

# Implementation Design Approach of Case Study 1 Using Open Source Tools

Sai Teja Bommi  
556620

# Solution Overview:

This section of the solution design provides an overview of the components of Continuous Integration and Continuous Deployment (CI/CD) solution for deploying the application using below mentioned tools.

- Terraform
- Jenkins
- Ansible
- Docker
- Maven
- Azure CLI
- GIT

# Design Consideration:

- Jenkins is available in the existing environment.
- The open source tools and application will be deployed and integrated in the Azure Cloud Environment.
- Continuous deployment of application will be achieved by integrating jenkins with other open source tools - Ansible, docker, maven etc.
- Jenkins will be configured in Master-Slave setup.

# Tools Functionality:

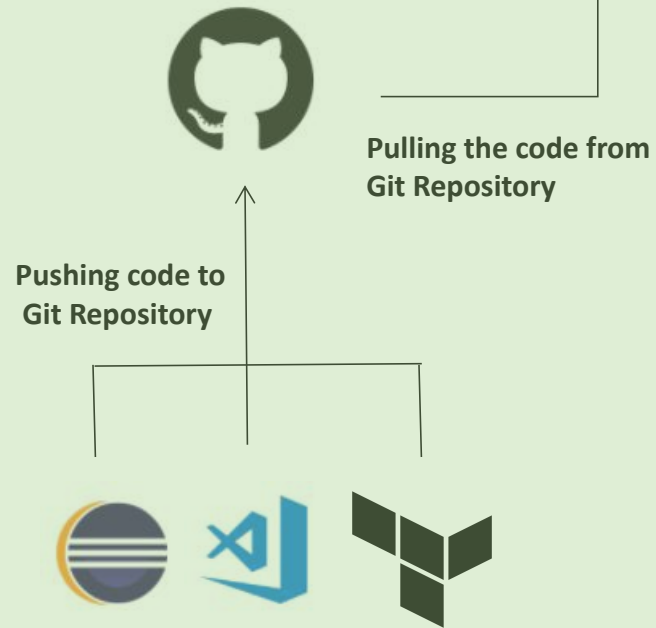
| Tool      | Functionality   |
|-----------|---|
| Jenkins   | For triggering the workflows                            |
| Maven     | For building and packaging the application              |
| GIT Hub   | For Source Code Management (SCM)                        |
| Terraform | For provisioning the Infrastructure                     |
| Ansible   | For provisioning and configuring the infrastructure     |
| Docker    | For building and package the application in a container |

# Architecture Details:

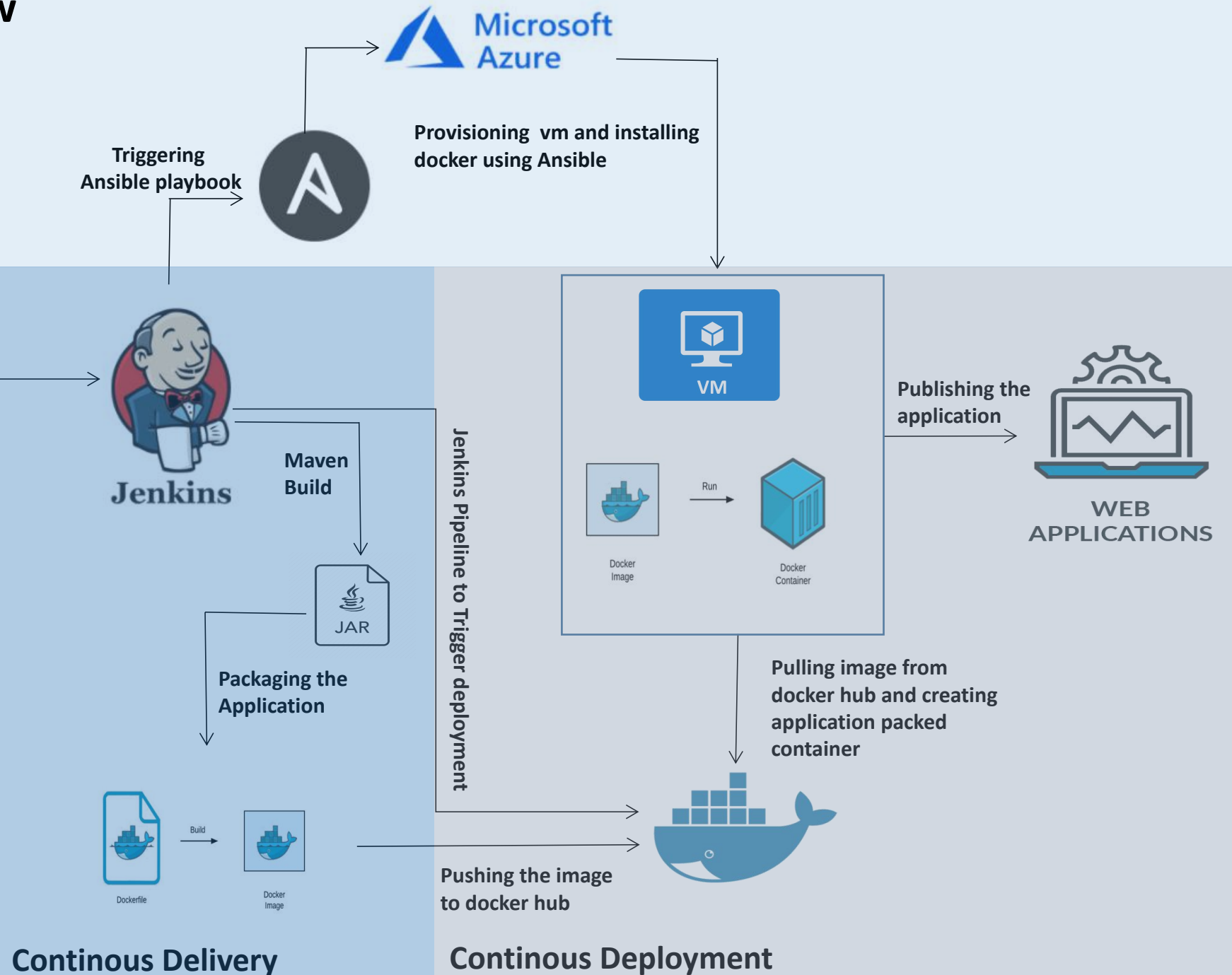
The suggested solution is achieved by configuring 4 VMS in the Infrastructure.

- **VM 1:** Developer VM on which Terraform, Visual Studio Code, Eclipse and GIT which will be used to create application code, terraform and ansible scripts and uploading to GIT repository.
- **VM 2:** Master Node VM on which jenkins will be installed and pipeline will be configured.
- **VM 3:** Slave Node VM on which GIT, Maven, Ansible and Docker will be installed.
- **VM 4:** Docker VM on which docker will installed and the docker container with packaged application will be deployed.

# Implementation Design Flow



Continuous Integration



Continuous Delivery

Continuous Deployment

# Design Flow:

- Using the terraform scripts to create Master and Slave VM's in Azure.
- Pushing the application code, terraform scripts and ansible playbooks to git repository.
- Starting and configuring the jenkins in Master-slave node configuration.
- Creating an jenkins pipeline which will pull the code from GitHub, packaging the application using maven build, then create an docker image with the packaged application and push to docker hub.
- Then triggering ansible to run ansible playbooks which will provision the docker VM and install docker. Execute docker commands to run the container with application image and publish the application.

Thank You