**Lesson 4 Demo 2**

**Deploying Multitier Application Postgres and Gogs**

**Objective:** Todeploy a multitier application Postgres and Gogs using Kubernetes

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

**Prerequisites:** A Kubernetes cluster must be set up (follow steps of Lesson 2 Demo 1)

Steps to be followed:

1. Creating Deployment for Postgres
2. Creating Deployment for Gogs
3. Exposing Service for Postgres and Gogs Deployment
4. Verifying the Deployment of application

**Step 1: Creating Deployment for Postgres**

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

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**creationTimestamp: null**

**labels:**

**app: postgres**

**name: postgres**

**spec:**

**replicas: 1**

**selector:**

**matchLabels:**

**app: postgres**

**strategy: {}**

**template:**

**metadata:**

**creationTimestamp: null**

**labels:**

**app: postgres**

**spec:**

**containers:**

**- image: postgres**

**name: postgres**

**env:**

**- name: POSTGRES\_PASSWORD**

**value: simplilearn**

**- name: POSTGRES\_DB**

**value: database1**

**resources: {}**

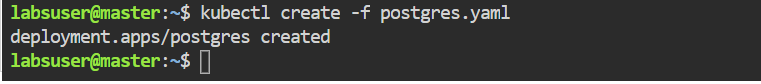
**status: {}**

**Text

Description automatically generated**

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

**kubectl create -f postgres.yaml**

****

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

**kubectl get pods**

**kubectl get deployments**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Step 2: Creating Deployment for Gogs**

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

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**creationTimestamp: null**

**labels:**

**app: gogs**

**name: gogs**

**spec:**

**replicas: 1**

**selector:**

**matchLabels:**

**app: gogs**

**strategy: {}**

**template:**

**metadata:**

**creationTimestamp: null**

**labels:**

**app: gogs**

**spec:**

**containers:**

**- image: docker.io/gogs/gogs**

**name: gogs**

**resources: {}**

**status: {}**

**Text

Description automatically generated with medium confidence**

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

**kubectl create -f gogs.yaml**

**Graphical user interface

Description automatically generated with medium confidence**

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

**kubectl get pods**

**kubectl get deployments**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Step 3: Exposing Service for Postgres and Gogs Deployment**

1. To expose a Service for **Postgres** and **Gogs** Deployment, run the following commands:

**kubectl expose deployment postgres --port=5432**

**kubectl expose deployment gogs --port=3000**

**Text

Description automatically generated**

1. Change the Service type for both **Postgres** and **Gogs** from **ClusterIP** to **Nodeport**.

**kubectl edit svc postgres**

**kubectl edit svc gogs**

**Text

Description automatically generated**

**Text

Description automatically generated**

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

1. To verify the Service type of **Postgres** and **Gogs**, run the following command:

**kubectl get svc**

**Graphical user interface

Description automatically generated**

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

|  |
| --- |
| Note: For more details follow the Lesson 4 Demo 1 |

**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, application, chat or text message

Description automatically generated

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

**172.31.7.122:32361 or 10.46.0.1:3000**

**Graphical user interface, text, application, email

Description automatically generated**

Complete the database set up by following the Deployment and Service configuration steps, then click **Install Gogs**.

* Host: **postgres**
* User: **postgres**
* Password: **simplilearn**
* Database Name: **database1**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

If an error occurs due to caching, open a new tab, and repaste the **IP address** and **NodePort**.

Graphical user interface, text

Description automatically generated

The **Gogs** application has been successfully installed and deployed, as shown in the screenshot above.