UGANDA TECHNOLOGY AND MANAGEMENT UNIVERSITY

UTAMU

CONTAINERIZATION TASK

CSC 2101 OPERATING SYSTEMS

SEP23BCS/3472U/F NABATANZI GORRET

GitHub Repository Link: https://github.com/gorretgolden/OS-Containerization-Task/tree/master

CONTAINERIZATION ASSIGNMENT

Explain the concept of containerization.

Containerization is a software deployment process that bundles an application's code with all the files and libraries it needs to run on any infrastructure. It is a lightweight virtualization technique that allows applications to run in isolated environments called containers.

What are the key use cases of containerization

The following are some use cases of containerization;

- **Cloud migration:** This is a software strategy that involves encapsulating legacy applications in containers and deploying them in a cloud computing environment. Organizations can modernize their applications without rewriting the entire software code.
- **Adoption of micro-service architecture:** A modern cloud application consists of multiple microservices. For example, a video streaming application might have microservices for data processing, user tracking, billing, and personalization. Containerization provides the software tool to pack microservices as deployable programs on different platforms.
- **IoT devices:** Internet of Things (IoT) devices contain limited computing resources, making manual software updating a complex process. Containerization allows developers to deploy and update applications across IoT devices easily.
- **CI/CD Pipelines:** Containers make CI/CD very easy by providing consistent environments from development to production.
- **Cloud-Native Applications:** Containers are the underlying infrastructure for cloud-native applications, while at the same time being portable across various cloud providers.
- **Dev/Test Environments:** Containers make it easier to set up consistent development and testing environments, significantly reducing the problems associated with "it works on my machine."

Explore different containerization technologies such as Docker, Podman, and Kubernetes.

- **Docker (Docker Engine),** is a popular open-source container runtime that allows software developers to build, deploy, and test containerized applications on various platforms. Docker containers are self-contained packages of applications and related files that are created with the Docker framework.
- **Podman:** Is a container management tool similar to Docker but daemonless (does not require a background service). It is more secure than Docker since it runs containers rootless by default.It is compatible with Docker images and uses OCI (Open Container Initiative) standards and it does not have a built-in orchestration system like Docker Swarm or Kubernetes.
- **Kubernetes:** Is a popular open-source container orchestrator that software developers use to deploy, scale, and manage a vast number of microservices. It has a declarative model that makes automating containers easier. The declarative model ensures that Kubernetes takes the appropriate action to fulfil the requirements based on the configuration files.

How containerization technologies (containerization) differ(s) from Virtual Machines

A virtual machine (VM) is a digital copy of the host machine's physical hardware and operating system. A host machine might have several VMs sharing its CPU, storage, and memory. A hypervisor, which is software that monitors VMs, allocates computing resources to all the VMs regardless of whether the applications use them.

Key differences

Architecture: VMs run a full operating system (OS) on virtualized hardware, including a hypervisor layer, whereas containers share the host OS kernel and isolate applications at the process level.

Resource Utilization: Containers are more lightweight, as they don't require a separate OS per instance, leading to faster startup times and reduced resource consumption.

Isolation: VMs provide strong isolation with separate kernels, which can be beneficial for running different operating systems on the same hardware. Containers, sharing the same kernel, offer lighter isolation, which is efficient but may pose security considerations in multi-tenant environments.

REFEREENCES

Amazon Web Services, Inc. (n.d.). *What is containerization?* AWS. Retrieved January 22, 2025, from https://aws.amazon.com/what-is/containerization/

GeeksforGeeks. (n.d.). What is a container? GeeksforGeeks. Retrieved January 25, 2025, from https://www.geeksforgeeks.org/what-is-a-container/