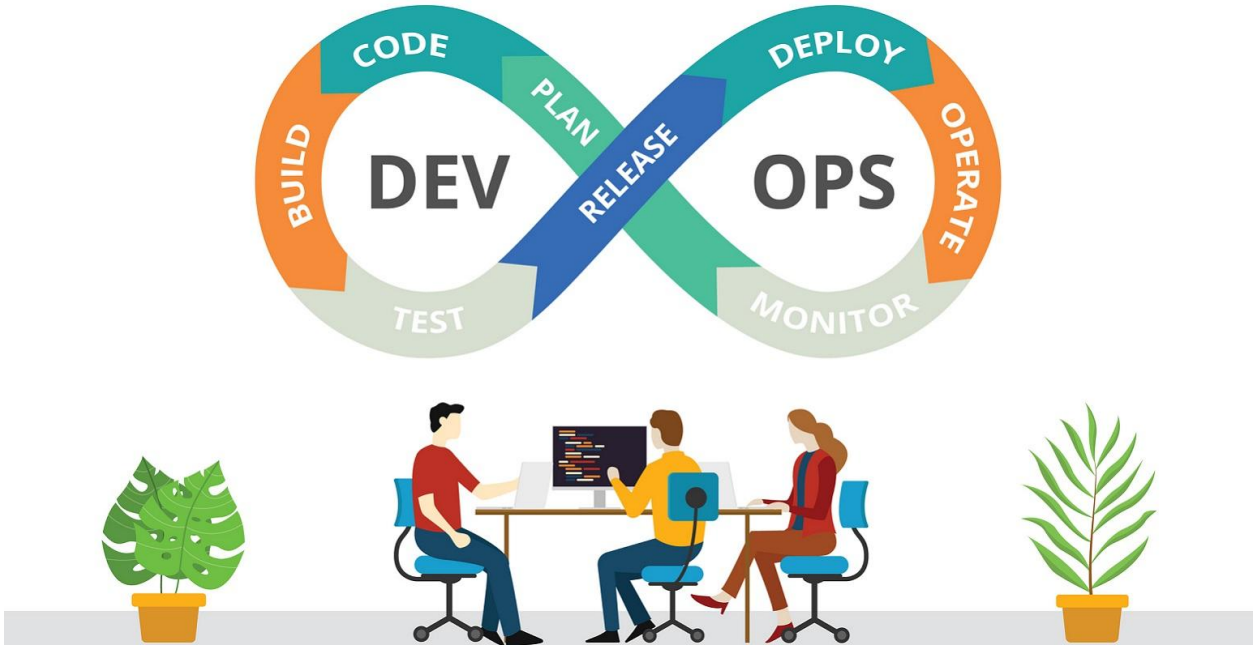

Secure DevOps - Project 1

Overview

In this project, you will build, configure, deploy and maintain a Saleor application suite on a Linux server using Docker Compose. Saleor is an open source e-commerce platform consisting of a Python API backend, a React (Javascript) front end and dashboard, a database backend (PostgreSQL) with an in-memory cache (Redis) and other supporting services. You will need to apply the knowledge learnt so far in this Unit to tackle the architectural complexity of Saleor.

This project is designed to test your skills on:

- Typical client-server application architecture
- Working with Linux servers (Ubuntu 20.04)
- Docker network, volumes, containers and images
- Docker Compose orchestration

What you need to complete this project

1. A Linux server running Ubuntu 20.04 with the latest Docker and Docker-Compose installed. The server should have at least 2 cores and 4GB RAM usable and 10GB storage space. Recommended to use Ubuntu 20.04 server edition.
2. Internet access

Basic understanding of the languages (Python and React) and tools (PostgreSQL, Redis) will be useful but not a prerequisite to successfully complete this project.

Task

1. Read the basic introduction of the following projects and make sure you understand what they are and their purposes.
    - Saleor API https://github.com/saleor/saleor
    - Saleor storefront https://github.com/saleor/react-storefront
    - Saleor dashboard https://github.com/saleor/saleor-dashboard
    - Saleor platform https://github.com/saleor/saleor-platform (Hint: This repo contains the Docker Compose artifacts to configure, build and run components of a Saleor stack. The repo uses Git submodules to reference the three repos above).
2. Create a personal account on Github.com and then fork the Saleor platform repo.

3. Follow the instructions in the Saleor platform to run a Saleor stack with example data.
4. Change the Compose file so that the React Storefront runs at port 3001, and the Saleor Dashboard runs at port 9001. Start the stack and confirm all services start successfully:
    - Saleor React Storefront - http://<Your-Linux-Server-IP>:3001
    - Saleor Dashboard - http://<Your-Linux-Server-IP>:9001
5. Commit the changes and push to the forked repo with the tag `isec6000-assignment1`.

Submission

1. Submission 1 (60 points) Write a Word/PDF document containing:
    - Link to your forked repo.
        - The repo **must be public** so that your work can be reviewed and marked.
        - The repo must contain the tag `isec6000-assignment1`.
        - It must be possible to run a Saleor stack following the Readme instructions.
2. Submission 2 (40 points): An architecture diagram outlining the core functionalities of each Saleor service and how they interact with each other in a stack. You are free to choose a diagram style and tools to help you draw the diagram. As long as the depicted information is clear and concise, the choice of tools is irrelevant. The architecture diagram must at least show
    - The volumes and networks used by different Saleor services.
    - How different Saleor services communicate with each other.
    - Exposed ports of each container