**Objective: Facilitate deployments across multiple Kubernetes clusters.**

**Tasks:**

1. Set up a local Kubernetes cluster.

2. Integrate local cluster with EKS using EKS Anywhere.

3. Deploy applications across clusters.

**Documentation:**

- Multi-cluster management benefits.

- Introduction to EKS Anywhere.

- Deployment strategies across clusters

**Task 1:**

**Set up a local Kubernetes cluster**

For this, we can use tools like **Kind** (Kubernetes in Docker) or **minikube**. I'll choose **Kind** for simplicity.

**Install Kind:** [**link**](https://kind.sigs.k8s.io/docs/user/quick-start/#installing-from-release-binaries)

| # For AMD64 / x86\_64 [ $(uname -m) = x86\_64 ] && curl -Lo ./**kind** https://**kind**.sigs.k8s.io/dl/v0.20.0/**kind**-linux-amd64 # For ARM64 [ $(uname -m) = aarch64 ] && curl -Lo ./**kind** https://**kind**.sigs.k8s.io/dl/v0.20.0/**kind**-linux-arm64 chmod +x ./**kind** sudo mv ./**kind** /usr/local/bin/**kind** |
| --- |

| kind create cluster # Default cluster context name is `kind`. ... kind create cluster --name local-kind-cluster |
| --- |

| kind get clusters  kubectl cluster-info --context local-kind-cluster |
| --- |

**Task 2:**

**Integrate local cluster with EKS using EKS Anywhere**

**Installation Requirements:**

1. [**Eksctl**](https://eksctl.io/introduction/#installation)
2. [**Kubectl**](https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/)
3. [**eksctl-anywhere**](https://anywhere.eks.amazonaws.com/docs/getting-started/docker/)
4. [**Docker**](https://docs.docker.com/engine/install/ubuntu/)
5. **Internet access**
6. **4 CPU cores**
7. **16GB memory**
8. **30GB free disk space**

To verify cgroups version

| docker info | grep Cgroup |
| --- |

To use cgroups v1 you need to sudo and edit /etc/default/grub to set GRUB\_CMDLINE\_LINUX to “**systemd.unified\_cgroup\_hierarchy=0**” and reboot.

| sudo nano /etc/**default**/grub |
| --- |

| **sudo update-grub sudo reboot now** |
| --- |

**Then verify you are using cgroups v1.**

| **docker info | grep Cgroup** |
| --- |

**Connect your local cluster using EKS Anywhere**

To set up an EKS Anywhere cluster, you need a declarative configuration file. Using EKS Anywhere to deploy over a **Minikube** cluster isn't standard practice. EKS Anywhere typically uses VSphere, Docker, or other infrastructure providers. However, for the sake of this lab, we'll approach it conceptually.

**Prepare a configuration file:**

Let's call this eks-anywhere-cluster.yaml:

**Prepare a configuration file:**

Let's call this eks-anywhere-cluster.yaml:

| CLUSTER\_NAME=mgmt eksctl anywhere generate clusterconfig $CLUSTER\_NAME \  --provider docker > $CLUSTER\_NAME.yaml |
| --- |

| apiVersion: anywhere.eks.amazonaws.com/v1alpha1 kind: Cluster metadata:  name: any-cluster spec:  clusterNetwork:  cniConfig:  cilium: {}  pods:  cidrBlocks:  - 192.168.0.0/16  services:  cidrBlocks:  - 10.96.0.0/12  controlPlaneConfiguration:  count: 1  datacenterRef:  kind: DockerDatacenterConfig  name: any-cluster  kubernetesVersion: "1.27" # Make sure this version is supported!  managementCluster:  name: any-cluster  workerNodeGroupConfigurations:  - count: 1  name: md-0  --- apiVersion: anywhere.eks.amazonaws.com/v1alpha1 kind: DockerDatacenterConfig metadata:  name: any-cluster spec: {} |
| --- |

**Create EKS Anywhere cluster:**

| eksctl-anywhere **create** **cluster** -f eks-anywhere-**cluster**.yaml |
| --- |

| export KUBECONFIG=$(**pwd**)/dev-**cluster**/dev-**cluster**.kubeconfig |
| --- |

| export KUBECONFIG=$(kind get kubeconfig-path --name=local-kind-cluster) |
| --- |

Now, try switching to the kind-any-cluster context:

| kubectl config use-context kind-any-cluster |
| --- |

| kubectl config current-context |
| --- |

To truly connect a **kind** cluster or any local Kubernetes cluster with EKS Anywhere, you'd need to bridge the two clusters, perhaps through some sort of multi-cluster control plane or a GitOps process that ensures applications are deployed uniformly across both clusters. However, as of now, EKS Anywhere does not inherently integrate with a pre-existing kind cluster.

**Task 3:**

**Install AWS CLI Install:**

**Deploy applications across clusters**

**Deploy an application to the local cluster:**

| kubectl config use-context kind-local-cluster |
| --- |

Deploy App

| **kubectl create deployment nginx-deployment --image=nginx kubectl expose deployment nginx-deployment --port=80 --type=LoadBalancer kubectl get services nginx-deployment** |
| --- |

**Deploy to EKS Anywhere cluster:**

First, switch to your EKS Anywhere cluster context, then deploy:

| kubectl config use-context my-eks-anywhere-cluster kubectl apply -f my-application.yaml |
| --- |

| kubectl create deployment nginx --image=nginx kubectl expose deployment nginx --port=80 --type=LoadBalancer |
| --- |

| kubectl **get** pods --all-namespaces |
| --- |

| kubectl **get** svc --all-namespaces |
| --- |

**Documentation**:

Multi-cluster management benefits:

* Resiliency: Deploying across multiple clusters can enhance disaster recovery strategies.
* Scaling: Utilize resources from multiple clusters when demand surges.
* Isolation: Different clusters can be used for different purposes (dev, staging, prod).
* Geo-Distribution: Serve content closer to the user for lower latency.

**Introduction to EKS Anywhere:**

EKS Anywhere is an Amazon offering that enables you to run Kubernetes clusters on-premises, including on your local machine. It's designed to provide a consistent experience between local deployments and cloud deployments on EKS.

This task is for DevOps engineers who want to run Kubernetes clusters locally (on their own infrastructure) but also want to take advantage of Amazon EKS (Elastic Kubernetes Service) features and management tools.

EKS Anywhere is a feature of Amazon EKS that allows you to create and manage EKS clusters on your private infrastructure, like your private cloud or on-site data center, instead of on Amazon's cloud.

In this task, you'll have to follow two main steps:

**Install EKS Anywhere CLI:** With the CLI (Command Line Interface) tool, you can create, delete, and manage EKS Anywhere clusters.

**Integrate the Local Cluster with EKS Anywhere:** Here, by "local cluster", we mean the Kubernetes cluster that you've already set up locally. You need to integrate this local cluster with EKS Anywhere so you can use EKS tools and features on it too.