SUSE

Cloud Native Fundamentals Scholarship Program

SUSE
SUSE Cloud Native Foundations Scholarship Program
Learn to apply modern cloud native application development and DevOps principles and accelerate your career growth.

Prepared by Sani Kamal

Lesson 1

Introduction to Cloud Native Fundamentals

CONCEPTS

1. Meet Your Instructor

2. Prerequisites

3. Course Outline

4. Introduction to Cloud-Native

5. CNCF and Cloud-Native Tooling

6. Stakeholders

7. Tools, Environment & Dependencies

8. Recap

9. Good Luck!

Meet Your Instructor

Summary

Welcome to the Microservice Fundamentals course! My name is Katie Gamanji, and I will be your instructor.

In this course, we will learn how to construct a CI/CD pipeline that will containerize an application and deployed it to a Kubernetes cluster.

I have been working as a cloud platform engineer for many years. In these roles, I built and maintained platforms that would enable developers to deploy and host their applications. This revolves around a high degree of automation, upskilling, and constant iterations on user feedback.

In my current role, I am working for CNCF (Cloud Native Computing Foundation) as Ecosystem Advocate, representing and growing the End User Community. Also, I am a member of the advisory board for Keptn, which currently is a sandbox CNCF project.

Prerequisites

Summary

For this course, students should be comfortable with:

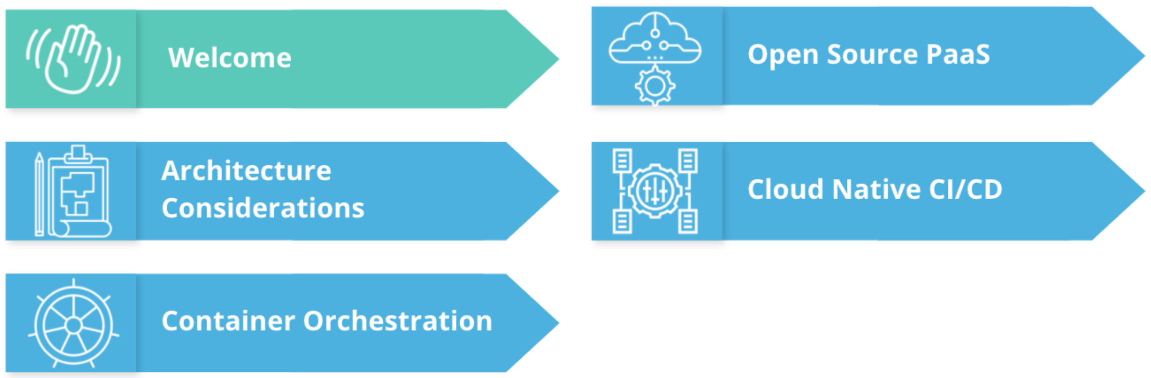
- **web application development with Python**
- using the **CLI** or command-line interface
- using **git commands**
- creating a **DockerHub account**

These prerequisites will be used throughout the course. Make sure you have them installed to closely follow the demos and exercise solutions.

Course Outline

Summary

Throughout the course, we will walk through a realistic example of applying good development practices and containerizing an application, before it's released to a Kubernetes cluster using an automated CI/CD pipeline.



Microservice Fundamentals course outline

This **course** has the following lessons:

- Welcome
- Architecture Considerations
- Container Orchestration
- Open Source PaaS
- Cloud Native CI/CD

In the first lesson, we will cover:

- Introduction to Cloud Native
- CNCF and Cloud Native tooling
- Stakeholders
- Tools, Environment & Dependencies

Introduction to Cloud-Native

Summary

Cloud-native refers to the set of practices that empowers an organization to **build and manage applications at scale**. They can achieve this goal by using private, hybrid, or public cloud providers. In addition to scale, an organization needs to be agile in integrating customer feedback and adapting to the surrounding technology ecosystem.

Containers are closely associated with cloud-native terminology. Containers are used to run a single application with all required dependencies. The main characteristics of containers are easy to manage, deploy, and fast to recover. As such, often, a microservice-based architecture is chosen in tandem with cloud-native tooling. Microservices are used to manage and configure a collection of small, independent services that can be easily packaged and executed within a container.

CNCF and Cloud-Native Tooling

Summary

Kubernetes had its first initial release in 2014 and it derives from Borg, a Google open-source container orchestrator. Currently, Kubernetes is part of **CNCF** or **Cloud Native Computing Foundation**. CNCF was founded in 2015, and it provides a **vendor-neutral home to open-source projects** such as Kubernetes, Prometheus, ETCD, Envoy, and many more.

Overall, Kubernetes is a container orchestrator that is capable to solutionize the integration of the following functionalities:

- Runtime
- Networking
- Storage
- Service Mesh
- Logs and metrics
- Tracing

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, Environment & Dependencies

Summary

For this course, make sure you have the listed prerequisites installed. These tools will be used throughout the course, so make sure you install them to follow the demo and exercise solutions closely.

Tools, Environment & Dependencies Checklist

Make sure you have the following dependencies installed:

Task List

Recap

Summary

In the first lesson, we went through:

- Introduction to Cloud Native
- CNCF and Cloud Native tooling
- Stakeholders
- Tools, Environment & Dependencies

Good Luck

Summary

Every organization aims to succeed! This is represented by providing customer value and the ability to be responsive to the surrounding ecosystem. Coincidentally, this is closely correlated with technological innovation, which translates into the adoption of containers, automation, and usage of cloud-native tooling.

By completing this course you will be equipped to lead the adoption of cloud-native tooling and principles within an organization.

Hope you will enjoy the course and **GOOD LUCK**!