## Contents

[Contents 1](#_Toc185410777)

[**Project Name:** 2](#_Toc185410778)

[**Project Overview:** 2](#_Toc185410779)

[**Environments:** 2](#_Toc185410780)

[**AWS Account Details:** 2](#_Toc185410781)

[**AWS Access for Users:** 2](#_Toc185410782)

[**BootStrapping:** 2](#_Toc185410783)

[**AWS boot strapping Process:** 2](#_Toc185410784)

[**Infrastructure Provisioning:** 3](#_Toc185410785)

[**AWS Infrastructure:** 4](#_Toc185410786)

[**SSL Certificates:** 5](#_Toc185410787)

[**OpenSearch configuration:** 5](#_Toc185410788)

[**Azure Pipeline for Infrastructure Setup:** 9](#_Toc185410789)

[**Azure Devops or CI/CD Setup for AWS infra:** 9](#_Toc185410790)

[**Login to Bastion Host Server:** 11](#_Toc185410791)

[**Application Deployment** 11](#_Toc185410792)

[**Git Repository:** 12](#_Toc185410793)

[**Branching Strategy for Deployment:** 12](#_Toc185410794)

[**Git CI info:** 12](#_Toc185410795)

[**Git CD info:** 12](#_Toc185410796)

[**CI/CD Setup for UI (nextjs-knowledge-search):** 13](#_Toc185410797)

[**CI/CD Setup for API (knowledge-search-api):** 15](#_Toc185410798)

[**Harbor Registry:** 17](#_Toc185410799)

[Login to Harbor Registry 17](#_Toc185410800)

[Project 17](#_Toc185410801)

[Repositories 17](#_Toc185410802)

[Helm Chart 18](#_Toc185410803)

# **Project Name:**

Knowledge Search

# **Project Overview:**

This project involves aggregating data from various sources, ingesting it into a searchable database (OpenSearch), and exposing it to different users through an API.

# **Environments:**

There are three environments

* Dev
* ITG
* Pro

# **AWS Account Details:**

* + **Dev Environment Account Number:** 851725399072
  + **Stage Environment Account Number:** 851725399072
  + **Production Environment Account Number:** 590183818485
  + Coordinate with the Public Cloud team for the creation of VPC, Subnets, and Networking by submitting the Network Form.·

# **AWS Access for Users:**

* + Required access includes administrative rights to create resources such as EC2, S3, IAM roles, EKS, Opensearch, Secrets Manager, Sagemaker and RDS.
  + Initial bootstrapping should be performed on the AWS account if they are new accounts

# **BootStrapping:**

Bootstrapping Setup Is required when new AWS account to be configured, If not please ignore boot strapping process.

## **AWS boot strapping Process:**

* **Boot strapping repo for knowledge search project Git repo -** [**https://github.azc.ext.hp.com/Knowledge-Search/aws-account-bootstrap-new**](https://github.azc.ext.hp.com/Knowledge-Search/aws-account-bootstrap-new)

**Bootstrap your AWS account to work with CodeWay and RunWay.**

**Below are the requirements to follow the steps**

* **Supported Operating System:**[**Linux**](https://www.linux.org/)**& Supported Operating System:**[**Windows**](https://www.microsoft.com/en-us/windows)
* **Install**[**Terraform**](https://www.terraform.io/downloads.html)**(version v0.13.6+).**
* **Install**[Terragrunt](https://terragrunt.gruntwork.io/docs/getting-started/install/)**(version v0.26.7+).**
* **Install**[**Python**](https://www.python.org/downloads/)**(version 3.6+).**
* **Install**[Authly-UI](https://authly-ui.mssi.corp.hpicloud.net/)**.**
* **Clone repo locally**
* **git clone https://github.azc.ext.hp.com/runway/aws-account-bootstrap.git**
* **Create a Personal Access Token follow the steps below to create it.**
  + **Go to**[Codeway](https://dev.azure.com/hpcodeway)**and click on your user (top-right corner).**
  + **Click on the 3 dots button and next click on "User settings".**
  + **Click on "Personal Access Token"**
  + **On the "Personal Access Token" window, click on "New Token"**
  + **On the "Create new personal access token" window, click on "Show all scopes"**
  + **Find the "Variable Groups" scope and select the option "Read, create, & manage"**
  + **Assign a name to your token, define the expiration time, click on "Create"**
  + **Note: Once you complete this process, make sure you export your token into ADO\_TOKEN environment variable.**
* **Assume the Security role**
* **Using Authly-UI, Assume the SECURITY role (i.e. the role with enough privileges to create IAM entities) in your AWS Account.**
* **Create an execution plan**
* **From the root-level folder, run below command to create an execution plan**

**terragrunt plan --terragrunt-source-update --terragrunt-non-interactive --var 'codeway\_project=<CODEWAY\_PROJECT\_NAME>'**

* **Apply the execution plan by running the below command from root-level folder**

**terragrunt apply --var 'codeway\_project=<CODEWAY\_PROJECT\_NAME>'**

# **Infrastructure Provisioning:**

EKS infrastructure Repo can be found below

* **Infrastructure Git Repository:** <https://github.azc.ext.hp.com/Knowledge-Search/eks-ks>
  + Change the branches as per the environment (Dev, Stage, Prod).

**Below are the resources that are defined in the repo**

* Bastion Host
* Tags
* EKS Cluster
* RDS
* OpenSearch
* S3
* Istio Addon
* KMS
* Secrets Manager

Below is list of resources which are created in Infrastructure

|  |  |  |  |
| --- | --- | --- | --- |
| Resoucres Name | Dev env | ITG Env | Prod Env |
| Bastion Host | ks-dev-dev-ks | ks-stage-bastion-ks-stage | ks-prod-bastion-ks-prod |
| Eks Cluster | ks\_dev\_cluster | ks-stage-cluster | ks-prod-cluster |
| RDS | ks-db-dev | ks-stage-db | ks-prod-db |
| OpenSearch | ks-dev-es | ks-stage-es | ks-prod-es |
| S3 | knowledgesearch-comman-infra | - | knowledgesearch-pro |
| S3 | ks-opensearch-manual-snapshot | [ks-opensearch-manual-snapshot-stage](https://us-west-2.console.aws.amazon.com/s3/buckets/ks-opensearch-manual-snapshot-stage?region=us-west-2) | ks-opensearch-manual-snapshot-production |
| S3 | api-batchjob-logs | - |  |
| Network load balancer(bastion) | ks-dev-dev-ks-lb | ks-stage-bastion-ks-stage-lb | [ks-prod-bastion-ks-prod-lb](https://us-west-2.console.aws.amazon.com/ec2/home?region=us-west-2#LoadBalancer:loadBalancerArn=arn:aws:elasticloadbalancing:us-west-2:590183818485:loadbalancer/net/ks-prod-bastion-ks-prod-lb/a5818f836ad2ef4e;tab=listeners) |
| Application load balancer (Istio) | k8s-istiosys-istioing-90a5ed5e76 | k8s-istiosys-istioing-e383ccd60d | [k8s-istiosys-istioing-c93edf27f0](https://us-west-2.console.aws.amazon.com/ec2/home?region=us-west-2#LoadBalancer:loadBalancerArn=arn:aws:elasticloadbalancing:us-west-2:590183818485:loadbalancer/app/k8s-istiosys-istioing-c93edf27f0/9376285fbab97196;tab=listeners) |
| Application load balancer (Ingress) | KnowledgeSearchTest | - | - |

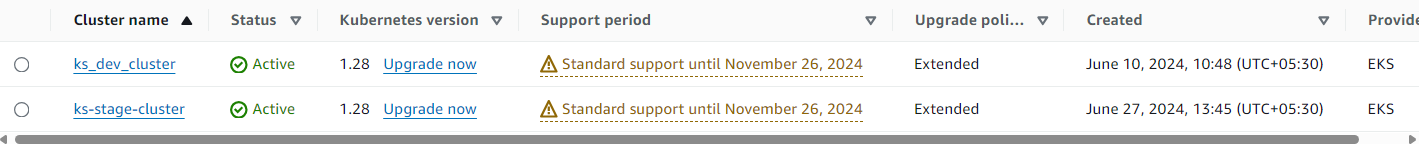
## **AWS Infrastructure:**

* **Infrastructure Git Repository:**

<https://github.azc.ext.hp.com/Knowledge-Search/eks-ks>

**EKS cluster name**

* [**ks\_dev\_cluster**](https://us-west-2.console.aws.amazon.com/eks/home?region=us-west-2#/clusters/ks_dev_cluster) **for Dev environment**
* [**ks-stage-cluster**](https://us-west-2.console.aws.amazon.com/eks/home?region=us-west-2#/clusters/ks-stage-cluster) **for stage environment**
* **ks-prod-cluster for production environment**

****

**A screenshot of a computer

Description automatically generated**

* **Resources Utilized:** EC2, EKS, S3, Secrets Manager, RDS, Certificate Manager, IAM roles, OpenSearch, WAF

## **SSL Certificates:**

**We need SSL certificates for the URLs hosted.**

**URLs hosted are as below**

* **Dev:**  [**https://kcs-dev.corp.hpicloud.net**](https://kcs-dev.corp.hpicloud.net)
* **ITG:** **https://kcs-itg.corp.hpicloud.net**
* **Prod:** **https://kcs-prod.corp.hpicloud.net**

**Certificate creation:**

* **Connect with HP counter parts for the creation of Certificate’s**
* **Provide the URL as per environment**
* **HP counter part will provide a bundle of certificates**
* **Upload the certificates in AWS certificate manager and map against the respective URL’s**

## **OpenSearch configuration:**

**Once opensearch cluster is created, two additional configurations to be followed that is**

* Fine-grained access control
* SAML integration

Steps to follow Fine-grained access control

* Go to AWS account and search for Amazon OpenSearch Service
* select on created open search cluster such as ks-dev-es , ks-stage-es or ks-prod-es
* Click on Security configuration and edit it
* Select Add Fine-grained access control
* Create master user
* Provide the Master username such as ks-stg-admin, ks-prod-admin
* Provide Master password and review the Access policy
* Save the changes
* User and password will be created and handover it to open search team to perform other required activities

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

A screenshot of a computer

Description automatically generated

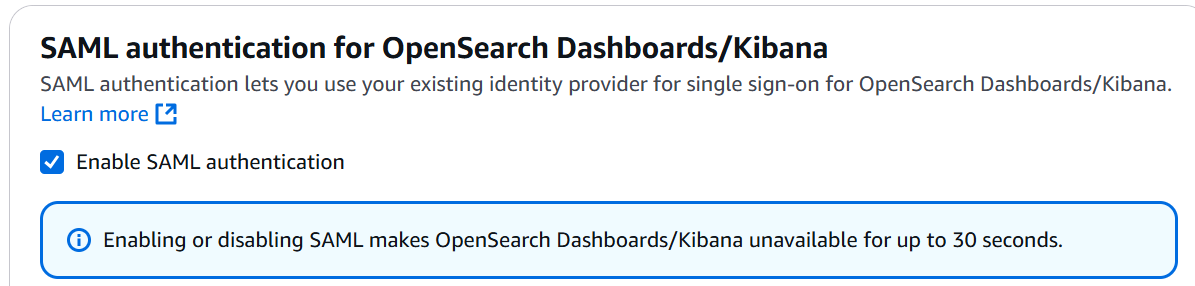
**Steps to follow saml integration**

We need to configure it for SAML. This is to enable SSO login into opensearch dashboards.

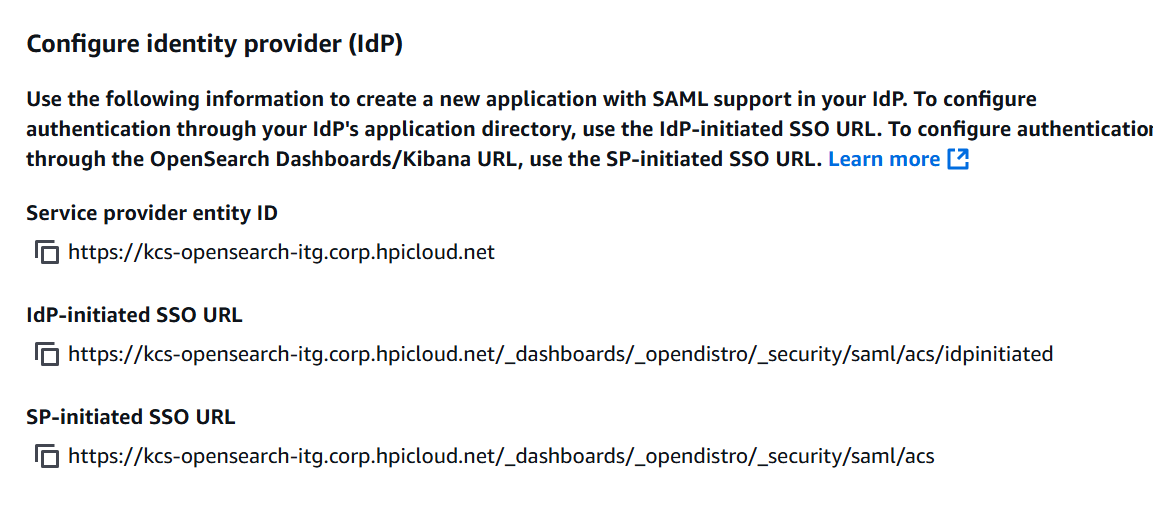
There are three steps to enable SAML

1. Create Custom endpoints
2. Configuring metadata
3. Update DNS configuration
4. Creating custom endpoints
5. Following custom endpoints to be created.
   1. **Dev: kcs-opensearch-dev.corp.hpicloud.net**
   2. **ITG: kcs-opensearch-itg.corp.hpicloud.net**
   3. **Prod: kcs-opensearch-prod.corp.hpicloud.net**
6. For each of the endpoints get the certificates from HP Team.
7. Add the certificates in AWS certificate Manager (ACM)
8. Go to edit Cluster
9. Select custom endpoints as yes
10. Enter the custom endpoints and select the appropriate certificates
11. Save. This will take approximately 10-15 mins to update.

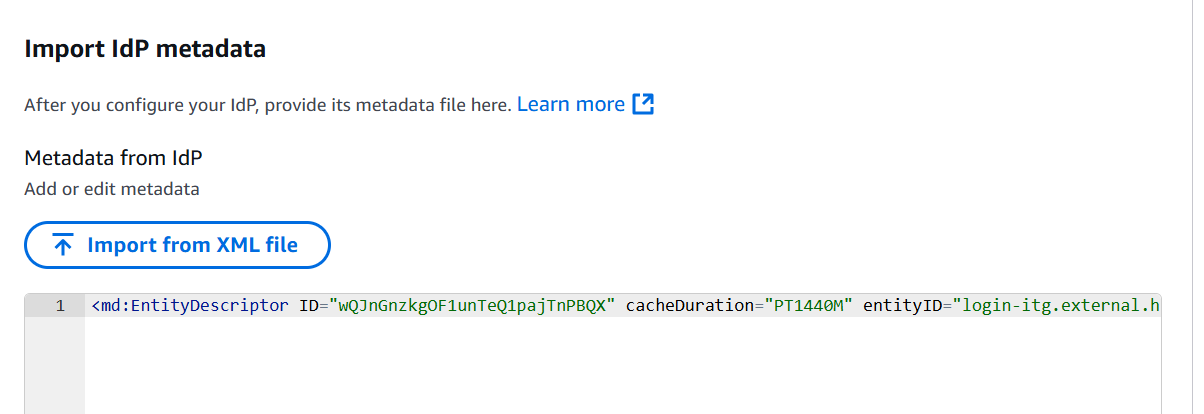
1. Configuring metadata
2. Go to cluster -> security configuration
3. Edit enable SAML configuration as yes



1. Get the URLs for saml and provide the same to HP team and ask for metadata xml file.



1. Once metadata xml file is available update the xml file here



1. Update DNS configuration
2. Update DNS to point the OpenSearch vpc endpoint to custom endpoint

**Refer the attachment for open search configuration on machine learning model deployments (deployment)**



**Refer to attachment for OpenSearch Automation (deployment)**



**Refer to attachment for OpenSearch-manual-snapshot (Configuration)**

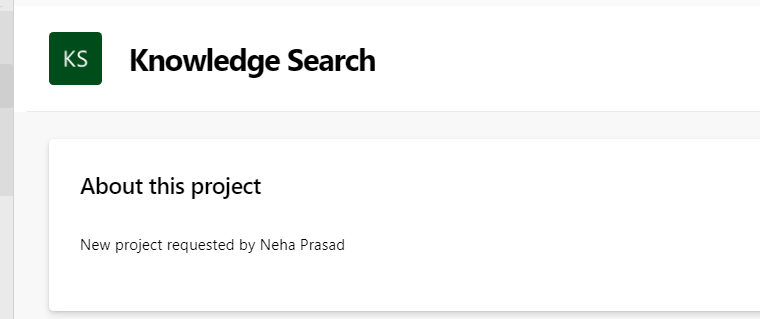


# **Azure Pipeline for Infrastructure Setup:**

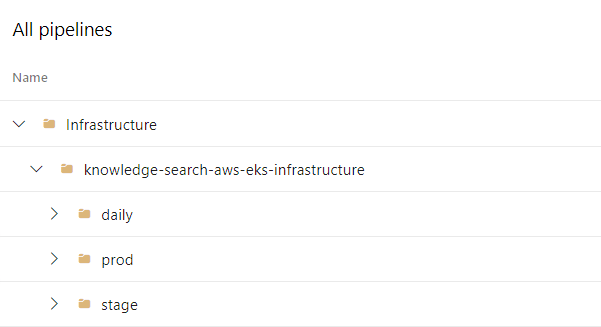
**Git Repo : https://github.azc.ext.hp.com/Knowledge-Search/ks-opensearch-dsl.git**

## **Azure Devops or CI/CD Setup for AWS infra:**

* **Via Azure Pipeline:** <https://dev.azure.com/hpcodeway/Knowledge%20Search/>



* **Project Name:** Knowledge Search
* **Pipeline Structure:**
  + Infrastructure – knowledge-search-aws-eks-infrastructure- Daily/Stage/Prod

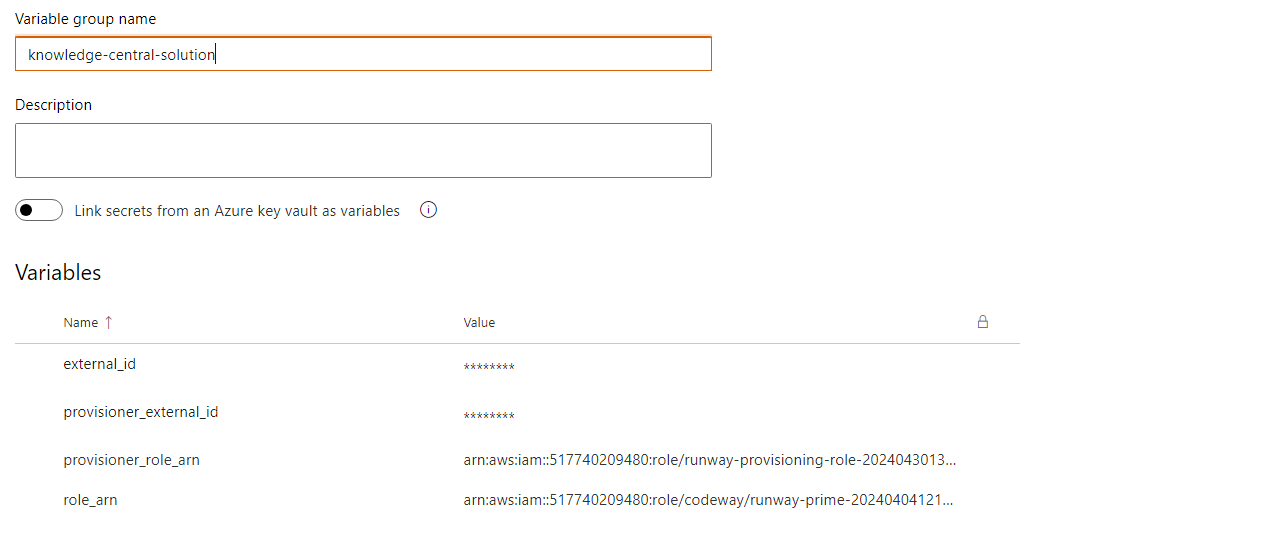


A screenshot of a computer

Description automatically generated

For Infrastructure Knowledge-Search.eks-ks is the pipeline information for creation of resources

* + Updated variables under the library as required (ex- knowledge-central-solution).



* + Add the IAM roles information from AWS accounts and resources will get created in AWS account via pipeline such as bastion host, tags, eks cluster, rds,opensearch,s3,istio,secrets manager, kms

## **Login to Bastion Host Server:**



A screenshot of a computer

Description automatically generated

* **Dev Server Name:** ks-dev-dev-ks
* **Stage Server Name:** ks-stage-bastion-ks-stage
* **Prod Server Name:** ks-prod-bastion-ks-prod
* After selecting the ks-dev-ks-dev sever click on connect option available on page and proceed with connect and switch the user using sudo su – ec2-user and authly and then give the command such as kubectl get namespaces knowledgesearch for in details step follow the below link.
* Configure the bastion host server and login to ec2 servers <https://rndwiki.inc.hpicorp.net/confluence/pages/viewpage.action?spaceKey=EngineeringSystems&title=aws_bastion>

# **Application Deployment**

**We have two repos for application deployment ie**

* **UI [nextjs-knowledge-search]**
* **Backend Application(API, Ingress batch processes) [knowledge-search-api]**

## **Git Repository:**

* <https://github.azc.ext.hp.com/knowledge-search>  
  There are two services deployed:
  + nextjs-knowledge-search
  + knowledge-search-api

## **Branching Strategy for Deployment:**

* release-to-daily
* release-to-stage
* release-to-prod

Strategy to be followed while deploying application for UI and API

Local Development → Raise PR to Main Branch → Release to Daily → Testing on Dev Environment → Raise PR from Daily to Stage → Release to Stage → Testing on Stage Environment → Raise PR from Stage to Prod → Release to Prod

Deployment of nextjs-knowledge-search refer to git repositories Knowledge-Search/nextjs-knowledge-search at branch for dev env is release\_to\_daily , branch for stage is release-to-stage , branch for prod is release-to-prod under folder Devops/Dev , Devops/stage , Devops/prod

the below CI and CD file for each stages according to each folders

### **Git CI info:**

* + codeway\_release\_to-daily\_ci.yaml defines stages such as docker,helm,run test case, sonarqube
  + codeway\_release\_to-stage\_ci.yaml defines stages such as docker,helm,run test case, sonarqube
  + codeway\_release\_to-prod\_ci.yaml defines stages such as docker,helm,run test case, sonarqube

### **Git CD info:**

* + codeway\_release\_to-daily\_cd.yaml defines stages such as helm deploy on specific environments.
  + codeway\_release\_to-stage\_cd.yaml defines stages such as helm deploy on specific environments.
  + codeway\_release\_to-prod\_cd.yaml defines stages such as helm deploy on specific environments.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

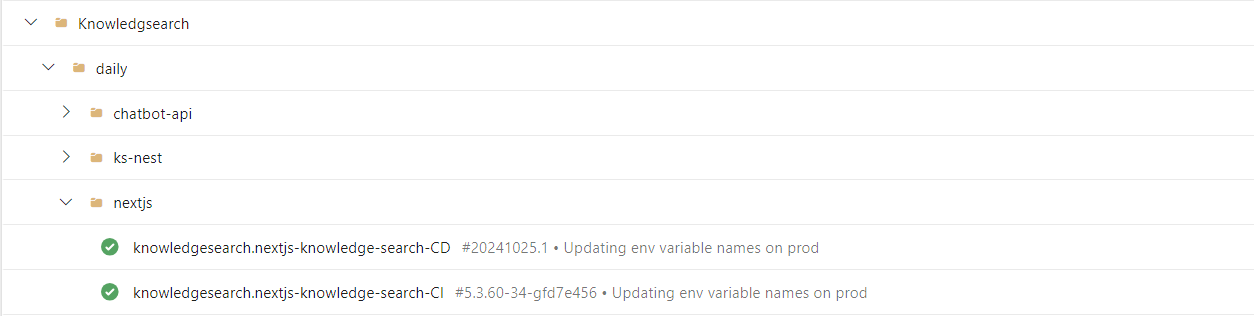
# **CI/CD Setup for UI (nextjs-knowledge-search):**

* **Via Azure Pipeline:** <https://dev.azure.com/hpcodeway/Knowledge%20Search/>
* **Project Name:** Knowledge Search
* **Pipeline Structure:**
  + Knowledge Search – Daily/Stage/Prod – (Deployment Service) – nextjs
  + Updated variables under the library such as (ex- Terragrunt, Terragrunt-Prod).

A screenshot of a computer

Description automatically generated

* + CI and CD files are defined in each service: codeway\_release\_to-daily\_ci.yaml and codeway\_release\_to-daily\_cd.yaml.
  + codeway\_release\_to-daily\_ci.yaml defines stages such as
* Git checkout
* Update git config user
* Versioning
* Docker login
* Docker build
* Trivy scan image
* Install trivy cli
* Check installation
* Trivy scan
* Copy files
* Docker push to the Harbor repository
* Helm charts stored in the Harbor repository,
* SonarQube scanning
* Extracting coverage reports.
  + codeway\_release\_to-daily\_cd.yaml defines stages such as deploying Helm and kubectl in the EKS cluster.
  + Review the CI/CD setup and ensure pipelines are successful.



# **CI/CD Setup for API (knowledge-search-api):**

Deployment of knowledge-search-api refer to git repositories Knowledge-Search/knowledge-search-api at release\_to\_daily, release-to-stage , release-to-prod under folder Devops/Dev you will find the below CI and CD file for each stages according to each folders

CI setup info

* + codeway\_release\_to-daily\_ci.yaml defines stages such as docker,helm,run test case, sonarqube
  + codeway\_release\_to-stage\_ci.yaml defines stages such as docker,helm,run test case, sonarqube
  + codeway\_release\_to-prod\_ci.yaml defines stages such as docker,helm,run test case, sonarqube

CD setup info

* + codeway\_release\_to-daily\_cd.yaml defines stages such as helm deploy on specific environments.
  + codeway\_release\_to-stage\_cd.yaml defines stages such as helm deploy on specific environments.
  + codeway\_release\_to-prod\_cd.yaml defines stages such as helm deploy on specific environments.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* **Via Azure Pipeline:** <https://dev.azure.com/hpcodeway/Knowledge%20Search/>
* **Project Name:** Knowledge Search
* **Pipeline Structure:**
  + Knowledge Search – Daily/Stage/Prod – (Deployment Service) – knowledge-search-api
  + Updated variables under the library such as (ex- nextjs-knowledgesearch, nextjs-prod-knowledgesearch).

A screenshot of a computer

Description automatically generated

* + CI and CD files are defined in each service: codeway\_release\_to-daily\_ci.yaml and codeway\_release\_to-daily\_cd.yaml.
  + codeway\_release\_to-daily\_ci.yaml defines stages such as

o    Git checkout

o    Update git config user

o    Versioning

o    Docker login

o    Docker build

o    Trivy scan image

o    Install trivy cli

o    Check installation

o    Trivy scan

o    Copy files

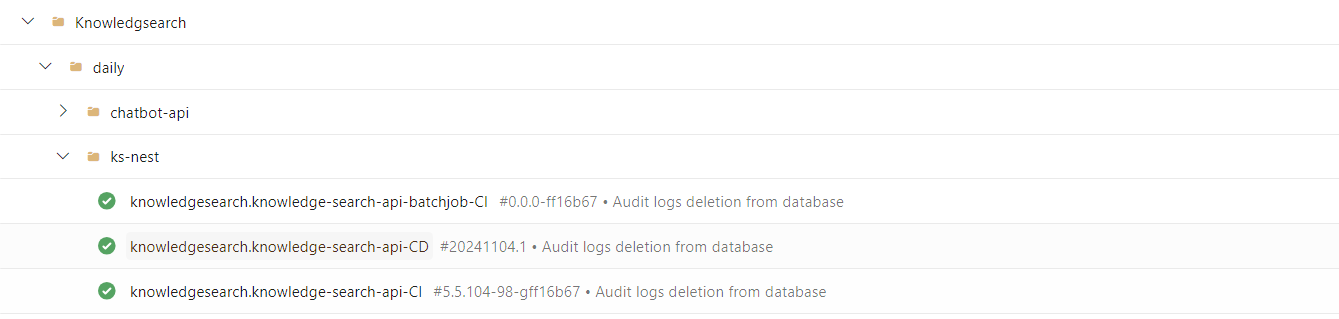
o    Docker push to the Harbor repository

o    Helm charts stored in the Harbor repository,

o    SonarQube scanning

o    Extracting coverage reports.

* + codeway\_release\_to-daily\_cd.yaml defines stages such as deploying Helm and kubectl in the EKS cluster.
  + Review the CI/CD setup and ensure pipelines are successful.



# **Harbor Registry:**

### **Login to Harbor Registry**

Access Link - <https://harbor.ext.hp.com>

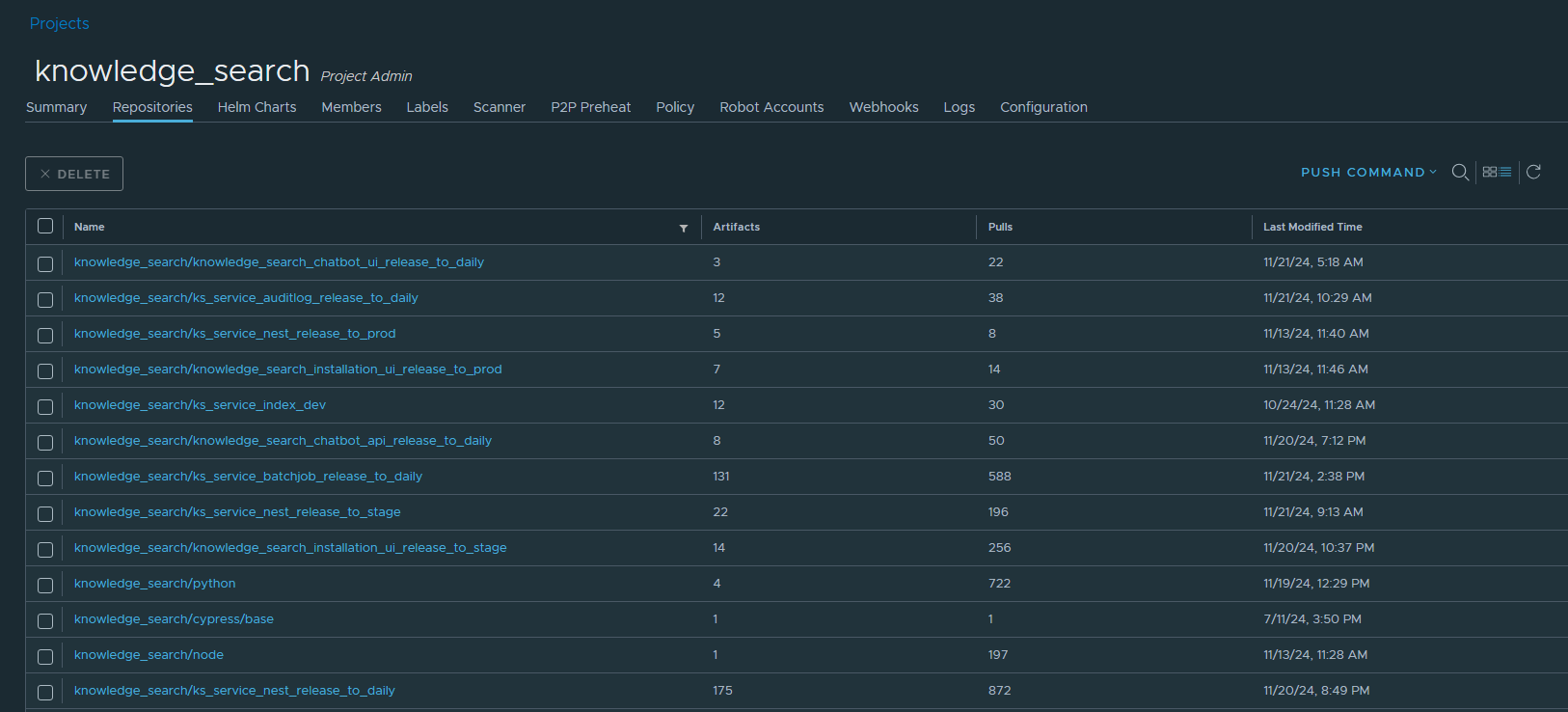
### **Project**

Select Project - knowledge\_search



### **Repositories**

Select Repositors - All the images will be available in Repositories selection



### **Helm Chart**

Select Helm Chart - All the Helm charts will be available in Helm charts option

