anthropic models you need (e.g., Claude 3.5 Sonnet, Claude 3.7 Sonnet).



3. **Provide Use Case Details for Anthropic Models**:
   - Anthropic models require additional information. Fill out the following fields:
     - **Company name**: Your organisation's name (e.g., "MyCompany Inc.").
     - **Company website URL**: Your organisation's website (e.g., "https://mycompany.com").
     - **Industry**: Select the relevant sector from the dropdown (e.g., "Technology" or "Other").
     - **Intended users**: Check the appropriate box (e.g., "Internal employees" or "External users").
     - **Use case description**: Provide a brief description of your use case (max 500 characters, avoid PII or IP information). For example:
       - "Using Anthropic models for internal chatbot development to assist employees with task automation."
     - **Important**: Do not mention specific model names like "Claude" in the description to avoid automatic denial.
   - Agree to the terms, including sharing aggregated metrics with Anthropic for trust and safety purposes.



4. **Review and Submit**:
   - Review your selections and the End User License Agreements (EULAs).
   - Click **Submit** to send the access request.
   - Wait for the **Access granted** status, which typically takes a few minutes.



5. **Verify Access**:
   - Return to the **Model access** page to confirm the status of your requested models. Anthropic models (e.g., Claude 3 Sonnet) should show **Access granted**.

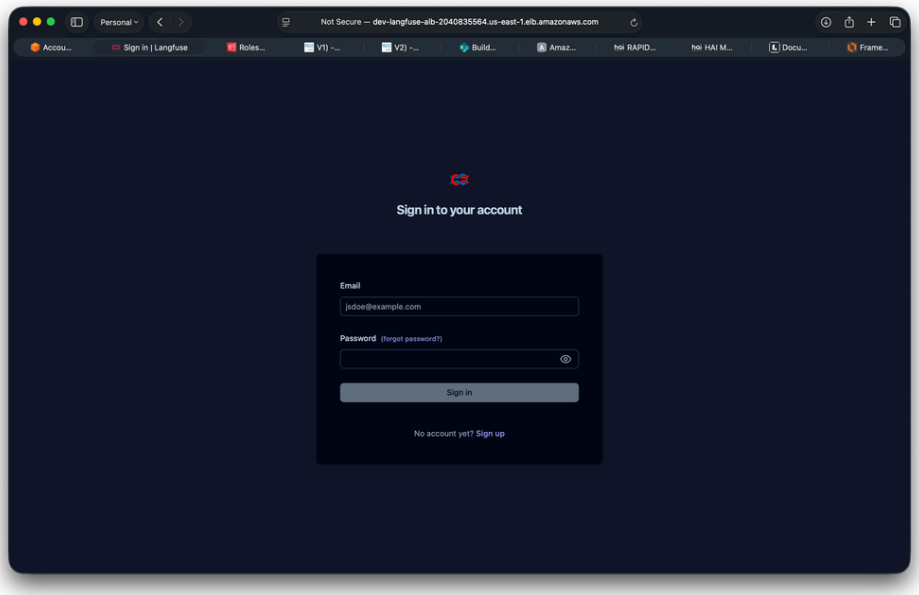
Application Setup and Langfuse Integration

This section outlines the essential steps for configuring your Langfuse environment, deployed via an AWS CloudFormation template, and subsequently integrating it with both Specifai and HAI Code Generator for robust LLM observability.

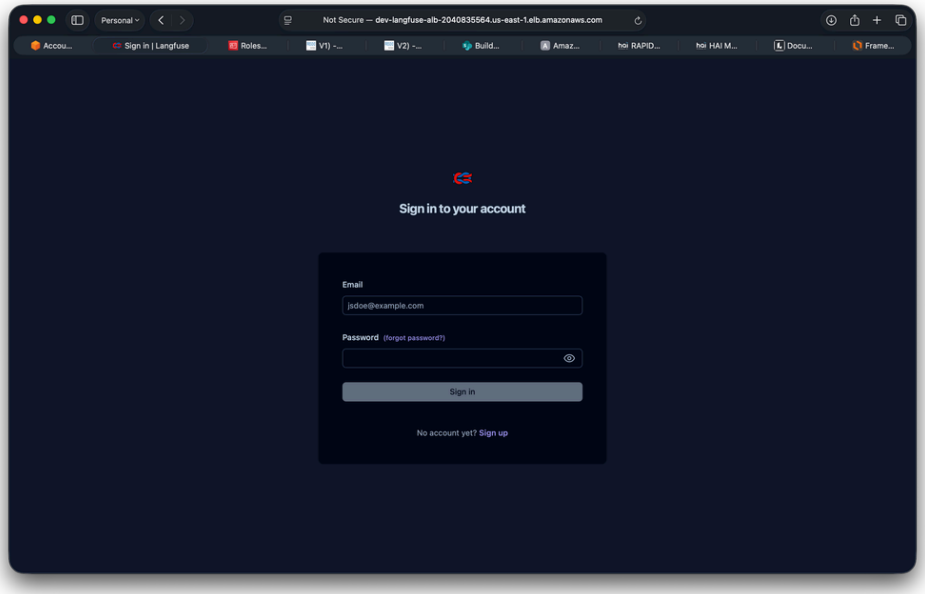
1. Accessing the Deployed Langfuse Application

Upon successful deployment of the Langfuse server through the provided CloudFormation template, the primary access URL will be available in the CloudFormation stack outputs.

- **Locate Application URL:** Retrieve the `ApplicationURL` from your CloudFormation stack's **Outputs** section. This URL represents your deployed Langfuse instance (e.g., `https://langfuse.example.com`).
- **Initial Account Creation:** Navigate to the `ApplicationURL` in your web browser. Follow the on-screen prompts to sign up for your Langfuse account using your email and a secure password.



- **Login:** After account creation, log in using your newly established Langfuse credentials.



2. Creating an Organization and Projects in Langfuse

Effective Langfuse integration necessitates logical organization of your observability data.

1. **Create Organisation**:
   - During your initial login, you may be prompted to create an Organization. Provide a descriptive name (e.g., `company-name`).
2. **Create Project**:
   - From the left sidebar within the Langfuse UI and click on **“Projects”**.
   - Click **“Create Project”**.
   - Assign a distinct name to each project (e.g., `hai-specifai`, `hai-code-generator`) and an optional description. Creating separate projects for each application (Specifai and Code Generator) is highly recommended to ensure isolation of trace logs, simplify debugging, and accurately track LLM costs per application.

3. Generating API Keys for Each Project in Langfuse

API keys are essential for authenticating your applications with the Langfuse server. Generate a unique set for each project.

- **Select Project:** Open the specific project you wish to configure (e.g., `hai-specifai`).
- **Navigate to API Keys:** Go to **Settings** > **API Keys**.
- **Generate Key:** Click **Generate API Key**.
- **Record Credentials:** Carefully copy and securely store the following:
  - **Public Key**
  - **Secret Key**
  - **Host URL:** Confirm the `ApplicationURL` (e.g., `https://langfuse.example.com`) which serves as your Langfuse host. This is crucial if your deployment is not `localhost`.

These recorded credentials (`Public Key`, `Secret Key`, `Host URL`) will be utilized in the telemetry configuration of both Specifai and Code Generator.

---

4. Specifai Setup

Integrate Langfuse with Specifai to capture telemetry for your LLM operations.

- **Getting Started:** Refer to the comprehensive Getting Started with Specifai guide.
- **AI Model Configuration:** Utilize the Bedrock IAM Keys provided in your **Specifai CloudFormation Outputs** within the AI Model Configuration section of the Specifai guide
- **Custom Langfuse Configuration:** Follow the instructions in the Specifai documentation's Custom Langfuse Configuration section. Updating configuration in the settings page with the `Public Key`, `Secret Key`, and `Host URL` obtained in Step 3 from Langfuse Server.

---

5. Code Generator Setup

Enable Langfuse telemetry for the HAI Code Generator VS Code Plugin.

- **Getting Started:** Consult the Getting Started with HAI Code Generator VS Code Plugin.
- **Telemetry Setup:** Refer to the Telemetry Setup for Langfuse Integration section within the guide.

---

5. Factifai Setup

- Explore Factifai Web https://github.com/presidio-oss/factif-ai
- Install Factifai Agent as npm package along with playwrite. https://factifai.io/getting-started/installation.html