**Objective**: Explore persistent storage options for Kubernetes in AWS.

**Tasks:**

**1.** Deploy a stateful application on EKS.

**2.** Attach EBS volumes.

**3.** Validate data persistence.

**Documentation:**

**-** Kubernetes StatefulSets.

**-** EBS basics.

**-** Integrating EBS with EKS."

**Requirements:**

1. An AWS account with required permissions to create and manage EKS clusters, EC2 instances, and EBS volumes.
2. AWS CLI installed and configured.
3. **kubectl** installed.
4. **eksctl** installed.

**Task 1: Deploy a stateful application on EKS**

**1. Set Up the EKS Cluster:**

**1.1 kubectl** installed. [URL Link](https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/#install-kubectl-binary-with-curl-on-linux)

| sudo apt-get update sudo apt-get full-upgrade -y curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl" curl -LO "https://dl.k8s.io/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sha256" echo "$(cat kubectl.sha256) kubectl" | sha256sum --check sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl chmod +x kubectl mkdir -p ~/.local/bin mv ./kubectl ~/.local/bin/kubectl |
| --- |

**1.2 eksctl** installed. [URL Link](https://eksctl.io/introduction/#installation)

| # for ARM systems, set ARCH to: `arm64`, `armv6` or `armv7` ARCH=amd64 PLATFORM=$(uname -s)\_$ARCH  curl -sLO "https://github.com/eksctl-io/eksctl/releases/latest/download/eksctl\_$PLATFORM.tar.gz"  # (Optional) Verify checksum curl -sL "https://github.com/eksctl-io/eksctl/releases/latest/download/eksctl\_checksums.txt" | grep $PLATFORM | sha256sum --check  tar -xzf eksctl\_$PLATFORM.tar.gz -C /tmp && rm eksctl\_$PLATFORM.tar.gz  sudo mv /tmp/eksctl /usr/local/bin |
| --- |

**1.3** **AWS cli Install.** [URL Link](https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html#getting-started-install-instructions)

| sudo apt-get install zip unzip curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip" unzip awscliv2.zip sudo ./aws/install  aws configure |
| --- |

If you don't already have an EKS cluster, you can set one up using **eksctl:**

| eksctl create cluster --name=kubernetes-clusters-labs --version=1.27 --region=us-east-2 --node-type=t2.medium --nodes=2 --nodegroup-name=muzammil-kubernetes-lab --managed  or  eksctl create cluster \ --name=myekscluster \ --region=us-east-2 \ --node-type=t2.medium \ --nodes=2 \ --nodes-min=2 \ --nodes-max=3 \ --version=1.27  or  eksctl create cluster --name=myekscluster --region=us-east-2 --node-type=t2.medium --nodes=2 --nodes-min=2 --nodes-max=3 --version=1.27 |
| --- |

**1.4 Deploy a StatefulSet:**

Here's a basic example of a StatefulSet. This will deploy a basic nginx server:

| apiVersion: apps/v1 kind: StatefulSet metadata:  name: nginx spec:  serviceName: "nginx"  replicas: 2  selector:  matchLabels:  app: nginx  template:  metadata:  labels:  app: nginx  spec:  containers:  - name: nginx  image: nginx  ports:  - containerPort: 80  name: web |
| --- |

**Apply the yaml:**

| kubectl apply -f statefulset.yaml |
| --- |

***# Get commands with basic output***

| *kubectl get services*  *# List all services in the namespace  kubectl get pods --all-namespaces*  *# List all pods in all namespaces  kubectl get pods -o wide*  *# List all pods in the current namespace,*  *kubectl get deployment my-dep*  *# List a particular deployment  kubectl get pods*  *# List all pods in the namespace  kubectl get pod my-pod -o yaml*  *# Get a pod's YAML* |
| --- |

**Troubleshooting EBS:**

| 1: Install the EBS CSI Driver: [Click Link](https://github.com/kubernetes-sigs/aws-ebs-csi-driver/blob/master/docs/install.md)  kubectl apply -k "github.com/kubernetes-sigs/aws-ebs-csi-driver/deploy/kubernetes/overlays/stable/?ref=release-1.22"  2: If the EBS CSI Driver is Installed kubectl -n kube-system logs -l app=ebs-csi-controller  3: Check for the EBS CSI Driver Installation kubectl get pods --all-namespaces | grep ebs-csi  4. Check Events: kubectl get events |
| --- |

**Task 2: Attach EBS volumes**

1. **Create a StorageClass for EBS:**

Create a file named **pv.yaml (Persistent Volume)** with the following content:

| apiVersion: v1 kind: PersistentVolume metadata:  name: ebs-pv spec:  capacity:  storage: 2Gi  volumeMode: Filesystem  accessModes:  - ReadWriteOnce  storageClassName: ""  persistentVolumeReclaimPolicy: Retain  csi:  driver: ebs.csi.aws.com  fsType: ext4  volumeHandle: vol-00cb3e690e68c2152 |
| --- |

**Apply the yaml:**

| kubectl apply -f pv.yaml |
| --- |

Create a file named **pvc.yaml (Persistent Volume Claims)** with the following content:

| apiVersion: v1 kind: PersistentVolumeClaim metadata:  name: ebs-pvc spec:  accessModes:  - ReadWriteOnce  storageClassName: ""  volumeMode: Filesystem  resources:  requests:  storage: 2Gi  volumeName: ebs-pv |
| --- |

**Apply the yaml:**

| kubectl apply -f ebs-pvc.yaml |
| --- |

**Note:** Update the Deployments to use EBS-backed storage.

| apiVersion: apps/v1 kind: Deployment metadata:  name: nginx-deployment  labels:  app: nginx spec:  replicas: 1  selector:  matchLabels:  app: nginx  template:  metadata:  labels:  app: nginx  spec:  containers:  - name: nginx  image: nginx  ports:  - containerPort: 80  volumeMounts:  - name: nginx-persistent-storage  mountPath: /usr/share/nginx/html  volumes:  - name: nginx-persistent-storage  persistentVolumeClaim:  claimName: ebs-pvc |
| --- |

**Task 3: Validate data persistence**

**1. Write Data to Nginx:**

Exec into one of the pods:

Create a file inside **/usr/share/nginx/html:**

| kubectl exec -it pod-name -- /bin/bash echo "Hello from Alnafi" > /usr/share/nginx/html/index.html |
| --- |

Exit from the pod.

**Delete the pod:**

| kubectl delete pod pod-name |
| --- |

Once the pod comes back up, validate the data is still there:

| kubectl exec pod-name -- cat /usr/share/nginx/html/index.html |
| --- |

**Documentation:**

* Kubernetes StatefulSets: These are used to manage stateful applications. They provide guarantees about the ordering and uniqueness of pods.
* EBS Basics: Elastic Block Store is a block storage service provided by AWS. It can be used to provide persistent storage for applications running in EKS.
* Integrating EBS with EKS: EKS can dynamically provision EBS volumes and attach them to pods. This allows data to persist even if pods are rescheduled or deleted.

**Troubleshooting Tips:**

1. If the EBS volume is not getting attached, check your EKS node IAM roles. They should have permissions for EBS operations.
2. Ensure that your AWS CLI is correctly configured with the necessary credentials and default region.
3. If data isn't persisting, make sure your pods are utilizing the correct PersistentVolumeClaim.
4. Check pod and StatefulSet logs for any potential issues.
5. Ensure the StorageClass provisioner is set correctly to kubernetes.io/aws-ebs.

**Tips and Tricks:**

1. kubectl describe is your friend. Use it to get detailed info about resources.
2. Always ensure you're working in the right AWS region.
3. Monitor the AWS Management Console's EBS section to view and manage volumes.