

Django CRM - Container Orchestration Final Assignment

Project Report

Richard Andrey Biscazzi - 8903530

Nikhil Shankar Chirakkal Sivasankaran - 9026254

Step 1: Gitea Repository Setup

Tasks Completed:

- Cloned repository to Gitea: <https://gitea.devsecmindset.dev/Richard/final-assignment-crm.git>
- Added teammates as collaborators
- Repository configured with proper access

Screenshots:

<https://gitea.devsecmindset.dev/admin/users>

ID	Username	Email Address	Activated	Restricted	2FA	Created	Last Sign-In
1	gitea-admin	admin@gitea.devsecmindset.dev	✓	✗	✗	Dec 12, 2025	Dec 12, 2025
2	Nikhil	email@email.com	✓	✗	✗	Dec 12, 2025	Never Signed-In
3	Richard	newmail@email.com	✓	✗	✗	Dec 12, 2025	Never Signed-In

Users created

Powered by Gitea Version: 1.22.3 Page: 33ms Template: 20ms [English](#) | [Licenses](#) | [API](#)

<https://gitea.devsecmindset.dev/Richard/final-assignment-crm>

Richard / Final-assignment-crm

Code Issues Pull Requests Actions Packages Projects Releases Wiki Activity Settings

Manage Topics

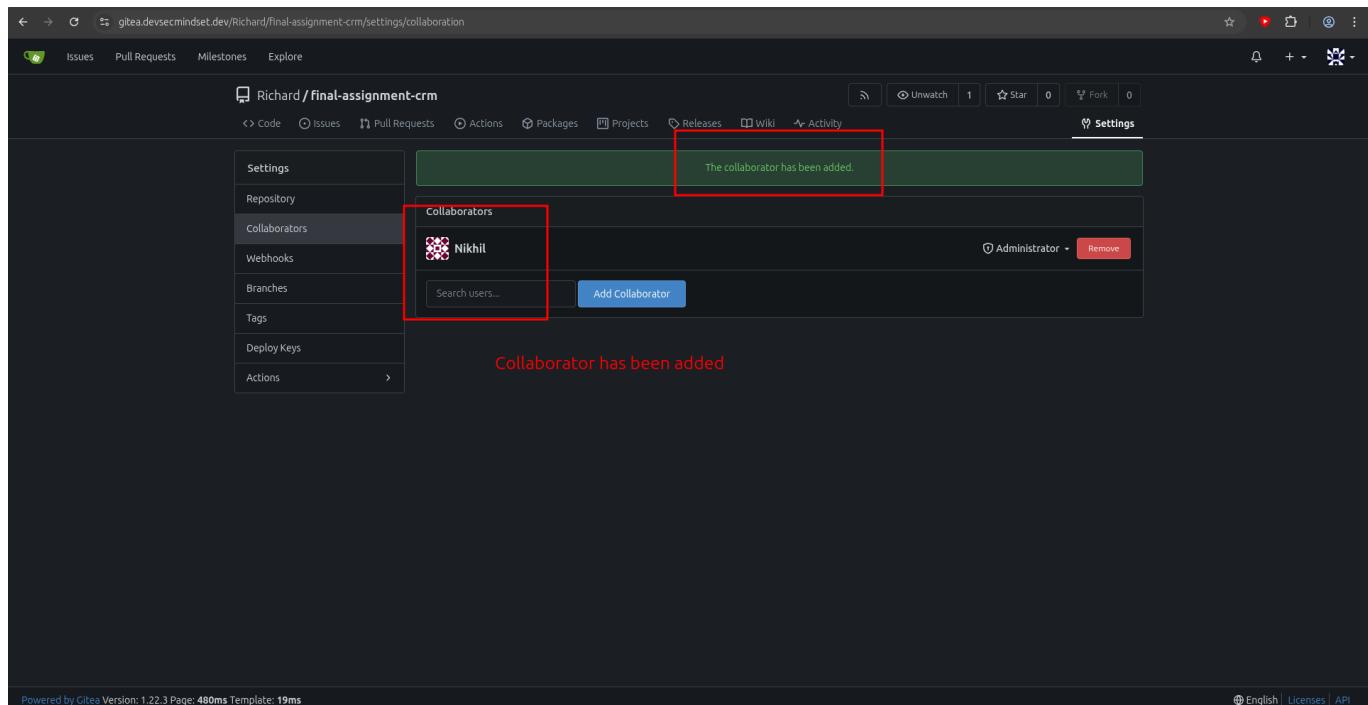
2 Commits 1 Branch 0 Tags 37 KB

main · Go to file Add File · Search code ... HTTPS SSH https://gitea.devsecmindset.dev/Richard/final-assignment-crm.git ⌂ ⌂

Richard 2e38e346d2 Update patch-settings.diff	2 weeks ago
.devcontainer	Initial commit
.github/workflows	Initial commit
django-crm @ bbd5f5e902	Initial commit
.gitignore	Initial commit
.gitmodules	Initial commit
docker-compose.yml	Initial commit
Dockerfile	Initial commit
down.yml	Initial commit
LICENSE	Initial commit
patch-settings.diff	Update patch-settings.diff
README.md	Initial commit
schema_changes.sql	Initial commit
settings.py	Initial commit
up.yml	Initial commit

Repository migrated

django-crm-helm



Step 2: NFS Storage Configuration

Tasks Completed:

- Configured TrueNAS NFS storage for database persistence
- Created StorageClass: `truenas-nfs`
- Verified NFS mount accessibility
- Database PVCs created and bound

Screenshots:

NFS share accessibility

```
ubuntu@ubuntu2204:~/source/container-assignment3$ showmount -e 10.172.27.23
/mnt/Application/mysql-data 10.172.27.0/24
```

Showing StorageClass

```
ubuntu@ubuntu2204:~/source/container-assignment3$ kubectl get storageclass truenas-nfs -o yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  annotations:
    meta.helm.sh/release-name: nfs-provisioner
    meta.helm.sh/release-namespace: default
    creationTimestamp: "2025-12-12T02:26:24Z"
  labels:
    app: nfs-subdir-external-provisioner
    app.kubernetes.io/managed-by: Helm
    chart: nfs-subdir-external-provisioner-4.0.18
    heritage: Helm
    release: nfs-provisioner
    name: truenas-nfs
    resourceVersion: "537252"
    uid: ecd01080-cbeb-44aa-82e0-5805e86531f1
provisioner: nfs-provisioner
  archiveOnDelete: "true"
  provisioner: cluster.local/nfs-provisioner-nfs-subdir-external-provisioner
  reclaimPolicy: Delete
  volumeBindingMode: Immediate
```

NFS mount inside pod

```
ubuntu@ubuntu2204:~/source/container-assignment3$ kubectl get pvc | grep postgresql
pvc-d75ea444-836f-47d6-8682-cf1fd621d97 10Gi RW0 truenas-nfs <unset> 14m
pvc-037ec9d-5224-48cd-9187-bfc1baa531fc 10Gi RW0 truenas-nfs <unset> 14m
pvc-9270da8e-63be-42c5-a9b2-b8e7487de17 10Gi RW0 truenas-nfs <unset> 14m
```

NFS mount inside pod

```
ubuntu@ubuntu2204:~/source/container-assignment3$ kubectl exec -it gitaa-prod-postgresql-ha-postgresql-0 -- mount | grep nfs
10.172.27.23:/mnt/Application/mysql-data/default/data/gitaa-prod-postgresql-ha-postgresql-ha-pvc-d75ea444-836f-47d6-8682-cf1fd621d97 on /bitnami/postgresql type nfs4 (rw,relatime,vers=4.2,rsize=1048576,wsize=1048576,hard,proto=tcp,timeo=500,retrans=2,sec=sys,clientaddr=10.172.27.3,local_lock=none,addr=10.172.27.23)
```

Helm values configuration

```
HOST: gitaa.devsecmindset.dev
NAME: gitaa
PASSWORD: gitaa
SSL MODE: disable
USER: gitaa
server:
  DOMAIN: gitaa.devsecmindset.dev
  HTTP_PORT: 3000
  ROOT_URL: https://gitaa.devsecmindset.dev
  SSH_DOMAIN: gitaa.devsecmindset.dev
  SSH_PORT: 22

postgresql-ha:
  enabled: true
  global:
    storageClass: truenas-nfs
    persistentVolume:
      accessModes:
        -ReadWriteOnce
      size: 10Gi
      storageClassName: truenas-nfs
    pgpool:
      resources:
```

```

PostgreSQL data directory from inside the pod
ubuntu@ubuntu2204:/source/container-assignment3$ kubectl exec -it gitea-prod-postgresql-ha-postgresql-0 -- ls -la /bitnami/postgresql/data/
total 61
drwx----- 19 1001 950 25 Dec 12 05:58 .
drwxrwxrwx  5 959 950 5 Dec 12 05:58 .
drwx----- 19 1001 950 3 Dec 12 05:58 base
drwx----- 1 1001 950 51 Dec 12 05:58 current_logfiles
drwx----- 2 1001 950 66 Dec 12 05:52 global
drwx----- 2 1001 950 2 Dec 12 05:50 pg_commit_ts
drwx----- 2 1001 950 2 Dec 12 05:58 pg_dynshmem
drwx----- 1 1001 950 2640 Dec 12 05:58 pg_ident.conf
drwx----- 4 1001 950 5 Dec 12 06:01 pg_logical
drwx----- 2 1001 950 950 3 Dec 12 05:58 pg_multixact
drwx----- 2 1001 950 2 Dec 12 05:58 pg_notify
drwx----- 4 1001 950 4 Dec 12 05:51 pg_replslot
drwx----- 2 1001 950 2 Dec 12 05:50 pg_serial
drwx----- 2 1001 950 2 Dec 12 05:58 pg_snapshots
drwx----- 2 1001 950 2 Dec 12 05:50 pg_stat
drwx----- 2 1001 950 2 Dec 12 05:58 pg_stat_tmp
drwx----- 2 1001 950 950 3 Dec 12 05:58 pg_stattrans
drwx----- 2 1001 950 2 Dec 12 05:58 pg_tbsync
drwx----- 2 1001 950 3 Dec 12 05:58 pg_twophase
drwx----- 1 1001 950 3 Dec 12 05:58 PG_VERSION
drwx----- 4 1001 950 14 Dec 12 05:52 pg_wal
drwx----- 2 1001 950 3 Dec 12 05:58 pg_xact
drwx----- 1 1001 950 88 Dec 12 05:58 postgresql.auto.conf
drwx----- 1 1001 950 248 Dec 12 05:58 postmaster.pid
drwx----- 1 1001 950 61 Dec 12 05:58 postmaster.pid

ubuntu@ubuntu2204:/source/container-assignment3$ kubectl exec -it gitea-prod-postgresql-ha-postgresql-0 -- bash
I have no name@gitea-prod-postgresql-ha-postgresql-0:/$ du -sh /bitnami/postgresql/data/
16M /bitnami/postgresql/data/
I have no name@gitea-prod-postgresql-ha-postgresql-0:/$ exit
exit
ubuntu@ubuntu2204:/source/container-assignment3$ 
```

Checking database size


```

Show PVC directories
truenas@truenas-nas:k3p.dev/ui/system/shell
trueNAS
Dashboard System
Storage Datasets Shares
Data Protection Network Credentials Instances Apps Reporting
System

RESULT IN SYSTEM FAILURE.
Welcome to TrueNAS
Last login: Thu Dec 11 22:19:47 PST 2025 on pts/4
truenas@truenas-nas:~$ sudo ls -la /mnt/Application/mysql-data/default-data/gitea-prod-postgresql-ha-postgresql-0-pvc-d75ea444-836f-47d6-8682-c7f1fd621d97
[sudo] password for truenas_admin:
total 28
drwxr-x 6 truenas_admin truenas_admin 7 Dec 11 21:54 .
drwxr-x 3 root root 3 Dec 11 15:54 ..
drwxrwxrwx 5 truenas_admin truenas_admin 5 Dec 11 21:50 default-data-gitea-prod-postgresql-ha-postgresql-0-pvc-c837ec3d-5224-48cd-9f87-bfcbaa531fc
drwxrwxrwx 5 truenas_admin truenas_admin 5 Dec 11 21:51 default-data-gitea-prod-postgresql-ha-postgresql-1-pvc-c837ec3d-5224-48cd-9f87-bfcbaa531fc
drwxrwxrwx 2 truenas_admin truenas_admin 2 Dec 11 21:54 default-test-truenas-nfs-pvc-ccaa15e7-9ab8-4c4c-8cc9-45aaeab07d1be
drwxrwxrwx 2 truenas_admin truenas_admin 2 Dec 11 21:54 default-test-truenas-nfs-pvc-ccaa15e7-9ab8-4c4c-8cc9-45aaeab07d1be
drwxrwxrwx 1 truenas_admin truenas_admin 0 Dec 11 18:24 test-file8c-?.

truenas@truenas-nas:~$ ls -la /mnt/Application/mysql-data/default-data/gitea-prod-postgresql-ha-postgresql-0-pvc-d75ea444-836f-47d6-8682-c7f1fd621d97/
total 61
drwxr-x 19 1001 truenas_admin 25 Dec 11 21:50 .
drwxrwxrwx 1 1001 truenas_admin 3 Dec 11 21:50 ..
-rw-r--r-- 1 1001 truenas_admin 3 Dec 11 21:50 PG_VERSION
drwx----- 8 1001 truenas_admin 8 Dec 11 21:51 base
drwx----- 1 1001 truenas_admin 91 Dec 11 21:50 current_logfiles
drwx----- 2 1001 truenas_admin 66 Dec 11 21:50 global
drwx----- 2 1001 truenas_admin 2 Dec 11 21:50 pg_commit_ts
drwx----- 2 1001 truenas_admin 2 Dec 11 21:50 pg_dynshmem
drwx----- 2 1001 truenas_admin 2 Dec 11 21:50 pg_multixact
drwx----- 2 1001 truenas_admin 2 Dec 11 21:50 pg_notify
drwx----- 4 1001 truenas_admin 4 Dec 11 21:50 pg_replslot
drwx----- 2 1001 truenas_admin 2 Dec 11 21:50 pg_serial
drwx----- 2 1001 truenas_admin 2 Dec 11 21:50 pg_snapshots
drwx----- 2 1001 truenas_admin 2 Dec 11 21:50 pg_stat
drwx----- 2 1001 truenas_admin 2 Dec 11 21:50 pg_stat_tmp
drwx----- 2 1001 truenas_admin 3 Dec 11 21:50 pg_stattrans
drwx----- 2 1001 truenas_admin 2 Dec 11 21:50 pg_tbsync
drwx----- 2 1001 truenas_admin 2 Dec 11 21:50 pg_twophase
drwx----- 4 1001 truenas_admin 14 Dec 11 21:52 pg_xact
drwx----- 2 1001 truenas_admin 3 Dec 11 21:50 pg_xlog
drwx----- 1 1001 truenas_admin 88 Dec 11 21:50 postgresql.auto.conf
drwx----- 1 1001 truenas_admin 249 Dec 11 21:50 postmaster.opts
drwx----- 1 1001 truenas_admin 81 Dec 11 21:50 postmaster.pid

truenas@truenas-nas:~$ 
```

Checking the data directory

Step 3: CI/CD Pipeline with Gitea Actions

Tasks Completed:

- Created Gitea personal access token with package write permissions
- Set up Gitea Actions runner using Docker
- Created `.gitea/workflows/build-and-push.yml` workflow file
- Configured automated Docker image build on code push
- Configured automated push to Gitea container registry
- Fixed Dockerfile to include g++ compiler for numpy dependencies
- Successfully built and pushed image to `gitea.devsecmindset.dev/nikhil/final-assignment-crm`

Runner Setup Process:

1. Generated Gitea token with `write:package` permission
2. Added token as repository secret (`GITEATOKEN`)
3. Created runner registration token from repository settings
4. Deployed runner using Docker:

```
docker run -d --name gitea-runner \
-e GITEA_INSTANCE_URL=https://gitea.devsecmindset.dev \
-e GITEA_RUNNER_REGISTRATION_TOKEN=<token> \
-v /var/run/docker.sock:/var/run/docker.sock \
gitea/act_runner:latest
```

5. Verified runner registered and active in repository settings

Screenshots:

The screenshot shows the 'User Settings' page in Gitea. On the left is a sidebar with options like Profile, Account, Appearance, Security, Blocked users, Applications, SSH / GPG Keys, Actions, Packages, Webhooks, Organizations, and Repositories. The main area is titled 'Manage Access Tokens' and contains instructions: 'These tokens grant access to your account using the Gitea API.' A red box highlights the 'Generate New Token' section, which includes a 'Token Name' input field containing 'container-final-project-containeregistrytoken'. Below it are radio buttons for 'Repository and Organization Access': 'Public only' (unchecked) and 'All (public, private, and limited)' (checked). A link 'Select permissions' is present. A blue 'Generate Token' button is at the bottom. Other sections include 'Authorized OAuth2 Applications' (with a note about granted access to third-party apps) and 'Manage OAuth2 Applications' (with a note about giving third-party apps access to user accounts).

The screenshot shows the 'Actions' settings page for the repository 'Richard / final-assignment-crm'. The sidebar on the left lists Settings, Repository, Collaborators, Webhooks, Branches, Tags, Deploy Keys, Actions (Runners, Secrets, Variables), and another Actions section. The main area is titled 'Runners Management (Total: 0)' and includes a search bar 'Search runners...'. A table with columns Status, ID, Name, Version, Type, Labels, and Last Online Time shows 'No runners available'. To the right is a 'Settings' sidebar with a 'Create new Runner' button, a 'How to start a runner' link, a 'REGISTRATION TOKEN' field containing 'TNltAJvSfnBcVsHZnR61Cy' with a copy icon, and a 'Reset registration token' link. A red box highlights the 'Settings' sidebar.

Screenshot of Docker Desktop showing the container details for "gitea-runner". The container ID is 06f073513a95 and it is running gitea/act_runner:latest.

```

2025-12-12 04:11:15 .runner is missing or not a regular file
2025-12-12 04:11:15 level=info msg="Registering runner, arch=amd64, os=linux, version=v0.2.13."
2025-12-12 04:11:15 level=debug msg="Successfully pinged the Gitea instance server"
2025-12-12 04:11:15 level=info msg="Runner registered successfully."
2025-12-12 04:11:15 SUCCESS
2025-12-12 04:11:15 time="2025-12-12T09:11:15Z" level=info msg="Starting runner daemon"
2025-12-12 04:11:15 time="2025-12-12T09:11:15Z" level=info msg="runner: 06f073513a95, with version: v0.2.13, with labels: [ubuntu-latest ubuntu-24.04 ubuntu-22.04], declare successfully"

```

Screenshot of a web browser showing the Gitea interface for managing runners. The page displays a table of runners, with one entry highlighted:

Status	ID	Name	Version	Type	Labels	Last Online Time	Edit
Idle	1	06f073513a95	v0.2.13	Repository	ubuntu-latest, ubuntu-24.04, ubuntu-22.04	1 minute ago	Edit

Powered by Gitea Version: 1.22.3 Page: 36ms Template: 4ms

English | Licenses | API

A screenshot of a web browser displaying the Gitea Actions interface. The URL is gitea.devsecmindset.dev/Richard/final-assignment-crm/actions/runs/5. The page shows a single job named "build-and-push" which completed successfully in 7m28s. The job steps are listed as follows:

Step	Description	Duration
Set up job		10s
Checkout code		4s
Set up Docker Buildx		7s
Log in to Gitea Container Registry		4s
Build and push Docker image		6m59s
Complete job		4s

The entire content area is highlighted with a red box.

A screenshot of a web browser displaying the Gitea Packages interface. The URL is gitea.devsecmindset.dev/Nikhil/-/packages. The page shows a package named "final-assignment-crm" published by Nikhil 1 minute ago. The package is associated with a Docker container. The entire content area is highlighted with a red box.

The screenshot shows a Gitea repository page for 'final-assignment-crm'. At the top, there's a navigation bar with links like 'Google Interview', 'AWS Certification', 'Job Sites', 'Cogitat', 'AWS', 'AmazonInterview', 'Keep screen aspect...', 'Projects | Giordano...', 'JSON with Java', '5 Deadly Sci-Fi Gad...', 'All Bookmarks', and a search bar. Below the navigation is a user profile for 'Nikhil' with a diamond-shaped icon.

The main content area has tabs for 'Repositories', 'Projects', 'Packages' (which is selected), 'Public Activity', and 'Starred Repositories'. The 'Packages' tab displays details for a Docker container:

- Installation:**
 - Pull the image from the command line:


```
docker pull gitea.devsecmindset.dev/nikhil/final-assignment-crm:58bd07117341af28cf5c78604deab9b6efedcbde
```
 - Digest:


```
sha256:d5c98d8def6648fbf5f83c46e76c442ed5666a5d7ed15249823392f26037b1bc
```
- OS / Arch:**

Digest:	OS / Arch	Size
sha256:b6d8946a8534a9d529d38725b51baba0564577473db849bb6b60980b4afbae95	linux/amd64	97 MiB
sha256:5954538f2a3242dd4b97ab92541bd46416420e6b2d9eaebe7693c8ee87a9ed08	unknown/unknown	18 KiB

On the right side, there's a 'Details' sidebar with information about the container, including its creation time ('3 minutes ago'), number of runs ('0'), and OCI/Docker compatibility. It also lists 'Versions' (latest and a specific commit) and a 'Settings' link.

At the bottom of the page, there's a footer with 'Powered by Gitea Version: 1.22.3 Page: 274ms Template: 173ms' and links for English, Licenses, and API.

Step 4: Cloudflare Tunnel - Exposing CRM to Internet

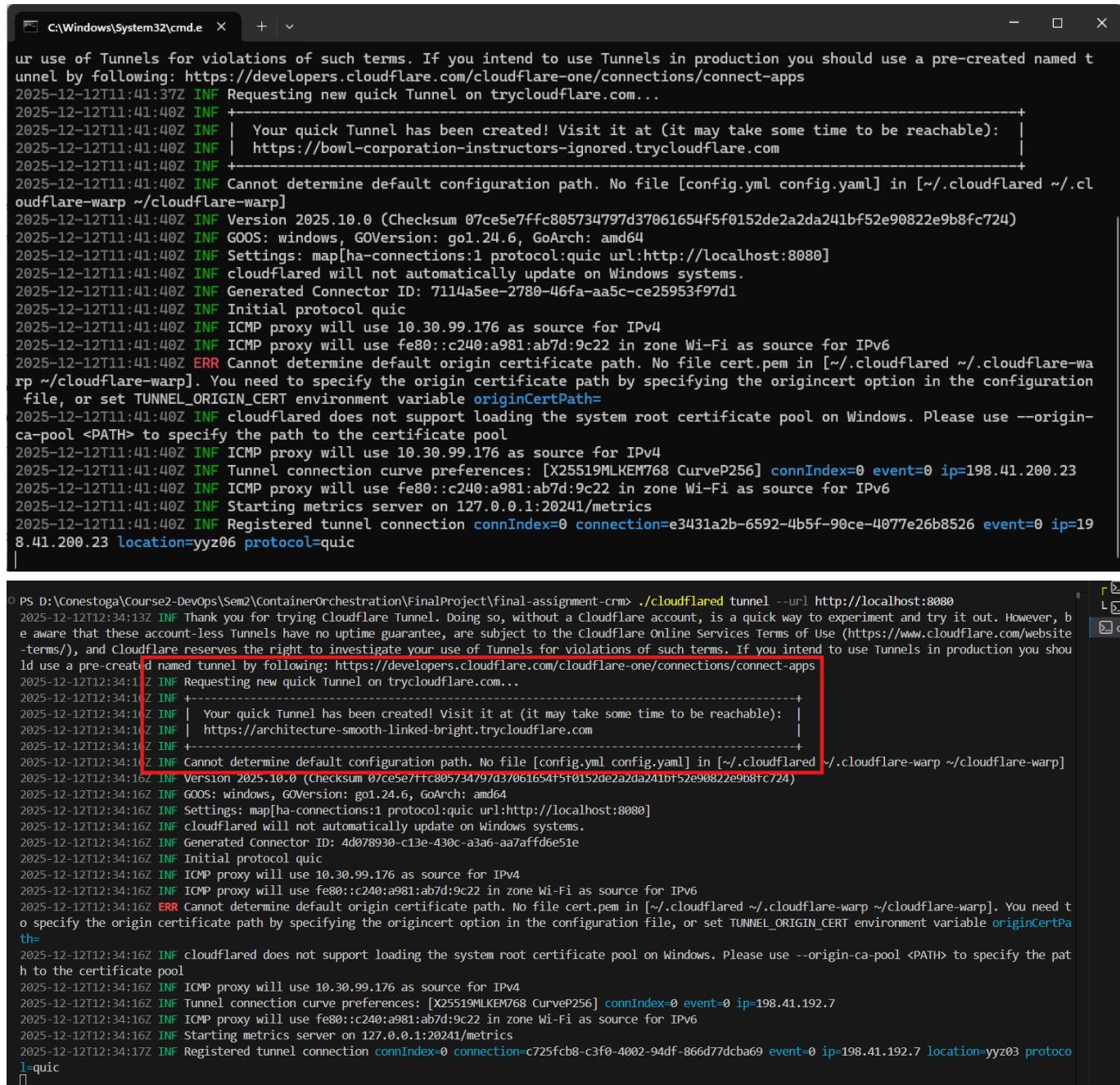
Tasks Completed:

- Downloaded and installed Cloudflare CLI
- Fixed Python version compatibility issue (changed from `python:alpine` to `python:3.12-alpine`)
- Configured Django ALLOWED_HOSTS to accept Cloudflare tunnel domains
- Added CSRF_TRUSTED_ORIGINS for Cloudflare URLs
- Created cloudflare tunnel using `./cloudflare.exe tunnel --url http://localhost:8080`
- Successfully exposed CRM application to public internet via Cloudflare tunnel
- Verified public URL accessibility

Configuration Changes:

- Dockerfile:** Updated base image to Python 3.12 for numpy/pandas compatibility
- settings.py:** Added Cloudflare tunnel URLs to ALLOWED_HOSTS:
 - `anaheim-bottles-hop-delayed.trycloudflare.com`
 - `architecture-smooth-linked-bright.trycloudflare.com`
- urls.py:** Added root URL redirect to `/en/123/` for better user experience

Screenshots:



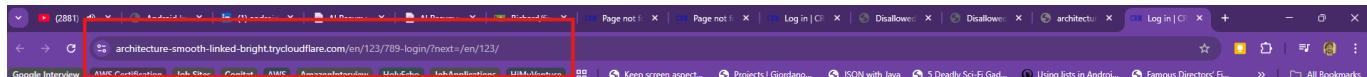
```

C:\Windows\System32\cmd.e X + -
ur use of Tunnels for violations of such terms. If you intend to use Tunnels in production you should use a pre-created named tunnel by following: https://developers.cloudflare.com/cloudflare-one/connections/connect-apps
2025-12-12T11:41:40Z INF Requesting new quick Tunnel on trycloudflare.com...
2025-12-12T11:41:40Z INF +-----+
2025-12-12T11:41:40Z INF | Your quick Tunnel has been created! Visit it at (it may take some time to be reachable): |
2025-12-12T11:41:40Z INF | https://bowl-corporation-instructors-ignored.trycloudflare.com |
2025-12-12T11:41:40Z INF +-----+
2025-12-12T11:41:40Z INF Cannot determine default configuration path. No file [config.yml config.yaml] in [~/.cloudflared ~/.cloudflare-warp ~/cloudflare-warp]
2025-12-12T11:41:40Z INF Version 2025.10.0 (Checksum 07ce5e7ffcc805734797d37061654f5f0152de2a2da241bf52e90822e9b8fc724)
2025-12-12T11:41:40Z INF GOOS: windows, GOVersion: go1.24.6, GoArch: amd64
2025-12-12T11:41:40Z INF Settings: map[ha-connections:1 protocol:quic url:http://localhost:8080]
2025-12-12T11:41:40Z INF cloudflared will not automatically update on Windows systems.
2025-12-12T11:41:40Z INF Generated Connector ID: 7114a5ee-2780-46fa-aa5c-ce25953f97d1
2025-12-12T11:41:40Z INF Initial protocol quic
2025-12-12T11:41:40Z INF ICMP proxy will use 10.30.99.176 as source for IPv4
2025-12-12T11:41:40Z INF ICMP proxy will use fe80::c240:a981:ab7d:9c22 in zone Wi-Fi as source for IPv6
2025-12-12T11:41:40Z ERR Cannot determine default origin certificate path. No file cert.pem in [~/.cloudflared ~/.cloudflare-warp ~/cloudflare-warp]. You need to specify the origin certificate path by specifying the originCert option in the configuration file, or set TUNNEL_ORIGIN_CERT environment variable originCertPath
2025-12-12T11:41:40Z INF cloudflared does not support loading the system root certificate pool on Windows. Please use --origin-ca-pool <PATH> to specify the path to the certificate pool
2025-12-12T11:41:40Z INF ICMP proxy will use 10.30.99.176 as source for IPv4
2025-12-12T11:41:40Z INF Tunnel connection curve preferences: [X25519MLKEM768 CurveP256] connIndex=0 event=0 ip=198.41.200.23
2025-12-12T11:41:40Z INF ICMP proxy will use fe80::c240:a981:ab7d:9c22 in zone Wi-Fi as source for IPv6
2025-12-12T11:41:40Z INF Starting metrics server on 127.0.0.1:20241/metrics
2025-12-12T11:41:40Z INF Registered tunnel connection connIndex=0 connection=e3431a2b-6592-4b5f-90ce-4077e26b8526 event=0 ip=198.41.200.23 location=yyz06 protocol=quic
|
```

PS D:\Conestoga\Course2-DevOps\Sem2\ContainerOrchestration\FinalProject\final-assignment-crm> ./cloudflared tunnel --url http://localhost:8080

```

2025-12-12T12:34:13Z INF Thank you for trying Cloudflare Tunnel. Doing so, without a Cloudflare account, is a quick way to experiment and try it out. However, be aware that these account-less Tunnels have no uptime guarantee, are subject to the Cloudflare Online Services Terms of Use (https://www.cloudflare.com/website-terms/), and Cloudflare reserves the right to investigate your use of Tunnels for violations of such terms. If you intend to use Tunnels in production you should use a pre-created named tunnel by following: https://developers.cloudflare.com/cloudflare-one/connections/connect-apps
2025-12-12T12:34:13Z INF Requesting new quick Tunnel on trycloudflare.com...
2025-12-12T12:34:16Z INF +-----+
2025-12-12T12:34:16Z INF | Your quick Tunnel has been created! Visit it at (it may take some time to be reachable): |
2025-12-12T12:34:16Z INF | https://architecture-smooth-linked-bright.trycloudflare.com |
2025-12-12T12:34:16Z INF +-----+
2025-12-12T12:34:16Z INF Cannot determine default configuration path. No file [config.yml config.yaml] in [~/.cloudflared ~/.cloudflare-warp ~/cloudflare-warp]
2025-12-12T12:34:16Z INF Version 2025.10.0 (Checksum 07ce5e7ffcc805734797d37061654f5f0152de2a2da241bf52e90822e9b8fc724)
2025-12-12T12:34:16Z INF GOOS: windows, GOVersion: go1.24.6, GoArch: amd64
2025-12-12T12:34:16Z INF Settings: map[ha-connections:1 protocol:quic url:http://localhost:8080]
2025-12-12T12:34:16Z INF cloudflared will not automatically update on Windows systems.
2025-12-12T12:34:16Z INF Generated Connector ID: 4d078930-c13e-430c-a3a6-aa7affdd6e51e
2025-12-12T12:34:16Z INF Initial protocol quic
2025-12-12T12:34:16Z INF ICMP proxy will use 10.30.99.176 as source for IPv4
2025-12-12T12:34:16Z INF ICMP proxy will use fe80::c240:a981:ab7d:9c22 in zone Wi-Fi as source for IPv6
2025-12-12T12:34:16Z ERR Cannot determine default origin certificate path. No file cert.pem in [~/.cloudflared ~/.cloudflare-warp ~/cloudflare-warp]. You need to specify the origin certificate path by specifying the originCert option in the configuration file, or set TUNNEL_ORIGIN_CERT environment variable originCertPath
2025-12-12T12:34:16Z INF cloudflared does not support loading the system root certificate pool on Windows. Please use --origin-ca-pool <PATH> to specify the path to the certificate pool
2025-12-12T12:34:16Z INF ICMP proxy will use 10.30.99.176 as source for IPv4
2025-12-12T12:34:16Z INF Tunnel connection curve preferences: [X25519MLKEM768 CurveP256] connIndex=0 event=0 ip=198.41.192.7
2025-12-12T12:34:16Z INF ICMP proxy will use fe80::c240:a981:ab7d:9c22 in zone Wi-Fi as source for IPv6
2025-12-12T12:34:16Z INF Starting metrics server on 127.0.0.1:20241/metrics
2025-12-12T12:34:17Z INF Registered tunnel connection connIndex=0 connection=c725fcbb-c3f0-4002-94df-866d77dcba69 event=0 ip=198.41.192.7 location=yyz03 protocol=quic
|
```



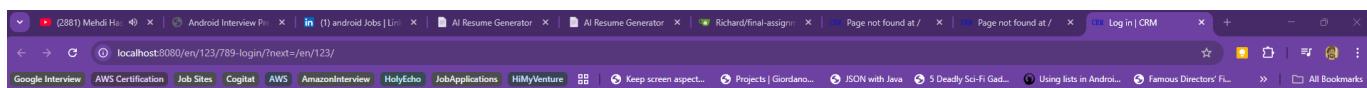
Django-CRM

Username:

Password:

Log in

Django-CRM. Copyright (c) 2025



Django-CRM

Username:

Password:

Log in

Django-CRM. Copyright (c) 2025

Step 5: Kubernetes Deployment with Helm

Tasks Completed:

- Updated docker-compose.yml to use Gitea registry image
([gitea.devsecmindset.dev/nikhil/final-assignment-crm:latest](#))
- Installed Kompose tool
- Converted docker-compose.yml to Kubernetes manifests using Kompose
- Fixed Kubernetes resource naming (replaced underscores with hyphens for DNS compliance)
- Created Helm chart structure with Chart.yaml and values.yaml
- Organized Kompose-generated manifests into [crm-helm-chart/templates/](#)

- Modified up.yaml to use Helm for Kubernetes deployment
- Modified down.yaml to use Helm for Kubernetes cleanup
- Installed Helm on local machine
- Successfully deployed application to Kubernetes cluster using `helm install crm-app ./crm-helm-chart`

Deployment Status:

- Database (MySQL) pod: Running
- Adminer pod: Running
- CRM application pod: Deployed (requires image rebuild with runtime dependencies)

Screenshots:

The screenshot shows the VS Code interface with the terminal tab active. The command `git commit -m "Converted docker compose to kubernetes using Kompose"` has been run, and its output is displayed. The output includes several warning messages about CR LF characters being replaced by LF, and it ends with a success message indicating 12 files changed.

```

CRLF the next time Git touches it
warning: in the working copy of 'crm-service.yaml', LF will be replaced by CR
LF the next time Git touches it
warning: in the working copy of 'db-deployment.yaml', LF will be replaced by CR
LF the next time Git touches it
warning: in the working copy of 'db-service.yaml', LF will be replaced by CR
LF the next time Git touches it
warning: in the working copy of 'mysql-db-data-persistentvolumeclaim.yaml', L
F will be replaced by CR LF the next time Git touches it
PS D:\Conestoga\Course2-DevOps\Sem2\ContainerOrchestration\FinalProject\final
● -assignment-crm git commit -m "Converted docker compose to kubernetes using
Kompose"
[main 887038] Converted docker compose to kubernetes using Kompose
12 files changed, 169 insertions(+), 1 deletion(-)
create mode 100644 adminer-container-deployment.yaml
create mode 100644 adminer-container-service.yaml
create mode 100644 crm-deployment.yaml
create mode 100644 crm-service.yaml
create mode 100644 db-deployment.yaml
create mode 100644 db-service.yaml
create mode 100644 kompose.exe
create mode 100644 mysql-db-data-persistentvolumeclaim.yaml
create mode 100644 screenshots\Task3.4.PushedWorkflowRunAndPushedImageToCR.p
ng
create mode 100644 screenshots\Task3.5.PushedDockerImageInContainerRegistry.
png
create mode 100644 screenshots\Task3.6.ImageDetails.png
○ PS D:\Conestoga\Course2-DevOps\Sem2\ContainerOrchestration\FinalProject\final
● -assignment-crm []

```

The screenshot shows the VS Code interface with the terminal tab active. It displays the output of several Helm commands related to the 'crm-helm-chart' application. These include running `helm install`, `helm uninstall`, and `helm status`. The status command shows the application was last deployed on Friday, December 12, at 06:02:59, 2025, in the default namespace, with a deployed status and revision 1.

```

PS D:\Conestoga\Course2-DevOps\Sem2\ContainerOrchestration\FinalProject\final
● -assignment-crm ./helm install crm-app ./crm-helm-chart
Error: INSTALLATION FAILED: cannot re-use a name that is still in use
PS D:\Conestoga\Course2-DevOps\Sem2\ContainerOrchestration\FinalProject\final
● -assignment-crm ./helm.uninstall crm-app
Error: unknown command "uninstall" for "helm"
Did you mean this?
  uninstall
Run 'helm --help' for usage.
PS D:\Conestoga\Course2-DevOps\Sem2\ContainerOrchestration\FinalProject\final
● -assignment-crm ./helm.uninstall crm-app
release "crm-app" uninstalled
PS D:\Conestoga\Course2-DevOps\Sem2\ContainerOrchestration\FinalProject\final
● -assignment-crm ./helm install crm-app ./crm-helm-chart
NAME: crm-app
LAST DEPLOYED: Fri Dec 12 06:02:59 2025
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
○ PS D:\Conestoga\Course2-DevOps\Sem2\ContainerOrchestration\FinalProject\final
● -assignment-crm []

```

```
TEST SUITE: None
PS D:\Conestoga\Course2-DevOps\Sem2\ContainerOrchestration\FinalProject\final
-assignment-crm> kubectl get pods
NAME                                     READY   STATUS    RESTARTS
RTS          AGE
adminer-container-c484cfb6-jgwgd        1/1     Running   0
      2m4s
cloudflared-7897557cc7-d9245         0/1     CreateContainerConfigError   0
      2m4s
crm-6b765d9587-czhst                0/1     CrashLoopBackOff       4 (25
s ago)   2m4s
db-56b6988977-6lmr9                  1/1     Running   0
      2m4s
PS D:\Conestoga\Course2-DevOps\Sem2\ContainerOrchestration\FinalProject\final
-assignment-crm>
```

Project URLs

- **Gitea Repository:** <https://gitea.devsecmindset.dev/Richard/final-assignment-crm.git>
- **Public GitHub:** <https://github.com/Richard-Conestoga/container-asssignment3.git>
- **Gitea Instance:** <https://gitea.devsecmindset.dev/>
- **Container Registry:** <https://gitea.devsecmindset.dev/Nikhil/-/packages/container/final-assignment-crm>

Technologies Used

- Docker & Docker Compose
- Kubernetes (K3s)
- Helm
- Kompose
- Gitea & Gitea Actions
- Gitea Container Registry
- TrueNAS (NFS Storage)
- Cloudflared Tunnel
- Django CRM Application
- MySQL Database
- Adminer (Database Management)

Team Contributions

Richard:

- Step 1: Gitea Repository Setup
- Step 2: NFS Storage Configuration

Nikhil:

- Step 3: CI/CD Pipeline with Gitea Actions
- Step 4: Cloudflare Tunnel Configuration
- Step 5: Kubernetes Deployment with Helm