UNIT V APP Implementation IN Cloud

UNIT V APP IMPLEMENTATION IN CLOUD

Cloud providers Overview – Virtual Private Cloud – Scaling (Horizontal and Vertical) – Virtual Machines, Ethernet and Switches – Docker Container – Kubernetes

5.1 Cloud providers Overview

Overview

Cloud service providers are companies that establish public clouds, manage private clouds, or offer on-demand cloud computing components (also known as cloud computing services) like Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service(SaaS). Cloud services can reduce business process costs when compared to on-premise IT.

These clouds aren't usually deployed as a standalone infrastructure solution, but rather as part of a hybrid cloud.

Why use a cloud provider?

Using a cloud provider is a helpful way to access computing services that you would otherwise have to provide on your own, such as:

- **Infrastructure:** The foundation of every computing environment. This infrastructure could include networks, database services, data management, data storage (known in this context as cloud storage), servers (cloud is the basis for serverless computing), and virtualization.
- **Platforms:** The tools needed to create and deploy applications. These platforms could include operating systems like Linux®, middleware, and runtime environments.
- **Software:** Ready-to-use applications. This software could be custom or standard applications provided by independent service providers.

Certified cloud providers

There are a handful of well-known, major public cloud companies—such as Alibaba Cloud, Amazon Web Services (AWS), Google Cloud Platform (GCP), IBM Cloud, Oracle Cloud, and Microsoft Azure—but there are also hundreds of other cloud computing providers all over the world.

What are the benefits of using a cloud service provider?

Using a cloud provider has benefits and challenges. Companies considering using these services should think about how these factors would affect their priorities and risk profile, for both the present and long term. Individual CSPs have their own strengths and weaknesses, which are worth considering.

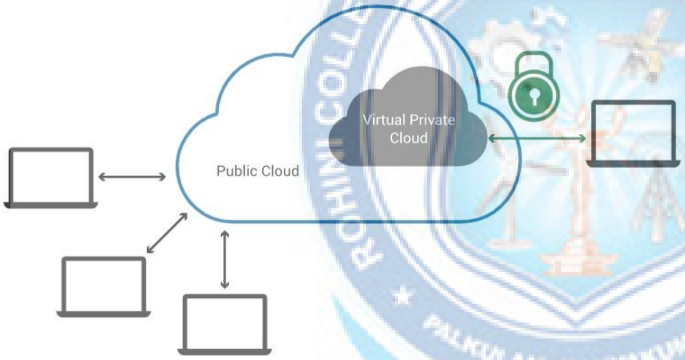
Benefits

- **Cost and flexibility.** The pay-as-you-go model of cloud services enables organizations to only pay for the resources they consume. Using a cloud service provider also eliminates the need for IT-related capital equipment purchases. Organizations should review the details of cloud pricing to accurately break down cloud costs.

- **Scalability.** Customer organizations can easily scale up or down the IT resources they use based on business demands.
- **Mobility.** Resources and services purchased from a cloud service provider can be accessed from any physical location that has a working network connection.
- **Disaster recovery.** Cloud services typically offer quick and reliable disaster recovery.

5.2 Virtual Private Cloud

What is a virtual private cloud (VPC)?



A virtual private cloud (VPC) is a secure, isolated private cloud hosted within a public cloud. VPC customers can run code, store data, host websites, and do anything else they could do in an ordinary private cloud, but the private cloud is hosted remotely by a public cloud provider. (Not all private clouds are hosted in this fashion.) VPCs combine the scalability and convenience of public cloud computing with the data isolation of private cloud computing.

Imagine a public cloud as a crowded restaurant, and a virtual private cloud as a reserved table in that crowded restaurant. Even though the restaurant is full of people, a table with a "Reserved" sign on it can only be accessed by the party who made the reservation. Similarly, a public cloud is crowded with various cloud customers accessing computing resources – but a VPC reserves some of those resources for use by only one customer.

What is a public cloud? What is a private cloud?

A public cloud is shared cloud infrastructure. Multiple customers of the cloud vendor access that same infrastructure, although their data is not shared – just like every person in a restaurant orders from the same kitchen, but they get different dishes. Public cloud service providers include AWS, Google Cloud Platform, and Microsoft Azure, among others.

The technical term for multiple separate customers accessing the same cloud infrastructure is "multitenancy" A private cloud, however, is single-tenant. A private cloud is a cloud service that is exclusively offered to one organization. A virtual private cloud (VPC) is a private cloud within a public cloud; no one else shares the VPC with the VPC customer.

How is a VPC isolated within a public cloud?

A VPC isolates computing resources from the other computing resources available in the public cloud. The key technologies for isolating a VPC from the rest of the public cloud are:

Subnets: A subnet is a range of IP addresses within a network that are reserved so that they're not available to everyone within the network, essentially dividing part of the network for private use. In a VPC these are private IP addresses that are not accessible via the public Internet, unlike typical IP addresses, which are publicly visible.

VLAN: A LAN is a local area network, or a group of computing devices that are all connected to each other without the use of the Internet. A VLAN is a virtual LAN. Like a subnet, a VLAN is a way of partitioning a network, but the partitioning takes place at a different layer within the OSI model (layer 2 instead of layer 3).

VPN: A virtual private network (VPN) uses encryption to create a private network over the top of a public network. VPN traffic passes through publicly shared Internet infrastructure – routers, switches, etc. – but the traffic is scrambled and not visible to anyone.

A VPC will have a dedicated subnet and VLAN that are only accessible by the VPC customer. This prevents anyone else within the public cloud from accessing computing resources within the VPC – effectively placing the "Reserved" sign on the table. The VPC customer connects via VPN to their VPC, so that data passing into and out of the VPC is not visible to other public cloud users.

Some VPC providers offer additional customization with:

- **Network Address Translation (NAT):** This feature matches private IP addresses to a public IP address for connections with the public Internet. With NAT, a public-facing website or application could run in a VPC.
- **BGP route configuration:** Some providers allow customers to customize BGP routing tables for connecting their VPC with their other infrastructure. (Learn how BGP works.)

What are the advantages of using a VPC instead of a private cloud?

Scalability: Because a VPC is hosted by a public cloud provider, customers can add more computing resources on demand.

Easy hybrid cloud deployment: It's relatively simple to connect a VPC to a public cloud or to on-premises infrastructure via the VPN. (Learn about hybrid clouds and their advantages.)

Better performance: Cloud-hosted websites and applications typically perform better than those hosted on local on-premises servers.

Better security: The public cloud providers that offer VPCs often have more resources for updating and maintaining the infrastructure, especially for small and mid-market businesses. For large enterprises or any companies that face extremely tight data security regulations, this is less of an advantage.

5.3 Scaling (Horizontal and Vertical)

If your business website or application becomes popular in the market or there is an increase in demand, you must broaden your view of user accessibility and performance. What do you do? The answer for this is usually some type of scalability of your IT infrastructure. When talking about scalability in cloud computing, you will often hear about **two ways of scaling: horizontal or vertical**. we will look deeper into these terms and also into **AWS (Amazon Web Services) scalability** and which services you can use.

What is scalability?
Cloud scalability refers to the ability to increase or decrease IT resources (virtual machines, databases, networks) as needed to meet changing needs. Scalability is one of the **main advantages of the cloud** and the main driving force for its popularity in businesses.

Public cloud providers such as **AWS (Amazon Web Services) already have all the infrastructure in place**; in the past, when scaling had to be done using on-premises infrastructure, the process could take weeks or months and require capital investment. Systems have four general areas that **scalability can apply to:**

- **CPU**
- **Disk I/O**
- **Memory**
- **Network I/O**

The **main benefit of the scalable architecture is performance** and the ability to handle bursts of traffic or heavy loads with little or no notice.

What is horizontal scaling?
To scale horizontally (scaling in or out), you add more resources like virtual machines to your system to spread out the workload across them. Horizontal scaling is especially important for companies that need **high availability** services with a requirement for minimal downtime.

Benefits of horizontal scaling
Horizontal scaling **increases high availability** because as long as you are spreading your infrastructure across multiple areas, if one machine fails, you can just use one of the other ones.

Because you're adding a machine, you need **fewer periods of downtime** and don't have to switch the old machine off while scaling. There may never be a need for downtime if you scale effectively.

And here are some simpler advantages of horizontal scaling:

- Easy to resize according to your needs
- Immediate and continuous availability
- Cost can be linked to usage and you don't always have to pay for peak demand

Disadvantages of horizontal scaling

The main disadvantage of horizontal scaling is that it **increases the complexity of the maintenance and operations** of your architecture, but there are services in the AWS environment to solve this issue.

- Architecture design and deployment can be very complicated
- A limited amount of software that can take advantage of horizontal scaling

What is vertical scaling?

Through vertical scaling (scaling up or down), you can increase or decrease the capacity of existing services/instances by upgrading the memory (RAM), storage, or processing power (CPU). Usually, this means that the expansion has an upper limit based on the capacity of the server or machine being expanded.

Vertical scaling benefits

- **No changes have to be made to the application code** and no additional servers need to be added; you just make the server you have more powerful or downsize again.
- **Less complex network** – when a single instance handles all the layers of your services, it will not have to synchronize and communicate with other machines to work. This may result in faster responses.
- **Less complicated maintenance** – the maintenance is easier and less complex because of the number of instances you will need to manage.

Vertical scaling disadvantages

- **A maintenance window with downtime is required** – unless you have a backup server that can handle operations and requests, you will need some considerable downtime to upgrade your machine.
- **Single point of failure** – having all your operations on a single server increases the risk of losing all your data if a hardware or software failure were to occur.
- **Upgrade limitations** – there is a limitation to how much you can upgrade a machine/instance.

Horizontal scaling vs. vertical scaling

In the cloud, you will usually use both of these methods, but horizontal scaling is usually considered a long-term solution, while vertical scaling is usually considered a short-term solution. The reason for this distinction is that you can usually add as many servers to the infrastructure as you need, but sometimes hardware upgrades are just not possible anymore.

Both horizontal and vertical scaling have their benefits and limitations. Here are some factors to consider:

- **Upgradability and flexibility** – if you run your application layer on separate machines (horizontally scaled), they are easier to decouple and upgrade without downtime.

- **Worldwide distribution** – if you plan to have national or global customers, it is unreasonable to expect them to access your services from one location. In this case, you need to scale resources horizontally.
- **Reliability and availability** – horizontal scaling can provide you with a more reliable system. It increases redundancy and ensures that you are not dependent on one machine.
- **Performance** – sometimes it’s better to leave the application as is and upgrade the hardware to meet demand (vertically scale). Horizontal scaling may require you to rewrite code, which can add complexity.

5.4 Virtual Machines, Ethernet and Switches

What is a virtual machine?

A **Virtual Machine** (VM) is a compute resource that uses software instead of a physical computer to run programs and deploy apps. One or more virtual “guest” machines run on a physical “host” machine. Each virtual machine runs its own operating system and functions separately from the other VMs, even when they are all running on the same host. This means that, for example, a virtual MacOS virtual machine can run on physical PC.

Virtual machine technology is used for many use cases across on-premises and cloud environments. More recently, public cloud services are using virtual machines to provide virtual application resources to multiple users at once, for even more cost efficient and flexible compute.

What are virtual machines used for?

Virtual machines (VMs) allow a business to run an operating system that behaves like a completely separate computer in an app window on a desktop. VMs may be deployed to accommodate different levels of processing power needs, to run software that requires a different operating system, or to test applications in a safe, sandboxed environment.

Virtual machines have historically been used for server virtualization, which enables IT teams to consolidate their computing resources and improve efficiency. Additionally, virtual machines can perform specific tasks considered too risky to carry out in a host environment, such as accessing virus-infected data or testing operating systems. Since the virtual machine is separated from the rest of the system, the software inside the virtual machine cannot tamper with the host computer.

How do virtual machines work?

The virtual machine runs as a process in an application window, similar to any other application, on the operating system of the physical machine. Key files that make up a virtual machine include a log file, NVRAM setting file, virtual disk file and configuration file.

Advantages of virtual machines

Virtual machines are easy to manage and maintain, and they offer several advantages over physical machines:

- VMs can run multiple operating system environments on a single physical computer, saving physical space, time and management costs.
- Virtual machines support legacy applications, reducing the cost of migrating to a new operating system. For example, a Linux virtual machine running a distribution of Linux as the guest operating system can exist on a host server that is running a non-Linux operating system, such as Windows.
- VMs can also provide integrated disaster recovery and application provisioning options.

Disadvantages of virtual machines

While virtual machines have several advantages over physical machines, there are also some potential disadvantages:

- Running multiple virtual machines on one physical machine can result in unstable performance if infrastructure requirements are not met.
- Virtual machines are less efficient and run slower than a full physical computer. Most enterprises use a combination of physical and virtual infrastructure to balance the corresponding advantages and disadvantages.

What is an Ethernet Switch?

Ethernet switching connects wired devices such as computers, laptops, routers, servers, and printers to a local area network (LAN). Multiple Ethernet switch ports allow for faster connectivity and smoother access across many devices at once.

An Ethernet switch creates networks and uses multiple ports to communicate between devices in the LAN. Ethernet switches differ from routers, which connect networks and use only a single LAN and WAN port. A full wired and wireless corporate infrastructure provides wired connectivity and Wi-Fi for wireless connectivity.

Hubs are similar to Ethernet switches in that connected devices on the LAN will be wired to them, using multiple ports. The big difference is that hubs share bandwidth equally among ports, while Ethernet switches can devote more bandwidth to certain ports without degrading network performance. When many devices are active on a network, Ethernet switching provides more robust performance.

Routers connect networks to other networks, most commonly connecting LANs to wide area networks (WANs). Routers are usually placed at the gateway between networks and route data packets along the network.

Most corporate networks use combinations of switches, routers, and hubs, and wired and wireless technology.

What Ethernet Switches Can Do For Your Network

Ethernet switches provide many advantages when correctly installed, integrated, and managed. These include:

1. Reduction of network downtime
2. Improved network performance and increased available bandwidth on the network
3. Relieving strain on individual computing devices
4. Protecting the overall corporate network with more robust security
5. Lower IT capex and opex costs thanks to remote management and consolidated wiring
6. Right-sizing IT infrastructure and planning for future expansion using modular switches

Most corporate networks support a combination of wired and wireless technologies, including Ethernet switching as part of the wired infrastructure. Dozens of devices can connect to a network using an Ethernet switch, and administrators can monitor traffic, control communications among machines, securely manage user access, and rapidly troubleshoot.

The switches come in a wide variety of options, meaning organizations can almost always find a solution right-sized for their network. These range from basic unmanaged network switches offering plug-and-play connectivity, to feature-rich Gigabit Ethernet switches that perform at higher speeds than wireless options.

How Ethernet Switches Work: Terms and Functionality

Frames are sequences of information, travel over Ethernet networks to move data between computers. An Ethernet frame includes a destination address, which is where the data is traveling to, and a source address, which is the location of the device sending the frame. In a standard seven-layer Open Systems Interconnection (OSI) model for computer networking, frames are part of Layer 2, also known as the data-link layer. These are sometimes known as “link layer devices” or “Layer 2 switches.”

Transparent Bridging is the most popular and common form of bridging, crucial to Ethernet switch functionality. Using transparent bridging, a switch automatically begins working without requiring any configuration on a switch or changes to the computers in the network (i.e. the operation of the switch is transparent).

Address Learning -- Ethernet switches control how frames are transmitted between switch ports, making decisions on how traffic is forwarded based on 48-bit media access control (MAC) addresses that are used in LAN standards. An Ethernet switch can learn which devices are on which segments of the network using the source addresses of the frames it receives.

Every port on a switch has a unique MAC address, and as frames are received on ports, the software in the switch looks at the source address and adds it to a table of addresses it constantly updates and maintains. (This is how a switch “discovers” what devices are reachable on which ports.) This table is also known as a forwarding database, which is used by the switch to make decisions on how to filter traffic to reach certain destinations. That the Ethernet switch can “learn” in this manner makes it possible for network administrators to add new connected endpoints to the network without having to manually configure the switch or the endpoints.

Traffic Filtering -- Once a switch has built a database of addresses, it can smoothly select how it filters and forwards traffic. As it learns addresses, a switch checks frames and makes decisions based on the destination address in the frame. Switches can also isolate traffic to only those segments needed to receive frames from senders, ensuring that traffic does not unnecessarily flow to other ports.

Multicast Traffic -- LANs are not only able to transmit frames to single addresses, but also capable of sending frames to multicast addresses, which are received by groups of endpoint destinations. Broadcast addresses are a specific form of multicast address; they group all of the endpoint destinations in the LAN. Multicasts and broadcasts are commonly used for functions such as dynamic address assignment, or sending data in multimedia applications to multiple users on a network at once, such as in online gaming. (Streaming applications such as video, which send high rates of multicast data and generate a lot of traffic, can hog network bandwidth.

5.5 Docker Container

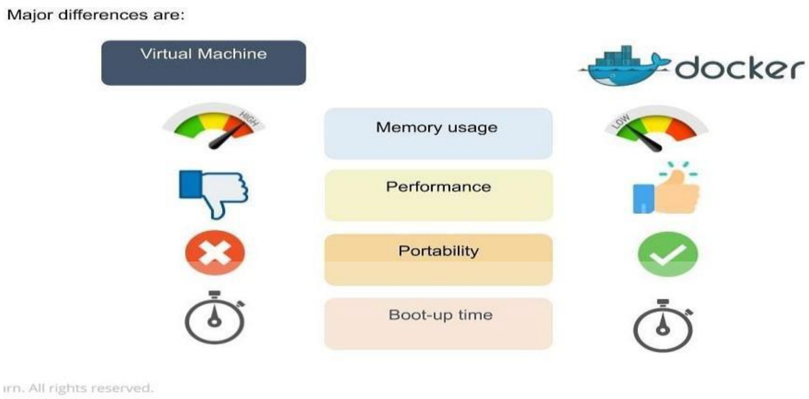
Docker is a container management service. The keywords of Docker are **develop, ship** and **run** anywhere. The whole idea of Docker is for developers to easily develop applications, ship them into containers which can then be deployed anywhere.

The initial release of Docker was in March 2013 and since then, it has become the buzzword for modern world development, especially in the face of Agile-based projects.

Features of Docker

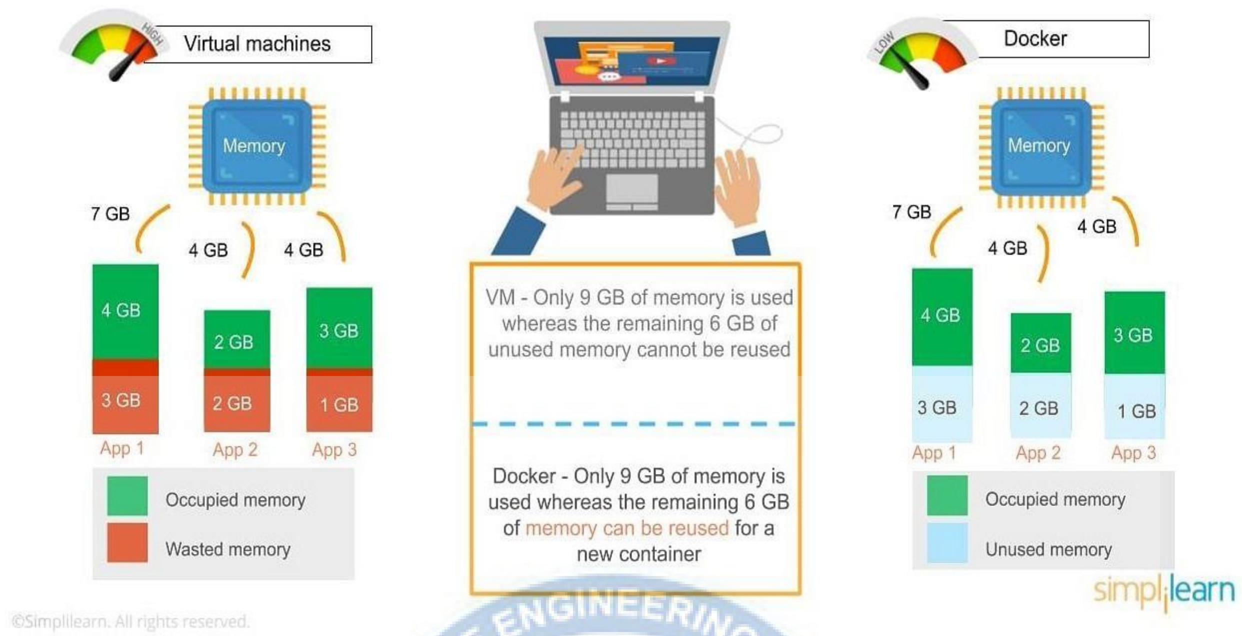
- Docker has the ability to reduce the size of development by providing a smaller footprint of the operating system via containers.
- With containers, it becomes easier for teams across different units, such as development, QA and Operations to work seamlessly across applications.
- You can deploy Docker containers anywhere, on any physical and virtual machines and even on the cloud.
- Since Docker containers are pretty lightweight, they are very easily scalable.

Docker vs Virtual Machines



In the image, you'll notice some major differences, including:

- The virtual environment has a hypervisor layer, whereas Docker has a Docker engine layer.
- There are additional layers of libraries within the virtual machine, each of which compounds and creates very significant differences between a Docker environment and a virtual machine environment.
- With a virtual machine, the memory usage is very high, whereas, in a Docker environment, memory usage is very low.
- In terms of performance, when you start building out a virtual machine, particularly when you have more than one virtual machine on a server, the performance becomes poorer. With Docker, the performance is always high because of the single Docker engine.
- In terms of portability, virtual machines just are not ideal. They're still dependent on the host operating system, and a lot of problems can happen when you use virtual machines for portability. In contrast, Docker was designed for portability. You can actually build solutions in a Docker container, and the solution is guaranteed to work as you have built it no matter where it's hosted.
- The boot-up time for a virtual machine is fairly slow in comparison to the boot-up time for a Docker environment, in which boot-up is almost instantaneous.



VM - Only 9 GB of memory is used whereas the remaining 6 GB of unused memory cannot be reused

Docker - Only 9 GB of memory is used whereas the remaining 6 GB of memory can be reused for a new container

©Simplilearn. All rights reserved.

- One of the other challenges of using a virtual machine is that if you have unused memory within the environment, you cannot reallocate it. If you set up an environment that has 9 gigabytes of memory, and 6 of those gigabytes are free, you cannot do anything with that unused memory. With Docker, if you have free memory, you can reallocate and reuse it across other containers used within the Docker environment.

- Running multiples of them in a single environment can lead to instability and performance issues. Docker, on the other hand, is designed to run multiple containers in the same environment—it actually gets better with more containers run in that hosted single Docker engine.

- Virtual machines have portability issues; the software can work on one machine, but if you move that virtual machine to another machine, suddenly some of the software won't work, because some dependencies will not be inherited correctly. Docker is designed to be able to run across multiple environments and to be deployed easily across systems.

- The boot-up time for a virtual machine is about a few minutes, in contrast to the milliseconds it takes for a Docker environment to boot up.

5.6 Kubernetes

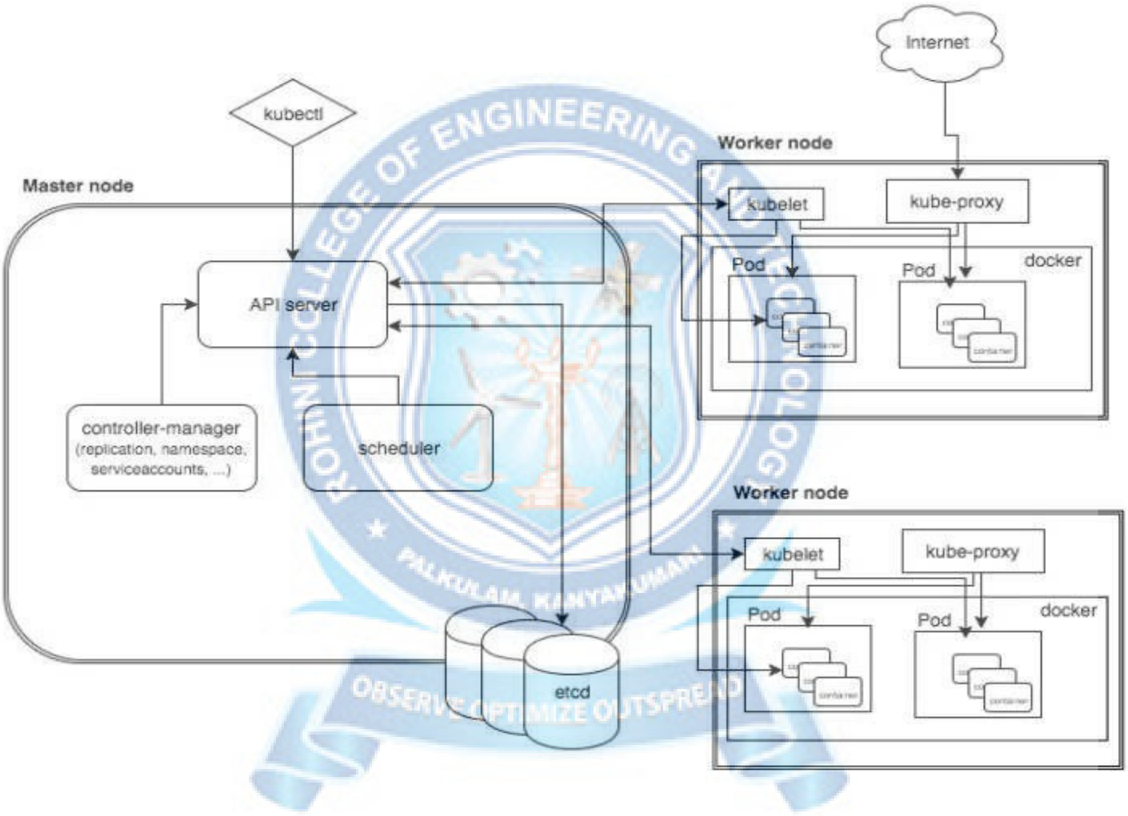
What is Kubernetes?

Kubernetes is an open source orchestration tool developed by Google for managing microservices or containerized applications across a distributed cluster of nodes. Kubernetes provides highly resilient infrastructure with zero downtime deployment capabilities, automatic rollback, scaling, and self-healing of containers (which consists of auto-placement, auto-restart, auto-replication , and scaling of containers on the basis of CPU usage).

The main objective of Kubernetes is to hide the complexity of managing a fleet of containers by providing REST APIs for the required functionalities. Kubernetes is portable in nature, meaning it can run on various public or private cloud platforms such as AWS, Azure, OpenStack, or Apache Mesos. It can also run on bare metal machines.

Kubernetes Components and Architecture

Kubernetes follows a client-server architecture. It’s possible to have a multi-master setup (for high availability), but by default there is a single master server which acts as a controlling node and point of contact. The master server consists of various components including a kube-apiserver, an etcd storage, a kube-controller-manager, a cloud-controller-manager, a kube-scheduler, and a DNS server for Kubernetes services. Node components include kubelet and kube-proxy on top of Docker.



A Kubernetes control plane is the control plane for a Kubernetes cluster. Its components include:

- **kube-apiserver**. As its name suggests the API server exposes the Kubernetes API, which is communications central. External communications via command line interface (CLI) or other user interfaces (UI) pass to the kube-apiserver, and all control planes to node communications also goes through the API server.
- **etcd**: The key value store where all data relating to the cluster is stored. etcd is highly available and consistent since all access to etcd is through the API server. Information in etcd is generally formatted in human-readable YAML (which stands for the recursive “YAML Ain’t Markup Language”).

- **kube-scheduler**: When a new Pod is created, this component assigns it to a node for execution based on resource requirements, policies, and ‘affinity’ specifications regarding geolocation and interference with other workloads.
- **kube-controller-manager**: Although a Kubernetes cluster has several controller functions, they are all compiled into a single binary known as kube-controller-manager.

What is Kubernetes node architecture?

Nodes are the machines, either VMs or physical servers, where Kubernetes place Pods to execute. Node components include:

kubelet: Every node has an agent called kubelet. It ensures that the container described in PodSpecs are up and running properly.

kube-proxy: A network proxy on each node that maintains network nodes which allows for the communication from Pods to network sessions, whether inside or outside the cluster, using operating system (OS) packet filtering if available.

container runtime: Software responsible for running the containerized applications. Although Docker is the most popular, Kubernetes supports any runtime that adheres to the Kubernetes CRI (Container Runtime Interface).

Kubernetes Concepts

Making use of Kubernetes requires understanding the different abstractions it uses to represent the state of the system, such as services, pods, volumes, namespaces, and deployments.

- **Pod** – generally refers to one or more containers that should be controlled as a single application. A pod encapsulates application containers, storage resources, a unique network ID and other configuration on how to run the containers.
- **Service** – pods are volatile, that is Kubernetes does not guarantee a given physical pod will be kept alive (for instance, the replication controller might kill and start a new set of pods). Instead, a service represents a logical set of pods and acts as a gateway, allowing (client) pods to send requests to the service without needing to keep track of which physical pods actually make up the service.
- **Volume** – similar to a container volume in Docker, but a Kubernetes volume applies to a whole pod and is mounted on all containers in the pod. Kubernetes guarantees data is

preserved across container restarts. The volume will be removed only when the pod gets destroyed. Also, a pod can have multiple volumes (possibly of different types) associated.

- **Namespace** – a virtual cluster (a single physical cluster can run multiple virtual ones) intended for environments with many users spread across multiple teams or projects, for isolation of concerns. Resources inside a namespace must be unique and cannot access resources in a different namespace. Also, a namespace can be allocated a resource quota to avoid consuming more than its share of the physical cluster's overall resources.
- **Deployment** – describes the desired state of a pod or a replica set, in a yaml file. The deployment controller then gradually updates the environment (for example, creating or deleting replicas) until the current state matches the desired state specified in the deployment file. For example, if the yaml file defines 2 replicas for a pod but only one is currently running, an extra one will get created. Note that replicas managed via a deployment should not be manipulated directly, only via new deployments.

➔ END ➔