Objective: Introduce students to Amazon EKS and its benefits.

Tasks:

1. Create an EKS cluster.

2. Set up kubectl for EKS.

3. Deploy a sample microservice.

**1. Create an EKS Cluster:**

Step 1: Open the AWS Management Console.

Step 2: Navigate to the Amazon EKS dashboard.

Step 3: Click on "Create cluster".

Step 4: Fill out the required details:

* Cluster name: (e.g., MyEKScluster)
* Kubernetes version: (e.g., Latest Version)
* Role: Choose an existing role or create a new role with the necessary permissions for EKS.
* VPC: Select the VPC where you want to deploy your EKS cluster.
* Subnets: Choose the subnets within the selected VPC.
* Security Group: Select or create a security group for your cluster.

Step 5: Click "Create" to create the EKS cluster.

**2. Set up kubectl for EKS:**

**Step 1:** Install kubectl: [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 |
| --- |

Install AWS cli [Clink 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 |
| --- |

| sudo apt-get update && sudo apt-get install -y kubectl |
| --- |

**Step 2:** Install aws-iam-authenticator:

| curl -Lo aws-iam-authenticator https://github.com/kubernetes-sigs/aws-iam-authenticator/releases/download/v0.5.9/aws-iam-authenticator\_0.5.9\_linux\_amd64  curl -Lo aws-iam-authenticator.txt https://github.com/kubernetes-sigs/aws-iam-authenticator/releases/download/v0.5.9/authenticator\_0.5.9\_checksums.txt  **awk '/aws-iam-authenticator\_0.5.9\_linux\_amd64/ {print $1}' aws-iam-authenticator.txt**  **openssl sha1 -sha256 aws-iam-authenticator**  **chmod +x ./aws-iam-authenticator**  **mkdir -p $HOME/bin && cp ./aws-iam-authenticator $HOME/bin/aws-iam-authenticator && export PATH=$HOME/bin:$PATH**  **echo 'export PATH=$HOME/bin:$PATH' >> ~/.bashrc**  **aws-iam-authenticator help** |
| --- |

Here this link [Click link](https://docs.aws.amazon.com/eks/latest/userguide/install-aws-iam-authenticator.html)

**Step 3:** Configure kubectl for Amazon EKS:

| aws eks --region region-code update-kubeconfig --name cluster\_name  kubectl get nodes |
| --- |

Replace region-code with your AWS region and cluster\_name with your EKS cluster name.

**3. Deploy a sample microservice:**

**Step 1:** Deploy a sample nginx application:

| kubectl create deployment nginx-deployment --image=nginx |
| --- |

**Step 2:** Expose the nginx service:

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

**Step 3:** Check the deployment and service status:

| kubectl get deployments kubectl get services |
| --- |

**Documentation:**

**1. Introduction to EKS:**

Amazon EKS (Elastic Kubernetes Service) is a managed Kubernetes service provided by AWS. EKS eliminates the need for users to install, operate, and maintain their own Kubernetes control plane. This means that EKS manages the master nodes, while users are responsible for their worker nodes.

**2. Comparison with self-managed Kubernetes:**

* Ease of setup: EKS is quicker to set up compared to self-managed Kubernetes.
* Maintenance: EKS automatically manages the master nodes, taking away the maintenance overhead.
* Scaling: EKS clusters can be easily scaled, while scaling self-managed Kubernetes requires manual effort.
* Integration: EKS is tightly integrated with other AWS services.

**3. AWS integrations with EKS:**

* Identity and Access Management (IAM): EKS integrates with IAM for authentication.
* CloudWatch: For monitoring the cluster.
* ELB: Load balancing for services within EKS.
* EBS: For persistent storage in EKS.

**Troubleshooting Tips:**

1. If kubectl commands fail, check if the kubeconfig is correctly set up.
2. If pods are stuck in "Pending" status, there may be insufficient resources or issues with the node.
3. For authentication issues, ensure that aws-iam-authenticator is correctly set up and that IAM roles are correctly configured.

**Student Performance:**

1. Let the students go through the steps one-by-one.
2. Monitor their progress and assist with issues.
3. Test their understanding by asking them to explain concepts or troubleshoot hypothetical issues.

**Requirements:**

* An active AWS account.
* An IAM user with appropriate permissions.
* An Ubuntu instance or workstation for setup.