UNIT V App Implement in Cloud

Cloud Implementation is the execution of a plan that finally gets you there and gets you the end product of a successful cloud implementation.

Cloud implementation Models is the answer to the question, “How” do you want your model implemented? The keyword is “How”.  You can implement your cloud as a subscription to Public Cloud, have it on-premise as a Private Cloud or subscribe to fully managed Private Cloud, have the best of both worlds, with a Hybrid Cloud, or have a Multi-Cloud setup.

Cloud Service Model is the answer to the question “What” do you want to be provisioned on the cloud? The keyword here is “What”. Is it your IT infrastructure? is it your software assets or web assets? Or a mix of both? The answers will usually result in it jazz-like IaaS, SaaS, PaaS, and all their variants.

**A cloud service provider, or CSP, is a company that offers some component of cloud computing; typically when you search the internet a cloud service is defined as, infrastructure as a service (IaaS), software as a service (SaaS) or platform as a service (PaaS) to other businesses or individuals.**

A Cloud Service is any system that provides on-demand availability of computer system resources, e.g; data storage and computing power, without direct active management by the user.

One benefit of using cloud computing services is that**firms can avoid the upfront cost and complexity**of owning and maintaining their own IT infrastructure, and instead simply pay for what they use, when they use it.

Today, rather than owning their own computing infrastructure or data centers, **companies can rent access to anything from applications to storage**. What that means is that if you have a supplier that handles and processes your companies healthcare data for instance, they may in fact be storing and processing your information in the cloud, either by outsourcing services or in some cases using an internal cloud or “private cloud” that they developed themselves by implementing it within the organization's dedicated resources, and infrastructure using “on-prem” services.

To add another twist, in other cases organizations may be using a diversified approach or “Hybrid Cloud” where they utilize both a private and public approach.

A large global bank has built their own private cloud. They wanted to take advantage of benefits of cloud computing like

* Rapid and simple deployment
* Less time to market for services
* Cost efficiency
* More utilization of server resources
* Less capital and operational costs
* This is managed by ABC bank Cloud datacenter services
* Better perceived security by managing and controlling it internally

Uses of Cloud Providers

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](https://www.redhat.com/en/topics/cloud-computing/what-is-iaas): The foundation of every computing environment. This infrastructure could include [networks](https://www.redhat.com/en/topics/hyperconverged-infrastructure/what-is-software-defined-networking), database services, [data management](https://www.redhat.com/en/topics/data-services/what-is-data-management), [data storage](https://www.redhat.com/en/topics/data-storage) (known in this context as [cloud storage](https://www.redhat.com/en/topics/data-storage/what-is-cloud-storage)), servers (cloud is the basis for [serverless computing](https://www.redhat.com/en/topics/cloud-native-apps/what-is-serverless)), and [virtualization](https://www.redhat.com/en/topics/virtualization).
* [Platforms](https://www.redhat.com/en/topics/cloud-computing/what-is-paas): The tools needed to create and deploy applications. These platforms could include operating systems like [Linux®](https://www.redhat.com/en/topics/linux), [middleware](https://www.redhat.com/en/topics/middleware/what-is-middleware), and [runtime environments](https://www.redhat.com/en/topics/cloud-native-apps/what-is-a-Java-runtime-environment).
* [Software](https://www.redhat.com/en/topics/cloud-computing/what-is-saas): Ready-to-use applications. This software could be custom or standard applications provided by independent service providers.

#### Public cloud providers

Public cloud providers virtualize their own infrastructure, platforms, or [applications](https://www.redhat.com/en/topics/cloud-native-apps) from hardware they own, and then pool all that into [data lakes](https://www.redhat.com/en/topics/data-storage/what-is-a-data-lake) that they [orchestrate](https://www.redhat.com/en/topics/automation/what-is-orchestration) with [management](https://www.redhat.com/en/topics/management) and [automation](https://www.redhat.com/en/topics/automation) software before transmitting it across the internet to their end users.

#### Managed private cloud

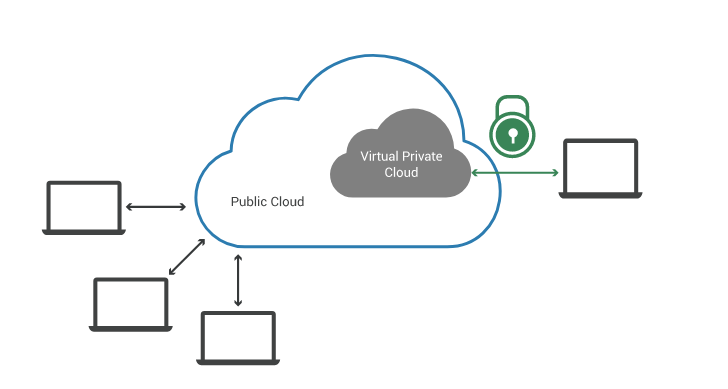
Also known as [managed cloud service](https://www.redhat.com/en/topics/cloud-computing/what-are-managed-it-services) providers, private cloud providers serve customers a private cloud that's deployed, [configured, and managed](https://www.redhat.com/en/topics/automation/what-is-configuration-management) by someone other than the customer. It's a cloud delivery option that helps enterprises with understaffed or underskilled IT teams provide better private [cloud services](https://www.redhat.com/en/topics/cloud-computing/what-are-cloud-services) and [cloud infrastructure](https://www.redhat.com/en/topics/cloud-computing/what-is-cloud-infrastructure) to users.

## Certified cloud providers

There are a handful of well-known, major public cloud companies—such as [Alibaba Cloud](https://www.redhat.com/en/partners/alibaba-cloud), [Amazon Web Services (AWS)](https://www.redhat.com/en/partners/aws/red-hat-on-aws), [Google Cloud Platform (GCP)](https://www.redhat.com/en/partners/google), [IBM Cloud](https://www.redhat.com/en/partners/ibm-alliance), Oracle Cloud, and [Microsoft Azure](https://www.redhat.com/en/partners/microsoft)—but there are also [hundreds of other cloud computing providers](https://redhat.secure.force.com/finder/) all over the world.

The [Red Hat Certified Cloud and Service Provider program](https://www.redhat.com/en/certified-cloud-and-service-providers) includes hundreds of cloud, system integrator, and managed service providers—along with software developers and hardware manufacturers—you can use to run Red Hat products, host physical and virtual machines, and set up private and public cloud environments.

Virtual Private Cloud



A virtual private cloud (VPC) is a secure, isolated [private cloud](https://www.cloudflare.com/learning/cloud/what-is-a-private-cloud/) hosted within a [public cloud](https://www.cloudflare.com/learning/cloud/what-is-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.

A public cloud is shared [cloud](https://www.cloudflare.com/learning/cloud/what-is-the-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" (see [What Is Multitenancy?](https://www.cloudflare.com/learning/cloud/what-is-multitenancy/) to learn more).

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.

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](https://www.cloudflare.com/learning/dns/glossary/what-is-my-ip-address/) 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](https://www.cloudflare.com/learning/ddos/glossary/open-systems-interconnection-model-osi/) (layer 2 instead of layer 3).

**VPN:** A [virtual private network (VPN)](https://www.cloudflare.com/learning/vpn/what-is-a-vpn/) uses [encryption](https://www.cloudflare.com/learning/ssl/what-is-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.](https://www.cloudflare.com/learning/security/glossary/what-is-bgp/))

## 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](https://www.cloudflare.com/learning/cloud/what-is-hybrid-cloud/) 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.

## How does Cloudflare support virtual private clouds

Cloudflare makes it easy to use any cloud service by providing a single plane of control for [performance](https://www.cloudflare.com/learning/performance/why-site-speed-matters/), security, and reliability services, including [bot management](https://www.cloudflare.com/products/bot-management/), [DNS](https://www.cloudflare.com/dns/), [SSL](https://www.cloudflare.com/ssl/), and [DDoS protection](https://www.cloudflare.com/ddos/) (even for [layer 3 traffic](https://www.cloudflare.com/magic-transit/)). The full Cloudflare stack sits in front of any cloud deployment and accelerates good traffic while blocking bad traffic.

## Benefits

Each VPC’s main features readily translate into a benefit to help your business achieve agility, increased innovation, and faster growth.

* **Flexible business growth:** Because cloud infrastructure resources—including virtual [servers](https://www.ibm.com/cloud/learn/cloud-server), [storage](https://www.ibm.com/cloud/learn/cloud-storage), and [networking](https://www.ibm.com/cloud/learn/networking-a-complete-guide)—can be deployed dynamically, VPC customers can easily adapt to changes in business needs.
* **Satisfied customers:** In today’s “always-on” digital business environments, customers expect uptime ratios of nearly 100%. The high availability of VPC environments enables reliable online experiences that build customer loyalty and increase trust in your brand.
* **Reduced risk across the entire data lifecycle:** VPCs enjoy high levels of security at the instance or subnet level, or both. This gives you peace of mind and further increases the trust of your customers.
* **More resources to channel toward business innovation:** With reduced costs and fewer demands on your internal IT team, you can focus your efforts on achieving key business goals and exercising core competencies.

## Architecture

In a VPC, you can deploy cloud resources into your own isolated virtual network. These cloud resources—also known as logical instances—fall into three categories.

* **Compute:** Virtual server instances (VSIs, also known as virtual servers) are presented to the user as virtual CPUs (vCPUs) with a predetermined amount of computing power, memory, etc.
* **Storage:** VPC customers are typically allocated a certain [block storage](https://www.ibm.com/cloud/learn/block-storage) quota per account, with the ability to purchase more. It is akin to purchasing additional hard drive space. Recommendations for storage are based on the nature of your workload.
* **Networking:** You can deploy virtual versions of various networking functions into your virtual private cloud account to enable or restrict access to its resources. These include public gateways, which are deployed so that all or some areas of your VPC environment can be made available on the public-facing Internet; [*load balancers*](https://www.ibm.com/cloud/learn/load-balancing), which distribute traffic across multiple VSIs to optimize availability and performance;  and routers, which direct traffic and enable communication between network segments. Direct or dedicated links enable rapid and secure communications between your on-premises enterprise IT environment or your private cloud and your VPC resources on public cloud.

Three-tier architecture in a VPC

The majority of today’s applications are designed with a three-tier architecture comprised of the following interconnected tiers:

* The web or presentation tier, which takes requests from web browsers and presents information created by, or stored within, the other layers to end users.
* The application tier, which houses the business logic and is where most processing takes place.
* The database tier, comprised of database servers that store the data processed in the application tier.

To create a three-tier application architecture on a VPC, you assign each tier its own subnet, which will give it its own IP address range. Each layer is automatically assigned its own unique ACL.

## Security

VPCs achieve high levels of security by creating [virtualized](https://www.ibm.com/cloud/learn/virtualization-a-complete-guide) replicas of the security features used to control access to resources housed in traditional data centers. These security features enable customers to define virtual networks in logically isolated parts of the public cloud and control which IP addresses have access to which resources.

Two types of network access controls comprise the layers of VPC security:

* **Access control lists (ACLs):** An ACL is a list of rules that limit who can access a particular subnet within your VPC. A subnet is a portion or subdivision of your VPC; the ACL defines the set of IP addresses or applications granted access to it.
* **Security group:** With a security group, you can create groups of resources (which may be situated in more than one subnet) and assign uniform access rules to them. For example, if you have three applications in three different subnets, and you want them all to be public Internet-facing, you can place them in the same security group. Security groups act like virtual firewalls, controlling the flow of traffic to your virtual servers, no matter which subnet they are in.

# Cloud Scalability

Cloud scalability in cloud computing refers to the ability to increase or decrease IT resources as needed to meet changing demand. Scalability is one of the hallmarks of the cloud and the primary driver of its

exploding popularity with businesses.

Data storage capacity, processing power and networking can all be scaled using existing cloud computing infrastructure. Better yet, scaling can be done quickly and easily, typically with little to no disruption or down time. Third-party cloud providers have all the infrastructure already in place; in the past, when scaling with on-premises physical infrastructure, the process could take weeks or months and require tremendous expense.

Cloud scalability in cloud computing refers to increasing or decreasing IT resources as needed to meet changing demand. Scalability is one of the hallmarks of the cloud and the primary driver of its explosive popularity with businesses.

Data storage capacity, processing power, and networking can all be increased by using existing cloud computing infrastructure. Scaling can be done quickly and easily, usually without any disruption or downtime.

Third-party cloud providers already have the entire infrastructure in place; In the past, when scaling up with on-premises physical infrastructure, the process could take weeks or months and require exorbitant expenses.

This is one of the most popular and beneficial features of cloud computing, as businesses can grow up or down to meet the demands depending on the season, projects, development, etc.

By implementing cloud scalability, you enable your resources to grow as your traffic or organization grows and vice versa. There are a few main ways to scale to the cloud:

If your business needs more data storage capacity or processing power, you'll want a system that scales easily and quickly.

Cloud computing solutions can do just that, which is why the market has grown so much. Using existing cloud infrastructure, third-party cloud vendors can scale with minimal disruption.

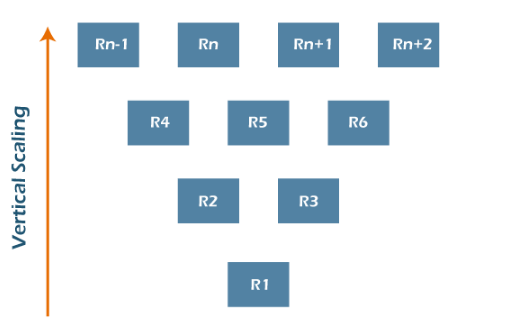
## Types of scaling

* Vertical Scalability (Scaled-up)
* horizontal scalability
* diagonal scalability

### Vertical Scaling

To understand vertical scaling, imagine a 20-story hotel. There are innumerable rooms inside this hotel from where the guests keep coming and going. Often there are spaces available, as not all rooms are filled at once. People can move easily as there is space for them. As long as the capacity of this hotel is not exceeded, no problem. This is vertical scaling.

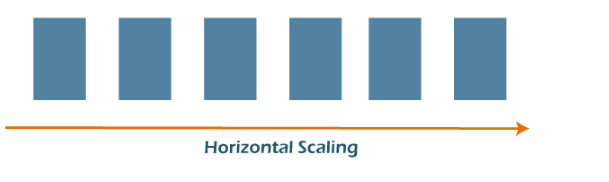
With computing, you can add or subtract resources, including memory or storage, within the server, as long as the resources do not exceed the capacity of the machine. Although it has its limitations, it is a way to improve your server and avoid latency and extra management. Like in the hotel example, resources can come and go easily and quickly, as long as there is room for them.



### Horizontal Scaling

Horizontal scaling is a bit different. This time, imagine a two-lane highway. Cars travel smoothly in each direction without major traffic problems. But then the area around the highway develops - new buildings are built, and traffic increases. Very soon, this two-lane highway is filled with cars, and accidents become common. Two lanes are no longer enough. To avoid these issues, more lanes are added, and an overpass is constructed. Although it takes a long time, it solves the problem.

Horizontal scaling refers to adding more servers to your network, rather than simply adding resources like with vertical scaling. This method tends to take more time and is more complex, but it allows you to connect servers together, handle traffic efficiently and execute concurrent workloads.



Benefits of cloud scalability

Key cloud scalability benefits driving cloud adoption for businesses large and small:

* Convenience: Often, with just a few clicks, IT administrators can easily add more VMs that are available-and customized to an organization's exact needs-without delay. Teams can focus on other tasks instead of setting up physical hardware for hours and days. This saves the valuable time of the IT staff.
* Flexibility and speed: As business needs change and grow, including unexpected demand spikes, cloud scalability allows IT to respond quickly. Companies are no longer tied to obsolete equipment-they can update systems and easily increase power and storage. Today, even small businesses have access to high-powered resources that used to be cost-prohibitive.
* Cost Savings: Thanks to cloud scalability, businesses can avoid the upfront cost of purchasing expensive equipment that can become obsolete in a few years. Through cloud providers, they only pay for what they use and reduce waste.
* Disaster recovery: With scalable cloud computing, you can reduce disaster recovery costs by eliminating the need to build and maintain secondary data centers.
* When to Use Cloud Scalability?
* Successful businesses use scalable business models to grow rapidly and meet changing demands. It's no different with their IT. Cloud scalability benefits help businesses stay agile and competitive.
* Scalability is one of the driving reasons for migrating to the cloud. Whether traffic or workload demands increase suddenly or increase gradually over time, a scalable cloud solution enables organizations to respond appropriately and cost-effectively to increased storage and performance.
* How do you determine optimal cloud scalability?
* Changing business needs or increasing demand often necessitate your scalable cloud solution changes. But how much storage, memory, and processing power do you need? Will you scale in or out?
* To determine the correct size solution, continuous performance testing is essential. IT administrators must continuously measure response times, number of requests, CPU load, and memory usage. Scalability testing also measures the performance of an application and its ability to scale up or down based on user requests.
* Automation can also help optimize cloud scalability. You can set a threshold for usage that triggers automatic scaling so as not to affect performance. You may also consider a third-party configuration management service or tool to help you manage your scaling needs, goals, and implementation.

A virtual machine

(VM) is a software-based computer that exists within another computer’s operating system, often used for the purposes of testing, backing up data, or running [SaaS](https://www.cloudflare.com/learning/cloud/what-is-saas/) applications. To grasp how VMs work, it’s important to first understand how computer software and hardware are typically integrated by an operating system.

## What is an operating system?

Traditional computers are built out of physical hardware, including hard disk drives, processor chips, RAM, and more. In order to utilize this hardware, computers rely on a type of software known as an operating system (OS). Some common examples of OSes are Mac OSX, Microsoft Windows, Linux, and Android.

The OS is what manages the computer’s hardware in ways that are useful to the user. For example, if the user wants to access the Internet, the OS directs the network interface card to make the connection. If the user wants to download a file, the OS will partition space on the hard drive for that file. The OS also runs and manages other pieces of software. For example, it can run a web browser and provide the browser with enough random access memory (RAM) to operate smoothly.

Typically, operating systems exist within a physical computer at a one-to-one ratio. for each machine there is a single OS managing its physical resources.

## Can you have two or more operating systems on one computer?

It is possible to run multiple operating systems on one computer. This can be achieved through a process called virtualization. In virtualization, a piece of software behaves as if it were an independent computer. This piece of software is called a virtual machine, also known as a ‘guest’ computer. (The computer on which the VM is running is called the ‘host’.) The guest has an OS as well as its own virtual hardware.

‘Virtual hardware’ may sound like an oxymoron. In fact, a VM's 'hard drive' is really just a file on the host computer’s hard drive. However, a virtual hard drive fulfills the same function as a physical one.

The number of VMs that can run on one host is limited only by the host’s available resources. The user can run the OS of a VM in a window like any other program, or they can run it in fullscreen so that it looks and feels like a genuine host OS.

## What are virtual machines used for?

Common use cases for virtual machines on single computers include:

* **Testing** - Software developers often want to test their applications in different environments. They can use virtual machines to run their applications in various OSes on one computer. This is simpler and more cost-effective than testing on several different physical machines.
* **Running software designed for other OSes** - Although certain software applications are only available for a single platform, a VM can run software designed for a different OS. For example, a Mac user who wants to run software designed for Windows can run a Windows VM on their Mac host.
* **Running outdated software** - Some pieces of older software can’t be run in modern OSes. Users who want to run these applications can run an old OS on a virtual machine.
* **Browser isolation** - [Browser isolation](https://www.cloudflare.com/learning/access-management/what-is-browser-isolation/) is the practice of 'isolating' web browser activity away from the rest of a computer's operating system to keep malware from affecting the computer's other files and programs. Some broswer isolation tools use VMs to establish this isolation — though this approach can slow down browsing activity.

## How does cloud computing use virtual machines?

Several cloud providers offer virtual machines to their customers. These virtual machines typically live on powerful servers that can act as a host to multiple VMs and can be used for a variety of reasons that wouldn’t be practical with a locally-hosted VM. These include:

* **Running SaaS applications** - [Software-as-a-Service](https://www.cloudflare.com/learning/cloud/what-is-saas/), or SaaS for short, is a cloud-based method of providing software to users, in which an application is served to user over the Internet rather than running on their computers. Often, it is virtual machines in [the cloud](https://www.cloudflare.com/learning/cloud/what-is-the-cloud/) that do the computation for SaaS applications as well as delivering them to users. If the cloud provider has a geographically distributed [network edge](https://www.cloudflare.com/learning/serverless/glossary/what-is-edge-computing/), then the application will run closer to the user, resulting in faster performance.
* **Backing up data** - Cloud-based VM services are popular for backing up data, because the data can be accessed from anywhere. Plus, cloud VMs provide better redundancy, require less maintenance, and generally scale better than physical data centers. (For example, it’s relatively easy to buy an extra gigabyte of storage space from a cloud VM provider, but much more difficult to build a new local data server for that extra gigabyte of data.)
* **Hosting services like email and access management** - Hosting these services on cloud VMs is generally faster and more cost-effective, and helps minimize maintenance and offload security concerns as well.
* **Browswer isolation** - Some browser isolation tools use cloud VMs to run web broswing activity and deliver safe content to users via a secure Internet connection

|  |
| --- |
| Virtual Machines, Ethernet and Switches  A large amount of computing services nowadays are migrating to virtualized environments, which offer significant advantages in terms of resource sharing and cost reduction. Virtual machines need to communicate and access peripherals, which for systems used as servers mostly means disks and network interfaces. The latter are extremely challenging to deal with even in non-virtualized environments, due to the high data and packet rates involved, and the fact that, unlike disks, traffic generation is initiated by external entities on which the receiver has no control. It is then not a surprise that virtual machines may have a tough time in operating network interfaces at wire speed in all possible conditions. As it is often the case, hardware assistance comes handy to improve performance. As shown in Section 2.2, some proposals rely on multiqueue network cards exporting resources to virtual machines through PCI passthrough, and/or on external switches to copy data between interfaces. However this solution is expensive and not necessarily scalable. On the other hand, software-only solutions proposed to date tend to have relatively low performance, especially for small packet sizes. |
|  |

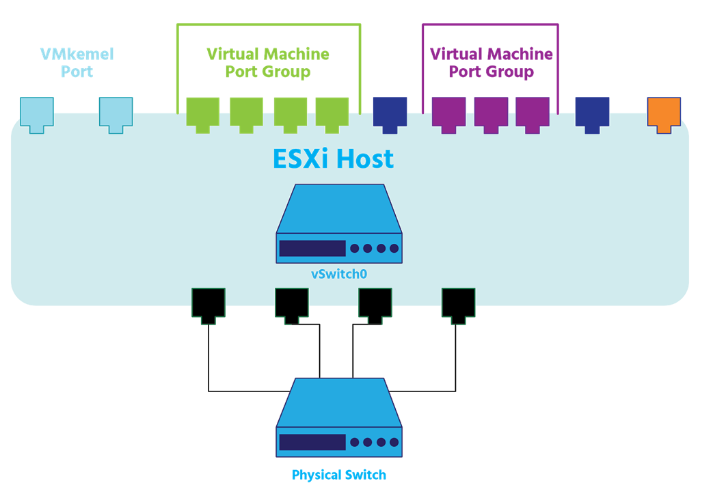
Cloudflare helps protect and manage any type of cloud deployment, including cloud VMs. SaaS providers can use [Cloudflare for SaaS](https://www.cloudflare.com/saas/) to improve their application's peformance, protect custom domains for end-users, and more.

Additionally, for users who want the functionality of running code on the edge without the overhead of virtual machines, the [Cloudflare Workers](https://www.cloudflare.com/products/cloudflare-workers/) a [serverless](https://www.cloudflare.com/learning/serverless/what-is-serverless/) platform provides edge computation to customers in a completely scalable way, allowing developers to augment existing applications or create entirely new ones without configuring or maintaining infrastructure.

At the core of vSphere networking are virtual switches (vSwitches). They allow virtual machines to connect to each other and to connect to the outside world. By default, each ESXi host has a single virtual switch called *vSwitch0*.

The connection between a virtual machine and a virtual switch is similar to the connection between a computer's physical network adapter (NIC) and a physical switch. But instead of using a wired Ethernet cable, the virtual machine is connected to the port on the virtual switch by a virtual wire.

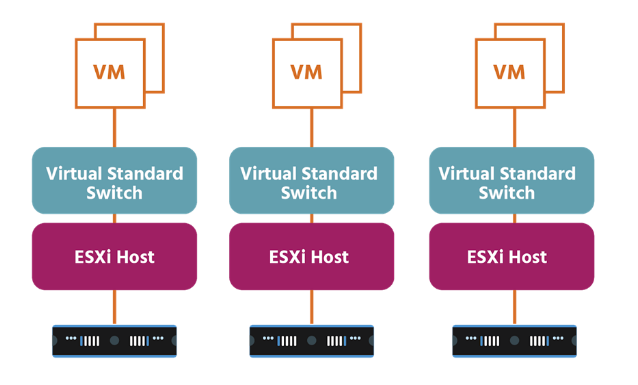
As with a physical switch, Layer 2 frames enter and exit a vSwitch. As with a physical switch, a vSwitch has ports organized into port groups. As with a physical switch, a vSwitch has uplink ports. These are physical network adapter ports found within the ESXi host, and connect the virtual switch within the ESXi host to a physical switch.



Uplinks connect the virtual switch to the physical world: they move physical 0s and 1s off the host and out into the world. A virtual switch can have one or more uplinks. Just as you can connect the uplink ports between the two physical switches in the virtual world, you can connect or uplink a virtual switch to a physical switch.

vSwitches allow you to make adjustments to your *Virtual Local Area Networks (VLANs)*, to some of your security settings, to your load balancing, and to your *Maximum Transmission Units (MTUs)*, which relate to the size of data frames, as well as to other settings which are beyond the scope of this course.

Standard Switches



vSphere supports two types of virtual switches: the standard virtual switch (the vSwitch or *VSS*) and the distributed virtual switch (or *VDS*).

A standard switch works like a physical Ethernet switch. It detects which virtual machines are logically connected to each of its virtual ports and uses that information to forward traffic to the correct virtual machines. A standard switch can forward traffic internally between VMs within the same ESXi host, between VMs on different ESXi hosts, and between VMs and physical machines, and can link to external networks.

A vSphere standard switch consists of port groups, VMkernel adapters, and uplink ports. To provide network connectivity to hosts and virtual machines, you connect the physical NICs of the hosts to uplink ports on the standard switch. Virtual machines have network adapters (or vNICs) that you connect to port groups on the standard switch.

Every port group can use one or more physical NICs to handle its network traffic. If a port group does not have a physical NIC connected to it, VMs on the same port group can only communicate with each other and not with the external network.

To ensure efficient use of host resources on ESXi hosts, the number of ports of standard switches are dynamically scaled up and down. A standard switch on such a host can expand up to the maximum number of ports supported on the host.

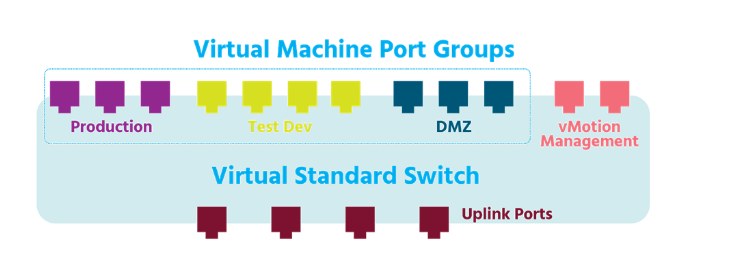
A VMkernel adapter is a port that is used by the hypervisor to attach a service to the network. Every VMkernel adapter has an IP address by which this service is accessible.

The uses of this VMkernel adapter include:

* VMware vMotion (which enables you to move VMs from one host to another while they're powered on with no downtime.
* Management port (which is used for ESXi management traffic and in most cases-except vSAN implementations- HA (or high availability) traffic)
* IP storage (which is any form of storage that uses TCP/IP network communication as its foundation)
* vSphere replication
* vSAN data replication

vSphere standard switches are created and configured on a per-host basis. So, if you have three hosts, you’ll need three virtual networks, three virtual switches, and three supporting port groups. Each host can have up to 4096 ports across both standard and distributed switches; a maximum of 1016 of these ports can be active at one time. Each standard switch can have up to 512 port groups.

Each logical port on the standard switch is a member of a single port group. Each port group on a standard switch is identified by a network label, which must be unique among the other port groups on a host, but consistent across hosts in order to ensure network connectivity.

Although by default vSwitch policies (security policies, for example) are automatically assigned to the vSwitch's port groups, port group policies can be configured manually.

vSphere network switches can be divided into two logical sections. The *data plane* carries out tasks such as packet switching, filtering (where a switch discards a frame that has the same source and destination MAC addresses), and tagging (where frames are tagged to indicate which VLAN they belong to). The *management plane* is where an administrator configures the functions of the data plane. Each vSphere standard switch contains both data and management planes, and the administrator configures and maintains each switch individually.

vSphere standard switches are supported by NSX-V, but not by NSX-T. Instead, VMware has developed a new virtual switch for the changing demands of modern networking.

Docker

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. By taking advantage of Docker’s methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production.

The Docker platform

Docker provides the ability to package and run an application in a loosely isolated environment called a container. The isolation and security allows you to run many containers simultaneously on a given host. Containers are lightweight and contain everything needed to run the application, so you do not need to rely on what is currently installed on the host. You can easily share containers while you work, and be sure that everyone you share with gets the same container that works in the same way.

Docker provides tooling and a platform to manage the lifecycle of your containers:

* Develop your application and its supporting components using containers.
* The container becomes the unit for distributing and testing your application.
* When you’re ready, deploy your application into your production environment, as a container or an orchestrated service. This works the same whether your production environment is a local data center, a cloud provider, or a hybrid of the two.

What can I use Docker for?

Fast, consistent delivery of your applications

Docker streamlines the development lifecycle by allowing developers to work in standardized environments using local containers which provide your applications and services. Containers are great for continuous integration and continuous delivery (CI/CD) workflows.

Consider the following example scenario:

* Your developers write code locally and share their work with their colleagues using Docker containers.
* They use Docker to push their applications into a test environment and execute automated and manual tests.
* When developers find bugs, they can fix them in the development environment and redeploy them to the test environment for testing and validation.
* When testing is complete, getting the fix to the customer is as simple as pushing the updated image to the production environment.

Responsive deployment and scaling

Docker’s container-based platform allows for highly portable workloads. Docker containers can run on a developer’s local laptop, on physical or virtual machines in a data center, on cloud providers, or in a mixture of environments.

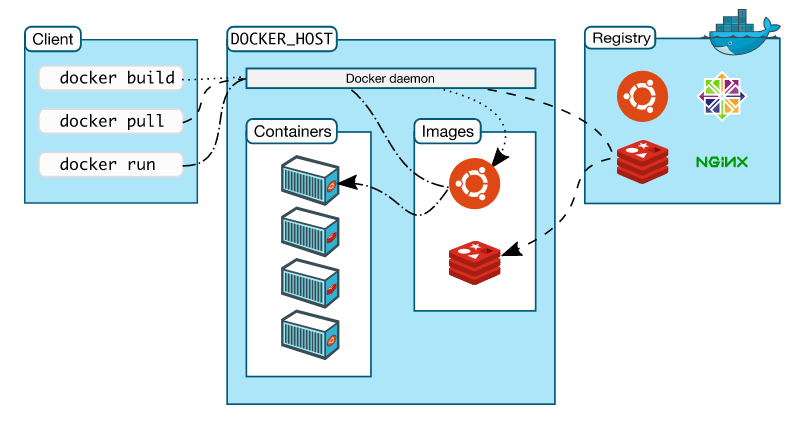
Docker’s portability and lightweight nature also make it easy to dynamically manage workloads, scaling up or tearing down applications and services as business needs dictate, in near real time.

Running more workloads on the same hardware

Docker is lightweight and fast. It provides a viable, cost-effective alternative to hypervisor-based virtual machines, so you can use more of your compute capacity to achieve your business goals. Docker is perfect for high density environments and for small and medium deployments where you need to do more with fewer resources.

Docker architecture

Docker uses a client-server architecture. The Docker *client* talks to the Docker *daemon*, which does the heavy lifting of building, running, and distributing your Docker containers. The Docker client and daemon *can* run on the same system, or you can connect a Docker client to a remote Docker daemon. The Docker client and daemon communicate using a REST API, over UNIX sockets or a network interface. Another Docker client is Docker Compose, that lets you work with applications consisting of a set of containers.



### The Docker daemon

The Docker daemon (dockerd) listens for Docker API requests and manages Docker objects such as images, containers, networks, and volumes. A daemon can also communicate with other daemons to manage Docker services.

### The Docker client

The Docker client (docker) is the primary way that many Docker users interact with Docker. When you use commands such as docker run, the client sends these commands to dockerd, which carries them out. The docker command uses the Docker API. The Docker client can communicate with more than one daemon.

### Docker Desktop

Docker Desktop is an easy-to-install application for your Mac or Windows environment that enables you to build and share containerized applications and microservices. Docker Desktop includes the Docker daemon (dockerd), the Docker client (docker), Docker Compose, Docker Content Trust, Kubernetes, and Credential Helper. For more information, see [Docker Desktop](https://docs.docker.com/desktop/).

### Docker registries

A Docker registry stores Docker images. Docker Hub is a public registry that anyone can use, and Docker is configured to look for images on Docker Hub by default. You can even run your own private registry.

When you use the docker pull or docker run commands, the required images are pulled from your configured registry. When you use the docker push command, your image is pushed to your configured registry.

### Docker objects

When you use Docker, you are creating and using images, containers, networks, volumes, plugins, and other objects. This section is a brief overview of some of those objects.

Kubernetes

## Kubernetes is software that automatically manages, scales, and maintains multi-container workloads in desired states

Modern software is increasingly run as fleets of containers, sometimes called microservices. A complete application may comprise many containers, all needing to work together in specific ways. Kubernetes is software that turns a collection of physical or virtual hosts (servers) into a platform that:

* Hosts containerized workloads, providing them with compute, storage, and network resources, and
* Automatically manages large numbers of containerized applications — keeping them healthy and available by adapting to changes and challenges

## How does Kubernetes work?

1. When developers create a multi-container application, they plan out how all the parts fit and work together, how many of each component should run, and roughly what should happen when challenges (e.g., lots of users logging in at once) are encountered.
2. They store their containerized application components in a container registry (local or remote) and capture this thinking in one or several text files comprising a*configuration*. To start the application, they “apply” the configuration to Kubernetes.
3. Kubernetes job is to evaluate and implement this configuration and maintain it until told otherwise. It:
   * Analyzes the configuration, aligning its requirements with those of all the other application configurations running on the system
   * Finds resources appropriate for running the new containers (e.g., some containers might need resources like GPUs that aren’t present on every host)
   * Grabs container images from the registry, starts up the new containers, and helps them connect to one another and to system resources (e.g., persistent storage), so the application works as a whole
4. Then Kubernetes monitors everything, and when real events diverge from desired states, Kubernetes tries to fix things and adapt. For example, if a container crashes, Kubernetes restarts it. If an underlying server fails, Kubernetes finds resources elsewhere to run the containers that node was hosting. If traffic to an application suddenly spikes, Kubernetes can scale out containers to handle the additional load, in conformance to rules and limits stated in the configuration.

## Why use Kubernetes?

One of the benefits of Kubernetes is that it makes building and running complex applications much simpler. Here’s a handful of the many Kubernetes features:

* Standard services like local DNS and basic load-balancing that most applications need, and are easy to use.
* Standard behaviors (e.g., restart this container if it dies) that are easy to invoke, and do most of the work of keeping applications running, available, and performant.
* A standard set of abstract “objects” (called things like “pods,” “replicasets,” and “deployments”) that wrap around containers and make it easy to build configurations around collections of containers.
* A standard API that applications can call to easily enable more sophisticated behaviors, making it much easier to create applications that manage other applications.

The simple answer to “what is Kubernetes used for” is that it saves developers and operators a great deal of time and effort, and lets them focus on building features for their applications, instead of figuring out and implementing ways to keep their applications running well, at scale.  
  
By keeping applications running despite challenges (e.g., failed servers, crashed containers, traffic spikes, etc.) Kubernetes also reduces business impacts, reduces the need for fire drills to bring broken applications back online, and protects against other liabilities, like the costs of failing to comply with Service Level Agreements (SLAs).

## Where can I run Kubernetes?

Kubernetes also runs almost anywhere, on a wide range of Linux operating systems (worker nodes can also run on Windows Server). A single Kubernetes cluster can span hundreds of bare-metal or virtual machines in a datacenter, private, or any public cloud. Kubernetes can also run on developer desktops, edge servers, microservers like Raspberry Pis, or very small mobile and IoT devices and appliances.  
  
With some forethought (and the right product and architectural choices) Kubernetes can even provide a functionally-consistent platform across all these infrastructures. This means that applications and configurations composed and initially tested on a desktop Kubernetes can move seamlessly and quickly to more-formal testing, large-scale production, edge, or IoT deployments. In principle, this means that enterprises and organizations can build “hybrid” and “multi-clouds” across a range of platforms, quickly and economically solving capacity problems without lock-in.

## What is a Kubernetes cluster?

The K8s architecture is relatively simple. You never interact directly with the nodes hosting your application, but only with the control plane, which presents an API and is in charge of scheduling and replicating groups of containers named Pods. Kubectl is the command line interface that allows you to interact with the API to share the desired application state or gather detailed information on the infrastructure’s current state.  
  
Let’s look at the various pieces.

#### Nodes

Each node that hosts part of your distributed application does so by leveraging Docker or a similar container technology, such as Rocket from CoreOS. The nodes also run two additional pieces of software: kube-proxy, which gives access to your running app, and kubelet, which receives commands from the k8s control plane. Nodes can also run flannel, an etcd backed network fabric for containers.

#### Master

The control plane itself runs the API server (kube-apiserver), the scheduler (kube-scheduler), the controller manager (kube-controller-manager) and etcd, a highly available key-value store for shared configuration and service discovery implementing the Raft consensus Algorithm.

## What is “enterprise Kubernetes?”

Kubernetes, by itself, provides a core software framework for container and resource management, default services, plus an API. It’s engineered to be extensible via standard interfaces to provide important capabilities like:

* Running containers – a container runtime or ‘engine’
* Letting containers communicate – a container network
* Providing persistent storage – a container storage solution
* Routing inbound traffic to containers in a secure and orderly way – an ingress solution
* Full-featured load balancing – distributing inbound traffic evenly to container workloads – via integration with an external load-balancing solution

… and many other components essential for efficient use and operations at scale. To make Kubernetes work at all — you or someone else needs to choose and integrate solutions to fill these critical slots.  
  
Kubernetes alternatives made available free of charge typically select from among open source alternatives to provide these capabilities. These are often very good solutions for learning and small-scale use.  
  
Organizations that want to use Kubernetes to run production software at scale need more, and more-mature functionality:

* They need Kubernetes that’s feature-complete, hardened and secure, and easily integrated with centralized IT resources like directory services, monitoring and observability, notifications and ticketing, and so on.
* They need Kubernetes that can be deployed, scaled, managed, and updated in consistent ways, perhaps across many different kinds of infrastructure.
* They need all the different parts of Kubernetes to be validated together, and supported by a single vendor.

“Enterprise Kubernetes” refers to products and suites of products that answer these needs: that fill all of Kubernetes’ feature slots with best-of-breed solutions, solve problems of Kubernetes management across multiple infrastructures, enable consistency, and provide complete support.

## How do I start using Kubernetes?

Mirantis makes several Kubernetes solutions, appropriate for different uses. Our open source products can be used free of charge, with community support. Our flagship products can be trialed free of charge and are available with tiered support up to fully-managed services.  
  
[Mirantis Container Cloud](https://www.mirantis.com/software/docker/docker-enterprise-container-cloud/) (formerly Docker Enterprise Container Cloud) is a solution for deploying, observing, managing, and non-disruptively updating Kubernetes (plus other applications that run on top of Kubernetes, like containerized OpenStack) on any infrastructure — ideal if you need to run Kubernetes reliably at scale with security, simplicity, and freedom of choice. ([Download Mirantis Container Cloud](https://www.mirantis.com/download/container-orchestration/mirantis-container-cloud/))  
  
[Mirantis Kubernetes Engine](https://www.mirantis.com/software/mirantis-kubernetes-engine/) (formerly Docker Enterprise/UCP) is fully-baked Enterprise Kubernetes for development, testing, and production. It includes the Universal Control Plane webUI for easy management, Mirantis Secure Registry (formerly Docker Trusted Registry) for private container image storage and security scanning, and runs on Mirantis Container Runtime (formerly Docker Engine – Enterprise) — a hardened container runtime with optional FIPS 140-2 encryption and other security and reliability features. ([Download Mirantis Kubernetes Engine](https://www.mirantis.com/download/container-orchestration/mirantis-kubernetes-engine/))  
  
[K0S](https://k0sproject.io/) – (pronounced “K-zeroes”) is zero-friction, open source Kubernetes that starts with a single command and runs on almost any Linux at almost any scale, from Raspberry Pis to giant datacenters. It’s our best choice for learners. ([Download k0s – zero friction Kubernetes](https://www.mirantis.com/download/open-source/k0s-zero-friction-kubernetes/))  
  
Finally, [Lens – the open source Kubernetes IDE](https://www.mirantis.com/software/lens/), accelerates Kubernetes learning and development. Lens lets you manage and interact with multiple Kubernetes clusters easily using a context-aware terminal, visualize object hierarchies inside them, view container logs, log directly into container command shells, and more.