NAME: Purvi Harniya

ROLL NO: 1814023

BATCH: A1, TY IT

DATE: 22/04/2021

1. Research Amazon EC2, Google Cloud Platform, and Azure and write a detailed comparison. Focus on the following comparative points:

a. Business model (IaaS, PaaS etc)

b. VM instance types offered (such as micro, small, medium, large etc.) along with their number of vCPUs and memory sizes

c. Storage

d. Development environment offered

e. OS environments offered

f. Security

g. Performance and scalability

h. Reliability and fault tolerance

i. Pricing model

j. Auto-Scaling/Elasticity

k. Monitoring tools/service provided

ANSWER:

| | **Amazon EC2** | **Google Cloud Platform** | **Azure** |
|---|---|---|---|
| Business model | IaaS, PaaS, SaaS | PaaS, IaaS | PaaS, IaaS |
| VM instance types offered | AWS EC2 provides- nano, medium, large and xlarge VM instances.<br>Hypervisor- Xen<br>Max vCPUs- 128 (X1)<br>Max Memory (GB)- 1952 (X1)<br>Max Attached Storage (GB)- 3840 (X1) | Compute Engine- nano, small ,medium, large VM instances.<br>Hypervisor- Hyper-V<br>Max vCPUs- 64<br>Max Memory (GB)- 416 (6.5/vCPU) | Azure Virtual Machines- nano, small ,medium, large VM instances.<br>Hypervisor- Hyper-V<br>Max vCPUs- 32 (G5)<br>Max Memory (GB)- 448 (G5) |

| | | | |
|---|---|---|---|
| | Max Instance Storage (GB)- 48000 (D2) | Max Attached Storage 4096 (64/vCPU)<br>Max Instance Storage (GB)- 64000 | Max Attached Storage 6598 (G5)<br>Max Instance Storage (GB)- 32000 |
| Storage | Main storage- Amazon Db<br>Object Storage- Amazon S3<br>Relational DB- Amazon RDS<br>Block Storage- Amazon EBS<br>File Storage- Amazon EFS | Main storage- Google db<br>Object Storage- Google cloud storage<br>Relational DB- Relational DBs<br>Block Storage- Persistent disks<br>File Storage- ZFS and Avere ZFS and Avere | Main storage- Microsoft Db<br>Object Storage- Block Blobs and files<br>Relational DB- Google Cloud SQL<br>Block Storage- Page Blobs<br>File Storage- Azure Files |
| Development environment offered | **AWS** Cloud9, an Integrated **Development Environment** (IDE) for writing, running, and debugging code | Google App Engine- a part of Google cloud platform, provides a developing and deploying environment for php, javascript, nodejs, java and GO programming languages. | Azure ML Studio provided by azure allows developing data science and machine learning applications on the go. |
| OS environments offered | Linux , Cent OS , Debian , Oracle Enterprise Linux , Red Hat Enterprise Linux , SUSE Enterprise Linux , Ubuntu , Windows Server | Linux , Cent OS , Debian , Linux , Red Hat Enterprise Linux , SUSE Enterprise Linux , Ubuntu , Windows Server | Cent OS , FreeBSD , openSUSE Linux ,Oracle Enterprise Linux , SUSE Enterprise Linux , Ubuntu ,Windows Server |
| Security | System Security- Firewall filters (Security groups) ,SSH , VPN access<br>User Security- User credentials provide through web interface | System Security- Firewall filters ,SSH , VPN access,<br>User Security- User credentials provide through web interface | System Security- Firewall filters ,SSH<br>User Security- User credential using certificate authority for VPN access |
| Performance and scalability | optimize network performance, increase network security, and reduce troubleshooting time. | optimize network performance, increase network security. | increase network security, and reduce troubleshooting time.<br>Host encryption keys |

| | | | |
|---|---|---|---|
| | Perform vulnerability scans on container images. | | and perform cryptographic operations in a cluster . |
| Reliability and fault tolerance | Amazon Simple Queue Service (SQS) offers a reliable way for messages to travel between applications. It stores in-flight messages and does not require applications to be always available. Amazon Simple Notification Service (SNS) provides another way for messaging. Messages published from an application will be delivered to subscribers immediately.<br>EC2 also offers Auto Scaling and Elastic Load Balancing services. Auto Scaling allows users to scale up/down their EC2 capacity automatically when pre-define events are triggered. Elastic Load Balancing automatically distributes incoming traffic across multiple Amazon EC2 instances, which brings improved responsiveness as well as fault tolerance. | GCP offers Auto Scaling and Elastic Load Balancing services by using Cloud Load Balancing for auto-scaling. Auto Scaling allows users to scale up/down their capacity automatically when pre-define events are triggered. Elastic Load Balancing automatically distributes incoming traffic across multiple. GCP instances, which brings improved responsiveness as well as fault tolerance. | Azure offers "Service Bus" for communications between applications or services. Service Bus provides three options to meet different communication requirements. Like Amazon SQS, queues provide FIFO guaranteed message delivery communication. But it is just one-directional and serves as a broker that stores sent messages until they are received. |
| Pricing model | Per hour<br>Smallest Instance: A basic instance includes 2 virtual CPUs and 8 GB of RAM. Its cost is around US$69 per month.<br>Largest Instance: AWS's largest instance includes 3.84 TB of RAM and 128 vCPUs. It costs around US$3.97/hour. | Per minute<br>Small Instance: In Comparison with AWS and Azure, GCP provides the most basic instance, including 2 virtual CPUs and 8 GB of RAM at a 25% cheaper rate. So, it's cost will be around US$52/month. | Per minute<br>Smallest Instance: Azure small instance includes 2 vCPUs and 8 GB of RAM. Its cost is around US$70/month.<br>Largest Instance: The Azure largest instance includes 3.89 TB of RAM and 128 vCPUs. Its |

| | | | |
|---|---|---|---|
| | | Largest Instance: GCP's largest instance includes 3.75 TB of RAM and 160 vCPUs. It will cost around US$5.32/hour. | pricing is around US$6.79/hour. |
| Auto-Scaling/Elasticity | Uses Elastic Load Balancing for auto-scaling | Uses Cloud Load Balancing for auto-scaling | Uses Load Balancer Application Gateway for auto-scaling |
| Monitoring tools/service provided | AWS CloudTrail , AWS CloudWatch | Cloud Stackdriver, Monitoring, Logging , Error Reporting, Trace | Azure Log Analytics, Azure Application Insights,<br>Azure Portal |

2. Compare Cloud Containers vs. VMs and explain where each should be used.

Discuss the scalability of containerized workloads and VM workloads.

Answer:

**Virtual Machine vs Container:**

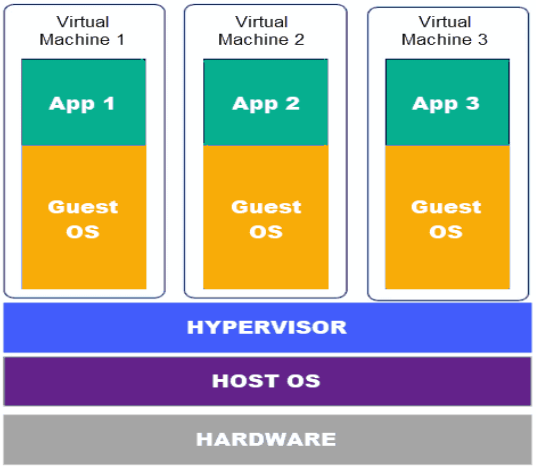
In traditional virtualization, a hypervisor virtualizes physical hardware. The result is that each virtual machine contains a guest OS, a virtual copy of the hardware that the OS requires to run and an application and its associated libraries and dependencies. VMs with different operating systems can be run on the same physical server. For example, a VMware VM can run next to a Linux VM, which runs next to a Microsoft VM, etc.

Instead of virtualizing the underlying hardware, containers virtualize the operating system (typically Linux or Windows) so each individual container contains *only* the application and its libraries and dependencies. Containers are small, fast, and portable because, unlike a virtual machine, containers do not need to include a guest OS in every instance and can, instead, simply leverage the features and resources of the host OS.

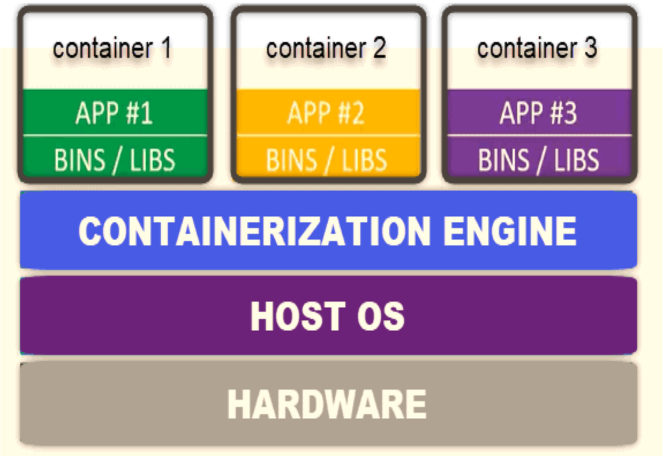
Just like virtual machines, containers allow developers to improve CPU and memory utilization of physical machines. Containers go even further, however, because they also enable microservice architectures, where application components can be deployed and scaled more granularly. This is an attractive alternative to having to scale up an entire monolithic application because a single component is struggling with load.

Virtual Machine vs Container:

VM:

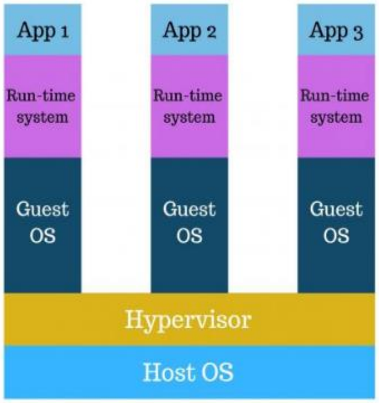


Container:

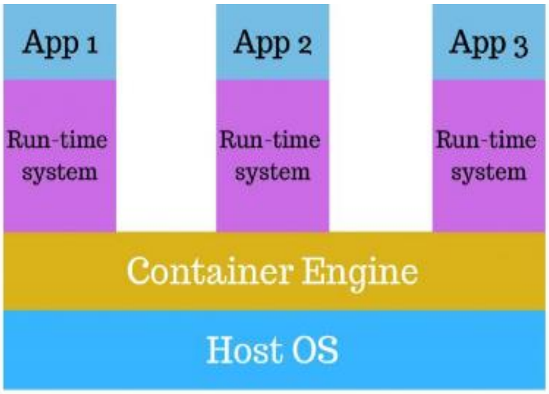


Simplified Architecture:

VM:



Container:



| VM | CONTAINER |
|---|---|
| A virtual machine (VM) is an operating system that shares the physical resources of one server. It is a guest on the host's hardware, which is why it is also called a guest machine. | A container is an environment that runs an application that is not dependent on the operating system. It isolates the app from the host by virtualizing it. This allows users to created multiple workloads on a single OS instance. |

| | |
|---|---|
| There are several layers that make up a virtual machine. The layer that enables virtualization is the hypervisor. A hypervisor is a software that virtualizes the server. | The kernel of the host operating system serves the needs of running different functions of an app, separated into containers. Each container runs isolated tasks. It cannot harm the host machine nor come in conflict with other apps running in separate containers. |
| PROS:<br>VMs reduce expenses. Instead of running an application on a single server, a virtual machine enables utilizing one physical resource to do the job of many. Therefore, you do not have to buy, maintain and store enumerable stacks of servers.<br>Because there is one host machine, it allows you to efficiently manage all the virtual environments with the centralized power of the hypervisor. These systems are entirely separate from each other meaning you can install multiple system environments.<br>Most importantly, a virtual machine is isolated from the host OS and is a safe place for experimenting and developing applications. | PROS:<br>Containers can be as small as 10MB and you can easily limit their memory and CPU usage. This makes containers<br>remarkably lightweight and fast to launch as opposed to deploying virtual machines, where the entire operating system needs to be deployed.<br>Because of their size, you can quickly scale in and out of containers and add identical containers.<br>Also, containers are excellent for Continuous Integration and Continuous<br>Deployment (CI/CD) implementation. They foster collaborative development by distributing and merging images among developers. |
| CONS:<br>Virtual machines may take up a lot of system resources of the host machine, being many GBs in size. Running a single app on a virtual server means running a copy of an operating system as well as a virtual copy of all the hardware required for the system to run. This quickly adds up to a lot of RAM and CPU cycles.<br>The process of relocating an app running on a virtual machine can also be complicated as it is always attached to the operating system. Hence, you have to migrate the app as well as the OS with it. Also, when creating a | CONS:<br>A container uses the kernel of the host OS and has operating system dependencies. Therefore, containers can differ from the underlying OS by dependency, but not by type. The host’s kernel limits the use of other operating systems.<br>Containers still do not offer the same security and stability that VMs can. Since they share the host’s kernel, they cannot be as isolated as a virtual machine. Consequently, containers are process-level isolated, and one container can affect others by compromising the stability of the kernel. |

| | |
|---|---|
| virtual machine, the hypervisor allocates hardware resources dedicated to the VM.<br>A virtual machine rarely uses all the resources available which can make the planning and distribution difficult. That’s still economical compared to running separate actual computers. | Moreover, once a container performs its task, it shuts down, deleting all the data inside of it. If you want the data to remain on the host server, you have to save it using Data Volumes. This requires manual configuration and provisioning on the host. |

WHERE SHOULD VM AND CONTAINER BE USED:

**Virtual machines are a better solution if we need to:**

1. Manage a variety of operating systems
2. Manage multiple apps on a single server
3. Run an app that requires all the resources and functionalities of an OS
4. Ensure full isolation and security

**Containers are suitable if we need to:**

1. Maximize the number of apps running on a server
2. Deploy multiple instances of a single application
3. Have a lightweight system that quickly starts
4. Develop an application that runs on any underlying infrastructure

**Scalability of containerized workloads and VM workloads:**

There is a vast difference between the scalability of a containerized workload and a virtual workload. The containers contain only those services which are basic in nature and which their functions require, but among those services, one can be a web server like Nginx and virtualization workload system, like cabernets, having the capability of judging that when there is a need of scaling out the number of containers based on the sequence of traffic follows and can copy the images of the container on its own and also remove them from the system.