



# Lesson 1: Introduction to Cloud Native Fundamentals

Introduction to Cloud Native

CNCF and Cloud Native Tooling

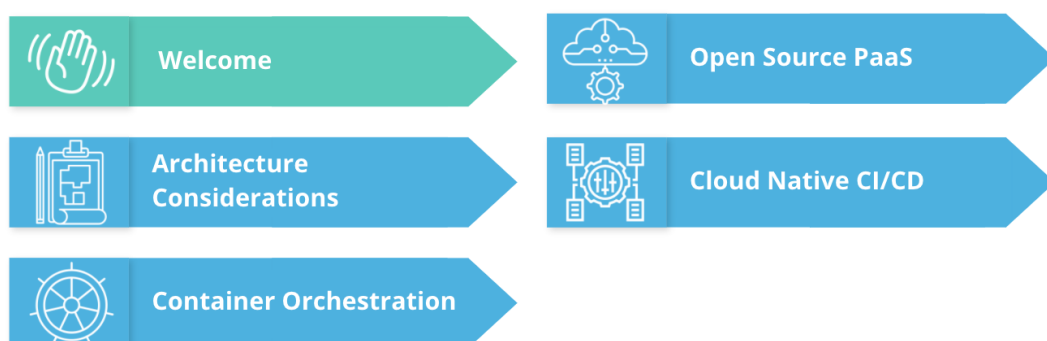
- Instructor: Katie Gamanji

## ▼ Prerequisites

- ✓ ~~Web Application Development using Python (Flask)~~
- ✓ ~~CLI~~
- ✓ ~~Git Commands~~
- ✓ ~~DockerHub Account~~

## ▼ Course Outline

- Containerize an application
- Release it to a Kubernetes cluster using an automated CI/CD pipeline



## Introduction to Cloud Native

- A set of practices that empowers an organization to build and manage applications at scale
- One can use private, hybrid, or public cloud providers and their services

- Essentially, the goal is to ensure an application that is fast and agile in terms of integrating customer feedback and adapting to the surrounding technology ecosystem
- Containers are used to run a single application with all the required dependencies. They are easy to manage, deploy, and fast to recover

## CNCF and Cloud Native Tooling

- With the increased use of containers, it was essential to bring in container management tools (Docker Swarm, Apache Mesos, Kubernetes) as well
- Kubernetes is a container orchestrator that automates the configuration, management, and scalability of an application
- It was derived from Google's Borg (in 2014), and is maintained by CNCF (Cloud Native Computing Foundation)
- Kubernetes has the following functionalities:
  - Runtime for application execution environment
  - Networking for application connectivity
  - Storage for application resources
  - Service Mesh for granular control of traffic within a cluster
  - Logs and metrics to construct the observability stack
  - Tracing for building the full request journey

### ▼ Stakeholders

## Summary

An engineering team can use cloud-native tooling to enable quick delivery of **value to customers** and **easily extend** to new features and technologies. These are the main reasons why an organization needs to adopt cloud-native technologies. However, when trialing cloud-native tooling, there are two main perspectives to address: business and technical stakeholders.

From a **business perspective**, the adoption of cloud-native tooling represents:

- Agility - perform strategic transformations
- Growth - quickly iterate on customer feedback
- Service availability - ensures the product is available to customers 24/7

From a **technical perspective**, the adoption of cloud-native tooling represents:

- Automation - release a service without human intervention
- Orchestration - introduce a container orchestrator to manage thousands of services with minimal effort
- Observability - ability to independently troubleshoot and debug each component

## ▼ Tools

- ✓ Python
- ✓ Git
- ✓ Docker
- ✓ Vagrant
- ✓ Virtual Box