

Application Deployment Using Containers (Podman)

Student Details

Name: Vivek

Registration Number: 12319700

Platform: Red Hat Enterprise Linux (RHEL)

1. Integrity Requirement

Before starting the project, set the shell prompt identifier to your registration number:

```
export PS1="12319700@[...]\$ "
```

To make it persistent:

```
echo 'export PS1="12319700@[...]\$ "' >> ~/.bashrc
source ~/.bashrc
```

2. Project Scenario

You are required to deploy an internal web application using **Podman** containers. The tasks include: - Creating and running an **nginx** container. - Serving a **custom web page** stored on the host. - Mounting the host directory to the container. - Ensuring the container auto-starts on boot using **systemd unit files**. - Verifying **data persistence** after container removal and reboot.

3. Project Directory Structure

```
project-container-deployment/
|
├── html/
│   └── index.html
|
└── systemd/
    └── container-nginx.service
```

4. Steps to Complete the Project

Step 1: Install Podman on RHEL

```
sudo dnf install -y podman  
podman --version
```

Step 2: Create a Custom Web Page

```
mkdir -p ~/project-container-deployment/html  
nano ~/project-container-deployment/html/index.html
```

Add the following content:

```
<h1>Welcome to Vivek's Container Project (12319700)</h1>
```

Step 3: Run Nginx Container with Volume Mount

```
podman run -d  
--name nginx-custom  
-p 8080:80  
-v ~/project-container-deployment/html:/usr/share/nginx/html:Z  
docker.io/library/nginx:latest
```

Check container status:

```
podman ps
```

Open in browser:

```
http://localhost:8080
```

Step 4: Generate systemd Unit File

```
podman generate systemd --name nginx-custom --files --new
```

This creates a file like:

```
container-nginx-custom.service
```

Move it to systemd directory:

```
sudo mv container-nginx-custom.service /etc/systemd/system/
```

Enable and start on boot:

```
sudo systemctl daemon-reload  
sudo systemctl enable --now container-nginx-custom.service
```

Check status:

```
systemctl status container-nginx-custom.service
```

Step 5: Verify Persistence After Reboot

Reboot the system:

```
sudo reboot
```

After reboot, check:

```
podman ps  
systemctl status container-nginx-custom.service
```

Now remove the container:

```
podman rm -f nginx-custom
```

Access the site again:

```
http://localhost:8080
```

The webpage still works because systemd recreated the container.

5. GitHub Upload Instructions

```
git init  
git add .  
git commit -m "Container Deployment Project - Vivek 12319700"  
git branch -M main  
git remote add origin <your-repository-url>  
git push -u origin main
```

6. Conclusion

This project demonstrates application deployment using **Podman**, including containerization, host-to-container volume mapping, automation using **systemd**, and verification of persistence. It provides a secure and manageable way to deploy internal web applications on RHEL.