

Gitea Production Deployment on Kubernetes

Assignment 3 - Orchestration with Gitea

Nikhil Shankar Chirakkal Sivasankaran 9026254

Overview

Production-ready Gitea deployment on k3s Kubernetes cluster with MySQL database, persistent storage, and public access via ngrok.

Prerequisites

- WSL2 (Ubuntu)
- Docker Desktop
- Python 3.x
- Ansible

Quick Start

1. Install Dependencies

```
sudo apt update  
sudo apt install -y ansible python3-kubernetes
```

2. Deploy Gitea

```
git submodule update --init --recursive  
ansible-playbook up.yml
```

Wait for all pods to be ready:

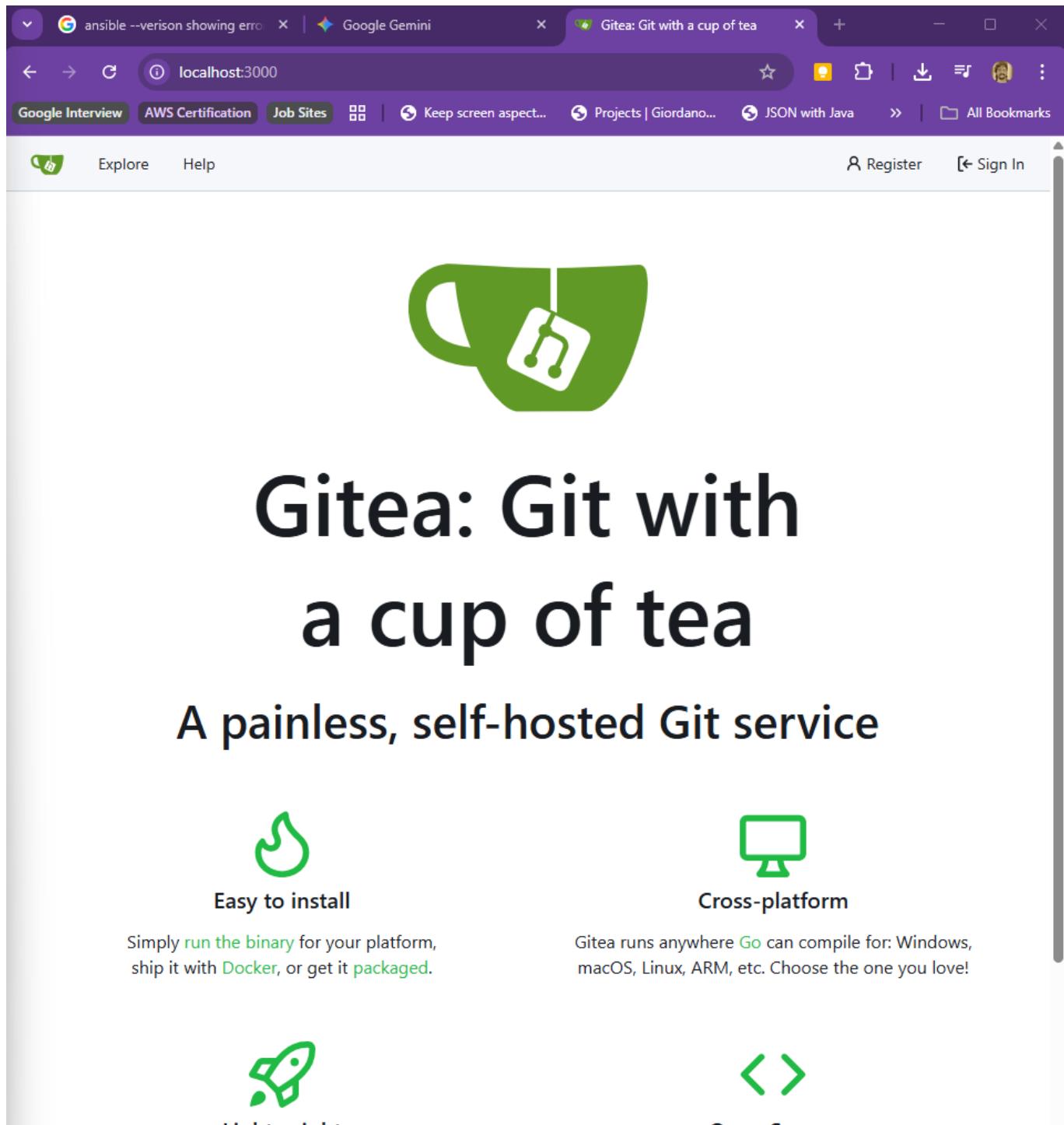
```
kubectl get pods -w
```

3. Access Gitea Locally

Get ingress IP:

```
kubectl get ingress
```

Access via: <http://<INGRESS-IP>>



The screenshot shows a Microsoft Edge browser window with three tabs open: "ansible --verison showing erro", "Google Gemini", and "Gitea: Git with a cup of tea". The main content area displays the Gitea landing page. It features a large green logo of a teacup with a small robot head inside. Below the logo, the text "Gitea: Git with a cup of tea" is displayed in a large, bold, black font. Underneath, it says "A painless, self-hosted Git service". There are four sections with icons and text: "Easy to install" (green flame icon), "Cross-platform" (monitor icon), "Lightweight" (rocket icon), and "Open Source" (code editor icon). Each section has a brief description.

Easy to install

Simply run the binary for your platform, ship it with Docker, or get it packaged.

Cross-platform

Gitea runs anywhere Go can compile for: Windows, macOS, Linux, ARM, etc. Choose the one you love!

Lightweight

Open Source

4. Setup Public Access with ngrok

Install ngrok:

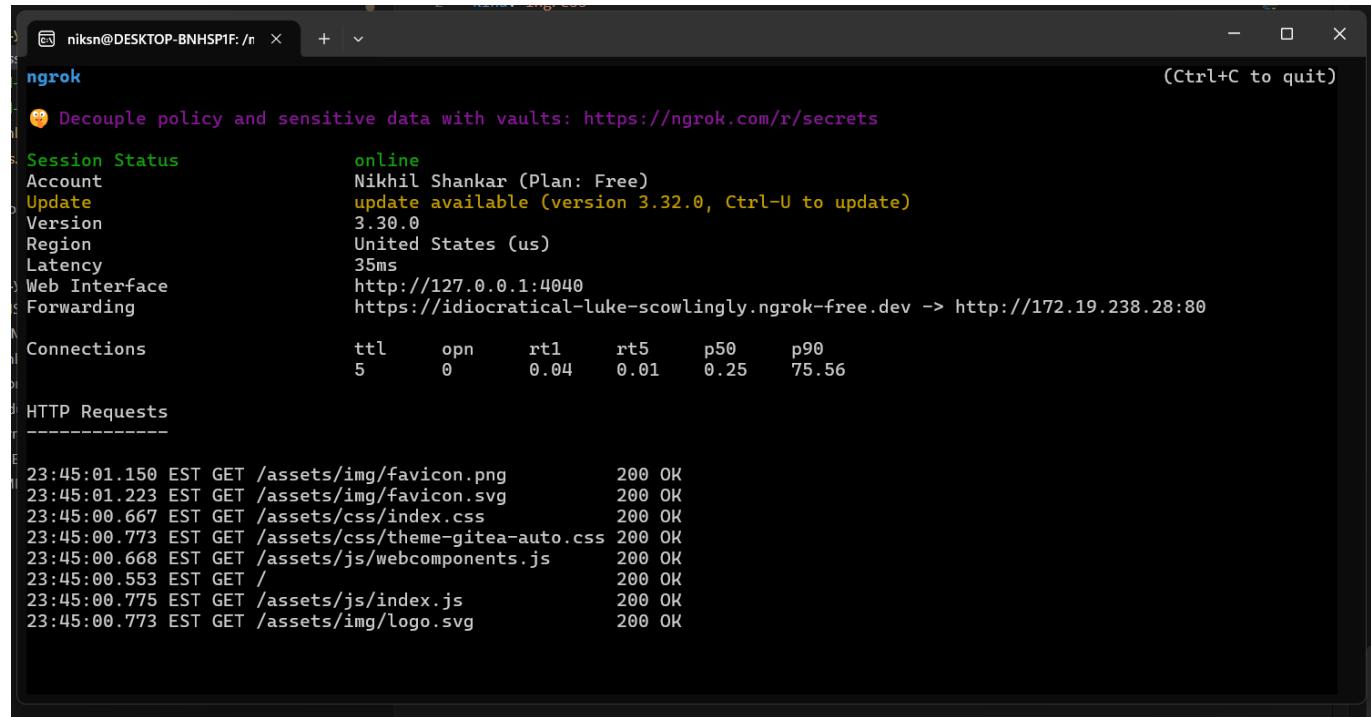
```
curl -s https://ngrok-agent.s3.amazonaws.com/ngrok.asc | sudo tee  
/etc/apt/trusted.gpg.d/ngrok.asc >/dev/null  
echo "deb https://ngrok-agent.s3.amazonaws.com buster main" | sudo tee  
/etc/apt/sources.list.d/ngrok.list  
sudo apt update && sudo apt install ngrok
```

Authenticate (get token from <https://ngrok.com>):

```
ngrok config add-authtoken <YOUR_TOKEN>
```

Get ingress IP and expose:

```
kubectl get ingress # Note the ADDRESS  
ngrok http http://<INGRESS-IP>:80
```



The screenshot shows a terminal window titled "ngrok" with the command "ngrok http http://<INGRESS-IP>:80" running. The output displays session status, including account information (Nikhil Shankar, Plan: Free), update availability (version 3.32.0), and connection statistics. It also lists recent HTTP requests made via the proxy.

```
ngrok (Ctrl+C to quit)  
Decouple policy and sensitive data with vaults: https://ngrok.com/r/secrets  
  
Session Status          online  
Account      Nikhil Shankar (Plan: Free)  
Update       update available (version 3.32.0, Ctrl-U to update)  
Version      3.30.0  
Region       United States (us)  
Latency      35ms  
Web Interface http://127.0.0.1:4040  
Forwarding   https://idiocritical-luke-scowlingly.ngrok-free.dev -> http://172.19.238.28:80  
  
Connections    ttl     opn     rt1     rt5     p50     p90  
               5       0     0.04    0.01    0.25   75.56  
  
HTTP Requests  
-----  
23:45:01.150 EST GET /assets/img/favicon.png      200 OK  
23:45:01.223 EST GET /assets/img/favicon.svg    200 OK  
23:45:00.667 EST GET /assets/css/index.css     200 OK  
23:45:00.773 EST GET /assets/css/theme-gitea-auto.css 200 OK  
23:45:00.668 EST GET /assets/js/webcomponents.js 200 OK  
23:45:00.553 EST GET /                         200 OK  
23:45:00.775 EST GET /assets/js/index.js       200 OK  
23:45:00.773 EST GET /assets/img/logo.svg     200 OK
```

Access Gitea publicly via the ngrok URL.

The screenshot shows a web browser window with the URL 'idiocritical-luke-scowlingly.ngrok-free.dev' highlighted in the address bar. The main content of the page is the Gitea landing page, which includes a large green icon of a teacup with a saucer, followed by the text 'Gitea: Git with a cup of tea' and 'A painless, self-hosted Git service'. Below this, there are four sections with icons and text: 'Easy to install' (flame icon), 'Cross-platform' (monitor icon), 'Lightweight' (rocket icon), and 'Open Source' (code brackets icon).

Production Features

Persistent Storage

- Gitea data stored in persistent volumes
- Survives pod restarts
- Configured in `gitea/values.yaml`

MySQL Database

- External MySQL deployment
- Persistent storage for database
- Configuration: `gitea/mysql-deployment.yaml`

```

[niksn@DESKTOP-BNHSP1F: /n + ^ - name: MYSQL_ROOT_PASSWORD
gitea-mysql-6c657bb598-24zm2      1/1   Running   0    43s
gitea-valkey-cluster-0             0/1   Running   0    16s
gitea-valkey-cluster-1             0/1   Running   0    16s
gitea-valkey-cluster-2             0/1   Running   0    16s
ingress-nginx-controller-6f449f6b9d-m77hg 1/1   Running   0    51s
gitea-846988c5d5-8t8dq           0/1   Init:2/3  0    17s
gitea-846988c5d5-8t8dq           0/1   PodInitializing 0    24s
gitea-846988c5d5-8t8dq           0/1   Running   0    25s
gitea-valkey-cluster-2             1/1   Running   0    28s
gitea-valkey-cluster-0             1/1   Running   0    28s
gitea-valkey-cluster-1             1/1   Running   0    28s
^Cniksn@DESKTOP-BNHSP1F:/mnt/d/Conestoga/Course2-DevOps/Sem2/ContainerOrchestration/Assignment3/cdevops-gitea$ kubectl get pods
NAME                               READY   STATUS   RESTARTS   AGE
gitea-846988c5d5-8t8dq           0/1   Running   0    35s
gitea-mysql-6c657bb598-24zm2     1/1   Running   0    62s
gitea-valkey-cluster-0             1/1   Running   0    35s
gitea-valkey-cluster-1             1/1   Running   0    35s
gitea-valkey-cluster-2             1/1   Running   0    35s
ingress-nginx-controller-6f449f6b9d-m77hg 1/1   Running   0    70s
niksn@DESKTOP-BNHSP1F:/mnt/d/Conestoga/Course2-DevOps/Sem2/ContainerOrchestration/Assignment3/cdevops-gitea$ kubectl get pods
NAME                               READY   STATUS   RESTARTS   AGE
gitea-846988c5d5-8t8dq           1/1   Running   0    50s
gitea-mysql-6c657bb598-24zm2     1/1   Running   0    77s
gitea-valkey-cluster-0             1/1   Running   0    50s
gitea-valkey-cluster-1             1/1   Running   0    50s
gitea-valkey-cluster-2             1/1   Running   0    50s
ingress-nginx-controller-6f449f6b9d-m77hg 1/1   Running   0    85s
niksn@DESKTOP-BNHSP1F:/mnt/d/Conestoga/Course2-DevOps/Sem2/ContainerOrchestration/Assignment3/cdevops-gitea$
```

Public Access

- Nginx Ingress Controller
- ngrok tunnel for external access
- Zero port-forwarding needed

Teardown

```
ansible-playbook down.yml
```

Architecture



Key Files

- `up.yml` - Main deployment playbook
- `gitea/values.yaml` - Gitea Helm configuration
- `gitea/mysql-deployment.yaml` - MySQL manifest
- `gitea/ingress.yml` - Ingress configuration