

Note

The exercises in this course will have an associated charge in your AWS account. In this exercise, you create or use the following resources:

- AWS Identity and Access Management (IAM) policy and user (Policies and users are AWS account features, offered at no additional charge)
- AWS Cloud9 integrated development environment (IDE) instance
- Amazon Elastic Kubernetes Service (Amazon EKS) cluster
- Amazon Elastic Container Registry (Amazon ECR) repositories

Familiarize yourself with [AWS Cloud9 pricing](#), [Amazon EKS pricing](#), [Amazon Elastic Container Registry pricing](#), and the [AWS Free Tier](#).

Exercise 5: Using Amazon EKS

In this exercise, you install kubectl (the Kubernetes command line tool) and eksctl (a command line tool for Amazon EKS). You push the images back to Amazon ECR. Finally, you create an Amazon EKS cluster and deploy to Kubernetes.

Task 1: Installing eksctl and kubectl

In this task, you will install eksctl and kubectl.

1. In the AWS Management Console, open AWS Cloud9.
2. Open your IDE instance.
3. Install [eksctl](#):

```
curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
sudo mv /tmp/eksctl /usr/local/bin
```

4. Install [kubectl](#):

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
```

You need to push the Docker images back to Amazon ECR because they were removed in the last exercise.

5. First, set the environment variables again.

```
export ACCOUNT_ID=$(aws sts get-caller-identity --output text --query Account)
export REGION=$(wget -q -O - http://169.254.169.254/latest/meta-data/placement/region)
aws ecr get-login-password | docker login --username AWS --password-stdin ${ACCOUNT_ID}.dkr.ecr.${REGION}.amazonaws.com
```

6. Create the Amazon ECR repositories.

```
aws ecr create-repository --repository-name "corpdirectory/service"
aws ecr create-repository --repository-name "corpdirectory/frontend"
```

7. Push the images to the repositories.

```
docker push ${ACCOUNT_ID}.dkr.ecr.${REGION}.amazonaws.com/corpdirectory/service:latest
docker push ${ACCOUNT_ID}.dkr.ecr.${REGION}.amazonaws.com/corpdirectory/frontend:latest
```

Task 2: Creating an IAM policy for DynamoDB

In this task, you will create the IAM policy that's needed to access the Amazon DynamoDB table.

1. Create the IAM policy.

```
aws iam create-policy \
--policy-name eks-directory-service \
--policy-document "{ \
\"Version\": \"2012-10-17\", \
\"Statement\": [ { \
\"Effect\": \"Allow\", \
\"Action\": [ \
\"dynamodb:GetItem\", \
\"dynamodb:Query\", \
\"dynamodb:Scan\", \
\"dynamodb>DeleteItem\", \
\"dynamodb:UpdateItem\", \
\"dynamodb:PutItem\" \
] \
\"Resource\": [ \
\"arn:aws:dynamodb:${REGION}:${ACCOUNT_ID}:table/Employees\" \
] \
} ] \
}"
```

2. Copy the policy's Amazon Resource Name (ARN).

The policy ARN will look similar to this example:

```
arn:aws:iam::012345678912:policy/eks-directory-service
```

3. From the IDE navigation pane, open the `corp-eks-cluster.yaml` file.

4. Replace the `attachPolicyARNs` Fill Me In (FMI) with the policy ARN you copied, and replace the FMI for `metadata.region` with your Region.

You can find your Region by running the following command:

```
aws ec2 describe-availability-zones --output text --query 'AvailabilityZones[0].[RegionName]'
```

The following shows an example of an updated `corp-eks-cluster.yaml` file:

```
metadata:
  name: corp-eks-cluster
  region: us-east-1

iam:
  withOIDC: true
```

```
serviceAccounts:
- metadata:
  name: dynamodb-read-only
  attachPolicyARNs:
- "arn:aws:iam::012345678912:policy/eks-directory-service"
```

5. Save the file.

Next, you will update the `deployment.yaml` file for the directory service.

6. From the navigation pane, open the `directory-service/kubernetes/deployment.yaml` file.

7. Replace the FMI for `spec.spec.containers.image` with the container image for the directory service.

You can find the information for this container image by running the following command:

```
docker images --format "{{.Repository}}:{{.Tag}}" | grep corpdirectory/service
```

The following shows an example of what the `directory-service/kubernetes/deployment.yaml` file should look like after you fill in the image information:

```
spec:
  serviceAccountName: dynamodb-read-only
  containers:
- image: 012345678912.dkr.ecr.us-east-1.amazonaws.com/corpdirectory/service:latest
  imagePullPolicy: Always
  name: directory-service
  ports:
- containerPort: 80
  protocol: TCP
```

1. Save the file.

Next, you will update the `deployment.yaml` file for the frontend.

2. From the navigation pane, open the `directory-frontend/kubernetes/deployment.yaml` file and replace the FMI for `spec.spec.containers.image` with the container image for the frontend.

You can find the information for this container by running the following command:

```
docker images --format "{{.Repository}}:{{.Tag}}" | grep corpdirectory/frontend
```

The following shows an example of what the `directory-frontend/kubernetes/deployment.yaml` file should look like after you fill in the image information:

```
spec:
  containers:
- image: 012345678912.dkr.ecr.us-east-1.amazonaws.com/corpdirectory/frontend:latest
  imagePullPolicy: Always
  name: directory-frontend
  ports:
- containerPort: 80
  protocol: TCP
```

3. Save the file.

Task 3: Creating the cluster

In this task, you will create the Amazon EKS cluster by using `eksctl`.

1. Create the cluster.

```
eksctl create cluster -f corp-eks-cluster.yaml
```

NOTE: This process can take 10–20 minutes to complete.

The `config` file was created for you, and it sets up the authentication.

2. View the `config` file.

```
cat ~/.kube/config
```

You can communicate with your Amazon EKS cluster with `kubectl`.

3. Retrieve all namespaces.

```
kubectl get all --all-namespaces
```

4. Deploy your application to Kubernetes.

```
kubectl apply -f directory-service/kubernetes/deployment.yaml
kubectl apply -f directory-service/kubernetes/service.yaml
```

5. Exec into a service pod.

```
kubectl exec -i -t service/directory-service -- /bin/bash
```

6. Run some commands inside the container, then exit the container.

```
ls -l
ps -x
exit
```

7. Port-forward the `corpapp-service` cluster IP address to your AWS Cloud9 environment.

```
kubectl port-forward service/directory-service 5000:80
```

8. In your AWS Cloud9 IDE, open another terminal tab and run this command:

```
curl http://localhost:5000/employee
```

9. Close the new terminal tab you opened.
10. Break out of the port-forward by pressing Ctrl+C.

Task 4: Deploying the frontend service

In this task, you will deploy the frontend service.

1. Deploy the frontend service.

```
kubectl apply -f directory-frontend/kubernetes/deployment.yaml  
kubectl apply -f directory-frontend/kubernetes/service.yaml
```

2. Find the endpoint for the frontend service.

You can find the endpoint by running the following:

```
kubectl get service directory-frontend
```

3. Open the endpoint URL in a new tab. **NOTE:** It can take some time to set up the load balancer. You might need to wait for a few minutes before you can access the endpoint.

4. Check the logs of the frontend pod.

```
kubectl logs service/directory-frontend
```

Challenge

Inspect `directory-service/kubernetes/deployment.yaml` and `directory-frontend/kubernetes/deployment.yaml`.

For this challenge, increase the number of replicas that are hosting the services, and redeploy them.

Cleaning up

1. Delete the Kubernetes and Amazon EKS resources.

```
kubectl delete -f directory-service/kubernetes/deployment.yaml  
kubectl delete -f directory-service/kubernetes/service.yaml  
kubectl delete -f directory-frontend/kubernetes/deployment.yaml  
kubectl delete -f directory-frontend/kubernetes/service.yaml  
  
eksctl delete cluster -f corp-eks-cluster.yaml
```

© 2022 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections, feedback, or other questions? Contact us at <https://support.aws.amazon.com/#/contacts/aws-training>. All trademarks are the property of their owners.