Assignment 2

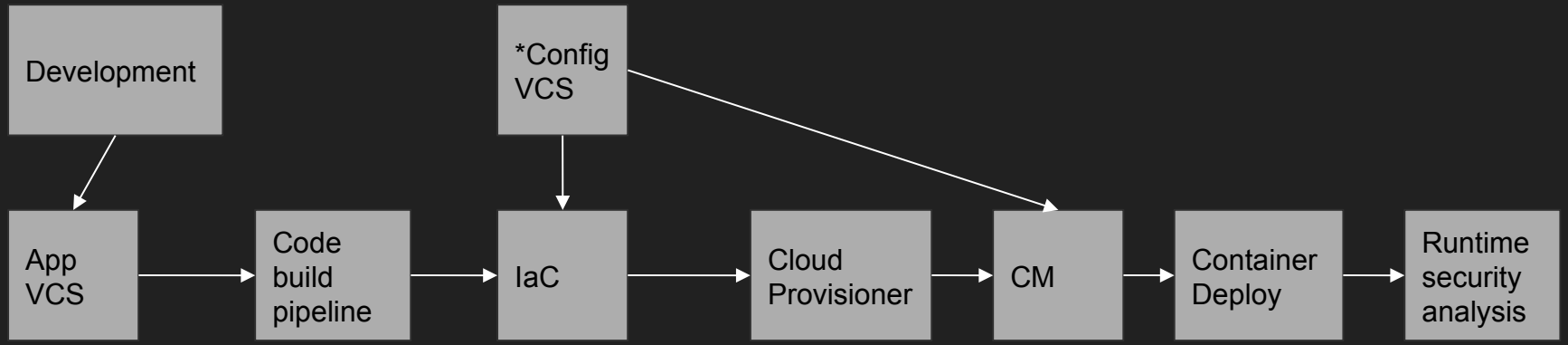
Justin Le
Kieran Loewen
Octavio Colantonio

Summary

The goal of this assignment is to use OSS tools to develop a Devops CI/CD pipeline. We will examine each of the phases of the and explain some of the tooling options available.

CI/CD flow diagram

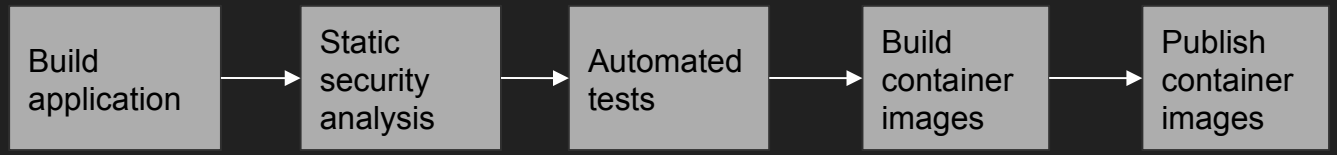
Release flow



*Configuration may or may not be in separate VCS

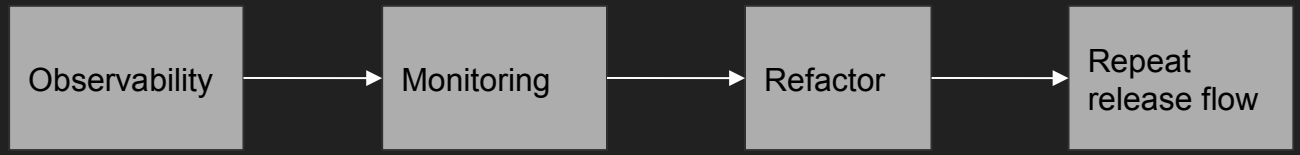
Code Build pipeline

Detailed overview of code build using automated build tool



Maintenance flow

Completed after successful deployment



Version Control

Version control is a very important part of the pipeline. It is the first step and very important process as without it, it would be hard to keep track of code changes and debugging would be a nightmare without it.

Github is the most popular tool used for version control and utilizes Git, a version control system, to manage code changes.

Infrastructure as Code

Using Infrastructure as code in a DevOps pipeline involves automating and the provisioning and management of infrastructure resources through code.

Popular tools used in creating IaC are Terraform, Microsoft Azure and AWS.

The Terraform tool is the most widely used tool and is mainly used to codify your infrastructure configuration, automate provisioning, manage dependencies, and maintain consistency across different environments and cloud providers. It provides a robust and flexible approach to infrastructure management, enabling IaC principals in your DevOps pipeline.

Configuration Management

The management of the configuration of all environments for an application, it ensures the consistency of systems and software using IaC to accelerate configuration changes and avoid configuration drift.

Ansible is a common automation tool used to facilitate CM, and consists of a control node and managed nodes. The control node sends and executes modules on the managed nodes to configure the node.

Continuous Integration, Automated build and testing

Continuous integration with automated test execution and trends has changed the way companies can build and test their software development practices. By using CI tool like jenkins coupled with automated testing (unit, integration) tool like selenium, we can build and test our software upon every changes committed to our version control.

Common tools

- Continuous Integration (Jenkins, CircleCI)
- Automated builds (gradle)
- Automated testing (selenium)

Containerization and Orchestration

We will adopt microservices architecture in the development of our pipeline. Containerization is a deployment technique that can help maintain the independence of a microservice. Lastly, Orchestration technologies help manage and maintain massive container infrastructures and application that run them on.

Some of the notable tools are the followings

- Containerization (Docker, Podman)
- Container Orchestration (Kubernetes, Docker Swarm, Red-hat Openshift)

Security

We would like to adopt shift-left testing approach means baking security into your applications at the very beginning, instead of waiting until the final stages of the delivery chain. Security toolings will be automatically built into our code development.

Common tools

- Static code analysis (snyk)
- Dynamic analysis (owasp zap)

Monitoring and Observability

Observability is the extent to which you can understand the internal state or condition of a complex system based only on knowledge of its external outputs which provides key metrics that can be monitored to understand the state of the application and infrastructure. This detailed information can be used to guide decisions based on how a system is used and where there can be performance gains.

Common tools

- Prometheus for data collection and alerting
- Grafana for querying and visualization