**About yourself**

I am Lakshmi Chalamalasetty. I did my Masters in Computer Applications and have more than 13 years of Software experience. I got to work in various technologies in all these years of my professional career. I started my career with Mainframes as a programmer, then worked as a QA Analyst, then worked on Content Management Systems along with QA and finally got into Cloud Computing. I am into both Microsoft Azure and AWS. I did Certifications like Azure Fundamentals, Azure Administrator and AWS Cloud Practitioner.

**What are your weeknesses?**

**What are your strengths?**

Discipline

Continuous learner

True

Quick

Determination

Self-motivated

Time bound

Curious

**EC2** – Elastic Compute Cloud - Amazon Elastic Compute Cloud is a part of Amazon Web Services, that allows users to rent virtual computers on which to run their own computer applications.

**Types:**

General purpose

Memory Optimized

Storage Optimized

Compute Optimized

Accelerated Computing

S3 - Amazon S3 or Amazon Simple Storage Service is a service offered by Amazon Web Services that provides object storage through a web service interface. Amazon S3 uses the same scalable storage infrastructure that Amazon.com uses to run its global e-commerce network

VPC - A virtual private cloud is an on-demand configurable pool of shared resources allocated within a public cloud environment, providing a certain level of isolation between the different organizations using the resources

**AWS CloudFormation** is a service that helps you model and set up your Amazon Web Services resources so that you can spend less time managing those resources and more time focusing on your applications that run in AWS. You create a template that describes all the AWS resources that you want (like Amazon EC2 instances or Amazon RDS DB instances), and CloudFormation takes care of provisioning and configuring those resources for you. You don't need to individually create and configure AWS resources and figure out what's dependent on what; CloudFormation handles that.

**CloudWatch:** Amazon CloudWatch is a monitoring and observability service built for DevOps engineers, developers, site reliability engineers (SREs), and IT managers. CloudWatch provides you with data and actionable insights to monitor your applications, respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health. CloudWatch collects monitoring and operational data in the form of **logs, metrics, and events,** providing you with a unified view of AWS resources, applications, and services that run on AWS and on-premises servers. You can use CloudWatch to detect anomalous behavior in your environments, set alarms, visualize logs and metrics side by side, take automated actions, troubleshoot issues, and discover insights to keep your applications  
running smoothly.

**CloudTrail:** AWS CloudTrail is a service that enables governance, compliance, **operational auditing, and risk auditing** of your AWS account. With CloudTrail, you can log, continuously monitor, and **retain account activity** related to actions across your AWS infrastructure. CloudTrail provides event history of your AWS account activity, including actions taken through the AWS Management Console, AWS SDKs, command line tools, and other AWS services. This **event history** simplifies security analysis, resource change tracking, and troubleshooting. In addition, you can use CloudTrail to detect unusual activity in your AWS accounts. These capabilities help simplify operational analysis and troubleshooting.

**DynamoDB:** Amazon DynamoDB is a fully managed proprietary NoSQL database service that supports key–value and document data structures that delivers single-digit millisecond performance at any scale. It's a fully managed, multi-region, multi-active, durable database with built-in security, backup and restore, and in-memory caching for internet-scale applications. DynamoDB can handle more than 10 trillion requests per day and can support peaks of more than 20 million requests per second.

**IAM:** **AWS** Identity and Access Management (**IAM**) enables you to manage access to **AWS** services and resources securely. Using **IAM**, you can create and manage **AWS** users and groups, and use permissions to allow and deny their access to **AWS** resources.

**RDS:** Amazon Relational Database Service is a distributed relational database service by Amazon Web Services. It is a web service running "in the cloud" designed to simplify the setup, operation, and scaling of a relational database for use in applications.

SQS – Simple Queue Service -Amazon Simple Queue Service is a distributed message queuing service introduced by Amazon.com in late 2004. It supports programmatic sending of messages via web service applications as a way to communicate over the Internet.

SNS – Simple Notification Service - Amazon Simple Notification Service is a notification service provided as part of Amazon Web Services since 2010. It provides a low-cost infrastructure for the mass delivery of messages, predominantly to mobile users.

**AZURE**

**Governance - https://docs.microsoft.com/en-us/azure/governance/azure-management**

**Azure Virtual Network:** Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks. VNet is similar to a traditional network that you'd operate in your own data center, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation.

**Storage Account:** An Azure storage account contains all of your Azure Storage data objects: blobs, files, queues, tables, and disks. The storage account provides a unique namespace for your Azure Storage data that is accessible from anywhere in the world over HTTP or HTTPS. Data in your Azure storage account is durable and highly available, secure, and massively scalable.

**Resource Group:** A **resource group** contains the **resources** required to successfully deploy a vSRX VM in **Azure**. ... In **Azure**, you logically **group** related **resources** such as storage accounts, virtual networks, and virtual machines (VMs) to deploy, manage, and maintain them as a single entity.

**Azure Active Directory:** Azure Active Directory (Azure AD) is Microsoft’s cloud-based identity and access management service, which helps your employees sign in and access resources in:

* External resources, such as Microsoft 365, the Azure portal, and thousands of other SaaS applications.
* Internal resources, such as apps on your corporate network and intranet, along with any cloud apps developed by your own organization.

**Azure Active Directory** (**Azure AD**) is Microsoft's enterprise cloud-based identity and access management (IAM) solution. **Azure AD** is the backbone of the Office 365 system, and it can sync with on-premise **Active Directory** and provide authentication to other cloud-based systems via OAuth.

The **OAuth** 2.0 is the industry protocol for authorization. It allows a user to grant limited access to its protected resources. ... **Azure Active Directory** (**Azure AD**) supports all **OAuth** 2.0 flows

**Azure Traffic Manager:** Azure Traffic Manager is a DNS(Domain Name System)-based traffic load balancer. This service allows you to distribute traffic to your public facing applications across the global Azure regions. Traffic Manager also provides your public endpoints with high availability and quick responsiveness.

**Traffic Manager** uses DNS to direct the client requests to the appropriate service endpoint based on a traffic-routing method. Traffic manager also provides health monitoring for every endpoint. The endpoint can be any Internet-facing service hosted inside or outside of Azure. Traffic Manager provides a range of [traffic-routing methods](https://docs.microsoft.com/en-us/azure/traffic-manager/traffic-manager-routing-methods) and [endpoint monitoring options](https://docs.microsoft.com/en-us/azure/traffic-manager/traffic-manager-monitoring) to suit different application needs and automatic failover models. Traffic Manager is resilient to failure, including the failure of an entire Azure region.

**Azure** **Application Gateway** is a web traffic load balancer that enables you to manage traffic to your web **applications**. ... **Application Gateway** can make routing decisions based on additional attributes of an HTTP request, for example URI path or host headers. For example, you can route traffic based on the incoming URL.

**Azure Load Balancer** is a high-performance, ultra low-latency Layer 4 **load**-balancing service (inbound and outbound) for all UDP and TCP protocols. It is built to handle millions of requests per second while ensuring your solution is highly available.Ma

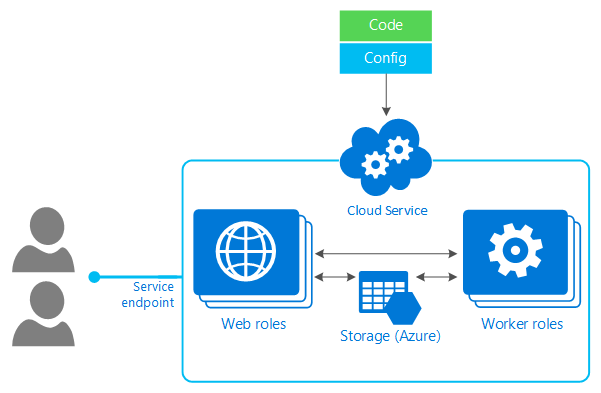
**Load balancing** refers to evenly distributing load (incoming network traffic) across a group of backend resources or servers.

Azure Load Balancer operates at layer 4 of the Open Systems Interconnection (OSI) model. It's the single point of contact for clients. Load balancer distributes inbound flows that arrive at the load balancer's front end to backend pool instances. These flows are according to configured load-balancing rules and health probes. The backend pool instances can be Azure Virtual Machines or instances in a virtual machine scale set.

A [**public load balancer**](https://docs.microsoft.com/en-us/azure/load-balancer/components#frontend-ip-configurations) can provide outbound connections for virtual machines (VMs) inside your virtual network. These connections are accomplished by translating their private IP addresses to public IP addresses. Public Load Balancers are used to load balance internet traffic to your VMs.

An [**internal (or private) load balancer**](https://docs.microsoft.com/en-us/azure/load-balancer/components#frontend-ip-configurations) is used where private IPs are needed at the frontend only. Internal load balancers are used to load balance traffic inside a virtual network. A load balancer frontend can be accessed from an on-premises network in a hybrid scenario.

**Azure Cloud Services** is an example of a [platform as a service](https://azure.microsoft.com/overview/what-is-paas/) (PaaS). Like [Azure App Service](https://docs.microsoft.com/en-us/azure/app-service/overview), this technology is designed to support applications that are scalable, reliable, and inexpensive to operate. In the same way that App Service is hosted on virtual machines (VMs), so too is Azure Cloud Services. However, you have more control over the VMs. You can install your own software on VMs that use Azure Cloud Services, and you can access them remotely.



[**Azure Cloud Services (extended support)**](https://docs.microsoft.com/en-us/azure/cloud-services-extended-support/overview) is a new Azure Resource Manager based deployment model for the Azure Cloud Services product. With this change, Azure Cloud Services running on the Azure Service Manager based deployment model have been renamed as Cloud Services (classic) and all new deployments should use [**Cloud Services (extended support)**](https://docs.microsoft.com/en-us/azure/cloud-services-extended-support/overview).

**Azure Migrate** is a Microsoft service that helps an enterprise assess how its on-premises workloads will perform, and how much they will cost to host, in the **Azure** public cloud. While **Azure Migrate** helps with planning a cloud **migration**, it does not actually transfer on-premises virtual machines (VMs) to the **Azure** cloud.

**Azure Site Recovery** offers ease of deployment, cost effectiveness and dependability. Deploy replication, failover and **recovery** processes through **Site Recovery** to help keep your applications running during planned and unplanned outages.

**Site Recovery** replicates workloads running on physical and virtual machines (VMs) from a primary **site** to a secondary location. When an outage occurs at your primary **site**, you fail over to secondary location, and access apps from there. After the primary location is running again, you can fail back to it.Mar 17, 2020

The **Azure Backup** service provides simple, secure, and cost-effective solutions to back up your data and recover it from the Microsoft Azure cloud.

* **Multiple storage options** - Azure Backup offers three types of replication to keep your storage/data highly available.
  + [Locally redundant storage (LRS)](https://docs.microsoft.com/en-us/azure/storage/common/storage-redundancy#locally-redundant-storage) replicates your data three times (it creates three copies of your data) in a storage scale unit in a datacenter. All copies of the data exist within the same region. LRS is a low-cost option for protecting your data from local hardware failures.
  + [Geo-redundant storage (GRS)](https://docs.microsoft.com/en-us/azure/storage/common/storage-redundancy#geo-redundant-storage) is the default and recommended replication option. GRS replicates your data to a secondary region (hundreds of miles away from the primary location of the source data). GRS costs more than LRS, but GRS provides a higher level of durability for your data, even if there's a regional outage.
  + [Zone-redundant storage (ZRS)](https://docs.microsoft.com/en-us/azure/storage/common/storage-redundancy#zone-redundant-storage) replicates your data in [availability zones](https://docs.microsoft.com/en-us/azure/availability-zones/az-overview#availability-zones), guaranteeing data residency and resiliency in the same region. ZRS has no downtime. So your critical workloads that require [data residency](https://azure.microsoft.com/resources/achieving-compliant-data-residency-and-security-with-azure/), and must have no downtime, can be backed up in ZRS.

An **Availability Zone** is a high-**availability** offering that protects your applications and data from datacenter failures. **Availability Zones** are unique physical locations within an **Azure** region. Each **zone** is made up of one or more datacenters equipped with independent power, cooling, and networking.

An **availability set** is a logical grouping of VMs that allows Azure to understand how your application is built to provide for redundancy and availability. We recommended that two or more VMs are created within an availability set to provide for a highly available application and to meet the [99.95% Azure SLA](https://azure.microsoft.com/support/legal/sla/virtual-machines/). There is no cost for the Availability Set itself, you only pay for each VM instance that you create.

**Fault domains** define the group of virtual machines and hardware components that share a common power source and network switch. By default, the virtual machines configured within your availability set are separated across up to three fault domains.

An **update domain** is a logical group of VMs and the underlying hardware that can undergo maintenance or be rebooted at the same time. As you create VMs within an availability set, the Azure platform automatically distributes your VMs across these **update domains**.

**Azure virtual machine scale sets** let you create and manage a group of load balanced VMs. The number of VM instances can automatically increase or decrease in response to demand or a defined schedule. Scale sets provide high availability to your applications, and allow you to centrally manage, configure, and update a large number of VMs. With virtual machine scale sets, you can build large-scale services for areas such as compute, big data, and container workloads.

**Azure role-based access control** (**Azure RBAC**) is a system that provides fine-grained access management of **Azure** resources. Using **Azure RBAC**, you can segregate duties within your team and grant only the amount of access to users that they need to perform their jobs.

**Azure VPN Gateway** connects your on-premises networks to **Azure** through Site-to-Site **VPNs** in a similar way that you set up and connect to a remote branch office. The connectivity is secure and uses the industry-standard protocols Internet Protocol Security (IPsec) and Internet Key Exchange (IKE).

**ExpressRoute** lets you extend your on-premises networks into the Microsoft cloud over a private connection with the help of a connectivity provider. With ExpressRoute, you can establish connections to Microsoft cloud services, such as Microsoft Azure and Microsoft 365.

Connectivity can be from an any-to-any (IP VPN) network, a point-to-point Ethernet network, or a virtual cross-connection through a connectivity provider at a colocation facility. ExpressRoute connections don't go over the public Internet. This allows ExpressRoute connections to offer more reliability, faster speeds, consistent latencies, and higher security than typical connections over the Internet.

You can use an **Azure network security group** to filter network traffic to and from Azure resources in an Azure virtual network. A network security group contains [security rules](https://docs.microsoft.com/en-us/azure/virtual-network/network-security-groups-overview#security-rules) that allow or deny inbound network traffic to, or outbound network traffic from, several types of Azure resources. For each rule, you can specify source and destination, port, and protocol.

**Virtual network peering** enables you to seamlessly connect two or more [Virtual Networks](https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-overview) in Azure. The virtual networks appear as one for connectivity purposes. The traffic between virtual machines in peered virtual networks uses the Microsoft backbone infrastructure. Like traffic between virtual machines in the same network, traffic is routed through Microsoft's private network only.

Azure supports the following types of peering:

* **Virtual network peering**: Connect virtual networks within the same Azure region.
* **Global virtual network peering**: Connecting virtual networks across Azure regions.

The benefits of using virtual network peering, whether local or global, include:

* A low-latency, high-bandwidth connection between resources in different virtual networks.
* The ability for resources in one virtual network to communicate with resources in a different virtual network.
* The ability to transfer data between virtual networks across Azure subscriptions, Azure Active Directory tenants, deployment models, and Azure regions.
* The ability to peer virtual networks created through the Azure Resource Manager.
* The ability to peer a virtual network created through Resource Manager to one created through the classic deployment model. To learn more about Azure deployment models, see [Understand Azure deployment models](https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/deployment-models?toc=/azure/virtual-network/toc.json).
* No downtime to resources in either virtual network when creating the peering, or after the peering is created.

A **Point-to-Site (P2S)** VPN gateway connection lets you create a secure connection to your virtual network from an individual client computer. A P2S connection is established by starting it from the client computer. This solution is useful for telecommuters who want to connect to Azure VNets from a remote location, such as from home or a conference. P2S VPN is also a useful solution to use instead of S2S VPN when you have only a few clients that need to connect to a VNet. This article applies to the Resource Manager deployment model.

A **Site-to-Site VPN** gateway connection is used to connect your on-premises network to an Azure virtual network over an IPsec/IKE (IKEv1 or IKEv2) VPN tunnel. This type of connection requires a VPN device located on-premises that has an externally facing public IP address assigned to it.

\*IPSec – Internet Protocol Security - IPsec refers to the Internet Protocol Security protocol suite for creating a secure Internet connection between two endpoints.

\*IKE – Internet Key Exchange

Configuring a **VNet-to-VNet** connection is a simple way to connect VNets. When you connect a **virtual network** to another **virtual network** with a **VNet-to-VNet** connection type (VNet2VNet), it's similar to creating a Site-to-Site IPsec connection to an on-premises location.

**Azure Resource Manager** is the deployment and management service for Azure. It provides a management layer that enables you to create, update, and delete resources in your Azure account. You use management features, like access control, locks, and tags, to secure and organize your resources after deployment.

**Azure Security:** Azure Security refers to security tools and capabilities available on Microsoft’s Azure cloud platform.

Azure Security Center is a unified security management system offered by Microsoft to Azure customers.

This is how the division of responsibilities changes across different [cloud service models](https://digitalguardian.com/blog/differences-among-infosec-cloud-delivery-models-iaas-saas-and-paas-%E2%80%93-and-how-choose):

* In IaaS (infrastructure as a service), Azure takes over physical security (hosts, networks, and datacenter).
* In PaaS (platform as a service), Azure takes over physical security and the operating system. Azure shares identity and directory infrastructure, network controls, and applications with the customers.
* In [SaaS (software as a service)](https://digitalguardian.com/blog/what-saas-company), Azure takes more responsibilities: physical security, operating system, network controls, and

**AZURE SECURITY BEST PRACTICES**

The Azure Security documentation is also a handy source for security recommendations and best practices. Here are some tips to get you started quickly:

* Upgrade your Azure subscription to Azure Security Center Standard to enjoy more functionality, like finding and fixing security vulnerabilities, detecting threats with [analytics and intelligence](https://digitalguardian.com/blog/what-security-analytics-learn-about-use-cases-and-benefits-security-analytics-tools), and quick response to an attack.
* Store your keys in the Azure Key Vault. This vault is designed to support passwords, database credentials, and other secrets.
* Install a web application firewall.
* Use Azure MFA ([Multi-factor Authentication](https://searchsecurity.techtarget.com/definition/multifactor-authentication-MFA)), especially for admin accounts.
* Encrypt virtual hard disk files.
* Connect Azure VMs ([virtual machines](https://azure.microsoft.com/en-us/overview/what-is-a-virtual-machine/)) to other networked devices by placing them on Azure virtual networks.
* Use Azure’s DDoS services to prevent and mitigate [DDoS (distributed denial of service) attacks](https://www.webopedia.com/TERM/D/DDoS_attack.html).
* Have [security policies](https://docs.microsoft.com/en-us/azure/security-center/faq-general) in place to prevent abuse. To help you get started, Azure can auto-generate a security policy per an Azure subscription.
* Regularly review the Azure Security Center dashboard. The dashboard provides a central view of your Azure resources and recommends actions.
* Implement Azure Security Center’s [Role-Based Access Control (RBAC)](https://www.appliedi.net/blog/what-is-azure-security-center/). There are five built-in roles (Subscription Owner, Resource Group Owner, Subscription Contributor, Resource Group Contributor, and Reader) and two unique security roles (Security Administrator and Security Reader). These roles vary in permissions.

Remember that cloud security is a shared responsibility between you and Azure. Depending on the cloud delivery model, the responsibilities you share with Azure will change. Don’t forget too to implement the security practices recommended by Microsoft.

**Azure Security:**

* + 1. Operations
    2. Applications
    3. Storage
    4. Networking
    5. Compute
    6. Identity

**Operations:**

1. **Azure Security Center -** [Security Center](https://docs.microsoft.com/en-us/azure/security-center/security-center-introduction) helps you prevent, detect, and respond to threats with increased visibility into and control over the security of your Azure resources.
2. **Azure Resource Manager -** [Azure Resource Manager](https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/overview) enables you to work with the resources in your solution as a group. You can deploy, update, or delete all the resources for your solution in a single, coordinated operation. You use an [Azure Resource Manager template](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/overview) for deployment and that template can work for different environments such as testing, staging, and production. Resource Manager provides security, auditing, and tagging features to help you manage your resources after deployment.
3. [**Application Insights**](https://docs.microsoft.com/en-us/azure/azure-monitor/app/app-insights-overview) is an extensible Application Performance Management (APM) service for web developers. With Application Insights, you can monitor your live web applications and automatically detect performance anomalies. Application Insights creates charts and tables that show you, for example, what times of day you get most users, how responsive the app is, and how well it is served by any external services that it depends on.
4. **Azure Monitor** - [Azure Monitor](https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/) offers visualization, query, routing, alerting, auto scale, and automation on data both from the Azure subscription ([Activity Log](https://docs.microsoft.com/en-us/azure/azure-monitor/essentials/platform-logs-overview)) and each individual Azure resource ([Resource Logs](https://docs.microsoft.com/en-us/azure/azure-monitor/essentials/platform-logs-overview)). You can use Azure Monitor to alert you on security-related events that are generated in Azure logs.
5. [Azure Advisor](https://docs.microsoft.com/en-us/azure/advisor/advisor-overview) is a personalized cloud consultant that helps you to optimize your Azure deployments. It analyzes your resource configuration and usage telemetry. It then recommends solutions to help improve the [performance](https://docs.microsoft.com/en-us/azure/advisor/advisor-performance-recommendations), [security](https://docs.microsoft.com/en-us/azure/advisor/advisor-security-recommendations), and [reliability](https://docs.microsoft.com/en-us/azure/advisor/advisor-high-availability-recommendations) of your resources while looking for opportunities to [reduce your overall Azure spend](https://docs.microsoft.com/en-us/azure/advisor/advisor-cost-recommendations).

## **Applications:**

### Penetration Testing

### Web Application firewall

### Authentication and authorization in Azure App Service

## **Storage:**

Azure role-based access control (Azure RBAC) - You can secure your storage account with [Azure role-based access control (Azure RBAC)](https://docs.microsoft.com/en-us/azure/role-based-access-control/overview). Restricting access based on the [need to know](https://en.wikipedia.org/wiki/Need_to_know) and [least privilege](https://en.wikipedia.org/wiki/Principle_of_least_privilege) security principles is imperative for organizations that want to enforce Security policies for data access.

Shared Access Signature - A [shared access signature (SAS)](https://docs.microsoft.com/en-us/azure/storage/common/storage-sas-overview) provides delegated access to resources in your storage account. The SAS means that you can grant a client limited permissions to objects in your storage account for a specified period and with a specified set of permissions. You can grant these limited permissions without having to share your account access keys.

Encryption in Transit - Encryption in transit is a mechanism of protecting data when it is transmitted across networks. With Azure Storage, you can secure data using:

* [Transport-level encryption](https://docs.microsoft.com/en-us/azure/storage/blobs/security-recommendations), such as HTTPS when you transfer data into or out of Azure Storage.
* [Wire encryption](https://docs.microsoft.com/en-us/azure/storage/blobs/security-recommendations), such as [SMB 3.0 encryption](https://docs.microsoft.com/en-us/azure/storage/blobs/security-recommendations) for [Azure File shares](https://docs.microsoft.com/en-us/azure/storage/files/storage-dotnet-how-to-use-files).
* Client-side encryption, to encrypt the data before it is transferred into storage and to decrypt the data after it is transferred out of storage.

Encryption at rest - For many organizations, data encryption at rest is a mandatory step towards data privacy, compliance, and data sovereignty. There are three Azure storage security features that provide encryption of data that is “at rest”:

* [Storage Service Encryption](https://docs.microsoft.com/en-us/azure/storage/common/storage-service-encryption) allows you to request that the storage service automatically encrypt data when writing it to Azure Storage.
* [Client-side Encryption](https://docs.microsoft.com/en-us/azure/storage/common/storage-client-side-encryption) also provides the feature of encryption at rest.
* [Azure Disk Encryption](https://docs.microsoft.com/en-us/azure/security/fundamentals/azure-disk-encryption-vms-vmss) allows you to encrypt the OS disks and data disks used by an IaaS virtual machine.

### Storage Analytics –

**Network:**

Network Layer Controls

Network Security Groups

VPN Gateway

Express Route

Web Application Firewall

Traffic Manager

Azure Load Balancer

**Azure VHD:** Just like any other computer, virtual machines in **Azure** use disks as a place to store an operating system, applications, and data. ... The operating system disk is created from an image, and both the operating system disk and the image are actually virtual hard disks (**VHDs**) stored in an **Azure** storage account.

Before you upload a Windows virtual machine (VM) from on-premises to Azure, you must prepare the virtual hard disk (VHD or VHDX). Azure supports both generation 1 and generation 2 VMs that are in VHD file format and that have a fixed-size disk. The maximum size allowed for the OS VHD on a generation 1 VM is 2 TB.

**VHD Set** files are a new shared Virtual Disk model for guest clusters in Windows Server 2016. VHD Set files support online resizing of shared virtual disks, support Hyper-V Replica, and can be included in application-consistent checkpoints.

**Powershell:**

Install-Module Az

Login-AzAccount - Login to Azure Account

Logout-AzAccount - Logout of the Azure account you are connected with in your session

Get-AzSubscription - List all subscriptions in all tenants the account can access

Get-AzSubscription -TenantId "xxxx-xxxx-xxxxxxxx" - Get subscriptions in a specific tenant

Select-AzSubscription –SubscriptionID “SubscriptonID” -Choose subscription

Get-AzSubscription - to get the subscription

Get-AzResourceGroup - Get all resource groups

Get-AzResourceGroup -Name "SkylinesRG” - Get a specific resource group by name

Get-AzResourceGroup | Where ResourceGroupName -like Skylines\* - Get a resource group where the name begins with “Skylines”

Get-AzResourceGroup |

Sort Location,ResourceGroupName |

Format-Table -GroupBy Location ResourceGroupName,ProvisioningState,Tags

* + Show resource groups by location

Get-AzResource -ResourceGroupName "SkylinesRG" - Find resources of a type in resource **groups** with a specific name

Get-AzResource -ResourceType "microsoft.web/sites" -ResourceGroupName "SkylinesRG" - Find resources of a type matching against the resource name string

New-AzResourceGroup -Name 'SkylinesRG' -Location

'northcentral' - Create a new Resource Group

Remove-AzResourceGroup -Name "SL-RGToDelete" – Delete a Resource Group

(Get-AzResourceGroup -Name "SkylinesRG").Tags - Display Tags associated with a

specific **resource group** name

(Get-AzResourceGroup -Tag @{ Owner="Skylines Academy"}).Name

* To get all Azure **resource groups** with a specific tag:

We distinguish three types of **brackets**:

* Parentheses ()
* Braces {}
* Square Brackets []

**Paranthesis()** : the command is wrapped with parentheses so that the result is already there when the DayofYear property comes into play …

To make a long story short, we can say:

**Objects (Commands, Variables, …) wrapped with parentheses are executed instantly.**

**Braces{}:** Objects (Commands, Variables, …) wrapped with braces are widely used in scriptblocks. They are executed when it is their turn.

**Square brackets []:** Square brackets retrieve elements in arrays or hashtables and serve as optional parameters.

1. Get the storage account and store it as a variable

➢ $storageaccount = Get-AzStorageAccount - ResourceGroupName "slstoragerg" -AccountName "slstorageaccount"

Get- AzVM - List all VMs in current subscription

Get -AzVM -ResourceGroupName $slResourceGroup - List VMs in a resource group

Get-AzVM -ResourceGroupName “slresourcegroup” -Name “my VM” - Get a specific virtual machine

New-AzVM -Name “vmname”  
Typing in this simple command will create a VM and populate names for all the associated

objects based on the VM name specified

$vmconfig = New-AzVMConfig -VMName “systemname” -VMSize "Standard\_D1\_v2" - Create a VM configuration

Move – Move resource from one RG to another or from one RG to another in a different subscription

Find

|  |  |
| --- | --- |
|  | **Command** |
| Start a VM | Start-AzVM -ResourceGroupName “slresourcegroup” -Name “vmname” |
| Stop a VM | Stop-AzVM -ResourceGroupName “slresourcegroup” -Name “vmname” |
| Restart a running VM | Restart-AzVM -ResourceGroupName “slresourcegroup” -Name “vmname” |
| Delete a VM | Remove-AzVM -ResourceGroupName “slresourcegroup” -Name “vmname” |

**Azure CLI Commands:**

### Create VMs

| **Task** | **Azure CLI commands** |
| --- | --- |
| Create a resource group | az group create --name myResourceGroup --location eastus |
| Create a Linux VM | az vm create --resource-group myResourceGroup --name myVM --image ubuntults |
| Create a Windows VM | az vm create --resource-group myResourceGroup --name myVM --image win2016datacenter |

### Manage VM state

| **MANAGE VM STATE** | |
| --- | --- |
| **Task** | **Azure CLI commands** |
| Start a VM | az vm start --resource-group myResourceGroup --name myVM |
| Stop a VM | az vm stop --resource-group myResourceGroup --name myVM |
| Deallocate a VM | az vm deallocate --resource-group myResourceGroup --name myVM |
| Restart a VM | az vm restart --resource-group myResourceGroup --name myVM |
| Redeploy a VM | az vm redeploy --resource-group myResourceGroup --name myVM |
| Delete a VM | az vm delete --resource-group myResourceGroup --name myVM |

### Get VM info

| **GET VM INFO** | |
| --- | --- |
| **Task** | **Azure CLI commands** |
| List VMs | az vm list |
| Get information about a VM | az vm show --resource-group myResourceGroup --name myVM |
| Get usage of VM resources | az vm list-usage --location eastus |
| Get all available VM sizes | az vm list-sizes --location eastus |

## Disks and images

| **DISKS AND IMAGES** | |
| --- | --- |
| **Task** | **Azure CLI commands** |
| Add a data disk to a VM | az vm disk attach --resource-group myResourceGroup --vm-name myVM --disk myDataDisk --size-gb 128 --new |
| Remove a data disk from a VM | az vm disk detach --resource-group myResourceGroup --vm-name myVM --disk myDataDisk |
| Resize a disk | az disk update --resource-group myResourceGroup --name myDataDisk --size-gb 256 |
| Snapshot a disk | az snapshot create --resource-group myResourceGroup --name mySnapshot --source myDataDisk |
| Create image of a VM | az image create --resource-group myResourceGroup --source myVM --name myImage |
| Create VM from image | az vm create --resource-group myResourceGroup --name myNewVM --image myImage |

A **tenant** represents an organization. It's a dedicated instance of **Azure** AD that an organization or app developer receives at the beginning of a relationship with Microsoft. That relationship could start with signing up for **Azure**, Microsoft Intune, or Microsoft 365, for example.

**Azure tenant** is a directory, **subscription** is an object that represents a "folder" that you can put resources in. **Subscriptions** are tied to **tenants**. so 1 **tenant** can have many **subscriptions**, but not vice versa

Blob storage – Binary Large Object

**Jenkins:** Jenkins is a free and open source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery. It is a server-based system that runs in servlet containers such as Apache Tomcat.

**TeamCity:** TeamCity is a build management and continuous integration server from JetBrains. It was first released on October 2, 2006 and is commercial software and licensed under a proprietary license: a freemium license for up to 100 build configurations and three free Build Agent licenses are available

**VMware** Workstation is the first product ever released by the software company. It enables users to create and run VMs directly on a single Windows or Linux desktop or laptop. Those VMs run simultaneously with the physical machine. Each VM runs its own OS such as Windows or Linux. VMware was the first commercially successful company to virtualize the x86 architecture.

**VirtualBox:** Oracle VM VirtualBox is a free and open-source hosted hypervisor for x86 virtualization, developed by Oracle Corporation. **VirtualBox** is a general-**purpose** virtualization tool for x86 and x86-64 hardware, targeted at server, desktop, and embedded **use**, that allows users and administrators to easily run multiple guest operating systems on a single host.

**Hyper**-**V** is Microsoft software that virtualizes a single hardware server into multiple virtual servers/machines. **Hyper**-**V** lets you share the underlying hardware resources (processor, hard drive, memory, etc.) across virtual machines (VMs) by assigning them virtual resources.

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Azure Devops**

**Kubernetes**We are using docker and kubernetes in our environment.Current version we are using is v 1.13

Writing the YAML file in Kubernetes , Installing, Configuring & Troubleshooting Kubernetes cluster – Master / Worker Nodes, Scheduling Pods , Logging and Monitoring Pods. Take down NODE for maintainence without taking down the application. Security , TLS Certificates for Cluster components and RBAC.Apart from that, Application failure , Work node failure and Control node failure troubleshooting in Kubernetes. Managin the ETCD cluster , Install docker on all the master and worker nodes.Installing the Kube-Apiserver, Kube controller manager, Kubelet.Creating PODS.

**How exactly the pipeline works?**

Our application is microservice based application.What is microservice based application?Blue green deployment – we call it prod A , PROD B – We provided 100% uptime to the customer.

**Challenging Interesting** :

our portal became very slow – running on kubernetes and everything is good but performance is down – it should scale up and scale down - one thing monitoring did not catch up – APM tool instana –

**What automation tools have you used to deploy resources in AWS?**

* Terraform and Cloudformation are for automatic infra provisioning on cloud
* Ansible is to install the application on EC2
* Jenkins is to create the pipeline

**What is an IAM Role in AWS?**

An IAM role is an IAM entity that defines a set of permissions for making AWS service requests. IAM roles are not associated with a specific user or group. Instead, trusted entities assume roles, such as IAM users, applications, or AWS services such as EC2.

**What have you done using lambda functions in AWS?**

Lambda is an Amazon compute service which allows you to run code in the AWS Cloud without managing servers.

AWS Lambda is a serverless compute service that runs your code in response to events and automatically manages the underlying compute resources for you.

You can use AWS Lambda to extend other AWS services with custom logic, or create your own back-end services that operate at AWS scale, performance, and security.

I have used aws lambda to create moogsoft tickets whenever there is a alarm generated by cloudwatch events

also used AWS lambda to push the alerts to slack channel

**How do you map out the journey to move from on prem to a secure AWS Cloud environment?**

What are the types of cloud migration strategies?

1. Rehosting ("lift and shift") As the name implies, this involves lifting your stack and shifting it from on-premises hosting to the cloud. ...

2. Replatforming. ...

3. Repurchasing. ...

4. Refactoring. ...

5. Retiring. ...

6. Retaining.

Will go with lift and shift in which these are the major steps

Lift and shift steps

1. First Create EC2 on AWS

2. Then Deploy Application on Ec2 (How you do – I do Rolling update)

3. Then Configure Application on Ec2

4. Then Test Application on EC2

5. and finally Migrate Traffic to AWS EC2 from On prem server (in aws console it will be route 53) for onprem if you are using cloudfare for example we do that there, but in aws it in route53

**Onprem – AWS Migration For DB**

1. First of all take the full db snapshot

2. then export it to S3

3. then create the RDS using the snapshot on S3

How do you deploy the application to AWS instance

<https://www.softwaretestinghelp.com/ansible-roles-jenkins-integration-ec2-modules/>

**Onprem to AWS Roadmap: How you do it?**

1. Define the Migration Plan
2. Overview of the Migration plan – Migration Cost & Application Disposition Options
3. AWS Migration Consideration – Methodology based on the speed and process, Native AWS tools or Partner tools
4. Building the Migration Plan
5. Estimating the Total Cost of Migration (TCM)
6. Lessons learned

Take onprem costs into consideration

1. Server costs
2. Storage costs
3. Network costs
4. IT Labour costs

**Application migration methodology**

Plan , Migrate and Run

* Plan = Assessment , Data requirements , Security and Risk assessment, Detailed Migration Plan and Effort
* Migrate = Migrate , Deploy, Infrastructure Integration , Application Integration (Rehosting – Lift & Shift)
* Run = Monitoring , Continuous Integration and Continuous Deployment.

Services Key to Enterprise Migrations – VPC , PIOPS, AWS Direct Connect , AWS CF

**Continuous Delivery** (CD) is a practice of automating the entire software release process. **Continuous Deployment** is a step up from **Continuous Delivery** in which every change **in the** source code is **deployed** to production automatically, without explicit approval from a developer.

**continuous integration automates build and testing stages**:

Continuous integration ensures new code is readily accessible in a central repository, integrating pieces of a project from multiple developers quickly to ensure code works well together and in multiple environments. It often helps reduce errors and uses automation to cut down on manual tasks. CI results in more accessible code and fewer bugs in the production pipeline as well as automated testing. Because of that, CI is ideal for businesses that struggle with manual tasks and complicated build processes.

**continuous delivery goes a step further and automates build, packaging, deployment, and testing during the full lifecycle.**

Continuous delivery works well for businesses that need to reduce their time to market, getting software to end users faster. CD alleviates the complexity of deploying new or upgraded software, automating tasks so your team doesn’t need to orchestrate releases manually.

**Application Deployment in AWS instance:**

<https://www.softwaretestinghelp.com/ansible-roles-jenkins-integration-ec2-modules/>

I have been a DevOps specialist for most of my life and have been working only in CI and CD.

**So for Example in Continuous Delivery where I am deploying a new build of my J2EE application (WAR file) to tomcat my steps would be as follows:**

* Stop the application
* Uninstall the application
* Deploy the new build of an application
* Start the application

So I would be creating a role with at least 4 tasks and one main file calling it. This way I am making my code more modular and reusable. So let’s call this role as **tomcat** and create it.

**$ cd /etc/ansible/roles**

**$ sudo ansible-galaxy init tomcat --offline**

### **Jenkins Integration with Ansible**

how Jenkins can be integrated with Ansible. The WAR file built using the build process will be used to deploy to Tomcat on the target machine using Ansible. We will be calling the Ansible role created.So once the build is done the deployment of WAR file will be automatically triggered using Ansible.

I am keeping this simple and have not configured Sonar or Artifactory or Junit during the continuous integration activities which can also be done.

**ONprem – AWS**

800 machines need to be moved from Onprem – AWS , which consists from VMware to AWS and other AWS to AWS.In our case , there is corporate DNS , from there we change to Route 53 , strategy we are following rehosting.Lift and Shift or Rehosting means – VM migrator ,Resimi (third party tool) using that one application at a time.Reengineering or rehosting: Traditional or Legacy applications we are doing rehosting

In our environment, for Redhatopenshift we have 6 prod clusters, 4 development clusters , 2 test and Lab clusters.

**Your Job Role:**

Our job is 50-60% on devops and remaining 40% on cloud operations. The technologies I am working are google cloud,Aws, Azure apart from devops tools Jenkins , deployment going through kubernetes and also having experience with sonarcube and artifactory.

**Q: What is DevOps ?**DevOps help organizations respond in a more agile manner to changing business requirements by:

* Automating and monitoring the process of software creation, from integration, testing, releasing to deploying and managing it.
* Streamlining the development and release pipeline
* Increasing the frequency of deployment
* Reducing the development cycles

**CI/CD Pipeline**

Entire Pipeline covers deployment docker and it deploys in kubernetes. I also have experience in APM monitoring tools also.

**Tools:**

We also use Cloud watch for AWS Monitoring and google monitoring we use.Application is mostly java based application.Monolithic application java.They want the microlithic application and

**Business Use Case:**

The use case , we need website should be up and running 99.99 % which is microservice based website. I got choice to automate and Java team did the development. Earlier it used to take 1 month for development , staging , production . Java team gave me different different images for microservices, which are already dockerised. They created the jaffr files. My job is with base image

Completely pipe line based, we are using terafrom , deploying google cloud engine ,

In application , we are using ansible

Jenkin pipeline – we use groovy script

Our deployment pipeline I will make sure it is successful , I closed work with the development team.

In kubernetes , in ansible

**AWS / GCP / Azure:**

Google is billing in seconds , AWS is billing hourly , if customer wants seconds go with google.

With Scale up ,it may be for 10mins , so it will charge only for 10mins in google , but in AWS you need to pay for an hour.

Azure – onprem Active directory – if you have application like outlook, Microsoft applications or tools, you can use it using AD connect and start using it.

You need high reputation , then AWS is best , AWS is the leader and are early in the market. More regions , more availability zones.

But I think google cloud is going fast as their kubernetes is better than other EKS or Azure kubernetes.

I am more interested in Kubernettes.

Some of the pipelines and playbooks already written. Here I will work from the scratch to write the pipeline end to end , built from the scratch. Pioneer about the devops process.

CI/CD pipeline:

We are using Jenkins heavily for CI/CD pipeline in AWS. Jenkins is market leader and for integration. For version control , git , big bucket , SVN , worked with all version controls.

Migrating from SVN to Jenkins –

We are looking for GITLAB , to replace GITHUB

GITLAB is getting popular , earlier using GITHUB , GITLAB is offering more features and pricing wise too. Support is also very good for GITLAB. GITLAB has very good monitoring when compared to GITHUB

From GIT to GITLAB – migrate our code based including all branched structure and everything.

We will be using GITLAB for the CI/CD 100% and using it in POC.

DBOS – deployment and orchestration – Not familiar but can learn, we are testing harness which has deployment feature (competitor for Jenkins)

For Azure we started using Azure devops

**Current projects:**

* Migrating to Ansible playbook
* Product called RUN DECK
* We are looking for GITLAB , to replace GITHUB
* DBOS – deployment and orchestration – Not familiar but can learn
* Day-Day work – Senior Role – Devops resources hired – trying to implement CI/CD – trying to hire the developer in dallas. Taking our tools and effective rolling it out. Leading the team and assign some work out for them as they are new. Prod deployment for customer implementations. We are trying to hire 1 senior , 2 mid level resource and 1 architect. Customer deployments , releases , working with development. Deployment should go through CI/CD pipeline instead of team member pushing it manually. We want to own the toolset and execution with others in the organization. Transforming aplications to the agile model. Taking care of the pipelines and generic failures.
* Rundeck is fairly new tool which they are using it – used to run the job , we are the ones to code it
* Rundeck and ansible doing everything behind

#########################################

**Terraform**

Terraform : infrastructure as code

Cloud automation

Alternative is EC2 Models in ANsible

Write a playbook

Idompotent for cloud, OS related &apis

\*.tf

provider "aws"{

access\_key = "ACCESS\_KEY\_HERE"

secret\_key = "SECRET\_KEY\_HERE"

}

**$terraform init**

download plug-ins

**$terraform plan**

**$terraform apply**

**$terraform show**

**$terraform destroy**

variables file save in below file

variables.tf

Provisioner

modules like packages

Providers

# Configure the AWS Provider

provider "aws" {

access\_key = "${var.aws\_access\_key}"

secret\_key = "${var.aws\_secret\_key}"

region = "us-east-1"

}

# Create a web server

resource "aws\_instance" "web" {

# ...

}

provider "vsphere" {

user = "${var.vsphere\_user}"

password = "${var.vsphere\_password}"

vsphere\_server = "${var.vsphere\_server}"

# if you have a self-signed cert

allow\_unverified\_ssl = true

}

data "vsphere\_datacenter" "dc" {

name = "dc1"

}

terraform module registry

Defining Variables

Let's first extract our access key, secret key, and region into a few variables. Create another file variables.tf with the following contents.

To persist variable values, create a file and assign variables within this file. Create a file named terraform.tfvars with the following contents:

################################################################

**Ansible**

Ansible use ssh

centralized tool

Idempotent behaviour - all configrationmanagemnt tools

Puppet

Solcestack

chef

**Ansible**

Ansible is Agentless rest all need to intall agents

Unix / linux - ssh

Windows - Winrm

Ansible control machine - Ansible is not a server or service

Ansible is push model other config management tools are pull model

Ansible --version

Ansible (control machine) can install only linux machine but can automate Windows tasks.

Ansible written is pythan

1st need to setup inventory

vi int-dev

web01 ansible\_ssh\_host=192.168.1.9 ansible\_ssh\_user=vagrant ansible\_ssh\_pass=vagarant

**2nd need to check connectivety**

**How to see if the server is connected to ansible server?**

ansible -i int-dev -m ping web01 (ping is a module)

error :hostkey error

To avoid this error disable hostkey check ...vi /etc/ansible/ansible.cfg

Hostgroup (to execute tasks by group)

vi int-dev

[websrvs]

web01

[bdsrvs]

db01

[webdbgrp:children]

websrvs

dbsrvs

[webdbgrp:vars]

ansible\_ssh\_user=vagrant

ansible\_ssh\_pass=vagarant

ansible -i int-dev -m ping websrvs

ansible -i int-dev -m ping all

ansible -i int-dev -m ping '\*'

one Comman user across all the machines

one password

sudoprevilages

Note that host variable high preority than group variables

modules of ansible

yum module

ansible -i int-dev -m yum -a "name=nginx state=installed" websrvs

ansible -i int-dev -m yum -a "name=nginx state=installed" websrvs --sudo

depracation warning

ansible -i int-dev -m yum -a "name=nginx state=installed" websrvs --become

Limits are with ssh not with ansible

ansible-doc yum

= argument is mandatory

- (optional)

---------------------------------------------------------------------------------------

Invetory file with out using password using ssh keys

cd ./ssh/

ssh-kegen

generate id\_rsa& id\_rsa.pub

ssh-copy-id dveops@192.168.1.9

whenevr use ssh use keys not the passowrds

manage the service

ansible -i int-dev -m service -a "name=nginx state=started enabled=yes" --become websrvs

ansible -i int-dev -m yum -a "name=httpd state=installed" websrvs --become

Stop IP table service

ansible -i int-dev - service -a "name=iptables state=stopped enabled=no" --become websrvs

index.html

**ansible -i int-dev -m copy -a "src=index.html dest=/var/www/html/index.html mode=0644" -become websrvs**

**YOu want to change default behaviour**

vi /etc/ansible/ansible.cfg

[defaults]

host\_key\_checking = false

inventory = int-dev

after that no need to mention int-dev in ansible command

Ansible does not store any logs by default

Playbook extention is yaml

yml file is key value pairs

vi web.yml

---

- hosts: websrvs

become: yes

tasks:

- name: Ensure Apache Installed

yum: name=httpd state=installed

- name: start httpd

service: name=httpd state=started enabled=yes

- hosts:dbsrvs

become: yes

tasks:

- name: install mysql

yum: name=mysql-server state=installed

- name start mysql

service: name=mysqld state=stated enabled=yes

**to execute playbook**

ansible-playbook web.yml

newstyle playbook

vi newstyle.yml

---

-hosts: websrvs

become: yes

tasks:

-yum:

name:httpd

state:present

playbooks are in YAML

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Variables

db.yml

----

- hosts:dbsrvs

become: yes

vars:

dbpackage: mysql-server

dbname: account

dbadmin: admin

tasks:

- name:installmysql

yum:

name: "{{dbpackage}}"

state : installed

- name:install admin tools

yum:

name:"{{item}}"

state:present

with\_items:

- mySQL-pythan

- wget

- git

- name: create a new database with name accounts

mysql\_db:

name: "{{dbname}}"

state: present

ansible-playbook db.yml

2nd method to define variables

#mkdir group\_vars

vi all

#mkdir host\_vars

Presidence

1. Playbook

2. host\_vars/hostname

3. group\_vars/groupname

4. group\_vars/all

including playbooks

master play book eg.: site.yml

-include:webservers.yml

-include:dbservers.yml

vi extramod.yml

---

- hosts:all

become:yes

gather\_facts:false

tasks:

-name :excutelinux command

shell: /usr/bin/whoami

register: usrnm

- debug:

msg: "{{usrnm}}"

note that shell commands are not idempoted

prompting

decisions

ansible -m setup web01

it will gather information and store that varaible

called fact variables

ansible -m setup web01 | grep ansible\_os\_family

candition

when:ansible\_os\_family=="RedHat"

when:ansible\_os\_family=="Debien""

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Handlers

after pushing the configuration need to restart the service.

- name: restart httpd

service:

name:httpsd

state: restarted

You want restart service only when configuration is changed

- name: deploy website

copy

notify:

handlers:

-name:restart httpd

service:

Templetes

to use variables

- name:deploy website

mv index.html index.j2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Vprofile project

One key for automation

one key for production

one key for Dev

call ansible as control machine

scp -i vprofileappNCali.pemvprofileappNCali.pem[ubuntu@18.144.57.249:/home/ubuntu/](mailto:ubuntu@18.144.57.249:/home/ubuntu/)

**Docker - containers**

**Docker** is a software platform that allows you to build, test, and deploy applications quickly. ... Running **Docker** on **AWS** provides developers and admins a highly reliable, low-cost way to build, ship, and run distributed applications at any scale.

**Deploy Docker Containers**

1. Step 1: Set up your first **run** with Amazon ECS. ...
2. Step 2: Create a task definition. ...
3. Step 3: Configure your service. ...
4. Step 4: Configure your cluster. ...
5. Step 5: **Launch** and view your resources. ...
6. Step 6: Open the Sample Application. ...
7. Step 7: Delete Your Resources.

Open source project.

Shipping with application

like a directory access with IP & port - isolation

light wight

no operating system

continers : OS virtuliztion

virtulization : Hardware virtualization

container : kernal trick

LXC - Linux continers

Docker Engine - Services - for containers

Docker - open source

on GIT HUB

Not like VM , you can not install application

Not Windows for docker engine

docker ce - community edition

docket ee - entrprise edition

docker commands run with root user by defalut

usermod -aG docker(group) ubuntu(user)

Commands

#docker images

Registry (like repository)- Docker hub

#docker pull jenkins

create container

# docker run -p 8080:8080 -p 50000:50000 jenkins

# docker pull ubuntu

# docker run -it ubuntu /bin/bash

#docker ps (shows the running containers)

#docker ps -a (shows all the containers)

Ctrl pq - out form container

#docker run -it -name myfrstcount ubuntu /bin/bash

attach container

# docker exec myfstcont ls /

# exit

container treat as service

# docker stop name

# docker start name

# docker rm container

#docker rmijenikins

Container running from & on image

Container is thin read & write laer

Container - AUFS driver

multiple containers from same image

# docker inspect jenikens

# docker run -p 7090:8080 jenkins

# docker inspect jenkins

#docker pull nginx:mainline

#docker run --name vpr-nginx -p 8090:80-v vpr-ngin-vol:/usr/share/nginx/html:ro -d nginx

docker file to create own image

#madir images

#vim Dockerfile

FROM ubuntu:latest

MINTAINER name

RUN apt update

RUN apt install git -y

RUN apt install weget tree zip unzip -y

#docker build -t visualpath/mygitimg:V1 .

https://drive.google.com/open?id=1fDA146rBiTzw-TkJNFPj2wyZG4cXoDeN

#docker images

#dcoker login

##########################################################

Scripting:

#########################################

AWS – Amazon Web Services

1) Explain what AWS is?

AWS stands for Amazon Web Service; it is a collection of remote computing services also known as a cloud computing platform. This new realm of cloud computing is also known as IaaS or Infrastructure as a Service.

2) Mention what the key components of AWS are?

The key components of AWS are

Route 53:A DNS web service

Simple E-mail Service:It allows sending e-mail using RESTFUL API call or via regular SMTP

Identity and Access Management:It provides enhanced security and identity management for your AWS account

Simple Storage Device or (S3):It is a storage device and the most widely used AWS service

Elastic Compute Cloud (EC2): It provides on-demand computing resources for hosting applications. It is handy in case of unpredictable workloads

Elastic Block Store (EBS):It offers persistent storage volumes that attach to EC2 to allow you to persist data past the lifespan of a single Amazon EC2 instance

CloudWatch: To monitor AWS resources, It allows administrators to view and collect key Also, one can set a notification alarm in case of trouble.

3) Explain what S3 is?

S3 stands for Simple Storage Service. You can use S3 interface to store and retrieve any amount of data, at any time and from anywhere on the web. For S3, the payment model is “pay as you go.”

4) What is AMI?

AMI stands for Amazon Machine Image. It’s a template that provides the information (an operating system, an application server, and applications) required to launch an instance, which is a copy of the AMI running as a virtual server in the cloud. You can launch instances from as many different AMIs as you need.

5) Mention what the relationship between an instance and AMI is?

From a single AMI, you can launch multiple types of instances. An instance type defines the hardware of the host computer used for your instance. Each instance type provides different computer and memory capabilities. Once you launch an instance, it looks like a traditional host, and we can interact with it as we would with any computer.

6) What does an AMI include?

An AMI includes the following things

A template for the root volume for the instance

Launch permissions decide which AWS accounts can avail the AMI to launch instances

A block device mapping that determines the volumes to attach to the instance when it is launched

7) How can you send a request to Amazon S3?

Amazon S3 is a REST service, and you can send a request by using the REST API or the AWS SDK wrapper libraries that wrap the underlying Amazon S3 REST API.

8) Mention what the difference between Amazon S3 and EC2 is?

The difference between EC2 and Amazon S3 is that

EC2 S3

It is a cloud web service used for hosting your application

It is a data storage system where any amount of data can be stored

It is like a huge computer machine which can run either Linux or Windows and can handle application like PHP, Python, Apache or any databases

It has a REST interface and uses secure HMAC-SHA1 authentication keys

9) How many buckets can you create in AWS by default?

By default, you can create up to 100 buckets in each of your AWS accounts.

10) Explain can you vertically scale an Amazon instance? How?

Yes, you can vertically scale on Amazon instance. For that

Spin up a new larger instance than the one you are currently running

Pause that instance and detach the root webs volume from the server and discard

Then stop your live instance and detach its root volume

Note the unique device ID and attach that root volume to your new server

And start it again

11) Explain what T2 instances is?

T2 instances are designed to provide moderate baseline performance and the capability to burst to higher performance as required by the workload.

12) In VPC with private and public subnets, database servers should ideally be launched into which subnet?

With private and public subnets in VPC, database servers should ideally launch into private subnets.

13) Mention what the security best practices for Amazon EC2 are?

For secure Amazon EC2 best practices, follow the following steps

Use AWS identity and access management to control access to your AWS resources

Restrict access by allowing only trusted hosts or networks to access ports on your instance

Review the rules in your security groups regularly

Only open up permissions that you require

Disable password-based login, for example, launched from your AMI

14) Explain how the buffer is used in Amazon web services?

The buffer is used to make the system more robust to manage traffic or load by synchronizing different component. Usually, components receive and process the requests in an unbalanced way. With the help of buffer, the components will be balanced and will work at the same speed to provide faster services.

15) While connecting to your instance what are the possible connection issues one might face?

The possible connection errors one might encounter while connecting instances are

Connection timed out

User key not recognized by the server

Host key not found, permission denied

An unprotected private key file

Server refused our key or No supported authentication method available

Error using MindTerm on Safari Browser

Error using Mac OS X RDP Client

16) What are key-pairs in AWS?

Key-pairs are secure login information for your virtual machines. To connect to the instances, you can use key-pairs which contain a public-key and private-key.

17) What are the different types of instances?

Following are the types of instances:

General purpose

Computer Optimized

Memory Optimized

Storage Optimized

Accelerated Computing

18) Is the property of broadcast or multicast supported by Amazon VPC?

No, currently Amazon VPI not provide support for broadcast or multicast.

19) How many Elastic IPs is allows you to create by AWS?

5 VPC Elastic IP addresses are allowed for each AWS account.

20) Explain default storage class in S3

The default storage class is a Standard frequently accessed.

21) What are the roles?

Roles are used to providing permissions to entities which you can trust within your AWS account. Roles are very similar to users. However, with roles, you do not require to create any username and password to work with the resources.

22) What are the edge locations?

Edge location is the area where the contents will be cached. So, when a user is trying to accessing any content, the content will automatically be searched in the edge location.

23) What is VPC?

VPC stands for Virtual Private Cloud. It allows you to customize your networking configuration. It is a network which is logically isolated from another network in the cloud. It allows you to have your IP address range, internet gateways, subnet and security groups.

24) Explain snowball

Snowball is a data transport option. It used source appliances to a large amount of data into and out of AWS. With the help of snowball, you can transfer a massive amount of data from one place to another. It helps you to reduce networking costs.

25) What is a redshift?

Redshift is a big data warehouse product. It is fast and powerful, fully managed data warehouse service in the cloud.

26) What are the advantages of auto-scaling?

Following are the advantages of autoscaling

Offers fault tolerance

Better availability

Better cost management

27) What is meant by subnet?

A large section of IP Address divided into chunks is known as subnets.

28) Can you establish a Peering connection to a VPC in a different region?

No, It’s only possible between VPCs in the same region.

29) What is SQL?

Simple Queues Services also known as SQL. It is distributed queuing service which acts as a mediator for two controllers.

30) How many subnets can you have per VPC?

You can have 200 subnets per VPC.

31) DNS and Load Balancer service comes under which type of cloud service?

DNS and Load Balancer and DNS services come under IAAS-storage cloud service.

32) What is the role of AWS CloudTrail?

CloudTrail is a specially designed tool for logging and tracking API calls. It helps to audit all S3 bucket accesses.

33) When EC2 officially launched?

EC2 officially launched in the year 2006.

34) What is SimpleDB?

SimpleDB is a data repository of structure record which encourages data doubts and indexing both S3 and EC2are called SimpleDB.

35) Explain Amazon ElasticCache

Amazon Elasticcache is a web service which makes it easy to deploy, scale and store data in the cloud.

36) What is AWS Lambda?

Lambda is an Amazon compute service which allows you to run code in the AWS Cloud without managing servers.

37) Name the types of AMI provided by AWS

The types of AMI provided by AWS are:

Instance store backed

EBS backed

38) Name the AWS service exists only to redundantly cache data and images?

AWS Edge locations are service which redundantly cache data and images.

39) Explain Geo Restriction in CloudFront

A Geo-restriction feature helps you to prevent users of specific geographic locations from accessing content which you’re distributing through a CloudFront web distribution.

40) What is Amazon EMR?

EMR is a survived cluster stage which helps you to interpret the working of data structures before the intimation. Apache Hadoop and Apache Spark on the Amazon Web Services helps you to investigate a large amount of data. You can prepare data for the analytics goals and marketing intellect workloads using Apache Hive and using other relevant open source designs.

41) What is boot time taken for the instance stored backed AMI?

The boot time for an Amazon instance store-backend AMI is less than 5 minutes.

42) Do you need an internet gateway to use peering connections?

Yes, the Internet gateway is needed to use VPC (virtual private cloud peering) connections.

43) How to connect EBS volume to multiple instances?

We can’t be able to connect EBS volume to multiple instances. Although, you can connect various EBS Volumes to a single instance.

44) List different types of cloud services

Various types of cloud services are:

Software as a Service (SaaS),

Data as a Service (DaaS)

Platform as a Service (PaaS)

Infrastructure as a Service (IaaS).

45) State the difference between An Instance and AMI

AMI is a template consisting software configuration part. For example Operating systems, applications, application server if you start an instance, a duplicate of the AMI in a row as an attendant in the cloud.

46) What are the different types of Load Balancer in AWS services?

Two types of Load balancer are:

Application Load Balancer

Classic Load Balancer

47) In which situation you will select provisioned IOPS over standard RDS storage?

You should select provisioned IOPS storage over standard RDS storage if you want to perform batch-related workloads.

48) What are the important features of Amazon cloud search?

Important features of the Amazon cloud are:

Boolean searches

Prefix Searches

Range searches

Entire text search

AutoComplete advice

49) Can vertically scaling is allows in Amazon Instance?

Yes, you can vertically estimate one Amazon instance.

50) What is the use of lifecycle hooks in Autoscaling?

Lifecycle hooks are used for autoscaling to put an additional wait time to a scale in or scale out event.

51) What are various layers of Cloud Architecture explained in AWS training?

Different layers of cloud architecture are:

Cloud controller

Cluster controller

Storage Controller

Node Controller

52) What are the storage class available in Amazon s3?

Storage classes available with Amazon s3 are:

Amazon S3 standard

Amazon S3 standard-infrequent Access

Amazon S3 Reduced Redundancy Storage

Amazon Glacier

53) Name some of the DB engines which can be used in AWS RDS

MS-SQL DB

MariaDB

MYSQL DB

OracleDB

PostgreDB

Redhat OPENSHIFT :

In our environment , for Redhatopenshift we have 6 prod clusters , 4 development clusters , 2 test and Lab clusters.

OpenShift is a Red Hat Service Platform. The service platform is suitable for offering certain services to users, such as auto-scaling. OpenShift Enterprise also supports various programming languages and related frameworks, applications lifecycle , and integrated tools.

So lets understand the need first : The move to modern application development and the need for rapid and continuous deployment make a strong DevOps-enabling platform, a key component in the IT arsenal.

Applications are designed around smaller, independent microservices modules. The decomposition of applications into smaller components has distinct advantages of designing software more efficiently and producing them in a more robust way - both of which are ideally matched with the market needs today. This approach however adds complexity in the IT operations application layer.

For IT operations to support this modern application architecture, the underlying middleware, runtime, and other application development resources, lifecycle management should be highly automated and abstract their complexity.

**Q: Do you know the most important features of Openshift ?**There are a lot of features offered by OpenShift. Here are a few main features :

* Software Defined Network
* Persistent Storage.
* Container Native Storage (CNS / SDS).
* Log Aggregation and Analysis.
* Monitoring | Telemetry.
* Capacity Management
* Egress Routing for Enterprise integration. Router Sharding.
* Full Stack Support.
* Automatic Application Scaling
* Rich Command-line Toolset
* Source Code Version Management
* System Certifications and Patching.
* Multi Environment Support
* Software Defined Network
* Multiple database and Language Support
* Remote Debugging of Applications
* Support for remote SSH login to application.
* Rest API Support
* Self-service on Demand Application Stack
* Remote Debugging of Applications
* Built-in Database Services
* Continuous Integration and Release Management
* Extensible Cartridge System
* IDE Integration

**Q: What Is Openshift Online?**OpenShift online is an OpenShift community service that allows one to create, deploy and scale containerized applications on the public cloud quickly.

It is the development and hosting platform of Red Hat's public cloud platform, which enables automated provisioning, management and application scaling to help developers concentrate on the development of framework logic.

**Q: What is in OpenShift Container Platform?**

Let's understand what OpenShift Container platform is

* RedHat OpenShift Container Platform is a PAAS offering from RedHat, which was formally known as OpenShift Enterprises
* It is a supported distribution of Kubernetes using Docker containers and DevOps tools for accelerated application development.
* Open Shift also allows you to have highly available, self-healing, and auto-scaling applications without any of the manual setup that would typically need to be done in a traditional environment whether they're onpremise or in the public cloud.
* OpenShift includes a full complement of open source programming languages giving polyglot choice to developers.

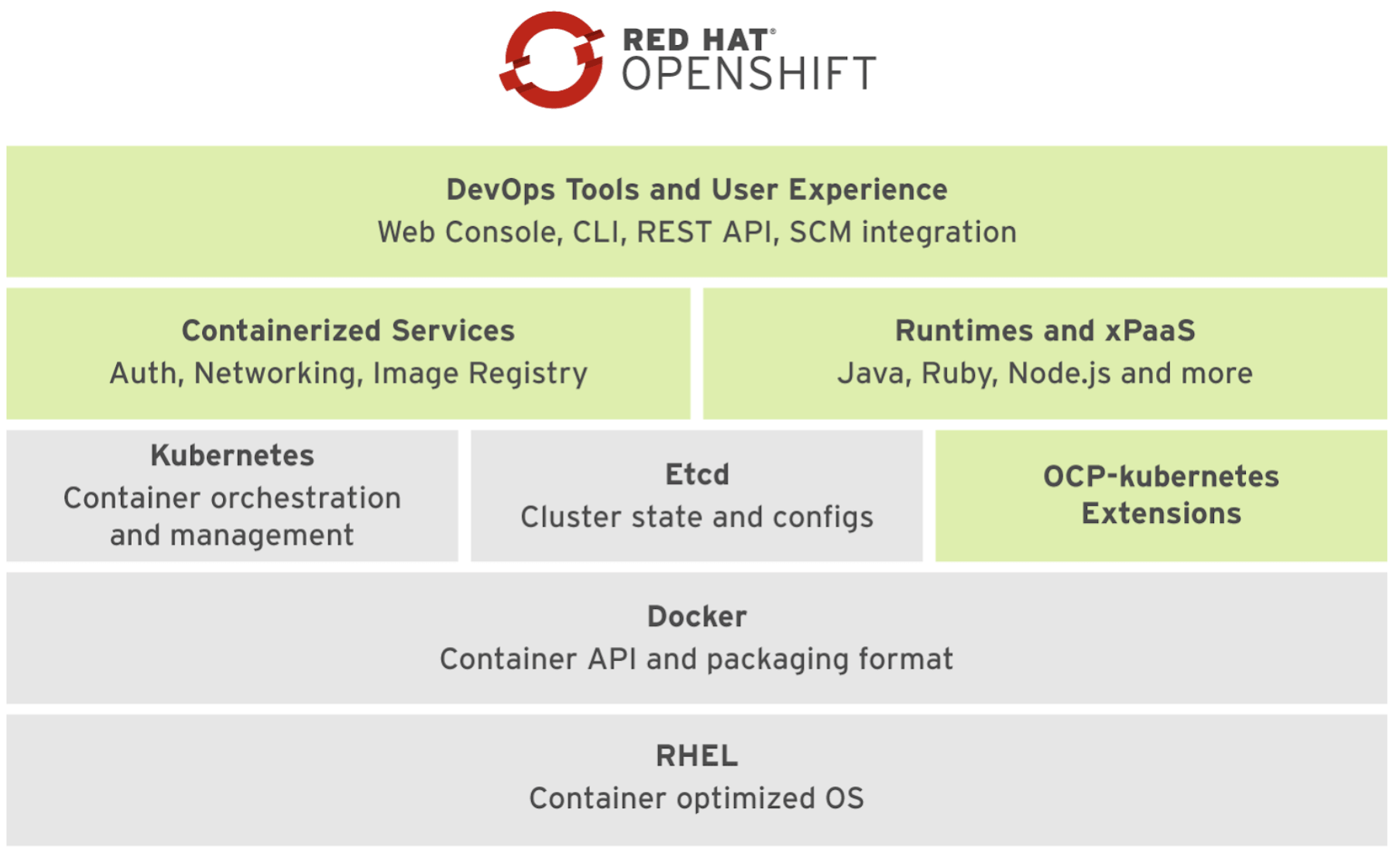
**Q: What are Benefits of OpenShift?**

Let's understand what OpenShift Container platform is

* Provides a container management platform for your complex IT infrastructure.
* Kubernetes is the industry leading open source container orchestration framework and Red Hat OpenShift Container Platform is the leading enterprise distribution of Kubernetes
* Enables your development team to focus on designing and testing applications rather than spending excessive time in managing and deploying containers.
* It includes containerization for multitenancy, automatic provisioning, automatic application scaling, continuous integration, and self-service for developers.

**Q: Explain OpenShift Architecture?**

OpenShift Container Platform is a set of modular components and services built on top of Red Hat Enterprise Linux, Docker, and Kubernetes.

OpenShift addKubernetess capabilities such as remote management, multitenancy, increased security, application life-cycle management and self-service interfaces for developers. 

In the above figure, going from bottom to top, and from left to right, the basic container infrastructure is shown, integrated and enhanced by Red Hat:

* The base OS is Red Hat Enterprise Linux (RHEL).
* Docker provides the basic container management API and the container image file format.
* Kubernetes manages a cluster of hosts (physical or virtual) that run containers. It works with resources that describe multi-container applications composed of multiple resources, and how they interconnect.
* Etcd is a distributed key-value store, used by Kubernetes to store configuration and state information about the containers and other resources inside the OpenShift cluster.

#### Q: What Is Pod?

The Kubernetes scheduling unit is the Pod, which is a grouping of containers sharing a virtual network device, internal IP address, TCP/UDP ports, and persistent storage. A Pod can be anything from a complete enterprise application, including each of its layers as a distinct container, to a single microservice inside a single container. Kubernetes manages replicas to scale pods. A replica is a set of pods sharing the same definition.

#### Q: What Are Deployment Strategies?

A deployment strategy is a way to change an application or to update it. The goal is to make the transition in a way that the consumer barely notices the changes, without downtime. Using a blue-green deployment is the commonest technique.  
  
The new version (the blue version) is being built for testing and evaluation, while the users are still using the stable version (the green). The users are moved to the blue version when it's available. You can go back to the Green version if a problem occurs.

#### Q: What Is Rolling Strategy?

A rolling rollout gradually replaces instances of an application's prior version with instances of the application's current version. A rolling deployment usually waits for new pods to get ready through a readiness test until the old modules are scaled down. The rolling deployment can be halted if a serious problem occurs.

#### Q: What Is Canary Deployments?

All rolling deployments in OpenShift Origin are canary deployments; it checks a new version (the canary) before replacing all the old instances. Unless the readiness test never works, then the canary instance and deployment will be disabled.

#### Q: What Is Routes In Openshift?

In OpenShift Routes is a way to externalize the services by providing externally reachable hostname. In OpenShift routes are created using routers developed by admin.

OpenShift CLI Commands

# subscription-manager register

# subscription-manager refresh

# subscription-manager attach --pool=<pool\_id>

# subscription-manager repos --enable="rhel-7-server-ose-3.9-rpms"

# yum install atomic-openshift-clients

$ oc login

OpenShift server [https://localhost:8443]: https://openshift.example.com

Username: alice

Authentication required for https://openshift.example.com (openshift)

Password: \*\*\*\*\*\*

Login successful.

You don't have any projects. You can try to create a new project, by running

$ oc new-project <projectname>

Welcome to OpenShift! See 'oc help' to get started.

$ oc logout

User, alice, logged out of https://openshift.example.com

$ oc login -u system:admin -n default

CLI Configuration Files

***~/.kube/config***. The current CLI configuration can be viewed using the following command:

*Example 2. Viewing the CLI Configuration*

$ oc config view

apiVersion: v1

clusters:

- cluster:

server: https://openshift.example.com

name: openshift

contexts:

- context:

cluster: openshift

namespace: aliceproject

user: alice

name: alice

current-context: alice

kind: Config

preferences: {}

users:

- name: alice

user:

token: NDM2N2MwODgtNjI1Yy10N3VhLTg1YmItYzI4NDEzZDUyYzVi

Projects

$ oc project

If you have access to multiple projects, use the following syntax to switch to a particular project by specifying the project name:

$ oc project <project\_name>

For example:

$ oc project project02

Now using project 'project02'.

$ oc project project03

Now using project 'project03'.

$ oc project

Using project 'project03'.

The oc status command shows a high level overview of the project currently in use, with its components and their relationships, as shown in the following example:

$ oc status

In project OpenShift 3 Sample (test)

service database-test (172.30.17.113:6434 -> 3306)

database-test deploys docker.io/library/mysql:latest

#1 deployed 47 hours ago

service frontend-test (172.30.17.236:5432 -> 8080)

frontend-test deploys origin-ruby-sample:test<-

builds https://github.com/openshift/ruby-hello-world with docker.io/openshift/ruby-20-centos7:latest

not built yet

#1 deployment waiting on image

To see more information about a service or deployment config, use 'oc describe service <name>' or 'oc describe dc <name>'.

You can use 'oc get pods,svc,dc,bc,builds' to see lists of each of the types described above.

**What are the types of cloud migration strategies?**

1. Rehosting ("lift and shift") As the name implies, this involves lifting your stack and shifting it from on-premises hosting to the cloud. ...
2. Replatforming. ...
3. Repurchasing. ...
4. Refactoring. ...
5. Retiring. ...
6. Retaining.

**Lift and shift steps**

1. Create EC2 on AWS
2. Deploy Application on Ec2 (How you do – I do Rolling update)
3. Configure Application on Ec2
4. Test Application on EC2
5. Migrate Traffic to AWS EC2 from On prem server (in aws console it will be route 53) for onprem if you are using cloudfare for example we do that there, but in aws it in route53

**Onprem – AWS Migration For DB**

1. First of all take the full db snapshot
2. then export it to S3
3. then create the RDS using the snapshot on S3

**Cloud Security:**

1. User Management using IAM service and enabling 2 factor authentication
2. Encryption via AWS Key Management Service (we encrypt EBS volumes)
3. Security Groups (Inbound rules & Outbound rules)
4. Encryption at rest and encryption in transit
5. IAM instance roles instead of access and secret key
6. TLS certs
7. WAF
8. AWS Shield
9. Auditing –AWSconfig

KMS for the REST and SSL for TRANSIT

For secure Amazon EC2 best practices, follow the following steps

* Use AWS identity and access management to control access to your AWS resources
* Restrict access by allowing only trusted hosts or networks to access ports on your instance
* Review the rules in your security groups regularly
* Only open up permissions that you require
* Disable password-based login, for example, launched from your AMI

**TLS certificate steps**

The 'client hello' message: The client initiates the handshake by sending a "hello" message to the server. The message will include which TLS version the client supports, the cipher suites supported, and a string of random bytes known as the "client random."

The 'server hello' message: In reply to the client hello message, the server sends a message containing the server's SSL certificate, the server's chosen cipher suite, and the "server random," another random string of bytes that's generated by the server.

Authentication: The client verifies the server's SSL certificate with the certificate authority that issued it. This confirms that the server is who it says it is, and that the client is interacting with the actual owner of the domain.

The premaster secret: The client sends one more random string of bytes, the "premaster secret." The premaster secret is encrypted with the public key and can only be decrypted with the private key by the server. (The client gets the public key from the server's SSL certificate.)

Private key used: The server decrypts the premaster secret.

Session keys created: Both client and server generate session keys from the client random, the server random, and the premaster secret. They should arrive at the same results.

Client is ready: The client sends a "finished" message that is encrypted with a session key.

Server is ready: The server sends a "finished" message encrypted with a session key.

Secure symmetric encryption achieved: The handshake is completed, and communication continues using the session keys.

## Tools for Managing Cloud-Native Applications

* [**Fluentd**](https://www.fluentd.org/)**–** Used for logging. It collects and shares log data and sends to log aggregations tools such as AWS CloudWatch.
* [**Prometheus**](https://prometheus.io/)**–** Monitoring tool which records Time-Series data for distributed Microservices.
* **Kubernetes –** Container orchestration system to deploy and manage Containers.
* **ELK Stack –** Provides complete monitoring solution.
* [**Grafana**](https://grafana.com/)**–** A Visualization tool.
* **AWS, Google Cloud –** Provides Cloud computing services.
* [**Istio**](https://istio.io/)**–** Enables Service Mesh.

Match.com

associated companies

they are looking for enterprise kind of environment

zach is not willing to cloud for the security reasons

Zach and Joey has put a ROADMAP to migrate to cloud

we need to educate them as they are not familiar with this.

Technical Screening questions

Google Cloud Platform (GCP)

**Difference between IaaS & PaaS?**

GCP provides IaaS as well, although it began only with PaaS. Google Compute Engine (GCE) is an Infrastructure as a Service (IaaS) offering that allows clients to run workloads on Google's physical hardware. Google Compute Engine lets you create and run virtual machines on Google infrastructure. The most distinct difference between IaaS and PaaS is that IaaS offers administrators more direct control over operating systems, but PaaS offers users greater flexibility and ease of operation. IaaS builds the infrastructure of a cloud-based technology.

**What are the main advantages of using Google Cloud Platform?**

1.Better Pricing Than Competitors.

2. Private Global Fiber Network.

3.Live Migration of Virtual Machines.

4.Improved Performance.

5.State of the Art Security.

6.Dedication to Continued Expansion.

7.Redundant Backups.

**Difference between Google Compute Engine and App Engine**

App Engine is a Platform-as-a-Service. It means that you simply deploy your code, and the platform does everything else for you. For example, if your app becomes very successful, App Engine will automatically create more instances to handle the increased volume.    
Compute Engine is an Infrastructure-as-a-Service. You have to create and configure your own virtual machine instances. It gives you more flexibility and generally costs much less than App Engine. The drawback is that you have to manage your app and virtual machines yourself. You can mix both App Engine and Compute Engine, if necessary. They both work well with the other parts of the Google Cloud Platform.   
 **How do you monitor GCP Infrastructure?**

Stackdriver Monitoring measures the health of cloud resources and applications by providing visibility into metrics such as CPU usage, disk I/O, memory, network traffic and uptime. It is based on collectd, an open source daemon that collects system and application performance metrics.

**What is Interconnect in GCP?**

Cloud Interconnect provides low latency, highly available connections that enable you to reliably transfer data between your on-premises and Google Cloud Virtual Private Cloud (VPC) networks. ... Dedicated Interconnect provides a direct physical connection between your on-premises network and Google's network.    
  
**What is IAM Framework?**

 Identity and Access Management (IAM) API: Manages identity and access control for Google Cloud Platform resources, including the creation of service accounts, which you can use to authenticate to Google and make API calls

**What is Network Tags in GCP?**

Network tags allow you to apply firewall rules and routes to a specific instance or set of instances: You make a firewall rule applicable to specific instances by using target tags and source tags. You make a route applicable to specific instances by using a tag.