



User Guide

OmniDeq™ Integration with AWS Migration Hub Refactor Spaces

November 2022

Table of Contents

About CloudHedge	3
Disclaimer	3
Introduction	3
Audience	3
Overview	4
AWS Migration Hub Refactor Spaces Service	4
Joint Solution	4
Prerequisites	5
Access to CloudHedge OmniDeq™	5
AWS Account Pre-requisites	5
Scenario	6
Steps to setup required AWS services	6
Deploying EKS Cluster	6
Fetching kubeconfig file	6
Steps to setup OmniDeq with your Credentials	7
Steps to Configure Application Blueprint	8
Verify the created Application Blueprint and setup Refactor Spaces Preferences	9
Application Deployment	12
Deploying AWS Migration Hub Refactor Spaces	15
Deleting Deployment	18
Deleting AWS Migration Hub Refactor Spaces Deployment	18
Deleting Blueprints	18
Known Issues	20

About CloudHedge

CloudHedge transforms clients' business, operating and technology models to be cloud-ready through its innovative suite of tools – Discover, Transform and Cruize. CloudHedge assists clients to:

- Envision, build and run efficient businesses in cloud,
- Modernizes monolithic applications to cloud-native by leveraging automated refactoring and containerization technology.

Disclaimer

The information contained in this document is confidential, privileged and only for the usage of the intended recipient and may not be used, published or redistributed without the prior written consent of CloudHedge Technologies.

Introduction

This document provides detailed step by step instructions of using CloudHedge OmniDeq™ with AWS Migration Hub Refactor Spaces.

Audience

The document is intended for the users of the CloudHedge OmniDeq™.

Overview

AWS Migration Hub Refactor Spaces Service

AWS Migration Hub Refactor Spaces is a refactor service that implements the stranger-fig pattern and enables its users to effectively deconstruct a monolith into various microservices as well as extending existing applications with new features built as microservices. As a managed service, Refactor Spaces provides managed strangler-fig refactor environments customers use to deconstruct monoliths into microservices or extending application with new features in microservices.

Joint Solution

OmniDeq Continuous Modernization platform integrates AWS Migration Hub Refactor Spaces in its automation driven modernization workflow. This is well integrated into OmniDeq Cruize in order to support strangler-fig pattern of incremental application modernization. As the application transitions in its natural multi-step, continuous modernization from one version to another, this integration takes the risk out of the cutover process. This helps to deliver one of the most crucial benefits of risk reduction in terms of service outages and smooth cutover to newer feature functionality.

Prerequisites

Access to CloudHedge OmniDeq™

1. In order to use the OmniDeq's lift shift solution, first we need to have access to Cloudhedge OmniDeq.
2. Open your web browser (recommended Chrome) and navigate to the <omnideq-appliance-URL> or app.cloudhedge.io to access CloudHedge OmniDeq™.
3. Login using credentials you received from CloudHedge team

AWS Account Pre-requisites

1. AWS account/s with programmatic access (Access Key, Secret Access Key)
2. EKS Cluster
3. Linux / Mac Workstation
4. aws cli
 - <https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>
5. kubectl cli
 - <https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/>

Scenario

We will be deploying a sample Kubernetes application which has 2 endpoints that clients can utilize.

Steps to setup required AWS services

Deploying EKS Cluster

- Using the below CloudFormation Script provision a new EKS Cluster
 - https://ch-aws-refactor-spaces.s3.amazonaws.com/v1/EKS_CloudFormation_v1.yaml
- Parameters
 - Stack name: Provide appropriate stack name
 - ClusterName: Keep it as default
 - VpcId: select VPC from dropdown
 - EKSClusterSubnets: Select appropriate subnet from same VPC as selected above. Select at least 2 public subnets
 - NodeGroupName: Keep it as default
 - EKSNodeGroupSubnets: Select appropriate subnet from same VPC as selected above. Select at least 2 public subnets
 - NodeInstanceType: Keep it as default
 - NodeVolumeSize: Keep it as defaults
- It usually takes around 15-20 minutes for CloudFormation script to execute successfully. Verify the cloudformation is in “**CREATE_COMPLETE**” state

Fetching kubeconfig file

- Download the script from <https://ch-aws-refactor-spaces.s3.amazonaws.com/v1/getEKSKubeconfig.sh>
- From your terminal navigate to the folder where you have downloaded getEKSKubeconfig.sh script
- Execute
 - `chmod -R 755 getEKSKubeconfig.sh`
- Configure AWS CLI
 - Execute
 - `aws configure`
 - For reference <https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-quickstart.html>
 - Make sure you are using the same AWS Account Credentials which were used to provision the EKS Cluster
- On your linux/mac workstation execute script with below parameters
 - `./getEKSKubeconfig.sh <<aws_region>> <<cluster_name>>`
 - Example
 - `./getEKSKubeconfig.sh us-east-1 AWS-RefactorSpaces`
- Post successful execution of the script last few lines should look as below

```
Updated context arn:aws:eks:us-east-1:307692620196:cluster/AWS-RefactorSpaces in /Users/anandk/.kube/config
serviceaccount "k8srefactorspaces" deleted
clusterrolebinding.rbac.authorization.k8s.io "k8srefactorspaces-binding" deleted
serviceaccount/k8srefactorspaces created
clusterrolebinding.rbac.authorization.k8s.io/k8srefactorspaces-binding created
Creating the K8S config file ./kubeconfig
Property "clusters.AWS-RefactorSpaces.server" set.
Property "clusters.AWS-RefactorSpaces.certificate-authority-data" set.
User "k8srefactorspaces" set.
Context "AWS-RefactorSpaces" created.
Switched to context "AWS-RefactorSpaces".
```

- This will also create a kubeconfig file by name **kubeconfig-AWS-RefactorSpaces-
<timestamp>.yaml**
- We will be using this file to perform BYOC [Bring Your Own Cluster] on CloudHedge OmniDeq

Steps to setup OmniDeq with your Credentials

- Login to CloudHedge OmniDeq and add AWS Credentials (With the right access privileges to create refactor spaces objects)
 - Follow the below userguide
 - <https://app.cloudhedge.io/api/ch-user-guide/#settings/vault/#vault>
 - <https://app.cloudhedge.io/api/ch-user-guide/#settings/vault/#aws>
- Once you have an EKS cluster in running state you can import it into OmniDeq using BYOC [Bring Your Own Cluster] feature
 - Follow the below userguide
 - <https://app.cloudhedge.io/api/ch-user-guide/#cruise/cluster/adding-new-clusters/#byoc-bring-your-own-cluster>

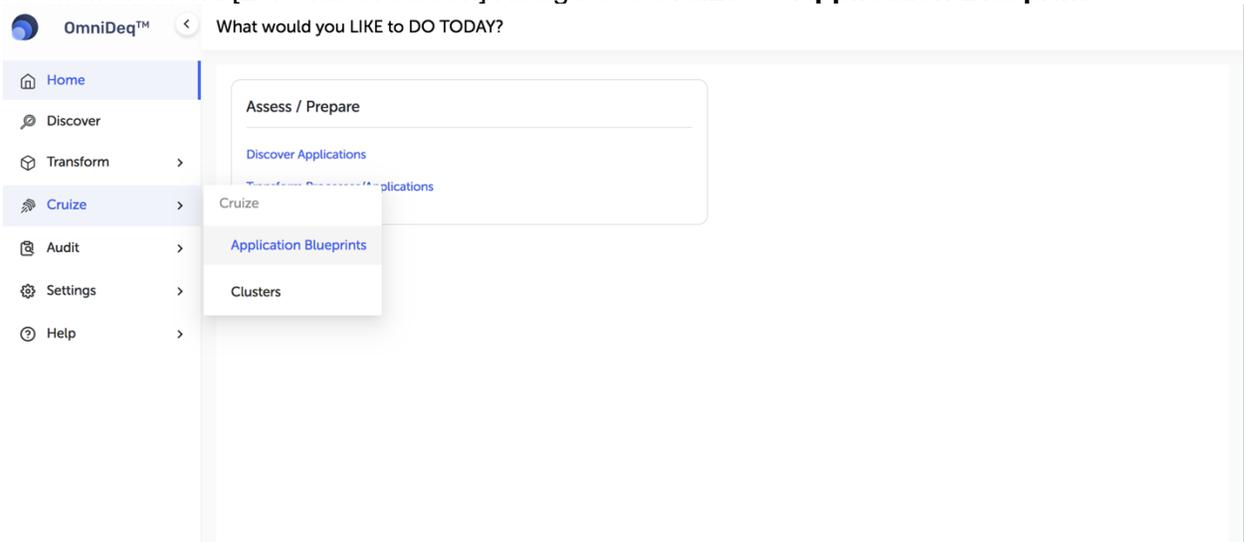
Steps to Configure Application Blueprint

- Navigate to **Cruise --> Application Blueprint**
- Click on **Create Application Blueprint**
- Enter below details
 - Name: **aws-sample-app**
 - Namespace: **aws-namespace**
 - Description: **Sample AWS Refactor Spaces Application**
 - Note : Keep rest of the values as default
 - Click on **Create Blueprint**
- Click on the **Deployments**
 - Click on **Add Deployment(s)**
- Enter **Nginx** as deployment name
- Click on **Add Deployment**
- Click on **Add Container**
- Enter below values
 - Container name : nginx
 - Image name : nginx
 - Image URL : nginx
 - Image Tag : latest
 - Image OS : Linux
 - Image pull policy : Always
 - Ports : 80
- Click on **Add**
- Click on **Service** on Top Menu
- Click on **Add new Service**
- Enter below values
 - Service name : nginx-srv
 - Service type : ClusterIp
 - Port Mapping : Click on **Add Container Port(s)**
 - Select **nginx** from pop-up
 - Click on **Add ports**
 - Click on **Save**
- **We will repeat the same procedure as above to add one more deployment containing httpd**
- Click on **Add Deployment(s)** [Top Menu Right side of screen]
- Enter **Httpd** as deployment name
- Click on **Add Deployment**
- Click on **Add Container**
- Enter below values
 - Container name : httpd
 - Image name : httpd
 - Image URL : httpd
 - Image Tag : latest
 - Image OS : Linux
 - Image pull policy : Always
 - Ports : 80
- Click on **Add**
- Click on **Service** on Top Menu

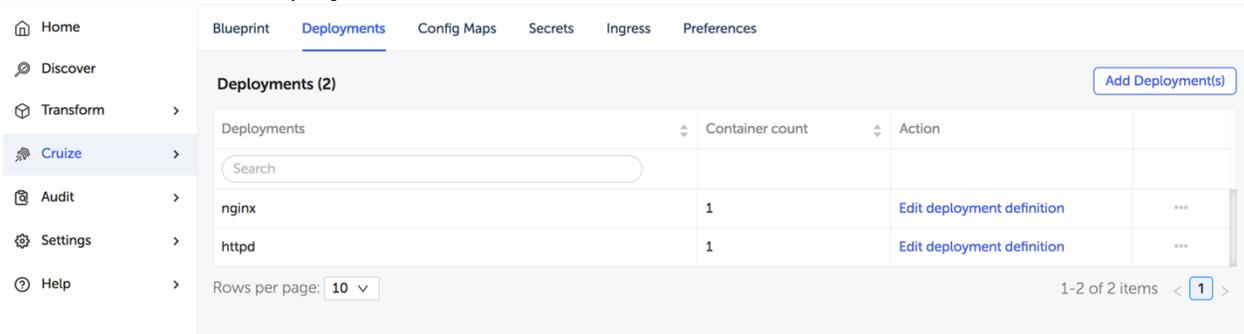
- Click on **Add new Service**
- Enter below values
 - Service name : httpd-srv
 - Service type : Clusterlp
 - Port Mapping : Click on **Add Container Port(s)**
 - Select **httpd** from pop-up
 - Click on **Add ports**
 - Click on **Save**

Verify the created Application Blueprint and setup Refactor Spaces Preferences

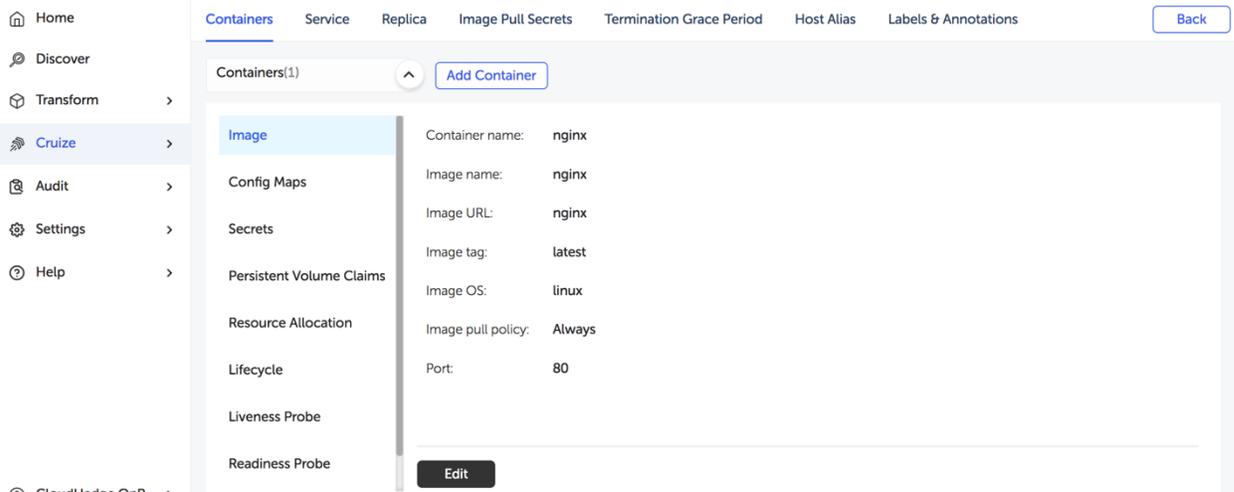
- From Main Menu [Left side of screen] Navigate to **Cruise --> Application Blueprint**



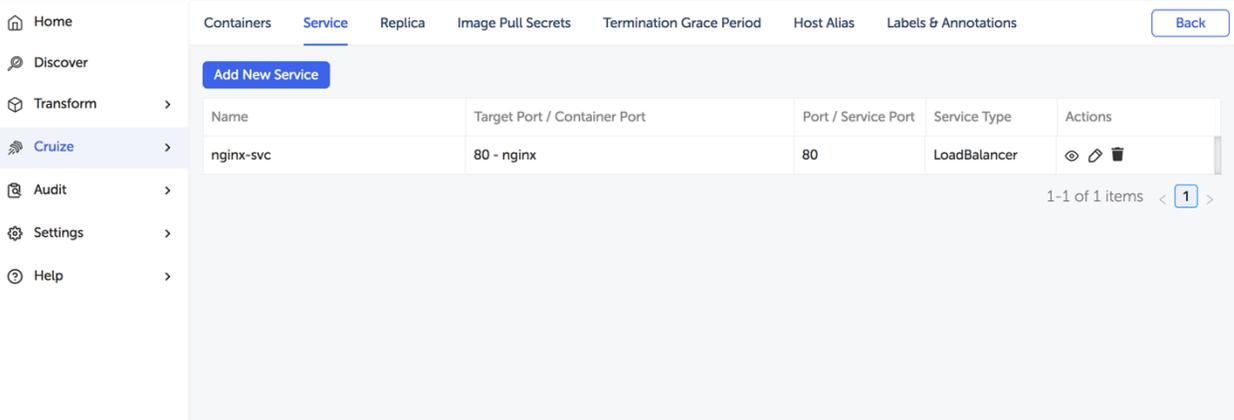
- You will notice an Application is created by name **aws-sample-app**
- Now click on the **versions** of **aws-sample-app** and click on **aws-sample-app**
- You can notice two deployments under it.



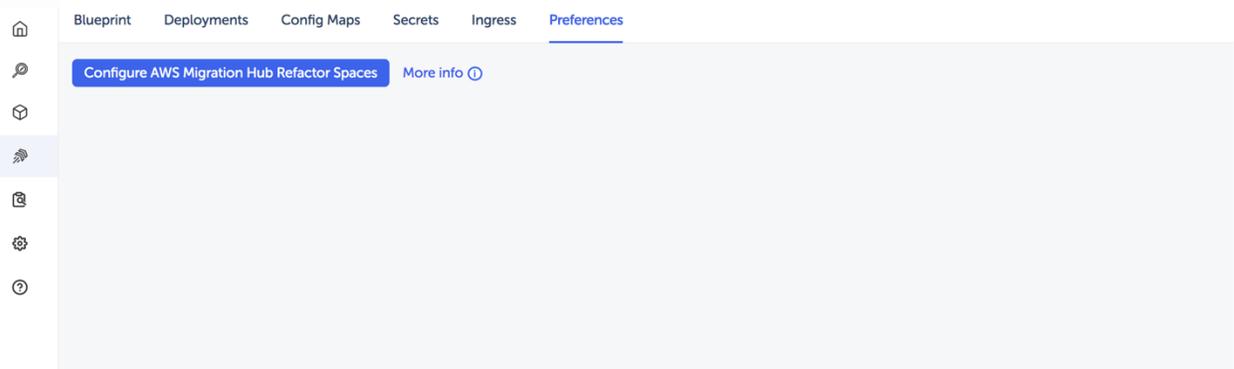
- Click on **“Edit deployment definition”** and notice the configuration of the container pointing to **“nginx”** or **“httpd”** docker image respectively



- Navigate to **“Service”** section on the same screen and notice that the service is being exposed via LoadBalancer



- This confirms that application has 2 services which will be exposed via LoadBalancer
- Let's configure "Refactor Spaces" for this application
- Navigate to **Cruize --> Application Blueprint --> aws-sample-app --> Click on the versions --> aws-sample-app**
- Click on **Preferences**



- Click on **Configure AWS Migration Hub Refactor Spaces**

- Select **AWS Vault** from dropdown
 - [Steps to setup OmniDeq with your Credentials](#)
- Select appropriate **AWS region**
- Provide appropriate **Environment name** eg: testEnv
 - Or Select “Use Existing” and click on existing environment from dropdown
- Select appropriate **Proxy VPC**
 - Note: Proxy VPC will be fetched from AWS Vault and region that you have specified
- Scroll below and click on “**Add Services**”
- On Pop-up check all the services and click on “**Add Services**”
 - Note: Services are fetched from the deployments

- Scroll below and enter the details for Services
 - Provide route: **“/”** to **nginx-svc** Verb: MATCH_ALL (This is default route)
 - Provide route: **“/httpd”** to **httpd-svc** Verb : MATCH_ALL

Service name	Source path (The path to the route)	Verb	Action
nginx-svc	/ <input checked="" type="checkbox"/> Include child paths	MATCH_ALL x	x
httpd-svc	/httpd <input checked="" type="checkbox"/> Include child paths	MATCH_ALL x	x

- Click on **Save**. AWS Migration Hub Refactor Spaces Preferences are saved successfully.

Application Deployment

- Navigate to **Cruize --> Application Blueprint --> aws-sample-app --> Click on the versions**

Name	Version	Type	Namespace	Description	Created	Action
aws-sample-app	v0	Container	refactor-spaces	sample app for refactor spaces	User Defined	Create workload

- Click on **Create Workload**
- Under **Default Cluster** select your EKS cluster
- Give appropriate value for **Workload name** (eg: **refactor-spaces**)
- Ensure **Enable AWS Migration Hub Refactor Spaces** is checked

- Click on **Validate** once its enabled
- Click on **Next**
- Verify configuration in the **Summary Screen**

- Click on **Start Workload Deployment**
- Wait for a while till kubernetes deployment succeeds.

- Click on **Refresh** icon and click on the **ellipsis** (3 dots icon) of your deployment
- Deployment of provided services usually takes around 3-5 minutes.

- **[Optional]** Click on the ellipsis and click on **View workload Summary** to check deployment status

The screenshot shows a table of workloads with the following columns: Deployed on, Version, Cluster, Namespace, K8S Deployment status, and Action. A single row is visible with the following data: 23 Aug 2022, v0, rupali-cluster, refactor-spaces, Accepted by k8s cluster. The Action column for this row has a dropdown menu open, showing options such as 'View workload Summary', 'Workload URL', 'Re-deploy workload', 'Delete workload', 'Force delete workload', 'AWS-Refactor deploy workload', 'AWS-Refactor re-deploy workload', 'Check AWS-Refactor status', and 'AWS-Refactor delete deployment'. The 'View workload Summary' option is highlighted.

- **[Optional]** Verify deployment is successful

The screenshot shows the details of a deployment. The top section displays: Application: aws-sample-app: v0, Cluster: rupali-cluster, Deployed At: 2022-08-23T12:52:27.000Z, Environment: development, Namespace: refactor-spaces, Status: success. Below this, a table lists the deployment components with 'View' links for each:

Config File	Deployment YAML	Deployed Config
httpd-dep	View	View
httpd-svc-loadbalancer-service	View	View
namespace	View	View
nginx-dep	View	View
nginx-svc-loadbalancer-service	View	View

The status message indicates: 'Application is successfully deployed.'

- Click on Ellipsis icon --> **Workload URL**

The screenshot shows the OmniDeq interface with a table of workloads. The table has columns for Deployed on, Version, Cluster, Namespace, K8S Deployment status, Refactor Spaces Status, and Action. A context menu is open over the first row, showing options like 'View workload Summary', 'Workload URL', 'Re-deploy workload', 'Delete workload', 'Force delete workload', 'AWS Refactor Spaces', 'View Refactor Spaces Details', 'Deploy Refactor Spaces', and 'Re-deploy Refactor Spaces'.

Deployed on	Version	Cluster	Namespace	K8S Deployment status	Refactor Spaces Status	Action
17 Oct 2022	v0	rupali-cluster	refactor-s	Accepted by k8s cluster	Not Applicable	Check deployment stat... ***

- If deployment is successful you will get endpoints of loadbalancer

The screenshot shows a dialog box titled 'Workload URL' with two endpoints listed:

- <http://a2a10fa4e02bf4360b23b69828ef7e32-1298855069.us-east-1.elb.amazonaws.com:80>
- <http://a3531195f784b451fb59ca14072eb108-1797901334.us-east-1.elb.amazonaws.com:80>

An 'OK' button is visible at the bottom right of the dialog.

- Click on **OK**

Deploying AWS Migration Hub Refactor Spaces

- Navigate to **Cruise --> Application Blueprint --> aws-sample-app --> Click on the Deployed Workloads**

The screenshot shows the OmniDeq interface with the 'Deployed Workloads' section. The table is the same as in the previous screenshot, but the context menu is not open. The 'Action' column for the first row shows 'Check deployment stat... ***'.

Deployed on	Version	Cluster	Namespace	K8S Deployment status	Refactor Spaces Status	Action
17 Oct 2022	v0	rupali-cluster	refactor-s	Accepted by k8s cluster	Not Applicable	Check deployment stat... ***

- Click on ellipsis icon --> Click **Deploy Refactor Spaces**

Filter by environments: Development (Workloads: 1) QA (Workloads: 0) Stage (Workloads: 0) Prod (Workloads: 0) Refresh

Deployed on	Version	Cluster	Namespac	K8S Deployment status	Refactor Spaces Status	Action
	All	All		All	All	Reset filter(s)
17 Oct 2022	v0	rupali-cluster	refactor-s	Accepted by k8s cluster	Not Applicable	Check deployment stat... <ul style="list-style-type: none"> Workloads View workload Summary Workload URL Re-deploy workload Delete workload Force delete workload AWS Refactor Spaces View Refactor Spaces Details Deploy Refactor Spaces Re-deploy Refactor Spaces

Rows per page: 10 1-1 of 1 items

- This will initiate deployment of refactor spaces
- To Check the status of deployment Click on **“View Refactor Spaces Details”** from above menu

Filter by environments: Development (Workloads: 1) QA (Workloads: 0) Stage (Workloads: 0) Prod (Workloads: 0) Refresh

Deployed on	Version	Cluster	Namespac	K8S Deployment status	Refactor Spaces Status	Action
	All	All		All	All	Reset filter(s)
17 Oct 2022	v0	rupali-cluster	refactor-s	Accepted by k8s cluster	AWS Refactor Spaces...	Check deployment stat... <ul style="list-style-type: none"> Workloads View workload Summary Workload URL Re-deploy workload Delete workload Force delete workload AWS Refactor Spaces View Refactor Spaces Details Deploy Refactor Spaces

Rows per page: 10 1-1 of 1 items

- Notice **Deployment Status** is “Creation in progress” you can click on the environment or application link and monitor the status on AWS Console

[Back](#) Refresh

Deployment Status: Creation inprogress [Last Call: appCreate]

Environment: [shared-to-qa \(env-llcu1kc7Ar\)](#)

Application: [aws-sample-app-v0 \(app-uHVkP08xhg\)](#)

Service name	Routes
nginx-svc	/
httpd-svc	/httpd

Rows per page: 10 1-2 of 2 items 1

- Alternatively you can n`avigate to your AWS account --> Region --> Refactor Space Service and notice Env getting created
 - <https://<your-region>.console.aws.amazon.com/migrationhub/refactor-spaces/>

- Make sure you are in appropriate AWS account and Region
- Deployment of Refactor spaces usually takes around **20 mins**. Wait till the deployment succeeds.
- You can also confirm the same from AWS Refactor Space Console
- Once the deployment is successful you can fetch the PROXY URL from “Workload URL”
 - Navigate to ellipsis of deployment and click on **“Workload URL”**

Filter by environments: Development (Workloads: 1) QA (Workloads: 0) Stage (Workloads: 0) Prod (Workloads: 0) Refresh

Deployed on	Version	Cluster	Namespace	K8S Deployment status	Action
23 Aug 2022	v0	rupali-cluster	refactor-spaces	Accepted by k8s cluster	Reset filter(s) Check deployment status ⋮

Rows per page: 10 | 1-1 of 1 items

- View workload Summary
- Workload URL
- Re-deploy workload
- Delete workload
- Force delete workload
- AWS-Refactor deploy workload
- AWS-Refactor re-deploy workload
- Check AWS-Refactor status
- AWS-Refactor delete deployment

View Workload URL ✕

Refactor Spaces (AWS) Proxy URL

- nginx-svc : <https://2khs1z8gn1.execute-api.us-east-1.amazonaws.com/prod/>
- httpd-svc : <https://2khs1z8gn1.execute-api.us-east-1.amazonaws.com/prod/httpd>

Workload URL

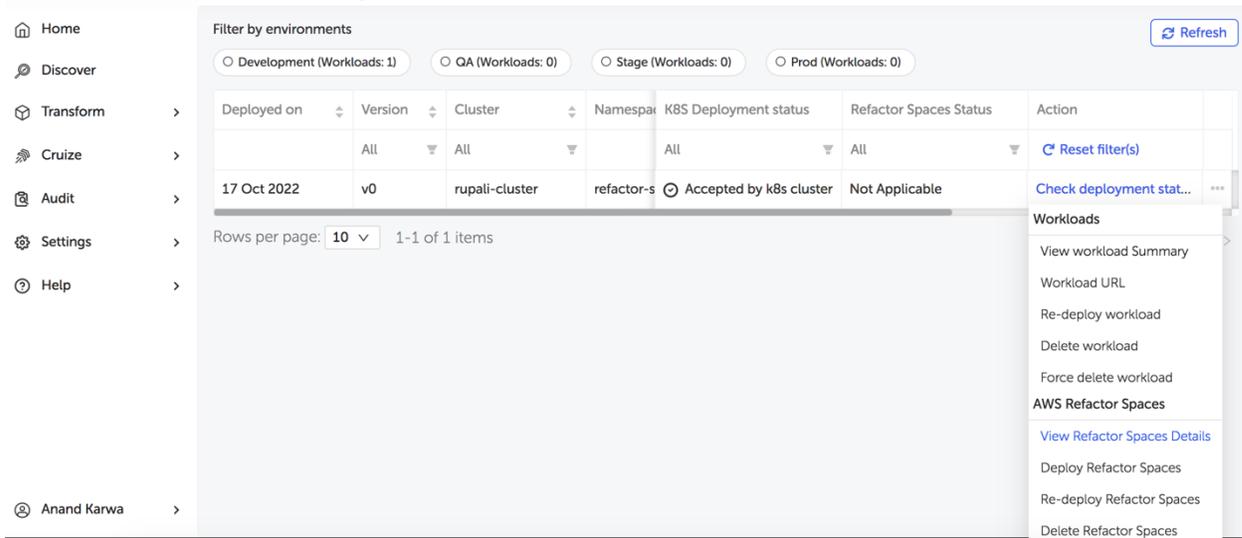
- <http://a2a10fa4e02bf4360b23b69828ef7e32-1298855069.us-east-1.elb.amazonaws.com:80>
- <http://a3531195f784b451fb59ca14072eb108-1797901334.us-east-1.elb.amazonaws.com:80>

[OK](#)

Deleting Deployment

Deleting AWS Migration Hub Refactor Spaces Deployment

- Navigate to **Cruise --> Application Blueprint --> aws-sample-app --> Click on the Deployed Workloads**
- Click on **Delete Refactor Spaces**



Filter by environments: Development (Workloads: 1) QA (Workloads: 0) Stage (Workloads: 0) Prod (Workloads: 0) [Refresh](#)

Deployed on	Version	Cluster	Namespace	K8S Deployment status	Refactor Spaces Status	Action
17 Oct 2022	v0	rupali-cluster	refactor-s	Accepted by k8s cluster	Not Applicable	Check deployment stat... <ul style="list-style-type: none"> Workloads View workload Summary Workload URL Re-deploy workload Delete workload Force delete workload AWS Refactor Spaces View Refactor Spaces Details Deploy Refactor Spaces Re-deploy Refactor Spaces Delete Refactor Spaces

Rows per page: 10 1-1 of 1 items

Anand Karwa

- Navigate to **AWS Console --> Refactor Spaces --> Env --> App**
- Notice Routes and Services getting deleted
- Once Routes and Services are deleted
 - App and Env will be deleted
- Complete deletion usually takes around **10-15 mins**

Deleting Blueprints

- Navigate to **Cruise --> Application Blueprint --> aws-sample-app --> Click on the Deployed Workloads**
- Click on ellipsis icon --> **Delete workload**

- Home
- Discover
- Transform >
- Cruize >
- Audit >
- Settings >
- Help >

Filter by environments [Refresh](#)

Development (Workloads: 1)
 QA (Workloads: 0)
 Stage (Workloads: 0)
 Prod (Workloads: 0)

Deployed on	Version	Cluster	Namespai	K8S Deployment status	Refactor Spaces Status	Action
17 Oct 2022	v0	rupali-cluster	refactor-s	Accepted by k8s cluster	Not Applicable	Check deployment stat...

Rows per page: 10 1-1 of 1 items

Workloads

- [View workload Summary](#)
- [Workload URL](#)
- [Re-deploy workload](#)
- [Delete workload](#)
- [Force delete workload](#)
- AWS Refactor Spaces**
- [View Refactor Spaces Details](#)

- Wait for 5-10 minutes your deployment should be deleted
- Navigate to CloudFormation dashboard and delete the provisioned CloudFormation Stack

Known Issues

- At times the application state is recorded as **Create_Paused**
 - This occurs because OmniDeq did not receive a success or failure response from AWS in stipulated time.
 - You have to click on **Redeploy** option to resume creation of AWS Migration Hub Refactor Spaces
- At times env delete shows below error
 - *"envDelete status is InProgress reason ConflictException: Environment: <<env_id>> in account: <<account_id>> can not be deleted because it contains proxies. Delete the proxies and try again"*
 - For this just delete the AWS-Refactor deployment again from OmniDeq Platform
- Note: For any step if delete fails just retry it again from OmniDeq Platform. Resources will get deleted gracefully