AWS Marketplace Offering (HAI Build V1) - How to 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. Specif AI
An intelligent application that redefines how teams gather, refine, and manage requirements. Powered by AI, it organizes 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 (VS Code Plugin)
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.

Introduction

This guide provides a detailed walkthrough for deploying **HAI Build** using an AWS CloudFormation Stack. It covers the following steps to unlock the full potential of AI-driven software development:

1. **Download and Setup**
   - **Specif AI**: The Frontend Electron App for intuitive requirement management.
   - **Code Generator**: The VS Code Plugin Build for AI-assisted code generation.
2. **Deployment Process**
   - Automates resource provisioning for the **Specif AI *Backend App*** using a pre-configured CloudFormation template.
   - Simplifies deployment to ensure rapid and efficient setup of the AI-powered infrastructure.

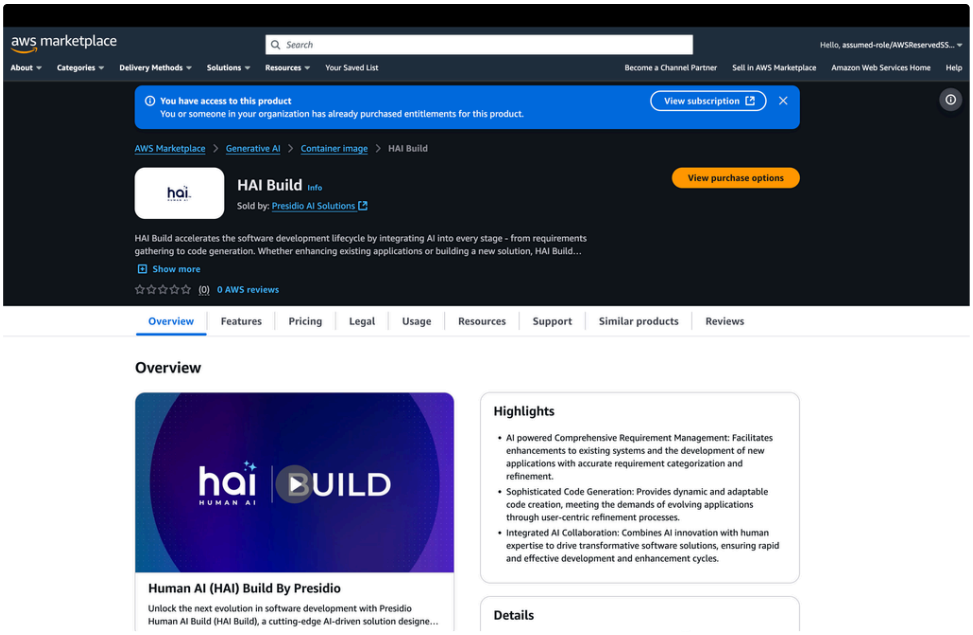
By following this guide, users can seamlessly integrate HAI Build into their workflow, enabling precision-driven requirement management and adaptive, high-quality code generation.

Prerequisites

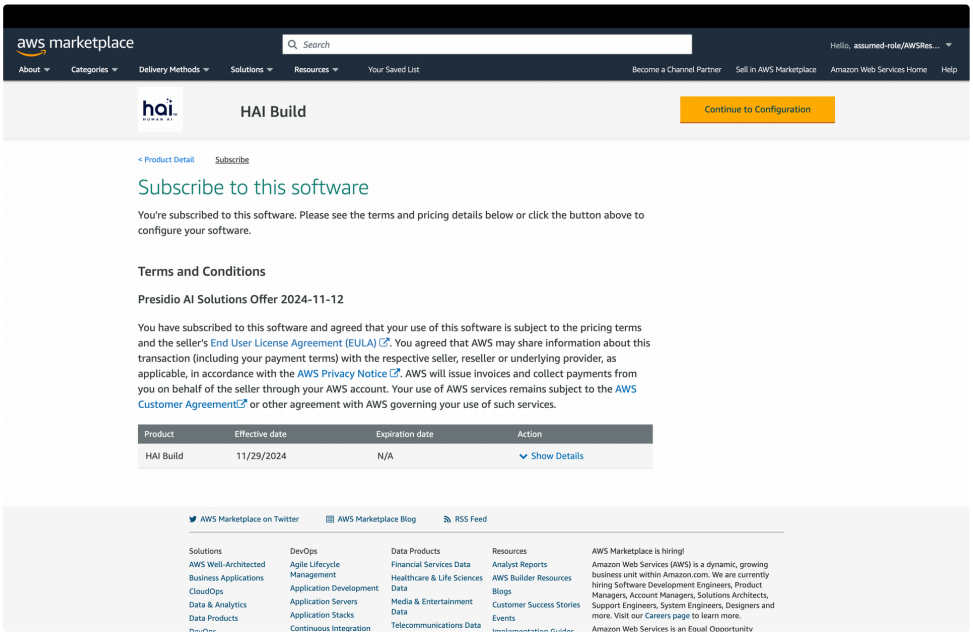
- **LLM Keys**: LLM keys are essential for the Hai Build Specif AI and Code Generator applications. Both support multiple models, including Azure OpenAI, OpenAI Native, and AWS Bedrock. You must obtain an LLM key from at least one LLM provider corresponding to the chosen model to enable the application to function.
- **AWS Account**: Access to an AWS account with required permissions.
- **IAM User Permissions**: AdministratorAccess or equivalent. https://aws-marketplace-cfn-template.s3.us-east-1.amazonaws.com/policy.json
- **Backend Hosting Domain**: A Domain for the Backend Service to be hosted. Link to register a domain using Route 53.
  - Certificate Name setup for your domain as detailed in the ACM Documentation.
- **AWS Marketplace Subscription**: Subscribe to Presidio HAI Build through AWS Marketplace.
- **Sentry Integration (Optional)**: Retrieve and monitor application logs in Sentry for enhanced debugging.

Steps to Deploy Presidio HAI Build

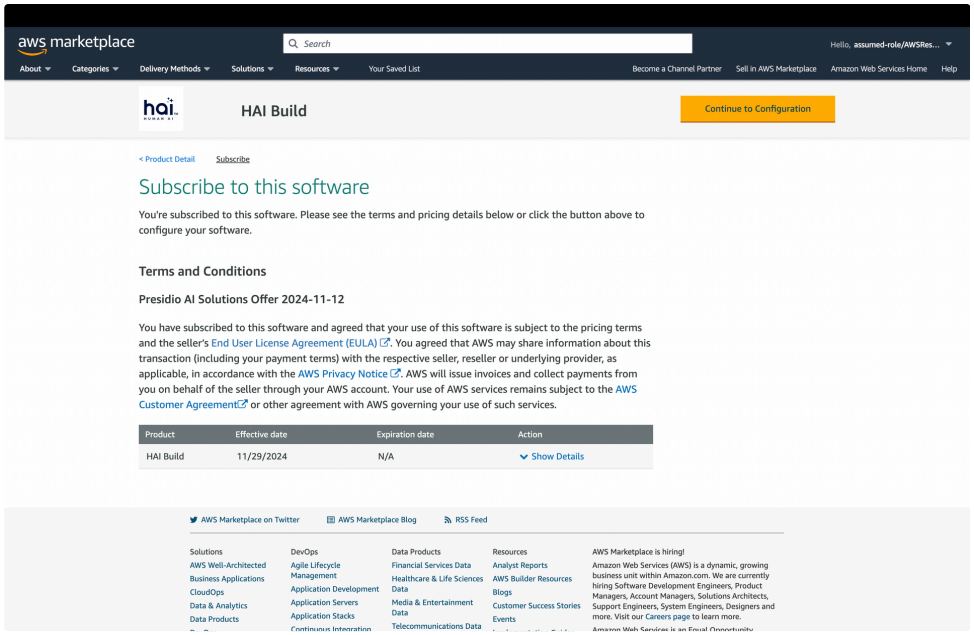
1. **Access AWS Marketplace**
   - Navigate to the Presidio HAI Build listing and click **“View Purchase Options”** for subscription options and pricing.



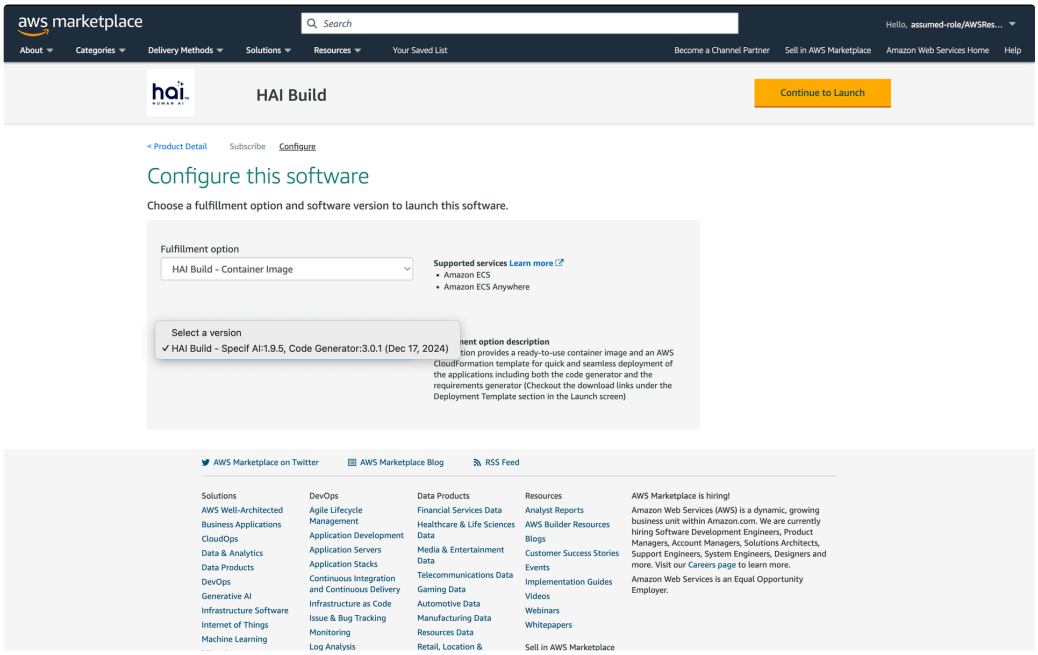
2. **Accept Terms and Conditions**
   - Review and accept the Terms and Conditions for the offering.



- Once accepted, click "***Continue to Configure***" to view the different Versions of the ***Presidio HAI Build*** to launch.



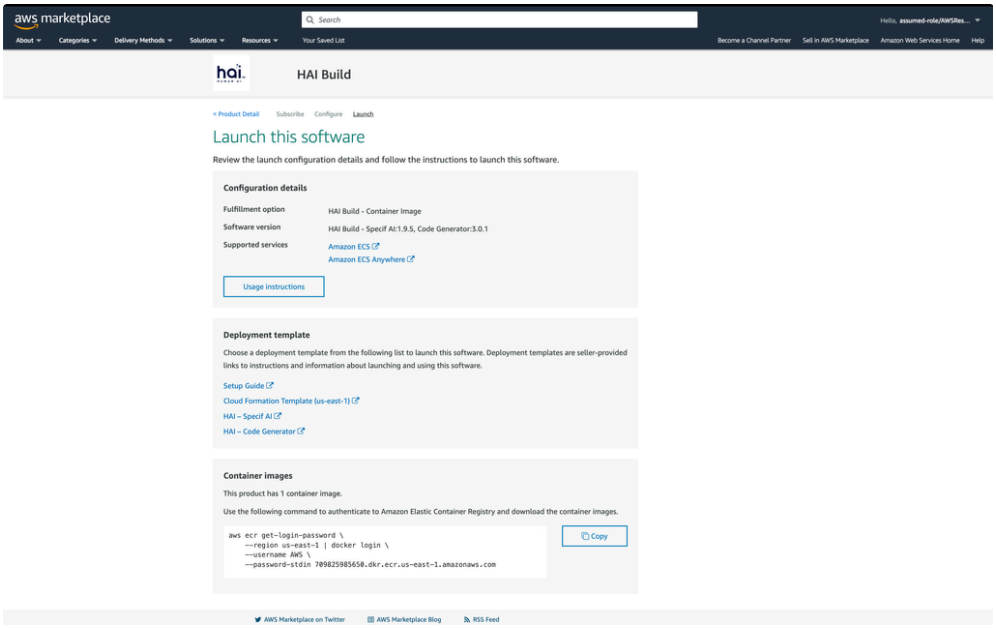
3. **Choose the Deployment Version**
   - Select the desired version of the Presidio HAI Build application and click **“Continue to Launch”**.



4. **Review Configuration Details**

   View the supported services and deployment configurations for your setup. The deployment template section includes the following links:

   - **Setup Guide**: Download the detailed guide for step-by-step instructions.
   - **CloudFormation Template**: Use the template to configure backend services for the Specif AI seamlessly.
   - : Download the Electron Desktop version for macOS (Intel, ARM) or Windows.
   - **Code Generator**: Download the VS Code Plugin compatible with macOS (Intel, ARM) or Windows.

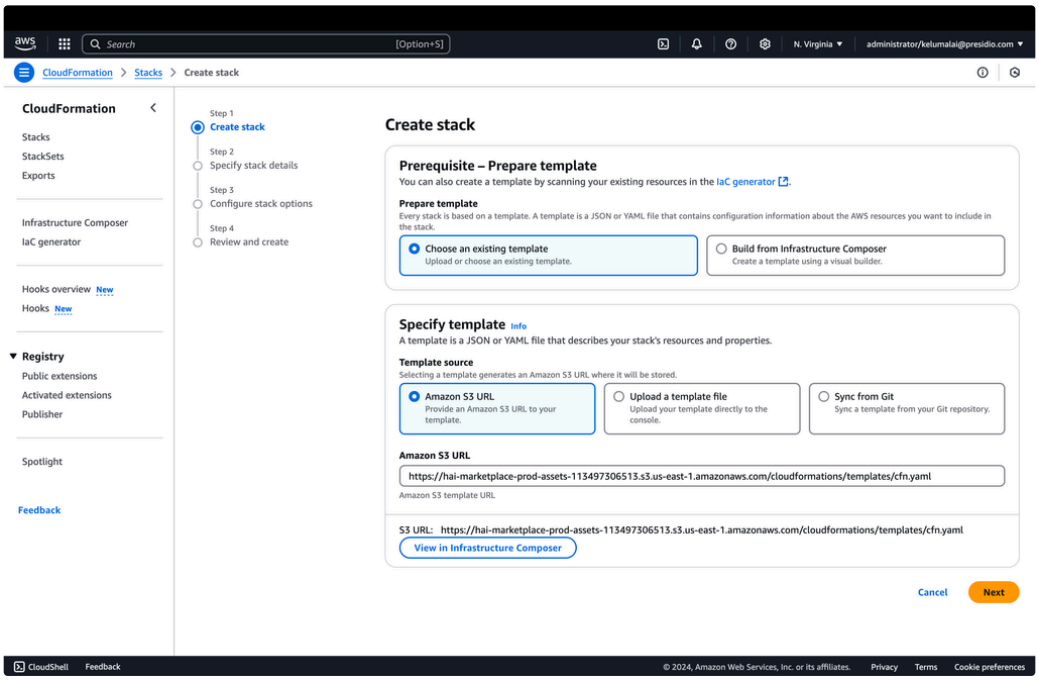


CloudFormation Stack Deployment

The CloudFormation Stack Deployment automates the creation of the required AWS infrastructure for the **Presidio HAI Build** in the customer's AWS account. Below is a step-by-step guide to deploying the CloudFormation Stack.

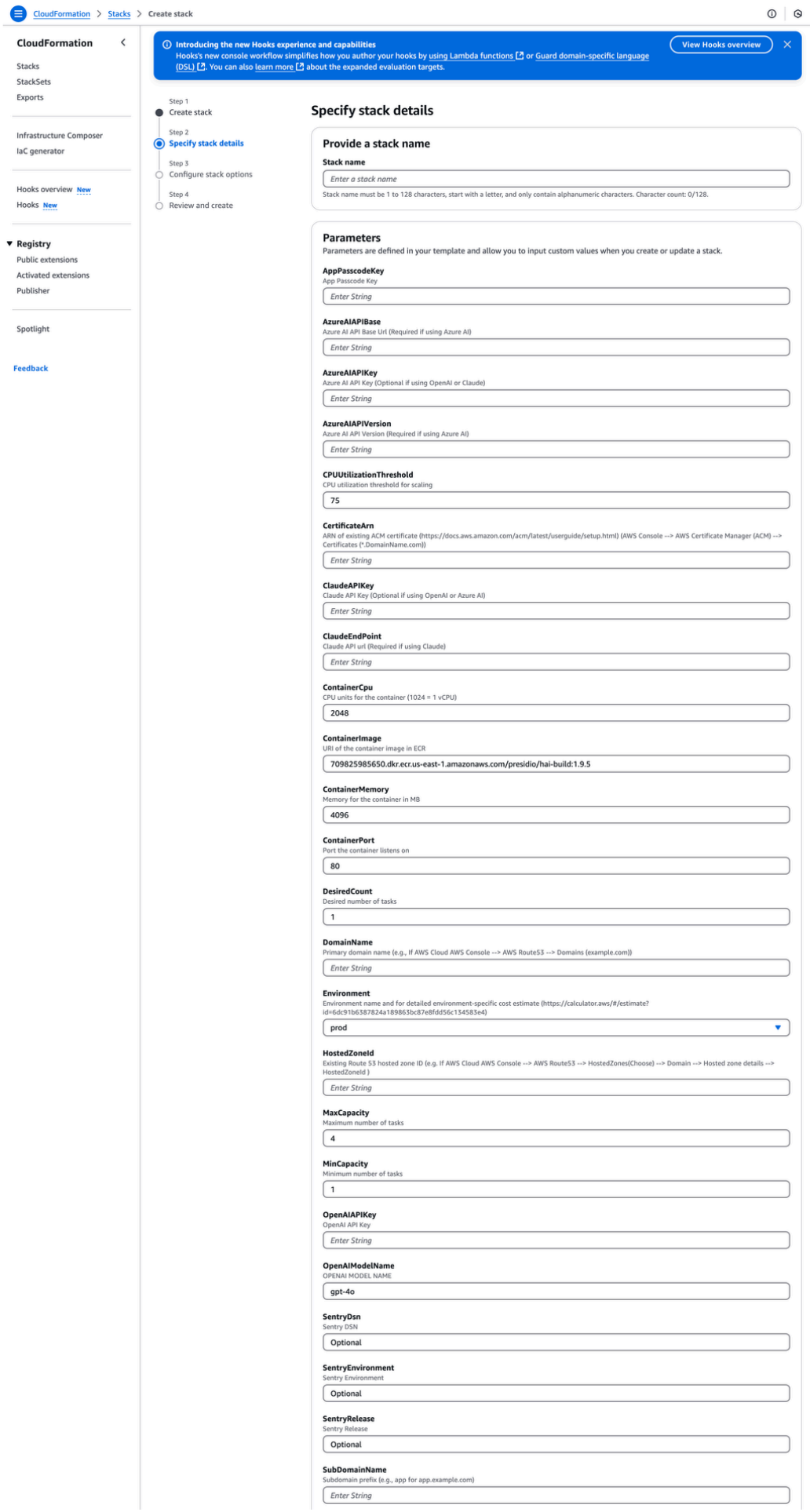
Step 1: Launch the CloudFormation Template

- Click on the provided **CloudFormation template** link for ECS deployment.
- This will navigate to the AWS CloudFormation page with the **Presidio HAI Build CFT template** pre-filled.
- Review the information and Click **Next** to proceed to Step 2

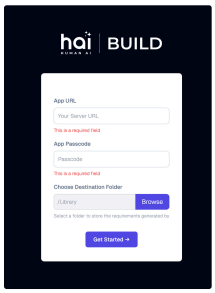


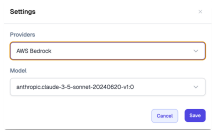
Step 2: Provide Stack and Parameter Details

1. **Review the Information**: Verify the pre-filled template and click **Next**.


CloudFormation > Stacks > Create stack
CloudFormation
Stacks
StackSets
Exports
Infrastructure Composer
IaC generator
Hooks overview New
Hooks New
Registry
Public extensions
Activated extensions
Publisher
Spotlight
Feedback
Introducing the new Hooks experience and capabilities
Hooks's new console workflow simplifies how you author your hooks by using Lambda functions or Guard domain-specific language (DSL). You can also learn more about the expanded evaluation targets.
View Hooks overview
Step 1
Create stack
Step 2
Specify stack details
Step 3
Configure stack options
Step 4
Review and create
Specify stack details
Provide a stack name
Stack name
Enter a stack name
Stack name must be 1 to 128 characters, start with a letter, and only contain alphanumeric characters. Character count: 0/128.
Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.
AppPasscodeKey
App Passcode Key
Enter String
AzureAIAPIBase
Azure AI API Base Url (Required if using Azure AI)
Enter String
AzureAIAPIKey
Azure AI API Key (Optional if using OpenAI or Claude)
Enter String
AzureAIAPIVersion
Azure AI API Version (Required if using Azure AI)
Enter String
CPUUtilizationThreshold
CPU utilization threshold for scaling
75
CertificateArn
ARN of existing ACM certificate (https://docs.aws.amazon.com/acm/latest/userguide/setup.html) (AWS Console --> AWS Certificate Manager (ACM) --> Certificates (*.DomainName.com))
Enter String
ClaudeAPIKey
Claude API Key (Optional if using OpenAI or Azure AI)
Enter String
ClaudeEndPoint
Claude API url (Required if using Claude)
Enter String
ContainerCpu
CPU units for the container (1024 = 1 vCPU)
2048
ContainerImage
URI of the container image in ECR
709825985650.dkr.ecr.us-east-1.amazonaws.com/presidio/hai-build:1.9.5
ContainerMemory
Memory for the container in MB
4096
ContainerPort
Port the container listens on
80
DesiredCount
Desired number of tasks
1
DomainName
Primary domain name (e.g., If AWS Cloud AWS Console --> AWS Route53 --> Domains (example.com))
Enter String
Environment
Environment name and for detailed environment-specific cost estimate (https://calculator.aws/#/estimate?id=6dc91b6387824a189863bc87e8fdd56c134583e4)
prod
HostedZoneId
Existing Route 53 hosted zone ID (e.g. If AWS Cloud AWS Console --> AWS Route53 --> HostedZones(Choose) --> Domain --> Hosted zone details --> HostedZoneId)
Enter String
MaxCapacity
Maximum number of tasks
4
MinCapacity
Minimum number of tasks
1
OpenAIAPIKey
OpenAI API Key
Enter String
OpenAIModelName
OPENAI MODEL NAME
gpt-4o
SentryDsn
Sentry DSN
Optional
SentryEnvironment
Sentry Environment
Optional
SentryRelease
Sentry Release
Optional
SubDomainName
Subdomain prefix (e.g., app for app.example.com)
Enter String

2. **Enter Stack Name and Parameters**: In Step 2, fill out the required fields as outlined below:

Parameter Name	Description	Default/Sample Value
AppPasscodeKey	App Passcode for logging into the **Specif AI** Frontend. 	Example: `987654`
AzureAIAPI Base	Azure AI API Base Url (Required if using Azure AI) for backend Services. The value can be inferred from the Azure Open AI Deployment Configuration. In Settings, select the Azure OpenAI provider and the appropriate deployed model to utilize the Azure Deloyed LLM Model. 	Example: `https://example-azure-ai.openai.azure.com/openai/deployments`
AzureAIAPI Key	Azure AI API Key (Required if using Azure AI) for backend Services.	Example: `xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx`
AzureAIAPI Version	Azure AI API Version (Required if using Azure AI) for backend Services.	`2024-08-01-preview`
CPUUtilizationThreshold	CPU utilization threshold for scaling.	Default: `75`
CertificateArn	ARN of the existing ACM certificate.	Example: `arn:aws:acm:us-east-1:xxxxxxxxxxx:certificate/xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx`
ClaudeAPIKey	Claude API Key (Optional if using OpenAI or Azure AI) In Settings, select the AWS Bedrock provider and the `anthropic.claude-3-5-sonnet-20240620-v1:0` deployed model to utilize the AWS Bedrock Deloyed LLM Model.	Example: `pk-xx-xxxxxxxx-xxxx-xxxx-xxxxxxxxxxx`

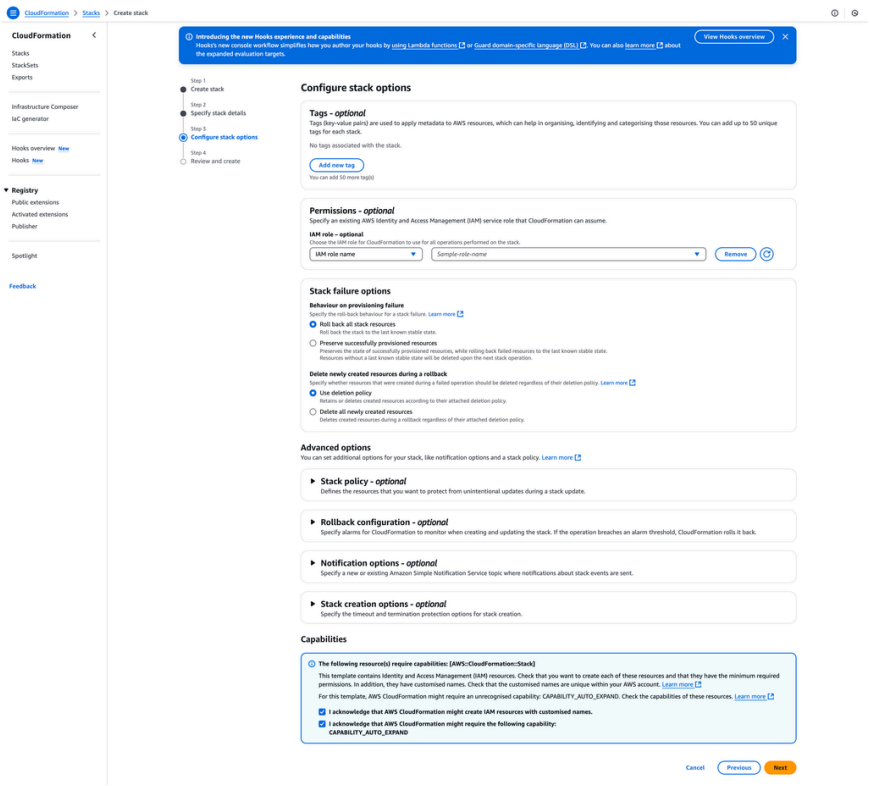
		
ClaudeEndPoint	Claude API URL (Required if using Claude)	Example: `http://example-bedrock-xxxx-xxxx.us-west-2.elb.amazonaws.com/api/v1/`
ContainerCpu	CPU units for the container (1024 = 1 vCPU).	Default: `2048`
ContainerImage	ECR URL of the container image.	Default: `709825985650.dkr.ecr.us-east-1.amazonaws.com/presidio/hai-build:1.9.4`
ContainerMemory	Memory for the container in MB.	Default: `4096`
ContainerPort	Port for HTTP server requests.	Default: `80`
Desired Count	Desired number of tasks to run.	Default: `2`
DomainName	Primary domain for hosting the Specif AI App.	Example: `http://example.com`
Environment	Deployment environment name.	Default: `prod`
HostedZoneId	Route 53 hosted the zone ID for the domain. - Backend App	Example: (Route 53 Hosted Zone Details)
MaxCapacity	Maximum number of tasks.	Default: `4`
MinCapacity	Minimum number of tasks.	Default: `1`
OpenAIAPIKey	OpenAI API key for backend services. In Settings, select the OpenAI Native provider and the appropriate model to utilize it. 	Example: `sk-xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx`
OpenAIModelName	Open AI Model Name for Backend Services LLM calls.	Example: gpt-4o
SentryDsn	Sentry DSN for monitoring and error tracking(Optional).	Example: `https://xxxxxxxxxxxx.ingest.sentry.io/xxxxxxxx`

SentryEnvironment	Environment name for Sentry integration(Optional).	Example: `prod`
SubDomainName	Subdomain prefix (e.g., `app` for `app.example.com`).	Example: `app`
VpcCidr	CIDR block for the VPC.	Default: (Ensure NAT Gateway availability)

3. Proceed to the Next Step.

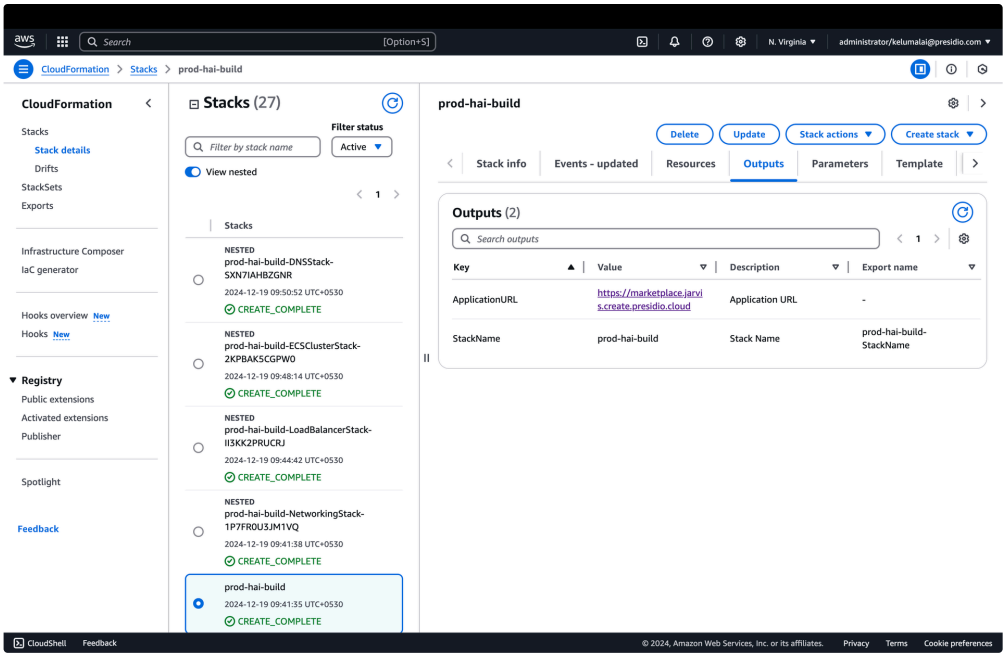
Step 3: Add Tags, Permissions, and Options

- Provide **tags** for better resource management.
- Configure stack permissions and specify failure handling options.
- Accept the required capabilities and click **Next**.



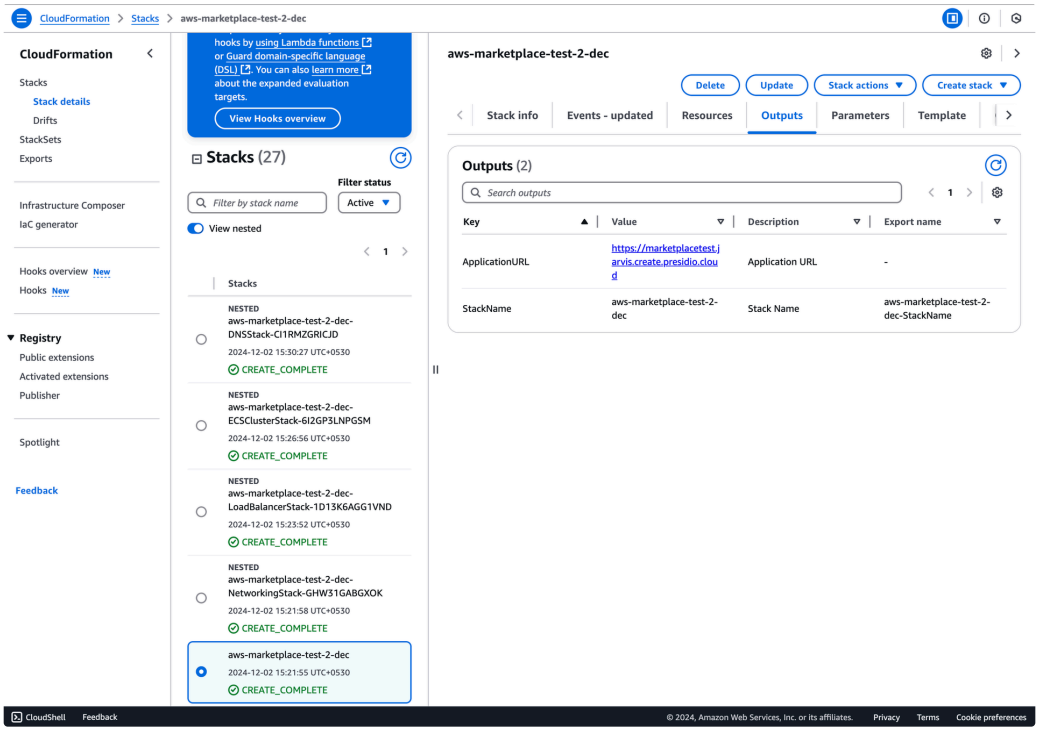
Step 4: Review and Execute

- **Review the Stack Configuration**: Ensure all parameter values and the complete template are correct.
- Click **Execute** to deploy the resources.



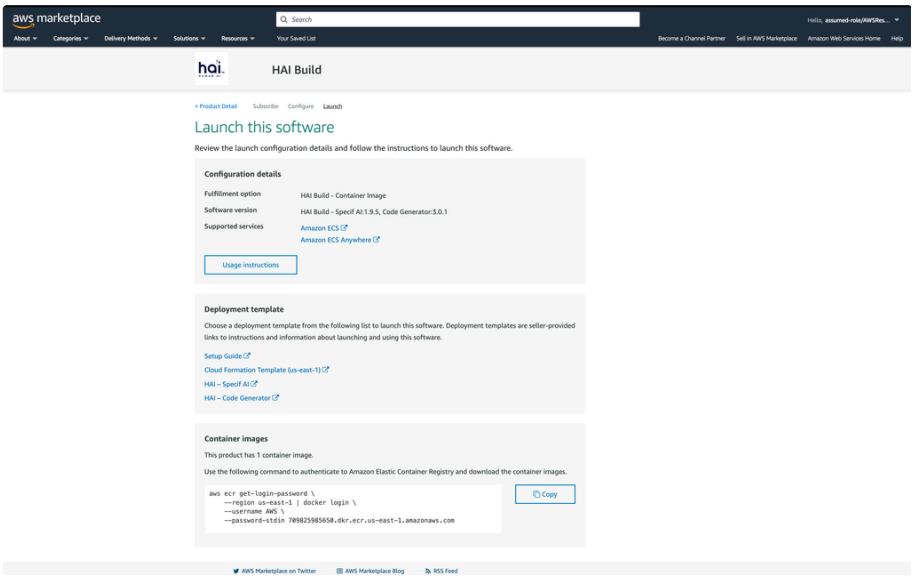
Post-Deployment Verification

1. **Check Stack Status**: Confirm that the parent CloudFormation stack and its four nested stacks have been successfully created.
2. **Retrieve the Backend URL**:
   - Go to the **Outputs** section of the parent stack.
   - Note the fully hosted URL for the **Presidio HAI Build Backend** (e.g., `https://app.example.com/`).



Steps to Download and Set Up HAI Build Specif AI App Frontend and Code Generator VS Code Plugin

The following steps guide you through the process of downloading and setting up the **HAI Build Specif AI Frontend** and the **Code Generator VS Code Plugin (VSIX)**.



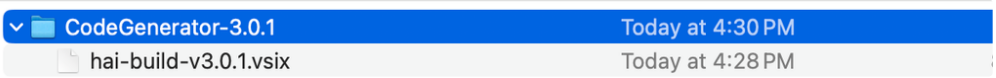
---

1. Download the Required Builds

Utilizing the provided download links from the AWS Marketplace - Deployment Template section, ***Download*** the appropriate builds for your platform:

- **HAI Build Specif AI App Frontend**: Select the Desktop App that matches your operating system (macOS Intel/ARM or Windows).

- **HAI Build Code Generator Plugin (VSIX)**: Select the VSIX package that matches your operating system (macOS Intel/ARM or Windows)



---

2. Set Up the HAI Build Specif AI Frontend

For Windows:

- Unzip the downloaded zip file and locate the **.exe** installer for the **Specif AI**.
- Run the installer and follow the on-screen instructions.
- Once installed, launch the **HAI Build Specif AI** from your desktop or the Start menu.

For macOS:

- Locate the downloaded **.dmg** file for the Specif AI App.
- Open the **.dmg** file and drag the application into your **Applications** folder.
- Launch the **HAI Build App** from the **Applications** folder.

---

3. Set Up the HAI Build Code Generator Plugin (VSIX)

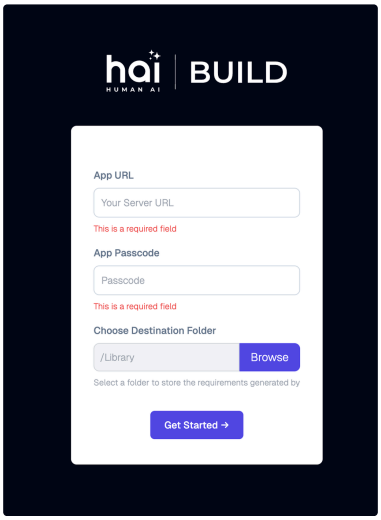
- Open **Visual Studio Code** on your machine.
- Click on the **Extensions** icon in the left sidebar to access the Extensions Marketplace.
- In the top-right corner of the Extensions panel, click the **three-dot menu** and select **Install from VSIX**.
- Locate the downloaded **VSIX** file and click **Install**.
- After installation, open the **HAI Build Code Generator Plugin** from the **Extensions** menu.

---

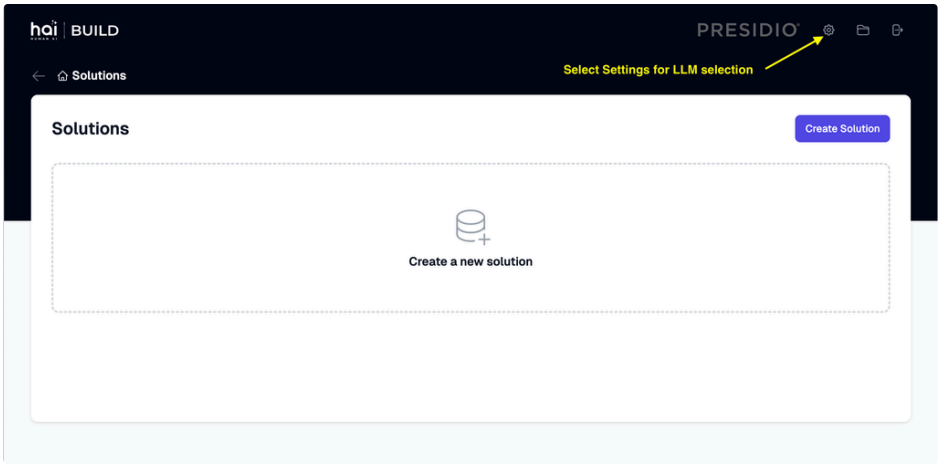
4. Verify the Setup for Specif AI:

Follow these steps to ensure the successful setup of the **HAI Build Specif AI**:

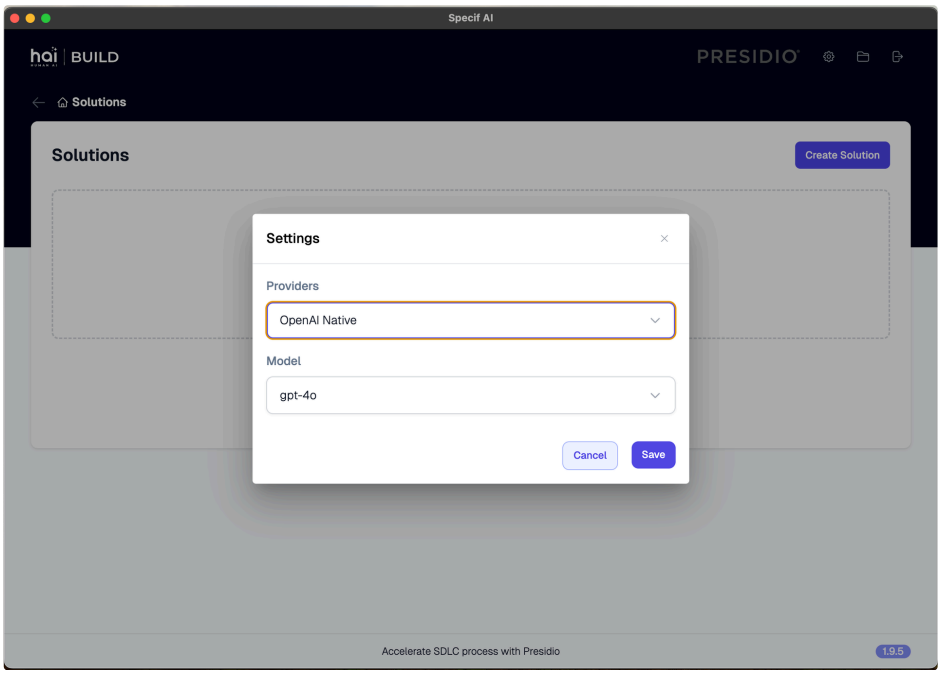
- **Access the Specif AI**:
  - Open the Specif AI installed on your system.
  - Use the **App URL** obtained from the deployed Backend URL during the CloudFormation deployment.
  - Enter the **App Passcode** which was provided as a parameter(**AppPasscodeKey**) during the CloudFormation deployment process to log in.
  - Choose the HAI Build **Destination Folder** that stores the requirements generated by HAI Build.



- **Check Functionality**
  - Verify successful login using the App URL and provided credentials.
  - Select LLM Config under the settings section.
    - Hit the Settings icon as shown below



- As shown below, the user can select the LLM Provider and appropriate Model on the settings page and ***save*** the changes. Choose the model that best suits your needs. Supported models include:
  - Azure OpenAI
    - gpt-4o
    - gpt-4o-mini
  - OpenAI Native
    - gpt-4o
    - gpt-4o-mini
  - AWS Bedrock
    - anthropic.claude-3-5-sonnet-20240620-v1:0



Note: Ensure the relevant Provider AI Keys are provided on the Backend Services via the Parameters Section.

- Create a sample solution and validate the generated BRD, PRD, and other specifications.
- Utilize the “Expand with AI” and “AI Chat” options by editing any of the solution requirement specifications (BRD, PRD, User Story, Task, User Interface, Business Process)

5. Verify the Setup for Code Generator Plugin

Follow these steps to confirm the successful setup of the **HAI Build Code Generator Plugin**:

- **Access the Plugin**:
  - Open Visual Studio Code.
  - Press **Ctrl + Shift + P** (or **Cmd + Shift + P** on macOS) to open the command palette.
  - Search for and select **"View: Show HAI Build"** to access the HAI Plugin.
- **Configure Settings**:
  - Navigate to the **Settings Page** within the plugin.
  - Add the **OpenAI API Key** to enable functionality.
- **Test Code Generation**:
  - Use the **chat option** within the plugin to execute a sample task from the Code Generator.
- **Import and Validate**:
  - Click on **HAI Tasks** and Import the **Solution folder** inside the **HAI Root Folder**.
  - Execute code generation for a sample task created in the Specif AI to ensure proper functionality and output accuracy.

This guide provides detailed steps to deploy, download, set up, and verify the proper functioning of the **HAI Build Specif AI Frontend** and the **Code Generator VS Code Plugin**. If you encounter any issues during deployment or setup, refer to the documentation for troubleshooting.