(Senior Academy - IT training center)

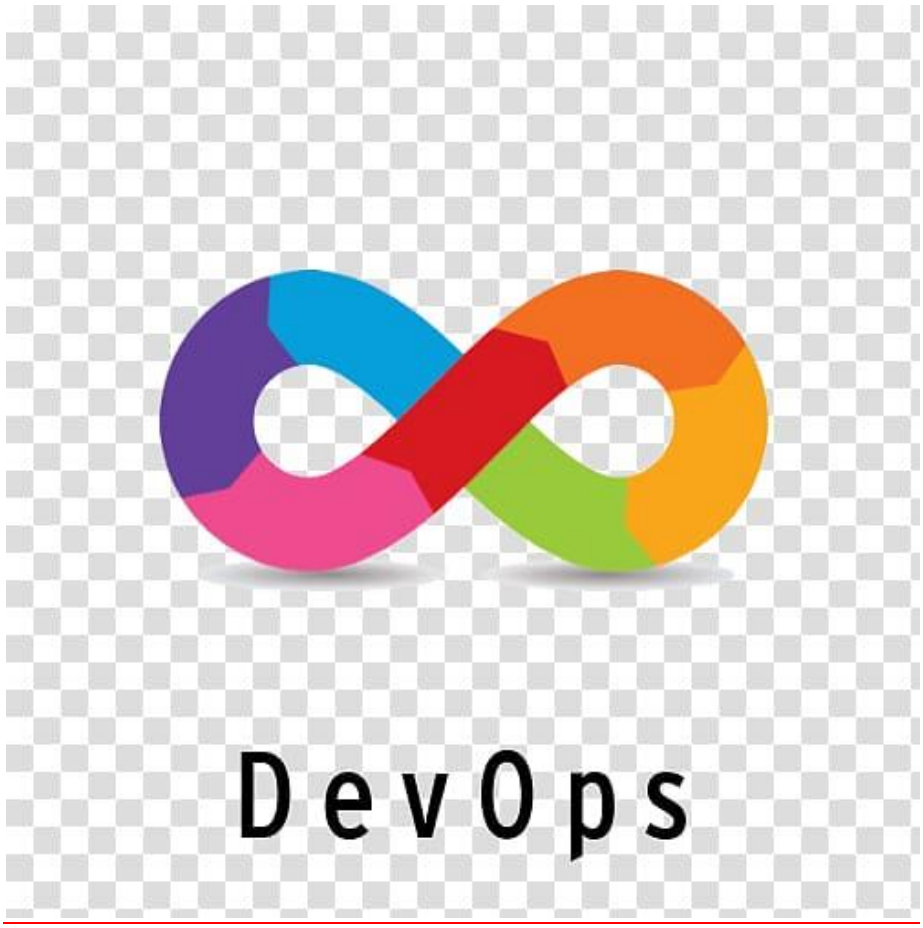
The Place You Can Be A Senior

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

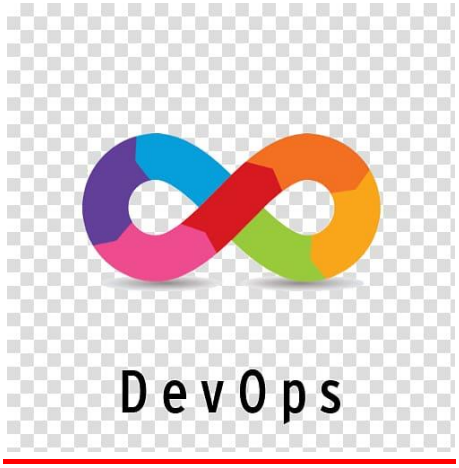
فرع مدينة نصر 1 : عمارة 4 – شارع محمد توفيق دياب – عباس العقاد - مدينة نصر – الدوران 1

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



OpenShift Labs

Lab 15

OpenShift Job Execution with Custom ServiceAccount and Elevated Privileges

Lab Objectives

- **Creating and Managing OpenShift Projects**
- **Using Custom ServiceAccounts for Job Execution**
- **Assigning Elevated Privileges via anyuid Security Context Constraint (SCC)**
- **Running One-Time Jobs with Root Access in OpenShift**
- **Verifying Job Execution and Permissions**
- **Cleaning Up OpenShift Resources After Job Completion**

Running a One-Time Job with Elevated Privileges in OpenShift

Objective

Create and manage a **one-time Job** in **OpenShift** that runs with elevated privileges.
The Job should use a **custom ServiceAccount** and execute a command requiring **root permissions**.

Scenario

As a **DevOps Engineer**, your team needs to perform a one-time system check Job inside OpenShift.
This Job must execute a simple system command that requires **root privileges** inside the container.

By default, all containers in OpenShift run with a **restricted Security Context Constraint (SCC)**, which prevents root execution.
To achieve this, you must:

1. Create a **custom ServiceAccount**.
2. Grant it higher privileges using the **anyuid SCC**.
3. Run a **Job** using that ServiceAccount to perform the system check.

Lab Tasks

Step 1 – Create a New Project

- Create a separate **OpenShift project** for this exercise.

Step 2 – Create a ServiceAccount

- Inside the project, create a new **ServiceAccount** that the Job will use.

Step 3 – Grant Elevated Privileges

- Assign the **anyuid SCC** to the new ServiceAccount so it can run containers as the **root user**.

Step 4 – Create a Job

- Create a **one-time Job** that runs a short system command (for example: whoami).
- Ensure the Job uses the **ServiceAccount** you created earlier.

Step 5 – Verify the Job

Check that:

- The **Job** completes successfully.
- The **Pod** used the correct ServiceAccount.
- The **container** runs with the expected permissions (root access).

Step 6 – Cleanup

- Delete all resources created during this lab once you verify the results.

You are Welcome