

## Project Goal

- The goal of this project is to explore the use of containerization in the deployment of 5G networks and develop enhancements related to container health monitoring, automatic recovery and distributed loads

## Tools used for the project: -

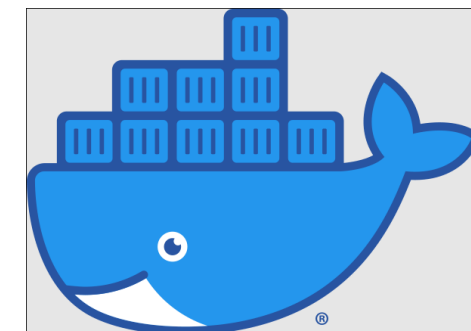
- Ansible



- Zabbix



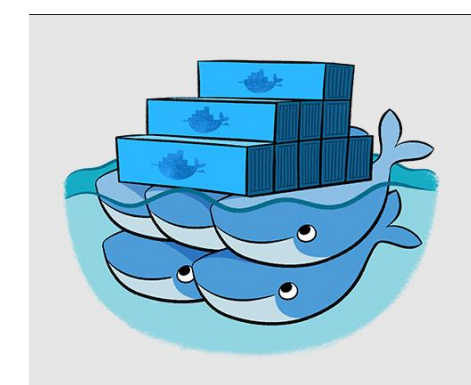
- Docker



- Docker – Compose



- Docker Swarm



## Ansible

- Ansible is a radically simple IT automation tool that simplifies the deployment of your apps and systems.
- It support configuration management with examples as below.
  - Configuration of servers
  - Application deployment
  - Continuous testing of already install application
  - Provisioning
  - Orchestration
  - Automation of tasks

## Why Ansible?

- It is an open-source application that is available for free.
- Agentless - There is no requirement for agent installation or management.
- Based on Python/YAML
- System configuration management that is highly versatile.
- Myriad of ready-to-use system management components
- If necessary, custom modules can be introduced.
- In the event of an error, the configuration can be rolled back.
- Straightforward and easy to read
- Self documenting

## ZABBIX

- Zabbix is an open-source software tool to monitor IT infrastructure such as networks, servers, virtual machines, and cloud services. Zabbix collects and displays metrics.
- Zabbix offers excellent reporting and data visualization features based on the stored data
- Zabbix is also able to perform actions such as sending notifications or perform recovery actions based on the data it collects

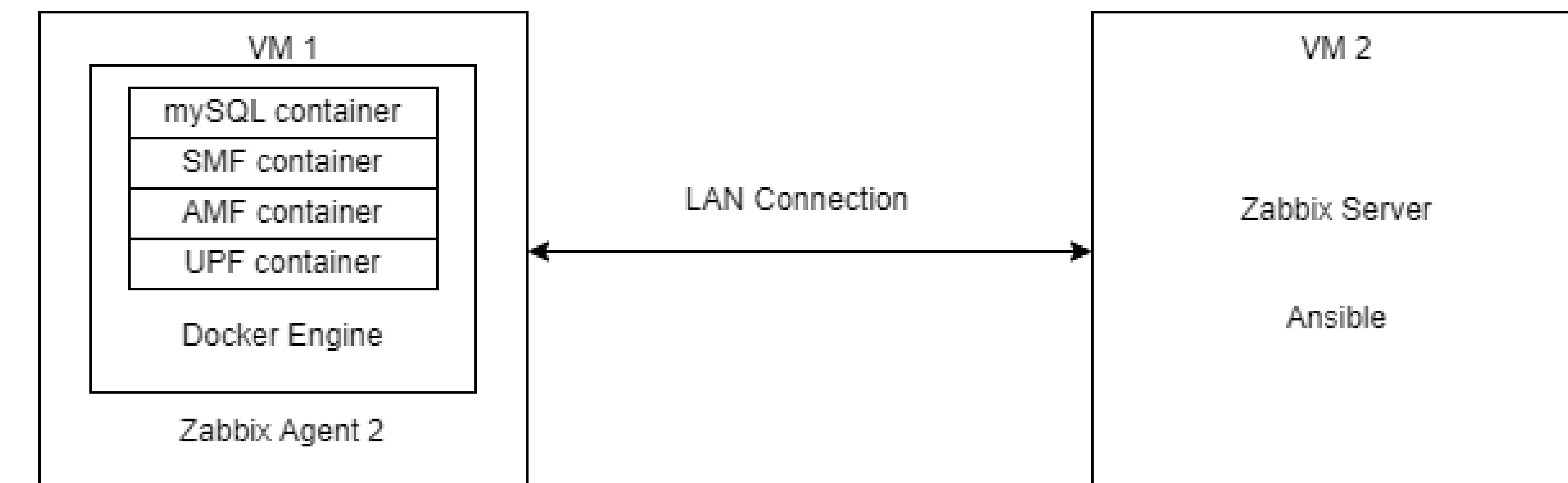
## Overview of Docker tools used

- Docker:** Docker is a software platform that allows you to build, test, and deploy applications quickly. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime.
- Docker compose:** Docker Compose is a tool that was developed to help define and share multi-container applications. With Compose, we can create a YAML file to define the services and with a single command, can spin everything up or tear it all down

- Docker Swarm:** Docker Swarm is a clustering and scheduling tool for Docker containers. With Swarm, IT administrators and developers can establish and manage a cluster of Docker nodes as a single virtual system

## Scenario/Use Case 1 – Container Recovery

- Local VM 1 running 5G core network function containers
- Local VM 2 running Zabbix server – monitoring operation of docker containers on VM 1

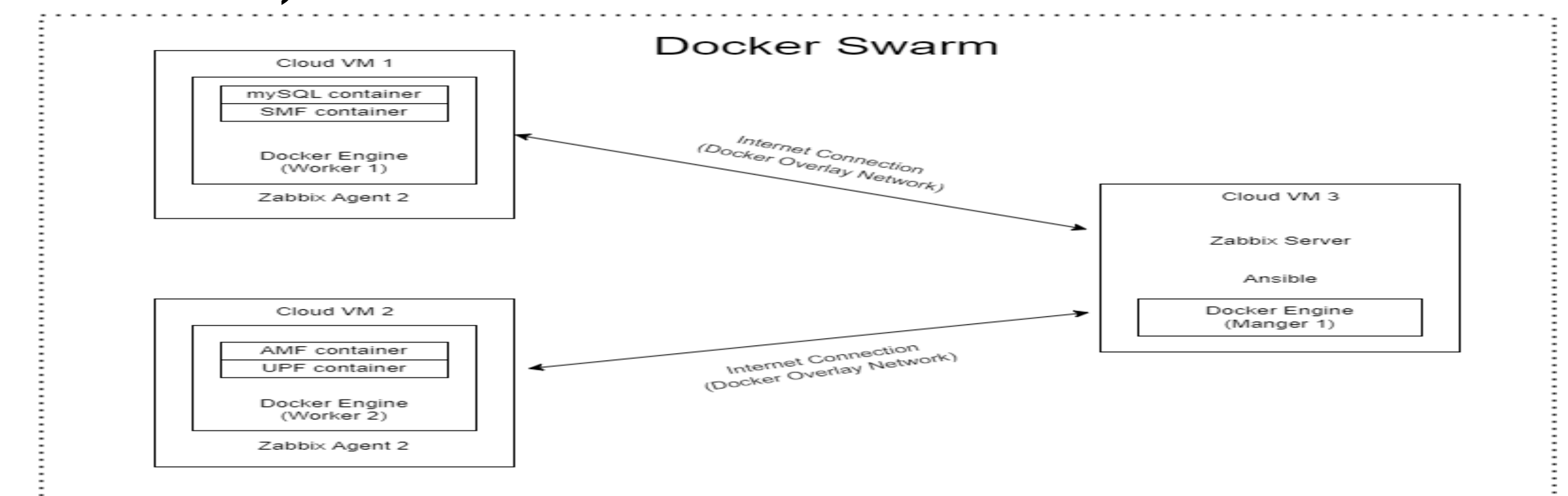


Scenario: If containers were to stop running on VM 1:

- Zabbix Server will detect change in container status
- Zabbix Server will run Ansible Playbook that will then SSH into VM 1 and run command to restart containers – the command to restart containers is a docker-compose command
- Zabbix Server will detect that containers have resumed operation

## Scenario/Use Case 2 – Distributed Load

- Three Cloud VMs included in Docker Swarm (1 manger, 2 workers)



Scenario: We can deploy all core network functions on the node closest to users, but what can we do if load at edge node becomes too much?

- We can move network functions that are not impacted by latency to a different edge node further away from the user while keeping network functions that will be impacted by latency close to them