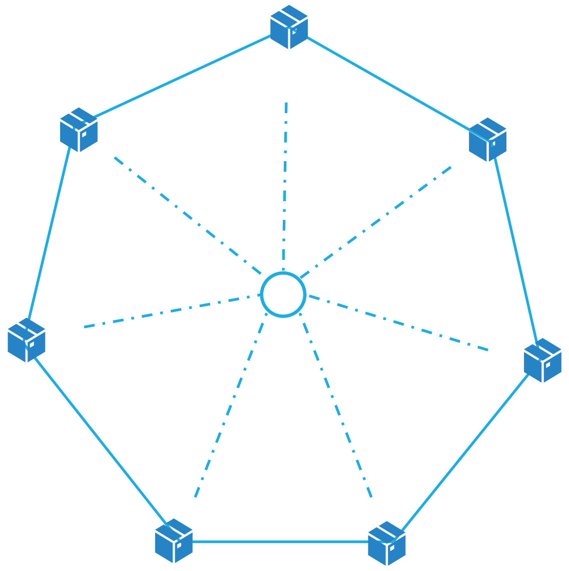
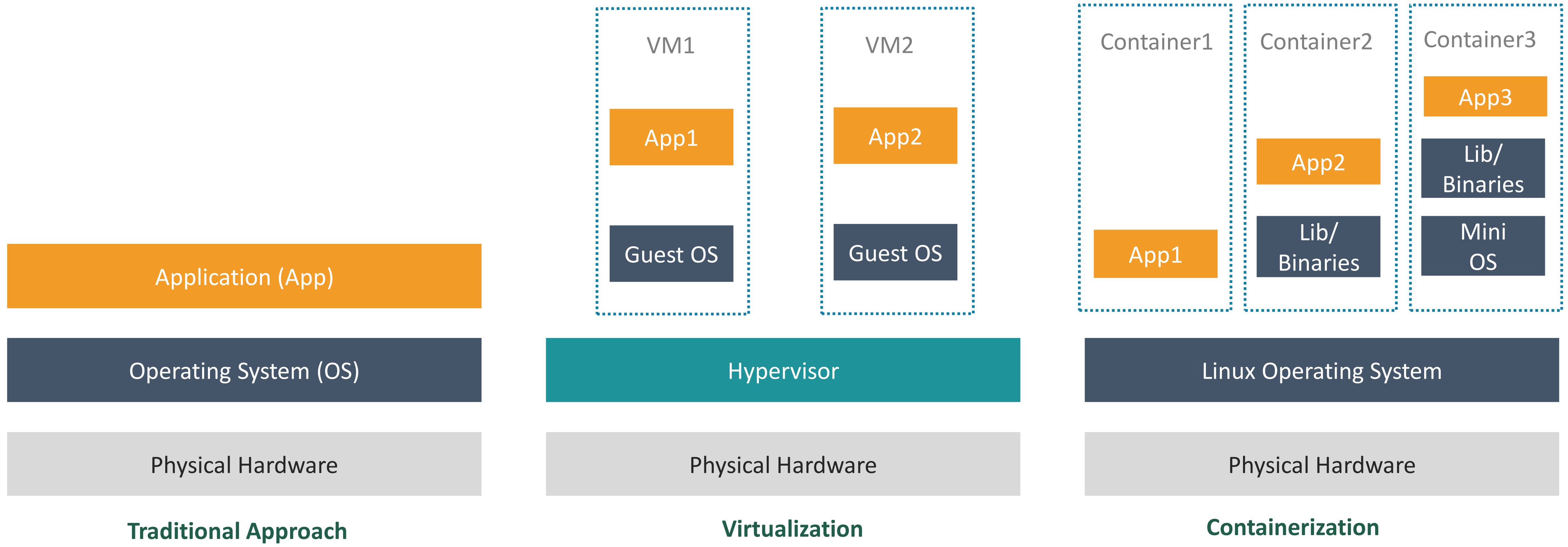


kubernetes

Containers

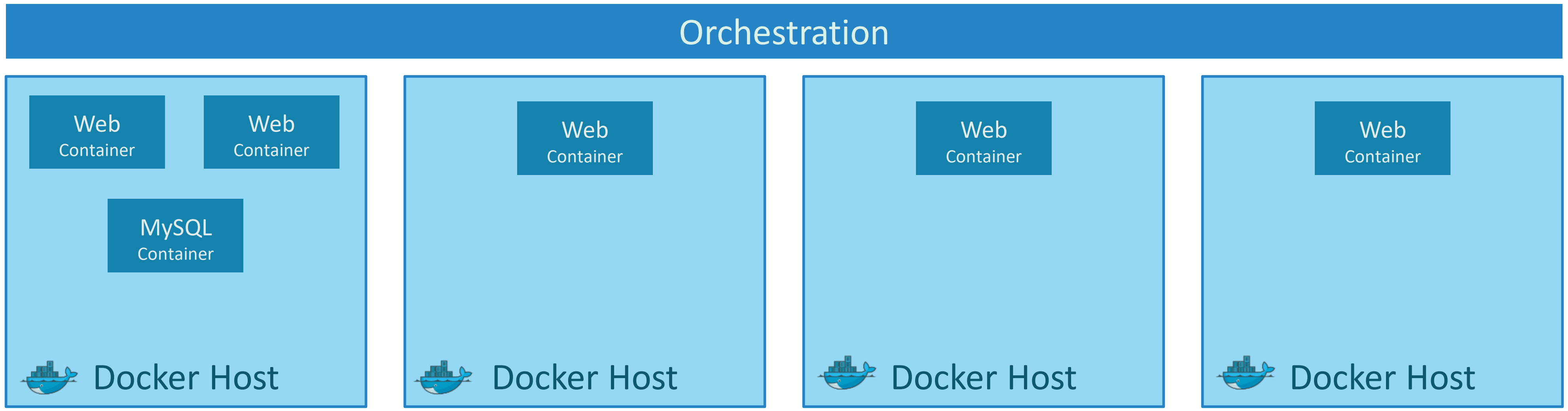

VM1
App1
Guest OS
VM2
App2
Guest OS
Container1
App1
Container2
App2
Lib/ Binaries
Container3
App3
Lib/ Binaries
Mini OS
Application (App)
Operating System (OS)
Physical Hardware
Hypervisor
Physical Hardware
Linux Operating System
Physical Hardware
Traditional Approach
Virtualization
Containerization

Container Orchestration

Containers Limitation?

High Availability?
Overlay Network?
Application Centric or Infra Centric?
Versioning of Application – Rollout, Rollback?
Scaling?
Autoscaling?
Monitoring?
Dependency between containers?
Load balancing among containers?
Network Security

Container orchestration



Orchestration Technologies

Docker Swarm

kubernetes

MESOS

What is Kubernetes?

The Kubernetes project was started by Google in 2014.

Kubernetes builds upon a decade and a half of experience that Google has with running production workloads at scale.

Kubernetes can run on a range of platforms, from your laptop, to VMs on a cloud provider, to rack of bare metal servers.

Kubernetes is an open-source platform for automating deployment, scaling, and operations of application containers across clusters of hosts, providing container-centric infrastructure.

portable: with all public, private, hybrid, community cloud

self-healing: auto-placement, auto-restart, auto-replication, auto-scaling

Why Kubernetes

Kubernetes can schedule and run application containers on clusters of physical or virtual machines.

host-centric infrastructure to a **container-centric** infrastructure.

Orchestrator

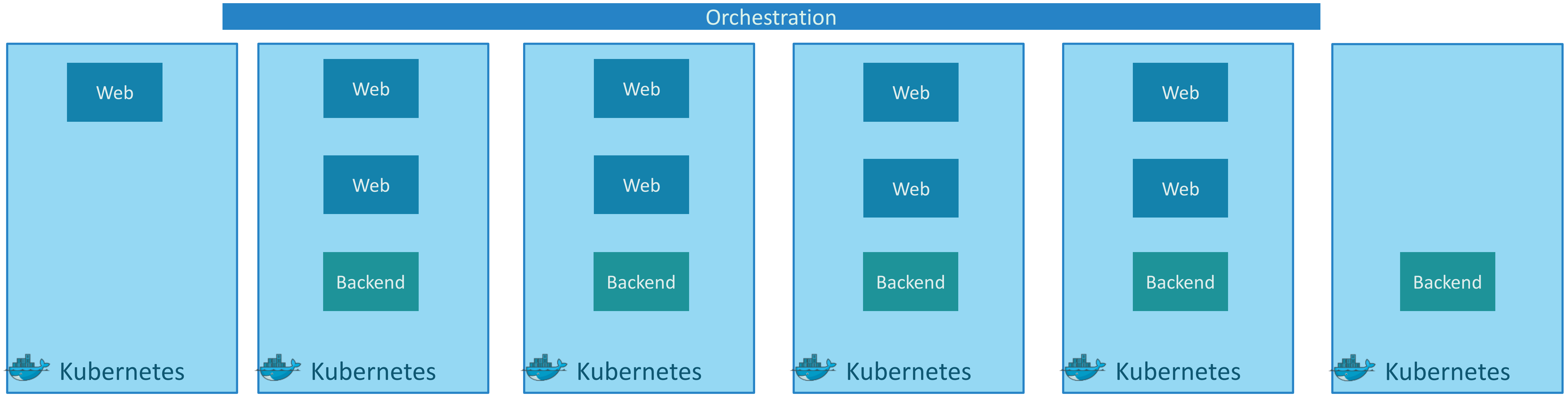
Load balancing

Auto Scaling

Application Health checks

Rolling updates

Kubernetes Advantage



Terraform with kubernetes

Terraform with Kubernetes

The Kubernetes (K8S) provider is used to interact with the resources supported by Kubernetes.

The provider needs to be configured with the proper credentials before it can be used.

Similar way like it manages resources on AWS, Azure, GCP, it can manage resources on K8S too.

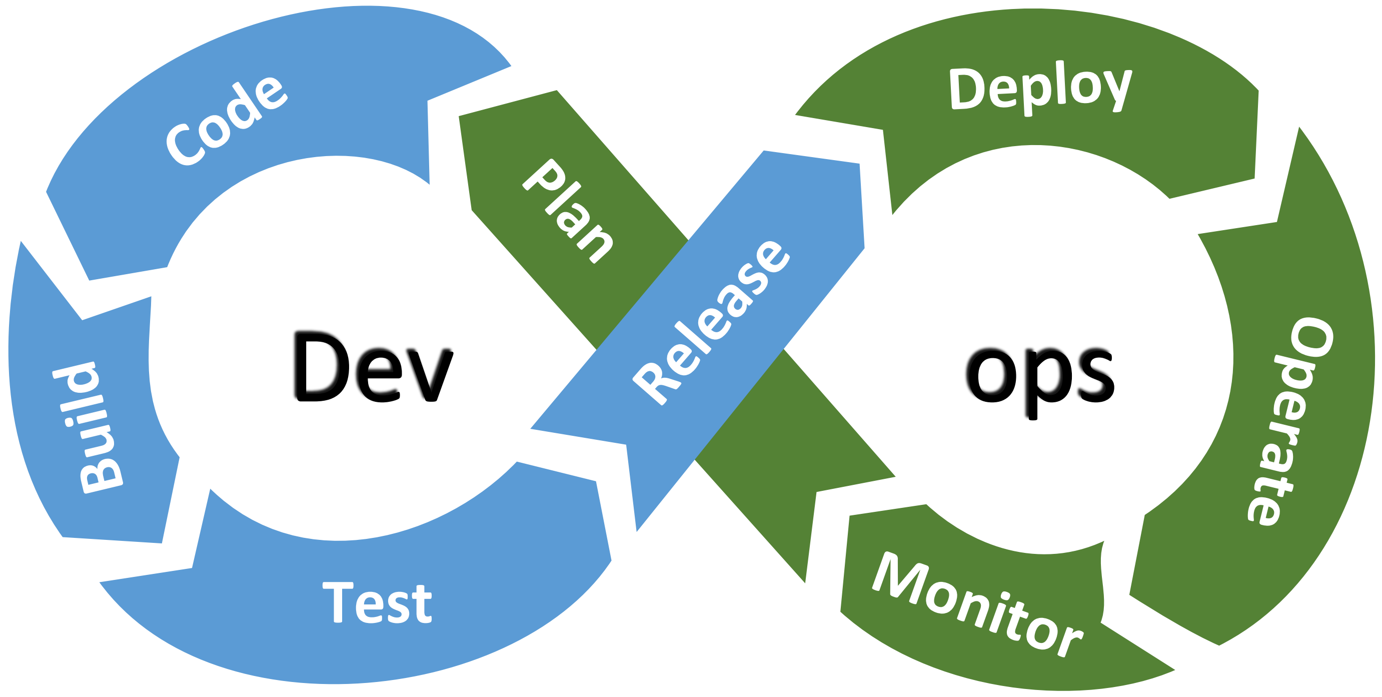
```
provider "kubernetes" {
  config_path = "~/.kube/config"
}
```

Terraform with Kubernetes

```
resource "kubernetes_namespace" "test" {
  metadata {
    name = "nginx"
  }
}
resource "kubernetes_deployment" "test" {
  metadata {
    name      = "nginx"
    namespace = kubernetes_namespace.test.metadata.0.name
  }
  spec {
    replicas = 2
    selector {
      match_labels = {
        app = "MyTestApp"
      }
    }
    template {
      metadata {
        labels = {
          app = "MyTestApp"
        }
      }
      spec {
        container {
          image = "nginx"
          name  = "nginx-container"
          port {
            container_port = 80
          }
        }
      }
    }
  }
}
```

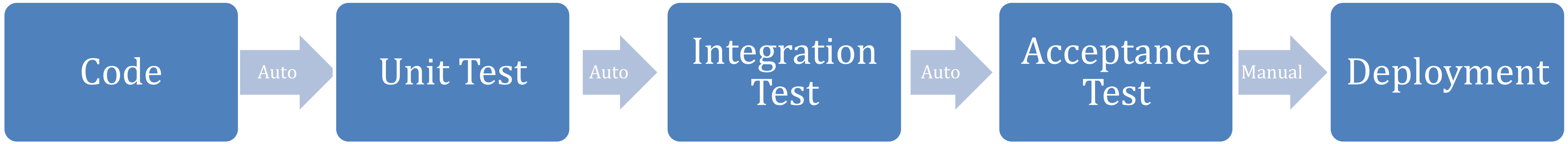
Terraform in DevOps

Terraform in DevOps

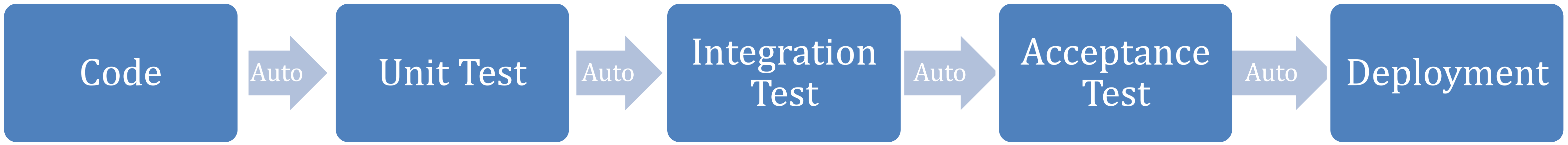


Terraform in DevOps

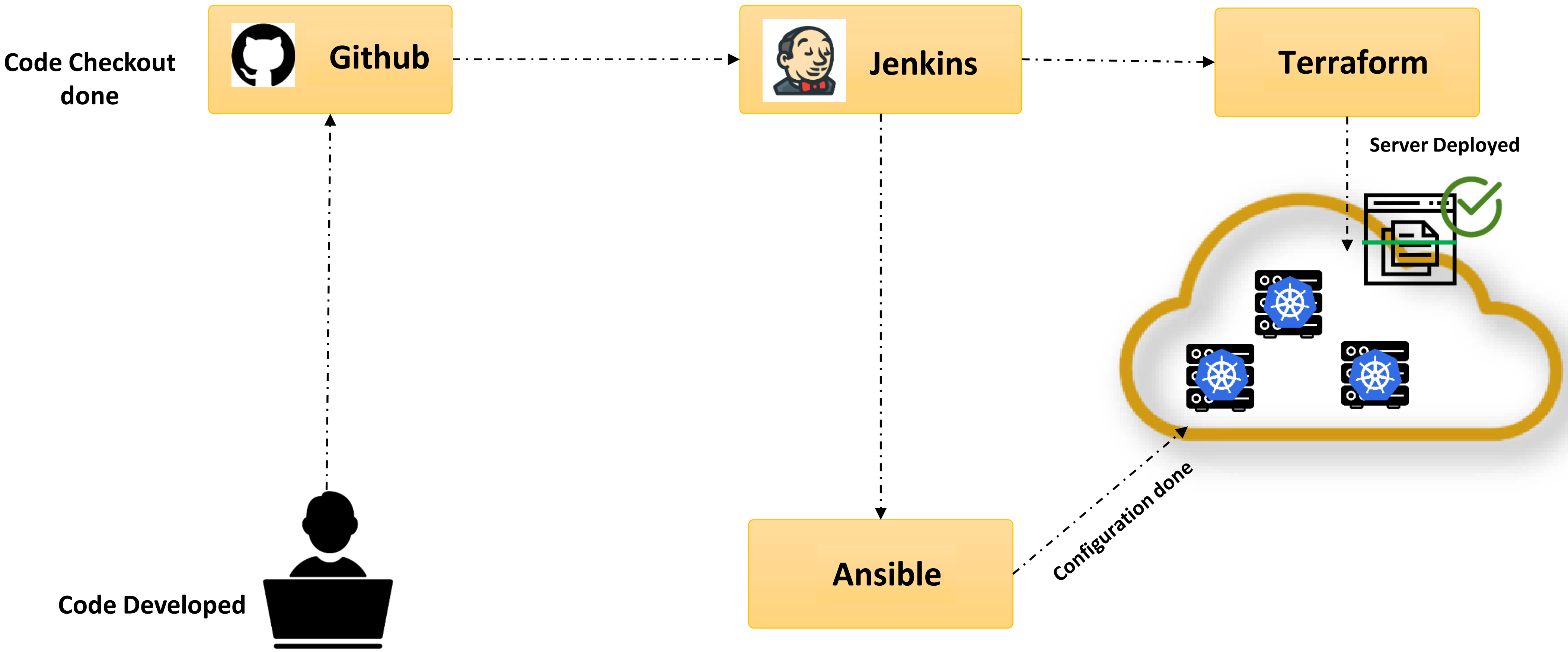
Continuous Integration



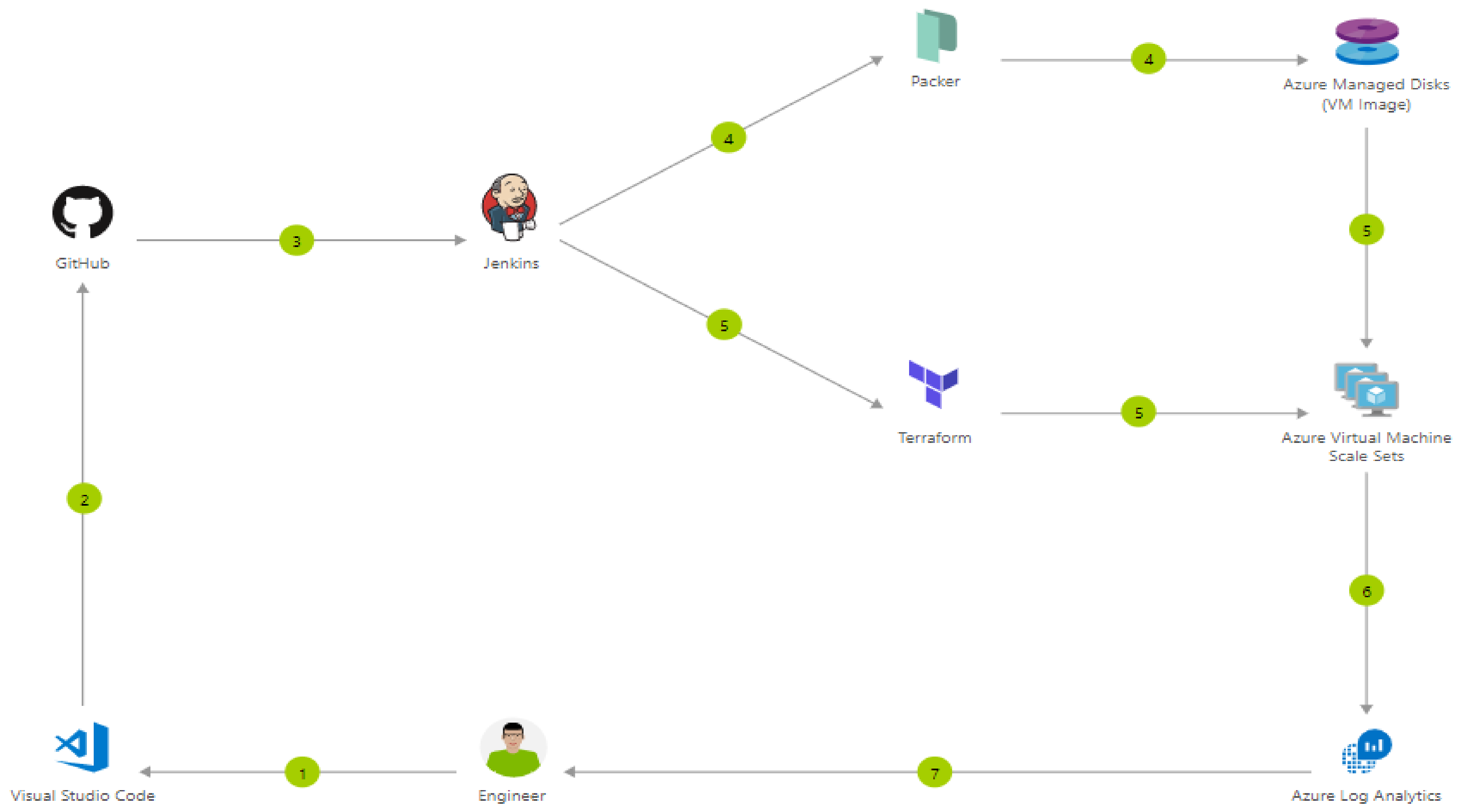
Continuous Deployment



Terraform in DevOps



DevOps IaC



Packer
4
Azure Managed Disks (VM Image)
4
5
GitHub
3
Jenkins
5
Terraform
5
Azure Virtual Machine Scale Sets
2
6
Visual Studio Code
1
Engineer
7
Azure Log Analytics

Terraform in DevOps

Install git client
yum install git

Install Jenkins client
sudo yum update -y
sudo yum install java-1.8.0-openjdk.x86_64 -y
java –version
Set JAVA_HOME and JRE_HOME
export JAVA_HOME=/usr/lib/jvm/jre-1.8.0-openjdk
export JRE_HOME=/usr/lib/jvm/jre
yum install wget -y

sudo wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins-ci.org/redhat-stable/jenkins.repo
sudo rpm --import http://pkg.jenkins-ci.org/redhat-stable/jenkins-ci.org.key
sudo yum install jenkins

sudo systemctl start jenkins.service
sudo systemctl enable jenkins.service

Run Jenkins with root

Lab: Ansible in DevOps

Create a new project in Jenkins:

1) Add Plugins – Git
2) Create repository on GitHub and add your playbooks on same
3) Add path of your GitHub in newly created project and set polling for every minute
4) Add build step to run the terraform apply statement
5) Build it for the first time
6) Do the changes in GitHub repo and observe the auto execution of next build on it own.

Questions & Answers