Instance(Folder, Deployment creation)

# Introduction

This document provides detailed instructions for setting up an instance and creating a deployment in Kubernetes. The steps include creating necessary folders, copying YAML files, and deploying the files to create and manage the deployment and services. Each step is accompanied by commands and their explanations to ensure a successful setup.

# Folder Creation

## Why:

Creating a folder structure and organizing YAML files is essential for managing configurations and deployments in a structured manner. This helps in maintaining consistency and ease of access to necessary files.

## Steps:

1. Log into the Instance using the public IP and ppk file.  
2. Navigate to the `/etc` folder:  
 cd /etc  
3. Create a folder named `sample`:  
 mkdir sample  
4. List the contents to verify the folder creation:  
 ls  
5. Navigate to the newly created `sample` folder:  
 cd /etc/sample  
6. Copy your local YAML files (`sample-deploy.yaml` and `sample-service.yaml`) to this location:  
 cd /etc/sample

# Deployment Creation

## Why:

Deploying YAML files is crucial to define and manage the desired state of applications and services in a Kubernetes cluster. It allows for automated deployment, scaling, and management of containerized applications.

## Steps:

1. After copying the YAML files to `/etc/sample`, list the contents to verify the files:  
 ls  
2. Deploy the deployment file:  
 kubectl apply -f sample-deploy.yaml  
 - This command creates the deployment as defined in the YAML file.  
3. Deploy the service file:  
 kubectl apply -f sample-service.yaml  
 - This command creates the service as defined in the YAML file.  
4. Verify the deployment by listing deployments:  
 kubectl get deployment  
 - Example Output:  
   
 ubuntu@k8smaster:/etc/sample$ kubectl get deployment  
 NAME READY UP-TO-DATE AVAILABLE AGE  
 springsample-deployment 2/2 2 2 20h  
   
5. List the pods to verify they are running:  
 kubectl get pod  
 - Example Output:  
   
 ubuntu@k8smaster:/etc/sample$ kubectl get pod  
 NAME READY STATUS RESTARTS AGE  
 springsample-deployment-577cd4fc57-dpf2d 1/1 Running 0 163m  
   
6. List the services to verify they are created:  
 kubectl get svc  
 - Example Output:  
   
 ubuntu@k8smaster:/etc/sample$ kubectl get svc  
 NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE  
 kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 20h  
 springsample-service NodePort 10.111.234.156 10.0.1.154 80:30558/TCP 20h

# Expose the Service

## Why:

Exposing the service allows external access to the application running inside the Kubernetes cluster. This step is necessary to make the application accessible over the internet using the public IP of the master instance.

## Steps:

1. Obtain the public IP of your master instance.  
 - Example : 13.233.62.83  
2. Expose the service in your browser using the following URL format:  
 http://<public-ip>:30558/appdata/api/get  
 - Example: http://13.233.62.83:30558/appdata/api/get  
3. The result will be displayed in the browser, indicating that the service is successfully exposed and accessible.