(Senior Academy - IT training center)

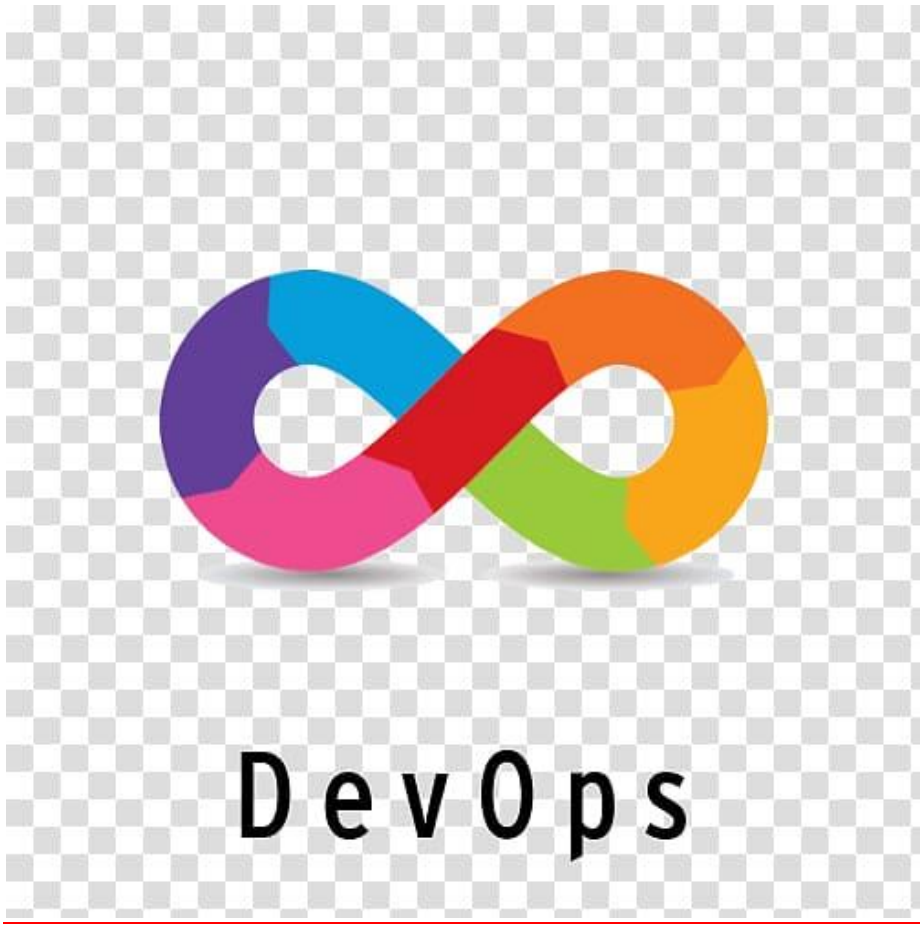
The Place You Can Be A Senior

www.seniorsteps.net
https://www.facebook.com/seniorsteps.it
contact us: 0224153419 - 01090873748

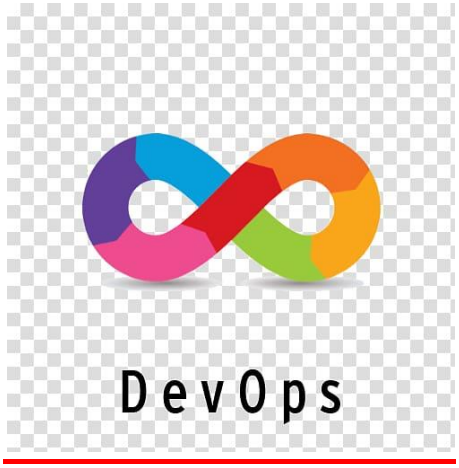
فرع مدينة نصر 1 : عمارة 4 – شارع محمد توفيق دياب – عباس العقاد - مدينة نصر – الدوران 1

Senior Academy - IT training center
www.seniorsteps.net
contact us: 0224153419 - 01090873748
عمارة 4 – شارع محمد توفيق دياب – عباس العقاد - مدينة نصر – الدورال 1

DevOps Engineer Diploma



DevOps Engineer Diploma



Kubernetes Labs

Lab 03

Managing Kubernetes Deployments, Services, and Resource Quotas in Dedicated Namespaces

Lab Objectives

- **Working with Namespaces**
- **Deploying Applications**
- **Service Management**
- **Scaling Deployments**
- **Managing Resource Quotas**
- **Verifying Resources:**
- **Environment Variables Management**

Deploy an Nginx application in a dedicated namespace, expose it using a Service, apply scaling, and configure a ResourceQuota to control resource usage.

---

1. Create a Namespace

- Define a new namespace called **web-app**.
- Verify that the namespace was created successfully.

---

2. Create a Deployment

- Define a Deployment for the Nginx application inside the **web-app** namespace.
- Set the number of replicas to **2**.
- Add resource requests and limits for **CPU** and **memory**.
- Deploy the YAML file and check the running pods.

---

3. Create a Service

- Expose the Nginx Deployment internally using a **Service** of type **NodePort**.
- Ensure it routes traffic to port **80** of the Nginx container.
- List the services in the namespace.

---

4. Test Connectivity

- Identify the **Node IP** and the assigned **NodePort**.
- Test access using **curl** or a browser.

---

5. Scale the Deployment

- Increase the number of replicas for the Deployment.
- Confirm that additional pods are created successfully.

---

6. Create a ResourceQuota

- Define a quota for the **web-app** namespace that limits:

  - The total number of **pods**.
  - **CPU** and **memory** requests.
  - **CPU** and **memory** limits.
- Apply the quota and describe it.

---

7. Test the Quota Enforcement

- Try to scale the Deployment beyond the allowed limits.
- Observe how Kubernetes prevents exceeding the quota.

---

8. Verify All Resources

- List all objects (Pods, Deployments, Services, Quotas) within the **web-app** namespace.
- Review their current state and confirm compliance with the quota.

---

9. Create a Namespace for db-app

- Create a namespace named **db-app**.
- Verify that it appears in the namespace list.

---

10. Create a MySQL Deployment

- Define a Deployment named **mysql-deployment** in the **db-app** namespace.
- Use the image **mysql:latest**.
- Set the container port to **3306**.
- Start with no environment variables.

---

11. Add Environment Variables Manually

- Use the command `kubectl set env` to manually add environment variables to the Deployment.
- Variables to set:
  - **MYSQL_ROOT_PASSWORD**
  - **MYSQL_DATABASE**
- Confirm that the Deployment was updated successfully.

You are Welcome