

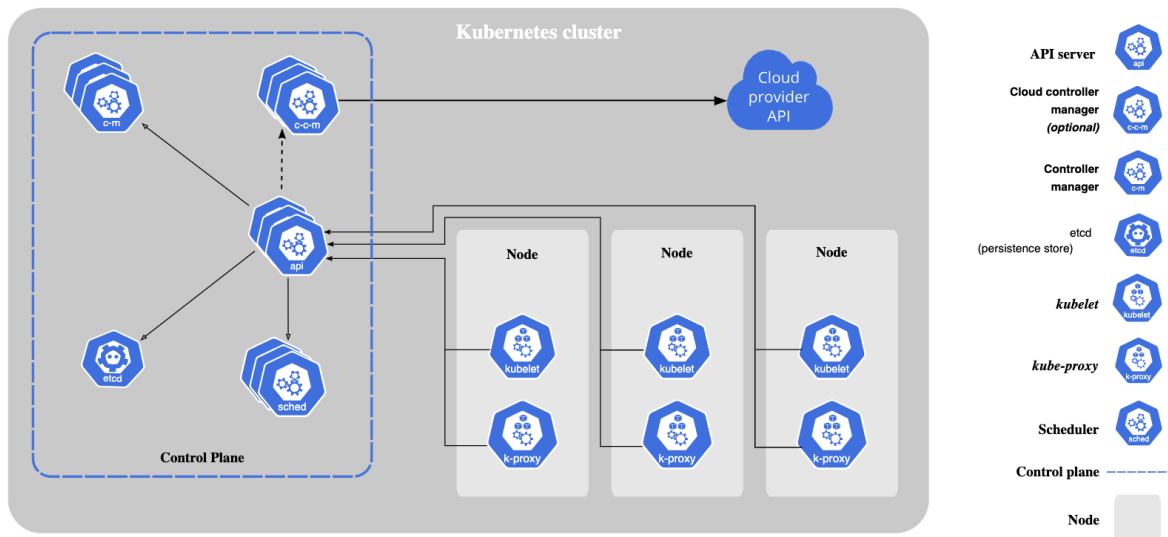
# COMP 4964 – Lab 5

## Kubernetes

### Kubernetes Components:

An overview of the key components that make up a Kubernetes cluster.

This part provides a high-level overview of the essential components that make up a Kubernetes cluster.



The components of a Kubernetes cluster

### Core Components

A Kubernetes cluster consists of a control plane and one or more worker nodes. Here's a brief overview of the main components:

#### Control Plane Components

Manage the overall state of the cluster:

##### [kube-apiserver](#)

The core component server that exposes the Kubernetes HTTP API

##### [etcd](#)

Consistent and highly available key value store for all API server data

##### [kube-scheduler](#)

Looks for Pods not yet bound to a node and assigns each Pod to a suitable node.

**[kube-controller-manager](#)**

Runs [controllers](#) to implement Kubernetes API behavior.

**[cloud-controller-manager](#) (optional)**

Integrates with underlying cloud provider(s).

## Node Components

Run on every node, maintaining running pods and providing the Kubernetes runtime environment:

**[kubelet](#)**

Ensures that Pods are running, including their containers.

**[kube-proxy](#) (optional)**

Maintains network rules on nodes to implement [Services](#).

**[Container runtime](#)**

Software responsible for running containers. Read [Container Runtimes](#) to learn more.

## Part 1- K8s Basics

Read through [concepts](#), Answer listed questions.

### 1. What is Kubernetes?

Kubernetes is a portable, extensible, open source platform for automating the deployment, scaling, and management of containerized applications.

### 2. What is a pod in Kubernetes?

A pod is the smallest deployable unit of computing that you can create and manage in Kubernetes. It consists of a group of one or more containers with shared storage and network resources.

### 3. What is the role of a Kubernetes node?

In Kubernetes, nodes are the physical/virtual machines that run the containerized applications. They're essentially a server, such as an EC2, that the nodes reside on. Each node is managed by the control plane and contains the services necessary to run Pods.

### 4. What is a namespace in Kubernetes?

While the names of resources need to be unique within a namespace, each Kubernetes cluster can have multiple namespaces. This is useful if there are multiple teams working within the same cluster. It's important to note that namespaces are isolated from each other though, so namespace restricted resources cannot directly access or interact with resources in another namespace.

5. What is a Kubernetes deployment?

A Kubernetes Deployment is a resource object that provides higher-level abstraction for managing and scaling pods. The Deployment Controller allows you to control the desired state of the pods.

6. How does a Service in Kubernetes work?

A service is a method/layer for exposing a network application that is running as one or more Pods in your cluster.

7. What is a kubelet?

The kubelet is a primary “node agent” that runs on each node and manages the lifecycle of containers within a K8 cluster. The Kubelet ensures that the containers are running in accordance with the specifications defined in the Kubernetes API

8. What is the role of etcd in Kubernetes?

Etcd is a key-value store used as Kubernetes backing storage for all cluster data.

## Part 2- Minikube Basics

Minikube is a lightweight tool that allows you to run a single-node Kubernetes cluster locally on your machine. It's commonly used for development and testing purposes, providing a simple environment to learn and experiment with Kubernetes.

Minikube supports various drivers, such as VirtualBox or Docker, to create the virtual machine or container where the cluster runs. It also includes features like the Kubernetes dashboard, and the ability to enable add-ons like Ingress controllers or metrics servers. Minikube is ideal for anyone looking to get hands-on experience with Kubernetes without needing a full-scale cluster.

Read more about Minikube [here](#), and answer listed questions.

1. What is Minikube?

Minikube is a lightweight tool for running Kubernetes locally; typically used for learning and experimenting Kubernetes.

2. What is the purpose of Minikube in Kubernetes development?

For providing a simple environment to learn and experiment with Kubernetes.

3. How do you start a Minikube cluster?

From terminal with admin access:  
“minikube start”

4. What command is used to stop a Minikube cluster?

minikube stop

5. How can you check the status of your Minikube cluster?

minikube status

6. What command is used to delete a Minikube cluster?

minikube delete --all

7. What driver options can Minikube use for running clusters?

Docker, VirtualBox, VMware, Hyperkit, Hyper-V, KVM, Parallels, SSH, None

8. How do you enable an add-on in Minikube?

minikube addons enable <addon-name>

9. How do you access the Kubernetes dashboard in Minikube?

minikube dashboard

10. What is the command to list the Minikube profiles?

minikube profile

### Part 3 – Minikube Hands on

Run this tutorial: <https://kubernetes.io/docs/tutorials/hello-minikube/>

Deliverables:

1. Screenshots of your console response to the commands:
  - View the Service you created `kubectl get services`
  - View the Pod and Service you created by installing that addon:  
`kubectl get pod,svc -n kube-system`

```

PS C:\Users\Marco> kubectl get services
NAME                                TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)
AGE
hello-minikube                     NodePort    10.103.250.165   <none>           8080:32144/TCP
173m
kubernetes                         ClusterIP   10.96.0.1        <none>           443/TCP
178m
PS C:\Users\Marco>

```

```

PS C:\Users\Marco> kubectl get pod,svc -n kube-system
NAME                                READY    STATUS    RESTARTS    AGE
pod/coredns-6f6b679f8f-s7drj       1/1     Running   0           178m
pod/etcd-minikube                   1/1     Running   0           178m
pod/kube-apiserver-minikube         1/1     Running   0           178m
pod/kube-controller-manager-minikube 1/1     Running   0           178m
pod/kube-proxy-wknvk                1/1     Running   0           178m
pod/kube-scheduler-minikube         1/1     Running   0           178m
pod/storage-provisioner             1/1     Running   2 (178m ago) 178m

NAME                                TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)
AGE
service/kube-dns                   ClusterIP   10.96.0.10      <none>           53/UDP,53/TCP,9153/TCP
178m

```

2. Screenshot on a browser window showing your app.  
[Hint: after running [minikube service hello-node](#)]

```

Command Prompt - minikube x Windows PowerShell x + v - □ x
C:\Users\Marco>kubectl get services
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
kubernetes    ClusterIP     10.96.0.1     <none>         443/TCP    3m49s

C:\Users\Marco>kubectl get pod.svc -n kube-system
error: the server doesn't have a resource type "pod"

C:\Users\Marco>kube

C:\Users\Marco>kubectl create deployment hello-minikube --image=kicbase/echo-server:1.0
deployment.apps/hello-minikube created

C:\Users\Marco>kubectl expose deployment hello-minikube --type=NodePort --port=8080
service/hello-minikube exposed

C:\Users\Marco>kubectl get services hello-minikube
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
hello-minikube NodePort      10.103.250.165 <none>         8080:32144/TCP 6s

C:\Users\Marco>kubectl port-forward service/hello-minikube 7080:8080

C:\Users\Marco>minikube service hello-minikube
|-----|-----|-----|-----|
| NAMESPACE | NAME          | TARGET PORT | URL              |
|-----|-----|-----|-----|
| default   | hello-minikube | 8080        | http://192.168.49.2:32144 |
|-----|-----|-----|-----|
* Starting tunnel for service hello-minikube.
|-----|-----|-----|-----|
| NAMESPACE | NAME          | TARGET PORT | URL              |
|-----|-----|-----|-----|
| default   | hello-minikube |             | http://127.0.0.1:55156 |
|-----|-----|-----|-----|
* Opening service default/hello-minikube in default browser...
! Because you are using a Docker driver on windows, the terminal needs to be open to run it.

```

## Part 4 – EKS Basics

Read through listed web pages to learn EKS basics:

- [What is EKS](#)
- [Kubernetes Concepts](#)
- [EKS Architecture](#)
- [EKS Deployment Options](#)
- [Common Use-cases](#)

Answer the following questions:

1. What is Amazon EKS?

AWS EKS is a managed Kubernetes service to run Kubernetes in the AWS cloud and on-premises data centers.

2. What is the main purpose of EKS?

To provide a fully managed Kubernetes service that runs on AWS infrastructure

3. What are the key components of an EKS cluster?

EKS Control Plane, ECS Data Plane, Fargate – serverless compute for containers, EKS Connector

4. How does EKS handle control plane management?

AWS EKS manages the K8s control plane for the user by handling control plane components such as K8s API server, etcd, scheduler, and controller manager.

5. What is the difference between EKS and self-managed Kubernetes?

The primary difference is who's responsible for managing the cluster infrastructure and how the control plane is handled: self-managed Kubernetes would require the user to manually control these aspects; EKS abstracts these details away and manages it for you.

6. How do you deploy a Kubernetes cluster on EKS?

Create the cluster (via CLI, SDK, or Web Console)

Example:

```
eksctl create cluster \  
  --name my-eks-cluster \  
  --region us-west-2 \  
  --nodegroup-name my-nodegroup \  
  --nodes 3 \  
  --nodes-min 1 \  
  --nodes-max 4 \  
  --managed
```

7. What networking solution does EKS use by default?

Amazon VPC Container Network Interface

8. How do you scale nodes in an EKS cluster?

Can manually adjust the node count (e.g. eksctl) or use automatic scaling methods AWS provides.

9. What is a managed node group in EKS?

A K8 cluster with its lifecycle managed by AWS EKS.

10. How does EKS handle high availability?

EKS handles high availability by running and scaling Kubernetes control plane across multiple AWS AZs. Instead of automatically scaling based on load, it detects and replaces unhealthy control plane instances, and automatically patches the control plane.

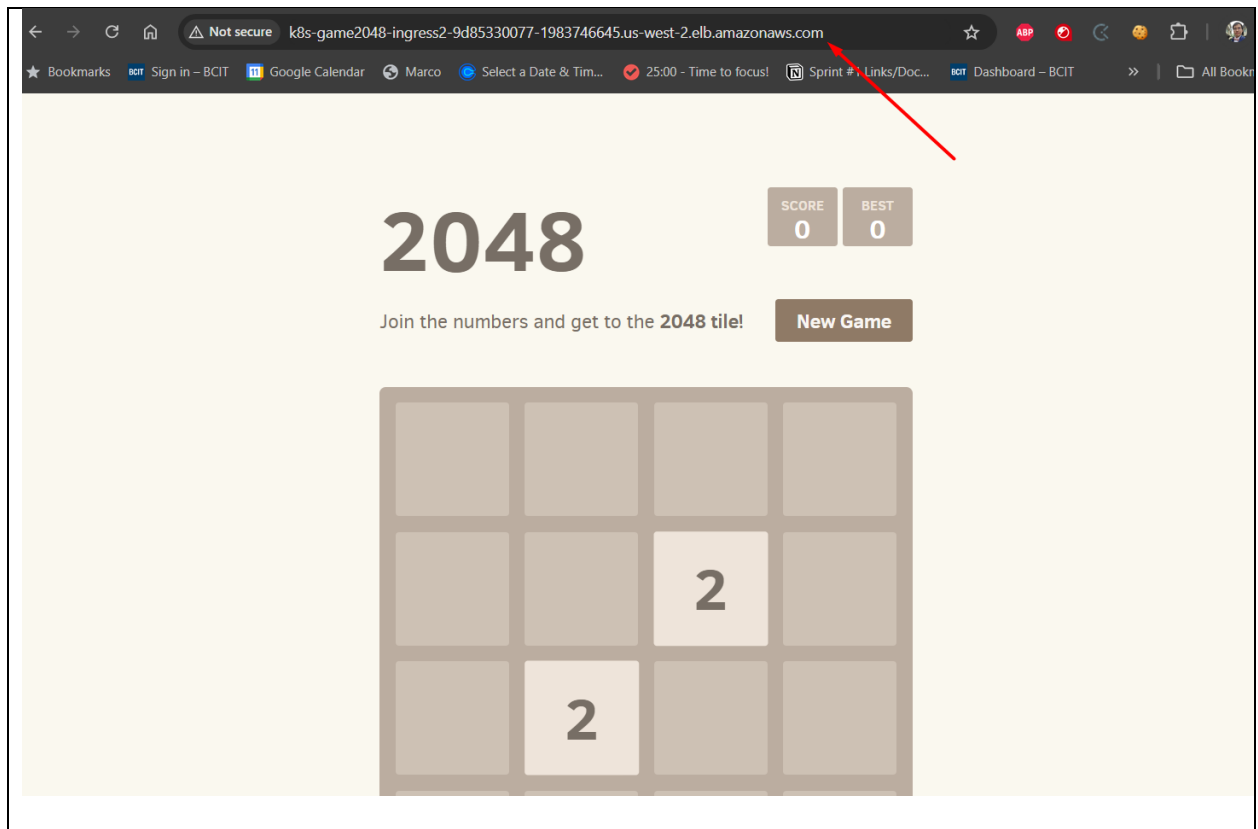
## Part 5 – EKS Hands on

Run this tutorial: <https://docs.aws.amazon.com/eks/latest/userguide/quickstart.html>

Deliverables:

1. Screenshot on your browser to show the deployment works





## 2. Other screenshots to showcase your deployment works

