Kubernetes documentation 1

1. Deploy a nginx deployment with name web1 and custom index.html. Expose it to nodeport of 30080 in “web” namespace for this scenario.

We first create a web namespace on Kubernetes:

Graphical user interface, application

Description automatically generated

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Now we deploy a nginx deployment with name web1 and custom index.html.  
We expose it to nodeport 30080 in web namespace which we created earlier.

The yaml file to perform this action is :

Text

Description automatically generated

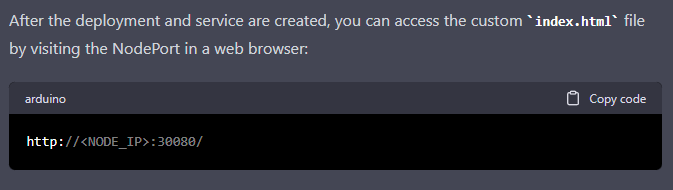
Text

Description automatically generated

In this yaml file, a **Service** resource is defined after the **Deployment** and **ConfigMap**. The **Service** resource is named **web1-service** and has a **type** of **NodePort**. The **ports** section specifies that port 80 of the **web1-service** service should be mapped to port 80 of the **nginx** container, and that the **nodePort** should be set to 30080.

To apply this updated YAML file, you can run:

* **kubectl apply -f nginx-deployment.yaml**

****

Replace **<NODE\_IP>** with the IP address of one of the nodes in your Kubernetes cluster. This should display the custom **index.html** file that you defined in the ConfigMap.

1. Create a WordPress deployment with mysql as a database and expose it to nodeport 30402. Use “volume” namespace for this scenario and login to wordpress with predefined credentials.

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Save the yaml file and create the resource by running:

* **kubectl apply -f wordpress.yaml**

1. Create a pod named sidecar with 2 containers named main and sidecar respectively. Fetch the logs from app container to sidecar container. Use nginx for main and busybox:1.28 image for sidecar container

Text

Description automatically generated

Graphical user interface, application

Description automatically generated