Lesson 1 - Intro to Cloud Native Fundamentals

Prerequisites For The course:

- **web application development with Python :**

  https://www.udacity.com/course/introduction-to-python--ud1110

  https://www.udacity.com/course/designing-restful-apis--ud388

  https://youtu.be/Qr4QMBUPxWo

- **using the CLI or command-line interface :**

  https://blog.testproject.io/2021/03/30/a-beginners-guide-to-command-line-interface-cli/

  https://www.w3schools.com/whatis/whatis_cli.asp

- **using git commands :**

  https://www.youtube.com/watch?v=RGOj5yH7evk

- **creating a DockerHub account :**
  https://hub.docker.com/

Tools Required:

- **Python:** https://www.python.org/downloads/
- **Git:** https://git-scm.com/downloads
- **Docker:** https://docs.docker.com/get-docker/
- **Vagrant:** https://www.vagrantup.com/downloads
- **Virtual Box:** https://www.virtualbox.org/wiki/Downloads

Containers : 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.
https://www.youtube.com/watch?v=A0g7I4A6GN4

Microservices : Microservices are used to manage and configure a collection of small, independent services that can be easily packaged and executed within a container.
https://youtu.be/gfWr2_H39N0

Cloud-Native : Cloud-native refers to the set of practices that empowers an organization to build and manage applications at scale
https://www.oracle.com/cloud/cloud-native/what-is-cloud-native/

Container Orchestrator : A container orchestrator is simply a tool to manage the containers. **Kubernetes** is a container orchestrator. It is capable to solutionize the integration of the following functionalities:

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

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