HCLSoftware

HCL MaxAI 25.1.0 Deployment Guide

Contents

Chapter 1. Overview

MaxAI is a suite of cloud-native applications, designed to run on Kubernetes clusters, deployed using Helm charts for simplified deployment and management.

This document provides comprehensive instructions for deploying MaxAI in kubernetes environments.

The deployment process leverages Kubernetes' container orchestration capabilities and Helm's package management features to ensure consistent, repeatable deployments across different environments. Understanding these foundational technologies will help you better manage your MaxAI installation throughout its lifecycle.

Chapter 2. Prerequisites

Before beginning the MaxAI installation, ensure your environment meets all the following requirements. These components work together to create the foundation for your MaxAI deployment:

Docker Engine (Version 20.10 or higher)

Docker serves as the container runtime that packages MaxAI and its dependencies. Verify your installation by running `docker --version` and ensure the Docker daemon is running properly.

A working Kubernetes cluster (Version 1.27 or higher)

Kubernetes cluster provides the orchestration platform where MaxAI will run. This can be a managed service like Amazon EKS, Google GKE, Azure AKS, or a self-hosted cluster using tools like kubeadm or k3s.

Helm (Version 3.8 or higher)

Helm acts as the package manager for Kubernetes, simplifying the deployment and management of MaxAI. Install Helm on the machine from which you will perform the deployment.

kubectl configured to connect to your cluster

The kubectl command-line tool allows you to interact with your Kubernetes cluster. Ensure it's configured to communicate with your target cluster by running `kubectl cluster-info`.

Images and Bundles

Access to the MaxAI docker images (for the applications) and Helm Deployment bundle.

Ingress Controller

NGINX Ingress Controller 2.0 must be installed.

Access Requirements

Cluster Administrative Access

You need cluster-admin privileges or sufficient RBAC permissions to create namespaces, deploy applications, and manage cluster resources.

Container Registry Access

Ensure your Kubernetes nodes can pull container images from the registry hosting MaxAI images. This may require configuring image pull secrets for private registries.

Network Connectivity

Verify that your cluster has outbound internet connectivity for downloading Helm charts and container images.

Note: Not applicable when you're using a fully air-gapped environment with local mirrors.

Minimum System Configurations

Since the MaxAI consists of multiple powerful components, we have a minimum system requirement to be able to install, configure and run the applications successfully. However, actual working configuration needs to be discussed and decided during the capacity planning and site-survey conducted jointly by HCL Software and Client teams.

Hardware Requirements

Memory

16 GB RAM

Storage

50 GB HDD

CPU

8 vCPUs

Chapter 3. Deployment Bundle

The MaxAI deployment bundle contains several interconnected components that work together to provide a complete deployment solution.

Helm Chart Structure

`Chart.yaml`

This file contains metadata about the MaxAI Helm chart, including version information, dependencies, and chart description. It serves as the primary descriptor for the entire package.

`Values.yaml`

The default configuration file that defines all customizable parameters for your MaxAI deployment. This includes resource allocations, service configurations, database connections, and feature toggles.

`templates/ directory`

Contains Kubernetes manifest templates that Helm processes to generate the actual deployment resources, and individual MaxAI components. These templates use Go templating syntax to incorporate values from the configuration files.

`charts/ directory`

Houses all dependent Helm charts that MaxAI requires, such as database charts.

Chapter 4. Pre-deployment Checklist

Complete these verification steps before beginning the deployment process. This systematic approach helps identify potential issues early and ensures a smooth deployment experience.

Environment Verification

Confirm your Kubernetes cluster is healthy by checking node status with `kubectl get nodes`. All nodes should show **Ready** status, and you should verify that system pods in kube-system namespace are running properly.

Test your Helm installation by running `helm version` and ensure it can communicate with your cluster by executing helm list in your target namespace.

Resource Availability Check

Verify sufficient resources are available in your cluster by reviewing current resource usage with `kubectl top nodes` and `kubectl describe nodes`. Ensure there is adequate CPU, memory, and storage capacity for the MaxAI component deployments.

Check that your chosen namespace exists or can be created, and verify that your kubectl context is set to the correct cluster and namespace for the deployment.

Security Considerations

Review your cluster's security policies, including Pod Security Standards, Network Policies, and RBAC configurations. Ensure these policies won't interfere with MaxAIs operation while maintaining appropriate security boundaries.

If using private container registries, verify that image pull secrets are properly configured and accessible from your target namespace.

Database Support

MaxAI supports MariaDB as database backends to accommodate different operational requirements.

If you are using an existing database server (external), ensure it's accessible from your Kubernetes cluster and that appropriate firewall rules and security groups allow connectivity. Create a dedicated database and user account for MaxAI with appropriate permissions for normal operation but restricted administrative access.

Chapter 5. Deployment

The deployment process follows a sequence that builds upon previous steps, ensuring each component is appropriately configured before proceeding to the next phase.

1: Environment Preparation

1. Create a dedicated namespace for MaxAI to provide resource isolation and easier management. Run the following command to establish a dedicated environment:

   ```
   kubectl create namespace maxai
   ```

2. Run the following command to configure the `kubectl` context to use the MaxAI namespace by default. It prevents accidental deployment of resources to the wrong namespace.

   ```
   kubectl config set-context --current --namespace=maxai
   ```

2: Customizing Configurations

1. Create a custom values file by copying the default `values.yaml` file and modifying it for your specific environment.
2. Name this file something descriptive like `maxai-production-values.yaml` to clearly identify its purpose.

3: Configuring the `values.yaml` File

About this task

Before installing the Helm charts, configure the downloaded values.yaml file as per your environment settings:

1. From the deployment bundle, open the `values.yaml` file.
2. Configure the following parameters as per your environment setup:

   Note: Some parameters should not be modified, some parameters do not need to be configured, and some parameters need to be configured. The details for the parameters are mentioned in the following table:

Parameter Name	Type	Description and Example Values
hclImagePullSecret	To be configured	**Example**: `regcred`
mariadb_ip	To be configured	The MariaDB hostname or IP address. **Example**: `maxai-suite-database`
mariadb_port	To be configured	The port number of the MariaDB hostname or IP address. **Example**: `3306`

Parameter Name	Type	Description and Example Values
securityContext	To be configured	The values for both sub-parameters must be the same: **Example**: ◦ **runAsUser** - `5002` ◦ **runAsGroup** - `5002`
BASIC_AUTH_USERNAME	To be configured	The authorization credentials. **Example**: `admin`
BASIC_AUTH_PASSWORD	To be configured	The authorization credentials. **Example**: `secret`
CIPHER_KEY	Not Applicable	
MODEL_ACCESS_KEY	Not Applicable	
TOKEN_EXEMPTION_USERS	To be configured	Configure the users that you want to be exempt from GenAI token limitation. You can configure multiple users. **Example**: `username1,username2`
PLATFORM_BASE_URL	To be configured	HCL Unica Platform URL or service name (if MaxAI is deployed on same network) **Example**: `http://unica-platform-domain:1111`
MAX_AI_CONFIG_PATH	To be configured	The path where MaxAI configurations are implemented on HCL Unica. **Example**: `Affinium|AION`
CLIENT_ID	To be configured	Credentials for accessing HCL Unica Platform. **Example**: `username`
ACCESS_KEY	To be configured	Credentials for accessing HCL Unica Platform.

Parameter Name	Type	Description and Example Values
		Example: `password`
chatBotServerURL	Do not change	**Example**: `CHATBOT_HOST_URL`
HOST	Do not change	**Example**: `localhost`
PORT	Do not change	**Example**: `9000`
CONTEXT_ROOT	Do not change	**Example**: `/platform`
REDIS_HOST	To be configured	The hostname of the Redis service. **Example**: `redis-leader`
REDIS_PORT	To be configured	The port number for the Redis service. **Example**: `6379`
REDIS_PASSWORD	To be configured	The password for Redis service. **Example**: *`password`*
REDIS_DB	Do not change	**Example**: `0`
REDIS_SESSION_TTL	Do not change	**Example**: `3600`
VERIFY_SSL	To be configured	Determines whether SSL certificates should be verified when making HTTPS requests. On Production is should be `true`. **Example**: `true`
SECURE_COOKIE	To be configured	Specifies whether cookies should be marked as secured (transmitted only over HTTPS). On Production is should be `true`. **Example**: `true`
TOKEN_EXEMPT_SERVICES	To be configured	Configure the services that you want to be exempt from GenAI token limitation. You can configure either the `email-generation` service, the `sla` service or both.

Parameter Name	Type	Description and Example Values
		Example: `email-generation,sla`
API_REQUEST_TIMEOUT	To be configured	The API request timeout, in seconds, for analytics service. **Example**: `300`
ANALYTICS	Do not change	The service name for Analytics. **Example**: `maxai-analytics`
INSIGHTS	Do not change	The service name for Insights. **Example**: `maxai-insights`
OFFER	Do not change	The service name for Offer. **Example**: `maxai-offer`
CLASSIFICATION	Do not change	The service name for Classification Engine. **Example**: `maxai-classification-engine`
EMAIL	Do not change	The service name for Email Generation. **Example**: `maxai-email-generation`
CONFIG	Do not change	The service name for Config Service. **Example**: `maxai-config-service`
SLA	Do not change	The service name for SLA. **Example**: `maxai-sla-service`
DOCUMENT_SEARCH	Do not change	The service name for Document search. **Example**: `maxai-document-search`
API_GATEWAY	Do not change	The service name for API Gateway. **Example**: `maxai-analytics`

Parameter Name	Type	Description and Example Values
LOG_LEVEL	Do not change	Used for configuring logger. **Example**: `DEBUG`
LOG_TO_FILE	Do not change	Used for configuring logger. **Example**: `False`
LOG_FILE	Do not change	Used for configuring logger. **Example**: `logs/app.log`
LOG_FILE_ROTATION	We recommend that you do not change. Can be modified, if required.	**Example**: `10 MB`
OLD_LOG_FILES_RETENTION	Not Applicable	
OLD_LOG_FILES_COMPRESSION	Not Applicable	
MAX_HISTORY	Do not change	**Example**: `2`
MODERATION_API	Do not change	**Example**: `https://api.openai.com/v1/moderations`
CONST_SQL_API_TIMEOUT	Do not change	**Example**: `200`
FEEDBACK_LOG_PATH	We recommend that you do not change. Can be modified, if required.	The path where the session logs are stored. **Example**: `/mnt/hclgenai_share/ce/chat_session_logs`
OPENAI_ASSISTANT_ID	To be configured	OpenAI RAG assistant for docsearch. **Example**: `assistant_XxXXXXXXxxXX0xX2XxX0X00`
USE_ASSISTANT	To be configured	Enable or disable OpenAI RAG assistant. Disabled by default.

4: Installing Helm Charts

1. Navigate to the directory containing the MaxAI chart.
2. Execute the following Helm installation command:

```
helm install genai ./hclgenai-hclnow -f maxai-production-values.yaml --namespace maxai
```

- The installation name `genai` can be customized based on your naming conventions.
- The directory name `hclgenai-hclnow` has to be the directory name where your deployment bundle is present.
- The `maxai-production-values.yaml` is the `values.yaml` file, customized for your environment.
- The namespace `maxai` is the one you have created for deploying the MaxAI suite of applications.

Run the following command to monitor the installation progress and to watch pods start up in real-time:

```
kubectl get pods -w
```

The deployment process typically takes several minutes as containers are pulled and started.

Change the permission for the shared mount point to `genaiuser` with **uid** and **gid** as `5002`.

```
chown 5002:5002 /mnt/hclgenai_share
```

5: Run the Provided SQL scripts

About this task

After installation and before verification, download and run the SQL script on MaxAI database `ce_metering_db`. Remember to run the script before moving on to verification. **Download Script**.

6: Initial Verification

1. If the installation is successful, all the pods will be up and running successfully. A Sample deployment output follows:

```
NAME                                                  READY   STATUS      RESTARTS   AGE
genai-hcl-genai-suite-database-0                      1/1     Running     0          7h25m
genai-hclgenai-classification-engine-544fdcdf6b-n7hhn 1/1     Running     0          7h25m
genai-hclgenai-documentsearch-6cd78484c9-d9jgv        1/1     Running     0          7h25m
genai-hclgenai-email-gen-64b5f8cf54-zjrfr             1/1     Running     0          7h25m
genai-hclgenai-maxai-analytics-5c95cfc5f8-zcjnq       1/1     Running     0          7h25m
genai-hclgenai-maxai-insights-6666c45568-7hwzn        1/1     Running     0          7h25m
genai-hclgenai-sla-56786f8d68-phpbn                   1/1     Running     0          7h25m
genai-maxai-api-gateway-7f4cc44788-snqwk              1/1     Running     0          7h25m
genai-maxai-config-service-5867d7d6d6-4l6zw           1/1     Running     0          7h25m
genai-maxai-offer-6b9788c7cf-jzrjf                    1/1     Running     0          7h25m
redis-leader-858cb8c796-kdp8p                         1/1     Running     0          7h25m
```

2. You can also check service status to ensure networking is configured correctly and that load balancers or ingress controllers have been properly provisioned. Run the following command:

```
kubectl get services
```

What to do next

After deploying MaxAI, you must configure it. For information related to MaxAI configurations, see *HCL MaxAI Administrator's Guide*.

Chapter 6. Uninstallation

To remove MaxAI from your cluster, follow these steps to ensure complete cleanup:

Run the following command:

```
helm uninstall genai --namespace maxai
```

Note:

- The installation name used during installation, was `genai`.
- The namespace in which you deployed MaxAI, was `maxai`.