

cloud computing

by Talwinder Kaur

Submission date: 29-Apr-2023 01:25PM (UTC+0100)

Submission ID: 203980429

File name: 45827_Talwinder_Kaur_cloud_computing_488474_477163080.docx (984.9K)

Word count: 5060

Character count: 30939



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

NAME: TALWINDER KAUR

STUDENT ID:2227879

SUBJECT: CLOUD ENGINEERING PRINCIPLE AND PRACTICE

PROFESSOR: Mike Dacey/Tim Bashford

Abstract

The definitions of terms and concepts used to describe "cloud computing" are constantly changing, and it is frequently necessary to provide clarification. Press coverage occasionally describes how businesses are making their solutions available in the "cloud" or how "cloud computing" is the way things are done, which can be ambiguous or fail to completely convey what cloud computing comprises or signifies. The characteristics, service models, and deployment patterns now in use, along with the advantages and difficulties of cloud computing, are explored in this white paper, which also introduces internet-based cloud computing. In addition, the relevance of scalability and flexibility in a cloud-based system as well as cloud communications services (including cloud access methods like web APIs and media management interfaces) are covered. The interface options available, including Web 2.0 APIs, media control interfaces, Java interfaces, and XML-based interfaces, catering to a wide spectrum of application and service creation developers, should also be recognized by organizations wishing to begin using communication services.

Table of contents

1.1. Introduction.....	
1.2. What is cloud computing?.....	
1.3. Why the Name Cloud?.....	
1.4. Types of cloud computing?.....	
1.5 What is cloud architecture?.....	
2. What is the docker?.....	
2.2 Why is docker?.....	
3.What is containerization?.....	
3.2. Benefits of containerization.....	
3.3. What is the scalability in cloud computing?.....	
3.4. Advantage Of the cloud scalability.....	
4 What is the load balancing in cloud computing?.....	
4.2 Objective of load balancing.....	
4.3Benefits of load balancing	
5.What are the application gateways?.....	
5.2. Application Gateway background.....	
5.3. Advantages.....	
5.4. Drawbacks.....	
6. Reference list.....	

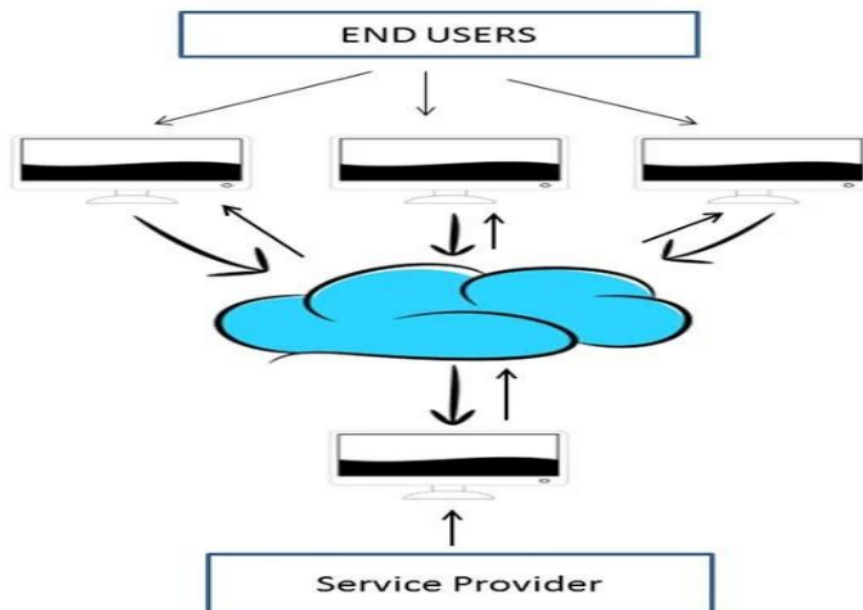
INTRODUCTION

What is cloud computing?

Utilizing hosted services including data storage, servers, databases, networking, and software over the internet is called cloud computing. Physical servers that are kept up by a cloud service provider house the data. In cloud computing, computer system resources, particularly data storage and processing power, are made on-demand and are not under the direct control of the user. Cloud computing services examples include AWS, Azure, and Google [1]

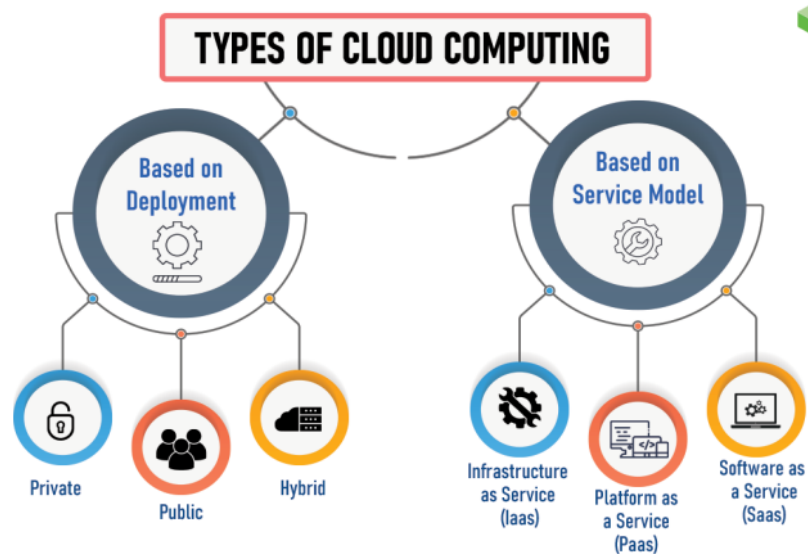
Why the name cloud?

The word "Cloud" was coined by network engineers to describe a network design that showed where different network devices were located and how they were connected. This network's architecture had a cloud-like shape



Types of cloud computing

The deployment model or the kind of service can be used to categories cloud computing. We can divide clouds into three categories: public, private, and hybrid, depending on the particular deployment architecture. The cloud model can be categorized as infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), and software-as-a-service (SaaS) depending on the service it provides.[1]



Private cloud

In a private cloud, computing services are provided through a private IT network just for the benefit of one particular business. A private cloud, which is also known as an internal, enterprise, or corporate cloud, is often operated using internal resources and is not available to anybody outside the company. All the advantages of a public cloud, like self-service, scalability, and elasticity, are also offered by private cloud computing, in addition to more control, security, and customization. Private clouds offer a higher level of security via corporate firewalls and internal hosting to guarantee that a company's sensitive data is inaccessible to outside providers. Private cloud has the disadvantage that the business is now in charge of managing and maintaining the data centers, which can be highly resource intensive.[1]

Public cloud

The term "public cloud" describes online computing services provided by independent vendors. The services offered by public clouds can be used or purchased by anybody, unlike private clouds. These services may be offered for free or on a pay-per-use basis, with customers only paying for the CPU time, storage space, or bandwidth that they actually use.[1]

Since the cloud service provider is in charge of system management, public cloud can assist enterprises in saving on the acquisition, administration, and maintenance of on-premises infrastructure. Additionally, they provide flexible bandwidth and scalable RAM, which makes it simpler for enterprises to grow their storage requirements.[1]

Hybrid cloud

A hybrid cloud combines elements from both public and private clouds. When computer and pricing requirements change, workloads can be moved between private and public clouds using the "best of both worlds" cloud concept. Hybrid clouds are used when the demand for processing and computation fluctuates. To handle the overflow, organizations can move their on-premises infrastructure up to the public cloud while ensuring that no outside data centers have access to their data. This opens a new window. Instead of purchasing and maintaining resources that may not be used for a long time, businesses using a hybrid cloud model only pay for the resources they temporarily need. A hybrid cloud provides public cloud advantages without the security dangers.[1]

Infrastructure as a service (IaaS)

In a form of cloud computing known as infrastructure as a service, or IaaS, a service provider oversees offering servers, storage, and networking over a virtual interface. With this service, the user retains control over the installed application, operating systems, and storage without having to manage the cloud infrastructure. A third-party vendor hosts the servers, storage, servers, and other infrastructure parts in place of the user. Additionally, the vendor hosts the user's applications and keeps a backup.[1]

Platform as a service (PaaS)

PaaS, or platform as a service, is a sort of cloud computing that offers an environment for development and deployment in the cloud that enables users to create and operate applications without having to worry about the complexity of creating or maintaining the infrastructure. It offers customers the tools they need to create cloud-based applications. In this kind of service, a user pays a vendor on a pay-as-you-go basis for the resources and can access them via a secure connection.[1]

Users are given control over the deployed apps but are not required to handle the underlying infrastructure, such as the network, servers, operating systems, or storage, thanks to PaaS. As a result, organizations are relieved of the duty of software maintenance, planning, and resource procurement, allowing them to concentrate on the deployment and management of their applications.[1]

Software as a service (SaaS)

SaaS or software as a service allows users to access a vendor's software on cloud on a subscription basis. In this type of cloud computing, users do not need to install or download applications on their local devices. Instead, the applications are located on a remote cloud network that can be directly accessed through the web or an API.[1]

In the SaaS model, the service provider manages all the hardware, middleware, application software, and security. Also referred to as 'hosted software' or 'on-demand software,' SaaS makes it easy for enterprises to streamline their maintenance and support.[1]

What is cloud computing architecture?

A crucial component of cloud construction is cloud architecture. It describes the organization and links all the elements and technology required for cloud computing. When compared to on-premises environments, moving to the cloud can provide many business advantages, including increased agility and scalability as well as cost savings. Many businesses may initially adopt a "lift-and-shift" strategy, in which existing on-premises applications are transferred to the cloud with little to no modification. However, in the end, it will be required to build and deploy apps that are compatible with cloud environments.

The elements of cloud architecture

Among the elements of cloud architecture are:

A frontend platform

A backend platform

A cloud-based delivery model

A network (internet, intranet, or intercloud)

Cloud architecture layers

Consider all these elements as different layers stacked on top of one another to form a cloud platform for a clearer grasp of how cloud architecture functions.

Hardware: The Hardware components that fuel the cloud, such as servers, storage, networks, and other things.

Virtualization: A layer of abstraction that transforms physical processing and storage resources into a virtual representation. Multiple programmers can use the same resources thanks to this.

Application and service: Assisting with requests from the frontend user interface, this layer coordinates and supports them while providing a variety of cloud service model-based services, such as resource management, application development tools, and web-based application.

What is the docker?

Containers are standardized, executable components that integrate application source code with the operating system (OS) libraries and dependencies necessary to run that code in any environment. Docker is an open-source platform that empowers developers to build, distribute, operate, update, and manage containers.

Distributed application development and delivery are made easier by containers. As businesses switch to cloud-native development and hybrid multicloud settings, they have grown in popularity. Developers can build containers without Docker by interacting directly with features included in Linux and other operating systems. However, Docker accelerates, simplifies, and secures containerization.

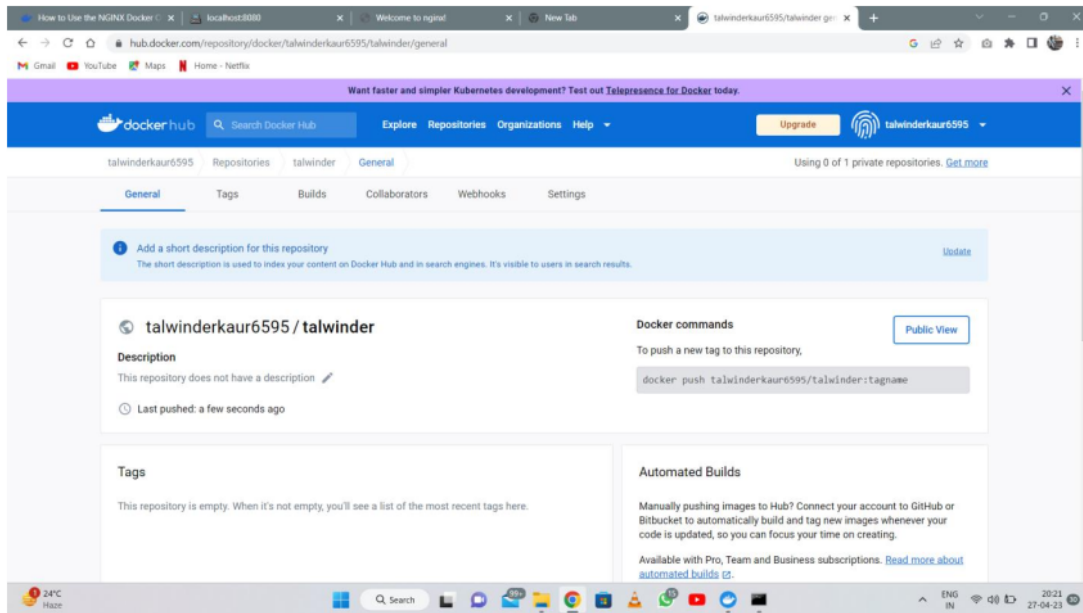
In addition, the term "Docker" can refer to both the open-source project that Docker, Inc. as well as numerous other companies and people participate in, as well as the business that sells Docker in its commercial form.

Why use Docker?

Nowadays, "Docker" and "containers" are used interchangeably because of Docker's widespread use. However, the initial container-related technologies existed for years, if not decades.

The Linux kernel's implementation of Linux Containers (LXC) in 2008 was particularly significant because it fully enabled virtualization for a single instance of Linux. Although LXC is still in use today, there are newer technologies that use the Linux kernel. This functionality is also offered by the contemporary, open-source Linux operating system Ubuntu.

Docker provides easy to use commands that provide developers access to these native containerization features, and an application programming interface (API) that automates them to save time. In contrast to LXC, Docker provides:



Improved and seamless container portability: LXC containers frequently make use of machine-specific parameters, whereas Docker containers operate unaltered on any desktop, data center, and cloud **environment**.

Even lighter weight and more granular updates: Multiple processes can be consolidated into a single container using LXC. This makes it feasible to create applications that can function even while one of their components is offline for maintenance or repair.

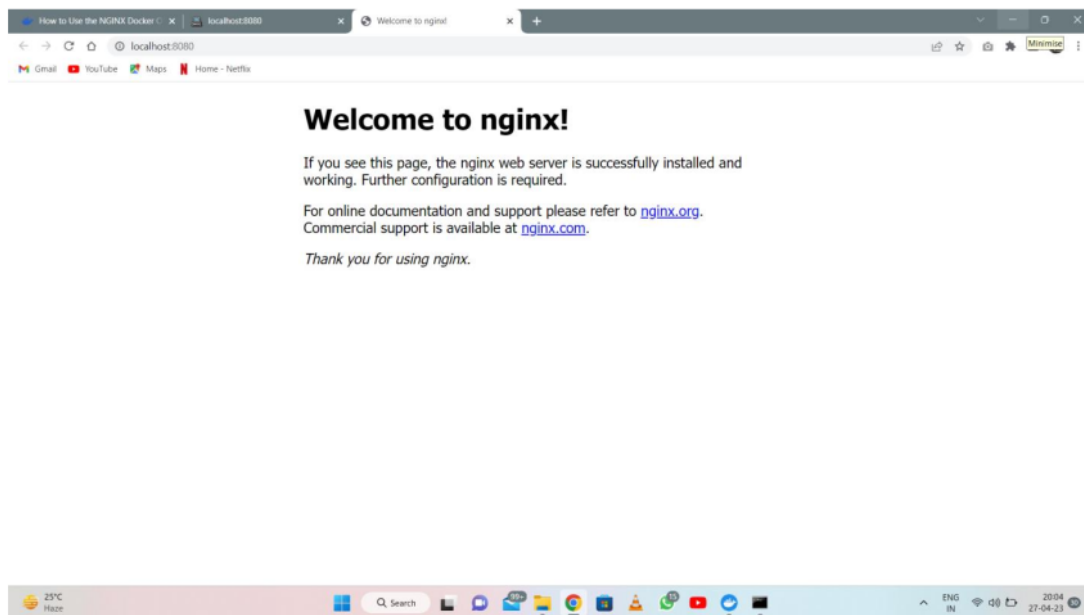
Automated container creation: Based on the application source code, Docker can automatically create a container.

Container versioning: Docker has the ability to log changes made to a container image, revert to a previous version, and discover who made a

version and how. Even just the differences between an existing version and an updated version might be uploaded.

Container reuse: Base images—essentially, templates for creating new containers—can be utilized with existing containers.

Shared container libraries: Thousands of user-contributed containers are available in an open-source registry that developers can access.



Today Both Apple MacOS and Microsoft Windows are compatible with Docker containerization. Every major cloud provider, including Amazon Web Services (AWS), Microsoft Azure, and IBM Cloud, offers specialized services to assist developers in creating, deploying, and running applications that are containerized with Docker. Developers can run Docker containers on any operating system.

What is containerization?

Software can be packaged such that it can be run in separate settings using a technique called containerization. The code, libraries, and dependencies required to operate an application are packaged in containers, which run on top of the host operating system. This implies that several containers can share the kernel of the host operating system while running on a single real or virtual machine. Containers offer a variety of advantages, such as the simplicity with which applications may be packaged and deployed, the consistency with which they can function in each environment, and the conflict-free operation of numerous apps on a single computer. The deployment, scaling, and networking of containers in a distributed system are managed by container orchestration tools, which are frequently used in conjunction with containers. Kubernetes and Docker Swarm are two common solutions for container orchestration. [2]

Benefits of containerization

Using containerization has several advantages.

- **Portability:** It is simple to deploy an application on any system running a container runtime, such as Docker, thanks to containers, which enable you to package an application together with all its dependencies in a single package.[2]
- **Consistency:** Whether an application is being created on a developer's local PC or is being deployed in a production environment, containers guarantee that it will function identically in any setting.[2]
- **Resource efficiency:** Virtual machines must have an operating system installed for each one, however containers are lighter than virtual machines. As a result, more containers can run on a single real or virtual computer, and scalability in container deployment and management are both facilitated.[2]

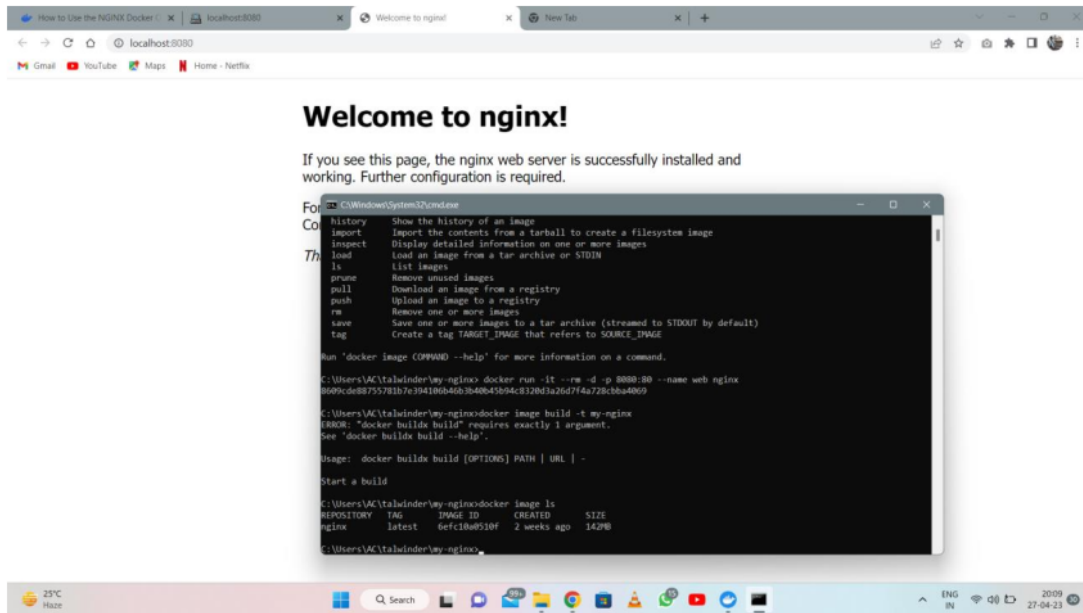
- **Isolation:** Applications can run independently of one another on a single system thanks to containers, which provide separation between them.[2]
- **Ease of use:** Application packaging, deployment, and management are made simple by containerization, and a variety of tools and platforms are available to assist with these processes.[2]
- **Security:** Because they keep applications apart from one another and from the host operating system, containers give an extra degree of protection.[2]

What the portability

Portability, in relation to software, is a measure of how easily an application can be transferred from one computer environment to another. A computer software application is considered portable to a unique environment if the effort required to adapt it to the pristine environment is within reasonable limits. The meaning of the abstract term 'reasonable' depends upon the nature of the application and is often difficult to express in quantifiable units.

The phrase "to port" means to modify software and make it adaptable to work on a different computer system. For example, to port an application to Linux means to modify the program so that it can be run in a Linux environment.

The term "portability" describes an application's capacity to move not just between platforms but also between environments. To be clear, a computer platform typically only relates to the hardware and operating system of a computer. The hardware, operating system, and interfaces with other applications, users, and programmers are all parts of the much larger computing environment.



WHAT IS THE CONTAINER IMAGE AND PUBLISH IMAGE?

A packed application, its dependencies, and details about the processes it launches are all contained in a container image.

```
C:\Users\AC>docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    6efc10a0510f   2 weeks ago    142MB

C:\Users\AC>docker run -d -v volume-high:/app/data nginx:latest
059d92a86979d8c85ceae37d513f608885892781d30ed5cdbc2ac7326aa7ee4a

C:\Users\AC>
```

How does that work?

It is crucial to comprehend the following ideas in this order if you want to understand container images:

1. Union file systems
2. Copy-on-Write
3. Overlay File Systems
4. Snapshotters

Union File Systems (Aufs) The Linux kernel includes the Union File System (UnionFS), which enables the merging of data from different file systems while maintaining the separation of the "physical" data. The data is actually organized in branches, but the end result is a unified file system.

The concept behind this is that if you have numerous photos with the same data, rather than copying it over again, it will be shared by using something called a layer.

Copy-on-write

A container looks to have its own whole file system when you first start it. This implies that each container you run on the system requires a separate copy of the

file system. This would consume a lot of disc space and take a long time to boot the containers. No, each container does not require a separate copy of the filesystem.

A copy-on-write method is used by containers and images to do this. The copy-on-write technique shares the same instance of data with numerous processes instead of copying files, and it only makes copies when a process needs to edit or write data. The original data would continue to be used in all other operations. A copy of the file that will be updated is first put on the container's writeable layer before any write operations are carried out.

Overlay File System

An overlay sits on top of an existing filesystem, combines an upper and lower directory tree, and presents them as a single directory. These directories are called *layers*. The lower layer remains unmodified. Each layer adds only the difference (the *diff*, in computing terminology) from the layer below it, and this unification process is referred to as a *union mount*.

Snapshotters

Layers and graph drivers are used by containers to build, manage, and distribute changes as a part of their container filesystem. However, using graph drivers is extremely challenging and prone to mistakes. Because they are unaware of images or containers, SnapShotters are distinct from graph drivers.

What are the container volumes

The parent container must be removed using the `docker rm -v` command (the `-v` is crucial) or the `-rm` flag must have been specified when running docker. Only then are volumes automatically erased. The volume will not be destroyed even in that case if another container does not relate to it. Volumes connected to user-specified host directories are never destroyed by Docker.

Although there are a lot of volumes, it should be fairly intuitive once you understand the underlying philosophy. Being able to use volumes successfully is crucial for a productive Docker workflow, therefore it is worthwhile to experiment with the commands until you fully grasp how they operate.

```
C:\Users\AC>docker volume create volume-high
volume-high
```

```
C:\Users\AC>_
```

```
C:\Users\AC>docker volume ls
DRIVER      VOLUME NAME
local       volume-high
```

```
C:\Users\AC>
```

What is container networking?

The notion of container networking, which is utilized in home desktops and web-scale enterprise networking solutions, is comparable to that of a virtual machine. Inside the container, a fully functional Linux environment with its own users, file system, processes, and network stack are isolated from the host and all other containers. All applications running inside the container are only allowed to access and alter resources present inside.

```
C:\Users\AC>docker network create waheguru
87352e2fa26cc0ed8bc31c23a8c4f02f693e23a44e6ea3dd958fb2626c7a9925

C:\Users\AC>docker run -d --network=waheguru --name=satnam nginx:latest
55f281c530f780c7e200b1251437daf9931625f82ec4cdffae8feb9574b27041
```

```
C:\Users\AC>docker network ls
docker: 'network' is not a docker command.
See 'docker --help'
```

```
C:\Users\AC>docker network ls
NETWORK ID        NAME        DRIVER        SCOPE
89aa905790df      bridge     bridge        local
b5dcf4ea2b7e      host       host          local
b7006c2e402f      none      null          local
87352e2fa26c      waheguru   bridge        local
```

```
C:\Users\AC>
```

Multiple containers, each with their own installations and dependencies, can run simultaneously. This is especially helpful when updating a dependency for newer versions of an application could conflict with updating dependencies for other applications that are already running on the server. Containers are smaller and faster than virtual machines and have less overhead because they share host resources rather than completely mimicking all the computer's hardware, unlike virtual machines. Containers were created as a replacement for VMs as a deployment platform for microservice architectures, particularly in the context of web-scale applications.

How Does Container Networking Work?

Virtualization in the form of a container network has distinct distinctions from virtual machines (VM) but is conceptually comparable. Comparing the container method to virtual machines (VMs), which are a type of hardware virtualization, the former is primarily an operating system virtualization.

Each virtual machine that runs on a hypervisor has its own operating system, programmers, and libraries and can encapsulate persistent data, install a new OS, use a different filesystem than the host, or use a different kernel version.

Containers, on the other hand, are a "running instance" of an image, a transient operating system virtualization that spins up to carry out some duties before being removed and forgotten. System users run far more instances of containers than virtual machines, which require a greater address space, due to the transitory nature of containers.

Benefits of Container Networking

Run Containerized Apps Alongside Existing Workloads: On the same infrastructure, machines can run containerized programs alongside traditional VMs, providing flexibility and speed.

Combine Portability with Security, Visibility, and Management: Sandboxing, resource transparency with the host, task management, and portable execution environments are all made possible by containers' inherent nature, which also provides for increased security.

Leverage Your Existing Infrastructure and Scale Easily: utilization of your current SDDC to avoid time-consuming and expensive infrastructure re-architecture that leads to silos. Silos happen when various departments retain their own IT infrastructure inside the same organization. Due to the variations in technical configurations across each department, this "silo effect" causes issues when implementing organization-wide IT policies and updates. By using container networking, the time- and money-consuming process of reintegrating silos can be avoided.

Provide Developers with a Docker-Compatible Interface: An interface that is compatible with Docker allows developers to create applications in containers, which can then be provisioned using a self-service management site or user interface.

Application Deployments at Scale Using Container Networking

Ease of Deployment: Host configurations can be embedded in containers making them ready to go once deployed.

Disposable: The setup and disposal of containers are made to be quick. A spare server can be easily added to resume application operation if the host fails.

Fault-Tolerant: Containers create easy redundancy for databases and web servers. Copying the same container over several nodes provides for high-availability and fault-tolerance.

Types of Container Networking

There are five types of container networking used today; their characteristics center around IP-per-container versus IP-per-pod models and the requirement of network address translation (NAT) versus no translation required.

None: The container is given a network stack, but it is disconnected from the outside world. This mode is helpful for assigning to containers that do not need external connectivity, testing containers, and staging a container for a future network connection.

Bridge: Containers that are enabled for communication with one another and are bridged on the internal host network of the same host. The host must be used to access containers. For Docker containers, the bridge network is the default.

Host: By sharing the host's network namespace, a generated container is given access to all its network interfaces thanks to this setup. Due to the common usage of networking interfaces, this sort of external networking configuration, which is the least sophisticated, is vulnerable to port disputes.

Underlay: Underlays are more effective than bridges since they do not require port-mapping and open the host interfaces directly to containers that are executing on the host.

Overlay: When containers are hosted on different hosts, overlays make advantage of networking tunnels to communicate between them and make it appear as though they are on the same machine.

What is the load balancing in cloud computing?

Workloads and computer resources are divided among one or more servers using cloud load balancing. The best performance with the quickest response time is ensured by this distribution style. To make better use of resources and reduce the amount of time spent reacting to system operations, two or more servers, hard drives, network ports, or other computer resources are divided. Therefore, effective cloud load balancing may ensure that a website with high traffic continues to operate.

The phrase "load" refers to a variety of factors, including server storage capacity, CPU load, network load, and website traffic.

Every machine in the network is guaranteed to get a comparable amount of load at some arbitrary moment thanks to load balancing. They are all equally exploited and not in any way overworked as a result.

The load balancer distributes data based on how active each server or node is. The client will have to wait for his operation to be finished in the absence of a load balancer, which could be too taxing and demoralizing for him.

OBJECTIVES OF LOAD BALANCING

The following are the primary goals of load balancing:

1. Managing and regulating traffic peaks on a single server
2. Improving user request response times

BENEFITS OF LOAD BALANCING

High-Performance Applications: Compared to their traditional local counterparts, cloud load balancing strategies are more affordable and simpler to implement. Businesses might perhaps lower their costs while improving the speed and performance of their customer apps.

Improved scalability: Cloud balancing aids in maintaining the scalability and agility of website traffic. With efficient load balancers, you can easily match rising user traffic and spread it across multiple servers or network devices. It is extremely important for websites that handle thousands of internet users every second. To distribute workloads during sales or other promotional activities, such efficient load balancers are required.

Ability to handle spikes in traffic: During any result announcement, a normally operational University site could completely go down. This is because there is a chance of getting a lot of requests. If they make use of cloud load balancers, they will not need to be concerned about traffic spikes. Whatever the request's size; it can be thoughtfully distributed across several servers to produce the greatest results in the quickest amount of time.

Continuity with complete flexibility: By establishing a load balancer, the main objective is to protect or save a site from unexpected failures. When distributed over numerous servers or network units, the workload can be transferred to another active node even if one node fails.

What is the scalability in cloud computing?

Virtualization makes it feasible to build scalable cloud architectures. Virtual machines (VMs) are extremely adaptable and can be scaled up or down with ease, in contrast to real machines whose resources and performance are largely fixed. Workloads and applications can be migrated to larger VMs as necessary, or they can be hosted on many servers simultaneously.

Additionally, independent businesses are unable to scale quickly or profitably on their own. Third-party cloud providers already have the huge hardware and software resources needed to do this.

ADVANTAGES OF THE CLOUD SCALABILITY

Businesses of all sizes are adopting the cloud because of its significant scalability benefits:

Convenience: IT managers may frequently add more VMs with a few clicks that are immediately accessible, tailored to a specific organization's requirements, and ready right away. Thus, IT staff members can save valuable time. Teams can work on other projects rather than spending days and hours setting up real hardware.

Flexibility and speed: Cloud scalability enables IT to react fast as business demands evolve, including unanticipated increases in demand. Even smaller businesses can now utilise high-powered resources that were previously out of reach due to expense. Businesses are no longer restricted by outdated hardware because it is simple to update systems and boost power and storage.

Cost savings: Businesses no longer must incur the upfront costs of buying expensive equipment that can become obsolete in a few years thanks to cloud scalability. They reduce waste by only paying for what they actually utilise through cloud service providers.

Disaster recovery: By doing away with the need to construct and manage additional data centres, scalable cloud computing allows you to lower disaster recovery expenses.

What are the application gateways?

Web application security is provided via an application gateway. They take the place of conventional online apps, which demand the same login information as the data center. Instead, consumers use mobile apps and cloud services to access application gateways using their login information from the app.

The modern worker who frequently works from home or on the go has a solution in application gateways. Users need to access programmers that are hosted by data centers and cloud solutions whether they are working on a laptop, smartphone, or another device. As VPNs (Virtual Private Networks) also aid in security against unauthorized access, several firms employ it for this purpose. VPN (Virtual Private Networks) maintenance, though, can be expensive.

A simple-to-install alternative that does away with the requirement for a VPN is an application gateway. Software programmers known as app gateways provide the same level of security as VPNs (Virtual Private Networks) without requiring configuration changes and can be utilized in corporate data centers and normal servers.

Instead of using specialized endpoint protection, application gateways use conventional SSL encryption to safeguard corporate data moving over the internet.

Application Gateway background

With an Application Gateway installed, your network system is completely protected. What is the true goal, and why do you actually require one? First, customers must connect to the server in order to request resources from databases. To reach the main server, the consumer must go even one more step as a result of this. The server's firewall is home to the Application Gateway. This safeguards the client's data for storage by maintaining the security of their IP addresses and other sensitive information.

Without the assistance of or knowledge of the client regarding the IP address, the Application Gateway and external computer operate. The truth is that there are actually two different connections active at once, one of which is between the client and the server. The second is situated in between the server and the last endpoint.

When everything is set up correctly, the server will ultimately make the decision. This is because computers that are secured by a firewall are actually safe.

ADAVANTAGES

Numerous advantages exist. You get the same result as a firewall because of their filtering abilities. Information may also be kept secret from the network with its assistance, and the authentication feature is fully protected to the required level.

The user is provided with a detailed log-in traffic itinerary from the Application Gateway that shows everything that enters and leaves the network. When you dissect this power, you have a full record that can document any malicious efforts to hack into the network. Without the Application Gateway installed, it is difficult to determine when or how a break-in attempt was undertaken without this helpful functionality.

DROWBACKS

The minuses accompany the pluses. In comparison to a packet filter, an Application Gateway may have a slower impact. Additionally, taking it a step further, not every available connection can support this type of technology, which results in a less effective result. In the end, your business will benefit significantly more from having an Application Gateway than from not having one.

REFERENCE LISTS

1. Patil, Prajakta, and Chiradeep BasuMallick. "What Is Cloud Computing? Definition, Benefits, Types, and Trends." *Spiceworks*, 9 Feb. 2022, www.spiceworks.com/tech/cloud/articles/what-is-cloud-computing/.
2. Peterson, Richard. "What Is Cloud Computing? Examples, Types & Benefits." *Www.guru99.com*, 2 Jan. 2020, www.guru99.com/what-is-cloud-computing-with-example.html.
3. "What Is Cloud Architecture? Benefits & Components." *Google Cloud*, cloud.google.com/learn/what-is-cloud-architecture#:~:text=Cloud%20architecture%20refers%20to%20how.
4. IBM. "What Is Docker? | IBM." *Www.ibm.com*, 25 Apr. 2023, www.ibm.com/topics/docker.
5. Sarkar, Taniya. "The Complete Guide to Cloud Containerization." *Cyfuture Cloud Blog*, 9 Jan. 2023, cyfuture.cloud/blog/all-you-need-to-know-about-cloud-containerization/. Accessed 23 Apr. 2023.
6. rouse, Margaret . "What Is Portability? - Definition from Techopedia." *Techopedia.com*, 18 Aug. 2011, www.techopedia.com/definition/8921/portability.
7. Velayudhan, Nived. "What Is a Container Image? | Opensource.com." *Opensource.com*, 31 Aug. 2021, opensource.com/article/21/8/container-image. Accessed 28 Apr. 2023.
8. Mouat, Adrian. "Understanding Volumes in Docker." *Blog.container-Solutions.com*, 9 Dec. 2014, blog.container-solutions.com/understanding-volumes-docker. Accessed 28 Apr. 2023.

9. VMWARE. "What Is Container Networking? | VMware Glossary." *VMware*, 17 Dec. 2021, www.vmware.com/topics/glossary/content/container-networking.html.
10. Ms. Shalini Joshi, Dr. Uma Kumari "Load Balancing in Cloud Computing: Challenges & Issues," Conference: 2016 2nd International Conference on Contemporary Computing and Informatics, DOI:10.1109/IC3I.2016.7917945.
11. Muhammad Asim Shahid, Noman Islam, Muhammad Mansoor Alam, Mazliham Mohd Su'ud, Shahrulniza Musa, "A Comprehensive Study of Load Balancing Approaches in the Cloud Computing Environment and a Novel Fault Tolerance Approach." DOI: 10.1109/ACCESS.2020.3009184
12. VMware. "What Is Cloud Scalability? | VMware Glossary." *VMware*, 23 Mar. 2022, www.vmware.com/topics/glossary/content/cloud-scalability.html#:~:text=Cloud%20scalability%20in%20cloud%20computing.
13. Brown, Schuyler. "What Is an Application Gateway (App Gateway)? | StrongDM." *Discover.strongdm.com*, 31 Oct. 2022, www.strongdm.com/what-is/application-gateway. Accessed 24 Apr. 2023.
14. Siegel |, Camille. "Application Gateway | Definition and Benefits of an Application Gateway." *Axway Blog*, 31 July 2019, blog.axway.com/learning-center/digital-security/risk-management/application-gateway. Accessed 24 Apr. 2023.

FINAL GRADE

50 / 100

GENERAL COMMENTS

Instructor

Docker Overview + Benefits (20): 16.5

Docker Examples (30): 21.5

Video (5): 0

Cloud Comp tech (25): 12

Cloud Comp functions (20): 0

Feedback:

=====

Marking correct report that was emailed to me.

There is quite a lot of discussion of docker but the benefits could have been evaluated in a bit more detail. Docker examples demonstrated the fundamental concepts. Wanted an example that was a little more interesting.

The discussion of load balancing and scalability could have been a little more technical. Also, the discussion of application gateway refers to a software app rather than the cloud gateway that acts as a single entry point for services.

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

