Lab 2: Deploying applications with Helm and Amazon S3

Helm is a tool that helps manage Kubernetes applications. It simplifies the deployment and management of applications on Kubernetes by using Helm charts.

Helm Chart: A collection of YAML files that describe the resources needed to run an application in Kubernetes. It's like a recipe that tells Kubernetes what to deploy and how to configure it.

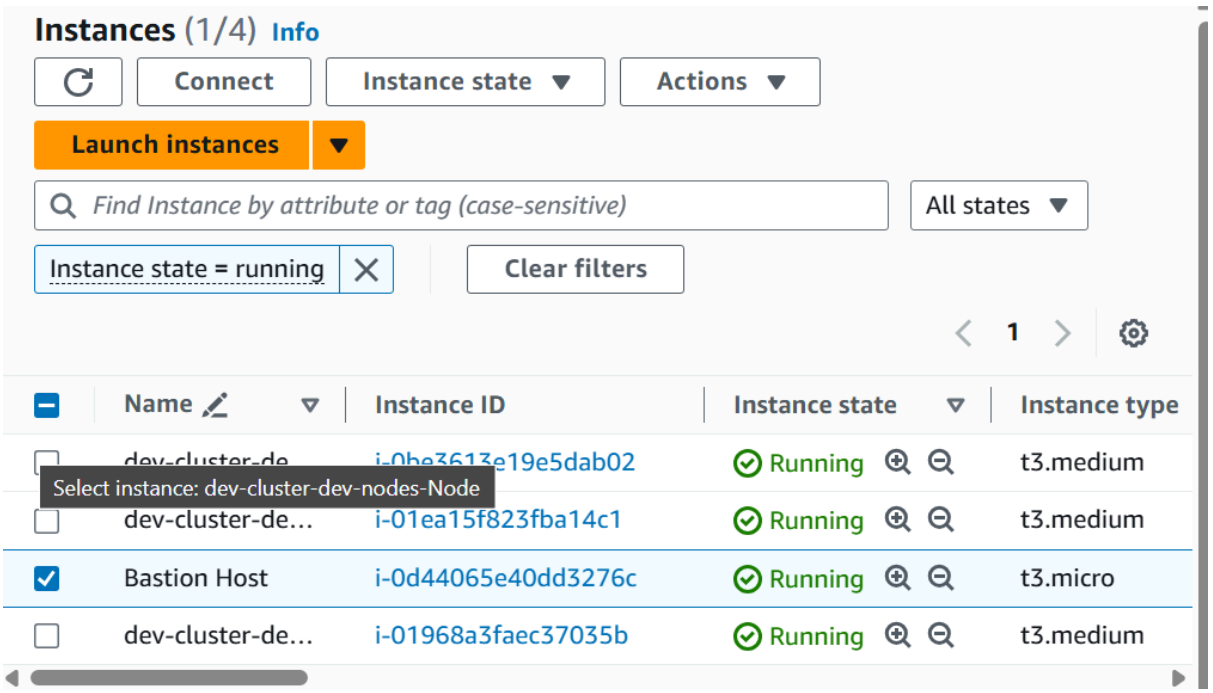
Charts: Contain templates that make the YAML files dynamic, allowing you to customize the configuration for different environments without changing the files themselves.

Objectives:

1. Define the basic structure of the Helm chart structure.
2. Set up an Amazon S3 bucket to use as a Helm repository.
3. Create and upload a chart to the S3 repository.
4. Package and install a Helm chart from an S3 repository.

Task 1: Install Helm and create an Amazon S3 bucket as a Helm repository

1.1 Open EC2, go to running instances, and connect to Bastion Host



1.2 To install Helm, at the command prompt, enter the following command:

```
sh-4.2$ curl -sSL https://raw.githubusercontent.com/helm/helm/master/scripts/get
-helm-3 | bash
Downloading https://get.helm.sh/helm-v3.15.1-linux-amd64.tar.gz
```

```
sh-4.2$ helm plugin install https://github.com/hypnoglow/helm-s3.git
Downloading and installing helm-s3 v0.16.0 ...
```

1.3 An Amazon S3 bucket for your hosted Helm repository has been pre-created by the lab and its name has been saved to the bastion host as an environment variable. To initialize the bucket as a Helm repository, enter the following command:

```
sh-4.2$ echo "Configuring $S3_BUCKET_NAME as a private Helm repository..."
Configuring workshop-854514635624 as a private Helm repository...
sh-4.2$ helm s3 init s3://$S3_BUCKET_NAME
Initialized empty repository at s3://workshop-854514635624
sh-4.2$
```

1.4 Verify the creation of index.yaml file

Note: The command creates an *index.yaml* file in the target bucket to track the chart information.

```
sh-4.2$ aws s3 ls $S3_BUCKET_NAME
2024-05-26 03:15:40         71 index.yaml
```

1.5 To add the S3 bucket as a chart repository for the Helm client, enter the following command:

```
sh-4.2$ helm repo add productcatalog s3://$S3_BUCKET_NAME
"productcatalog" has been added to your repositories
```

A Helm chart repository is a place where Helm charts are stored and from which they can be fetched and installed.

> **Task complete:** You have successfully installed *Helm*, the *Helm-S3 plugin* and initialized an *S3 bucket* as a Helm repository.

Task 2: In this task, you package a sample application into a Helm chart and push it to the Amazon S3 repository that you created.

Overview:

> You start by cloning the application code from GitHub. This contains the Kubernetes resource manifests needed to deploy the application. Next, you use the Helm CLI to package these resources into a single Helm chart archive. This bundles everything the application needs to run into a reusable format. The *helm package* command compresses the chart files into a versioned .tgz file. The chart version number comes from the *Chart.yaml* file included in the chart.

2.1 To clone the application Helm chart from the Containers on AWS Github page, enter the following command:

```
sh-4.2$ git clone https://github.com/aws-containers/eks-app-mesh-polyglot-demo.g
it
Cloning into 'eks-app-mesh-polyglot-demo'...
remote: Enumerating objects: 653, done.
remote: Counting objects: 100% (98/98), done.
remote: Compressing objects: 100% (65/65), done.
remote: Total 653 (delta 49), reused 76 (delta 33), pack-reused 555
Receiving objects: 100% (653/653), 856.63 KiB | 2.62 MiB/s, done.
Resolving deltas: 100% (344/344), done.
```

2.2 TO see the files in the helm chart directory:

```
The chart contains the following directory structure and files...

helm-chart/
├── Chart.yaml
├── productcatalog_workshop-1.0.0.tgz
├── security
│   ├── values-psa-pss-baseline-ns.yaml
│   ├── values-psa-pss-baseline.yaml
│   ├── values-psa-pss-priv.yaml
│   ├── values-psa-pss-restricted-ns.yaml
│   ├── values-psa-pss-restricted.yaml
│   └── values-psa-pss.yaml
├── templates
│   ├── NOTES.txt
│   ├── _helpers.tpl
│   ├── catalog_deployment.yaml
│   ├── catalog_service.yaml
│   ├── detail_deployment.yaml
│   ├── detail_service.yaml
│   ├── frontend_deployment.yaml
│   ├── frontend_service.yaml
│   ├── ingress.yaml
│   ├── namespace.yaml
│   └── tests
│       └── test-connection.yaml
├── values-aurora.yaml
├── values-ebs.yaml
├── values-efs.yaml
├── values-k8s-secret.yaml
├── values-secrets-manager.yaml
└── values.yaml

3 directories, 25 files
```

1. The directory name is the name of the chart

- *Chart.yaml:* A YAML file containing information about the chart
- *productcatalog_workshop-1.0.0.tgz*: A versioned chart archive that can be applied to a Kubernetes cluster
- *security/*: A directory containing manifests for pod security groups
- *templates/:* A directory of templates that, when combined with values, will generate valid Kubernetes manifest files
- *templates/NOTES.txt:* OPTIONAL: A plain text file containing short usage notes
- *values.yaml:* A series of YAML files defining default configuration values

2.3 To package the application Helm chart, enter the following command:

```
sh-4.2$ helm package helm-chart/
Successfully packaged chart and saved it to: /home/ssm-user/eks-app-mesh-polyglot-demo/
workshop/productcatalog_workshop-1.0.0.tgz
```

Our Helm Chart has been created:

```
-rw-r--r-- 1 ssm-user ssm-user 11330 May 26 03:31 productcatalog_workshop-1.0.0.tgz
```

2.4 To push the packaged application Helm chart to the S3 Helm repo, enter the following command:

```
sh-4.2$ helm s3 push ./productcatalog_workshop-1.0.0.tgz productcatalog
Successfully uploaded the chart to the repository.
```

2.5 Verify the contents of the S3 bucket (helm repo), to verify that the application Helm chart was pushed to the S3 bucket successfully:
Using this command:

```
sh-4.2$ aws s3 ls $S3_BUCKET_NAME
2024-05-26 03:32:35        446 index.yaml
2024-05-26 03:32:34      11330 productcatalog_workshop-1.0.0.tgz
```

Task complete: You have successfully packaged the application Helm chart and pushed it to S3 bucket Helm repository.

Task 3: Search and install a Helm chart from the Amazon S3 repository

3.1 Use this to check the version of your helm chart

```
sh-4.2$ helm search repo
NAME                                    CHART VERSION   APP VERSION     DESCRIPTION

productcatalog/productcatalog_workshop  1.0.0           1.0             Helm Chart for
Product Catalog Workshop
```

3.2 Use the “dry run” to test installing the chart without deploying any resources:

```
sh-4.2$ helm install productcatalog s3://$S3_BUCKET_NAME/productcatalog_workshop-1.0.0.
tgz --version 1.0.0 --dry-run --debug
```

Truncated output:

```
install.go:222: [debug] Original chart version: "1.0.0"
install.go:239: [debug] CHART PATH: /home/ssm-user/.cache/helm/repository/productcatalo
g_workshop-1.0.0.tgz

NAME: productcatalog
LAST DEPLOYED: Sun May 26 03:40:45 2024
NAMESPACE: default
STATUS: pending-install
REVISION: 1
USER-SUPPLIED VALUES:
{}

COMPUTED VALUES:
catalog:
  affinity: {}
  env:
  - name: AGG_APP_URL
    value: http://proddetail.workshop:3000/catalogDetail
  - name: AWS_XRAY_DAEMON_ADDRESS
    value: xray-service.default:2000
  - name: DATABASE_SERVICE_URL
    value: mysql.workshop
  fullnameOverride: ""
  image:
    pullPolicy: Always
```

Note from the lab:

Note: Based on the output, it looks like your --dry-run was successful.

3.3 To install the Helm chart, enter the following command:

```
sh-4.2$ helm install productcatalog s3://$S3_BUCKET_NAME/productcatalog_workshop-1.0.0.
tgz --version 1.0.0
NAME: productcatalog
LAST DEPLOYED: Sun May 26 03:44:21 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
1. Get the application URL by running these commands:
     NOTE: It may take a few minutes for the LoadBalancer to be available.
           You can watch the status of by running 'kubectl get --namespace workshop svc
 -w frontend'
  export LB_NAME=$(kubectl get svc --namespace workshop frontend -o jsonpath="{.status.
loadBalancer.ingress[*].hostname}")
  echo http://$LB_NAME:80
```

3.4 To list the resources deployed by the chart, enter the following command:

```
sh-4.2$ kubectl get pod,svc,deploy -n workshop -o name
pod/frontend-778b579c5b-5gj8g
pod/prodcatalog-7fc4b697f6-c7x58
pod/proddetail-d7847c958-xzkxq
service/frontend
service/prodcatalog
service/proddetail
deployment.apps/frontend
deployment.apps/prodcatalog
deployment.apps/proddetail
sh-4.2$
```

Helm chart has installed 3 pods, 3 services, and 3 deployments representing the frontend, productcatalog, and productdetail components of the application.

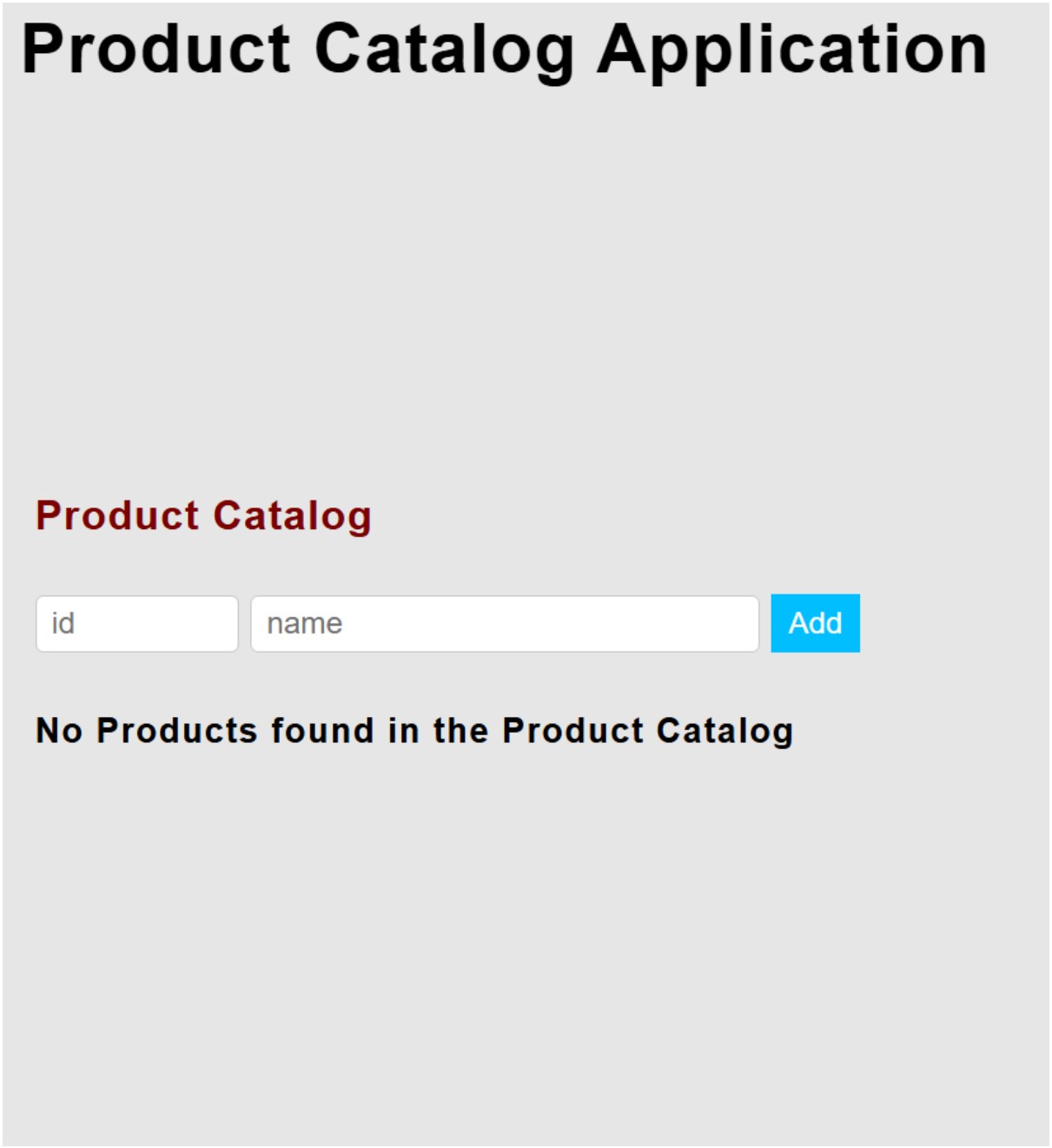
3.5 To view the frontend service:

```
sh-4.2$ FRONTEND_URL=http://$(kubectl get svc -n workshop frontend -o jsonpath='{.statu
s.loadBalancer.ingress[0].hostname}')

sh-4.2$ echo "The URL pointing to the frontend is: $FRONTEND_URL"
The URL pointing to the frontend is: http://a867080a29ddc4ffcb583ce4bd6704bc-1332788505
.ap-southeast-2.elb.amazonaws.com
```

Open the link

3.6



Task complete: You have successfully installed the application Helm chart.