**Deploying Multitier Application Using Kubernetes**

**Objective:** Todeploy a multitier application with Kubernetes

**Tools required:** kubeadm, kubectl, kubelet, and etcd

**Prerequisites:** A Kubernetes cluster must be set up

Steps to be followed:

1. Creating Deployment for MySQL
2. Creating Deployment for WordPress
3. Exposing Service for WordPress and MySQL Deployment
4. Verifying the Deployment of application

**Step 1: Creating Deployment for MySQL**

1. Write the following code in the **mysql.yaml** file:

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**creationTimestamp: null**

**labels:**

**app: mysql**

**name: mysql**

**spec:**

**replicas: 1**

**selector:**

**matchLabels:**

**app: mysql**

**strategy: {}**

**template:**

**metadata:**

**creationTimestamp: null**

**labels:**

**app: mysql**

**spec:**

**containers:**

**- image: mysql:5.6**

**name: mysql**

**env:**

**- name: MYSQL\_ROOT\_PASSWORD**

**value: simplilearn**

**- name: MYSQL\_DATABASE**

**value: database1**

**resources: {}**

**status: {}**

Text

Description automatically generated

1. Create Deployment for **MySQL** using the following command:

**kubectl create -f mysql.yaml**

**A screenshot of a computer

Description automatically generated with medium confidence**

1. To verify Pods and Deployments, run the following commands:

**kubectl get pods**

**kubectl get deployments**

A screenshot of a computer

Description automatically generated

**Step 2: Creating Deployment for WordPress**

1. Write the following code in the **wordpress.yaml** file:

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**creationTimestamp: null**

**labels:**

**app: wordpress**

**name: wordpress**

**spec:**

**replicas: 1**

**selector:**

**matchLabels:**

**app: wordpress**

**strategy: {}**

**template:**

**metadata:**

**creationTimestamp: null**

**labels:**

**app: wordpress**

**spec:**

**containers:**

**- image: wordpress**

**name: wordpress**

**env:**

**- name: WORDPRESS\_DB\_HOST**

**value: mysql**

**- name: WORDPRESS\_DB\_PASSWORD**

**value: simplilearn**

**- name: WORDPRESS\_DB\_USER**

**value: root**

**- name: WORDPRESS\_DB\_NAME**

**value: database1**

**resources: {}**

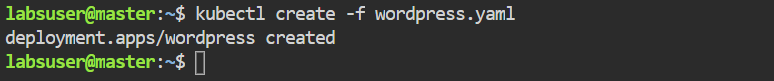
**status: {}**

Text

Description automatically generated with medium confidence

1. Create Deployment for **WordPress** using the following command:

**kubectl create -f wordpress.yaml**

****

1. To verify Pods and Deployments, run the following commands:

**kubectl get pods**

**kubectl get deployments**

Graphical user interface, text

Description automatically generated

**Step 3: Exposing Service for WordPress and MySQL Deployment**

1. To expose Service for **WordPress** and **MySQL** Deployment, run the following commands:

**kubectl expose deployment mysql --port=3306**

**kubectl expose deployment wordpress --port=80**

**Text

Description automatically generated**

1. Change the Service type for both **MySQL** and **WordPress** from **ClusterIP** to **Nodeport**.

**kubectl edit svc mysql**

**kubectl edit svc wordpress**

**Text

Description automatically generated**

|  |
| --- |
| Note: Edit Service type **ClusterIP** to **NodePort** |

Text

Description automatically generated

1. To verify the Service type of **MySQL** and **WordPress**, run the following command:

**kubectl get svc**

**Text

Description automatically generated with medium confidence**

1. To get detailed information on the Pods, use the following command:

**kubectl get pods -o wide**

**kubectl get nodes -o wide**

**A screenshot of a computer

Description automatically generated with medium confidence**

Copy the **WordPress** Pod IP address or any of the nodes **Internal-IP** and Service **NodePort** of **WordPress** and paste them into a browser to check the Deployment's success.

**Step 4: Verifying the Deployment of application**

1. Graphical user interface, text, application, chat or text message

   Description automatically generatedIn the **master** node, go to the **desktop** mode.
2. Open **Firefox** web browser.

Graphical user interface, text, chat or text message

Description automatically generated

1. In the browser, enter the **IP address** and Service **NodePort**.

**172.31.7.122:31891 or 10.46.0.7:80**

Graphical user interface, application, Word

Description automatically generated

The **WordPress** default page will appear. Finish the installation process and provide the necessary account information.

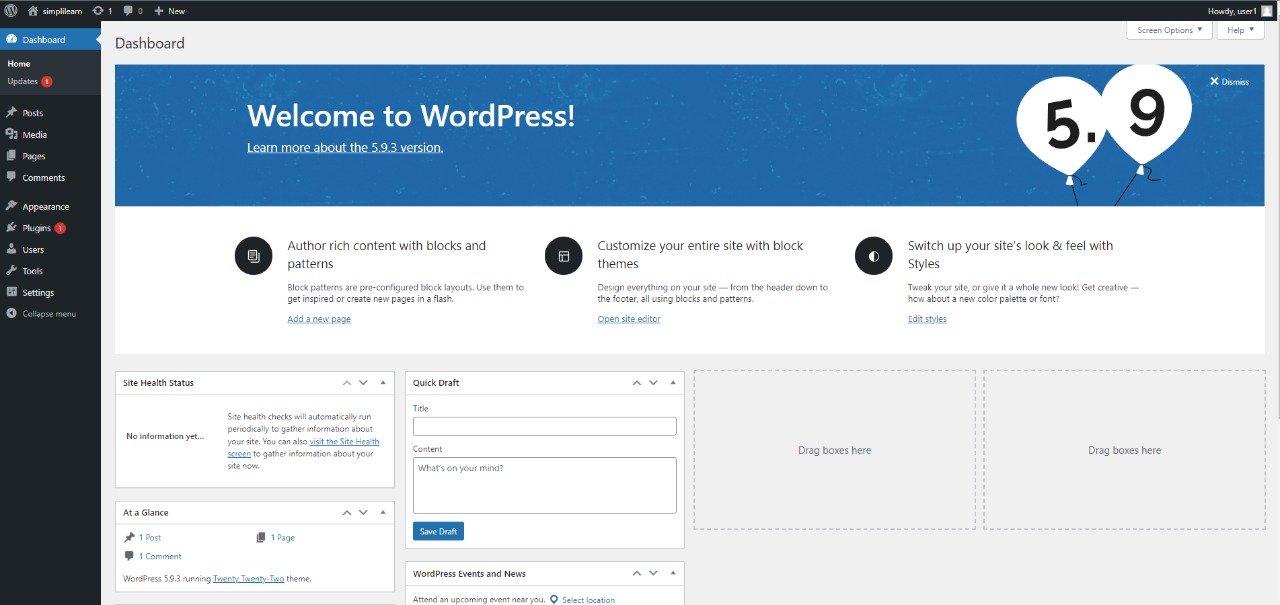
Graphical user interface, text, application

Description automatically generated

1. On the **login** **page,** enter the **username** and **password** provided during the installation process.

Graphical user interface, application

Description automatically generated



The **WordPress** application has been successfully deployed, as seen in the screenshot above.