Basic Questions

# Tell me about yourself ?

Hello, Im ??? I have ??? years of Experience on AWS & DEVOPS

EC2, S3, IAM, VPC, LB, ASG, RDS, CloudFront, Lambda, Code Deploy, Code Pipeline, DynamoDB

Git, Jenkins, Ansible, Docker, Terraform, Splunk, AppD, Python

# Roles and Responsibilitiesn

Jira – Build Dashboard & Issue Tracking Dashboard

Service now – Process Tracking

I interact with developer and desgin the infrastructure from them using terraform

Create the application flow.

Once Infra development is done we just we try to automate it using CI/ CD pipelines from Jenkins

deployment using Ansible.

# What were you working on previously ?

I was working on server level on web server and on App server

JBoss, Apache, IHS, Tomcat tools

Deploying the Source Code and maintaing the application

Loading the SSL certifacate and setting up Life cycles

# What is your Daily Activities ?

1. Mail Check
2. Ticketing tool -> Jira
3. deployment the given code from git hub
4. Env Monitoring
5. Webx Meeting
6. Dev Support if any servers are failed

# Jira & Service Now

**SLA - Service Level Aggrement**

## Status

* Open
* TO DO
* IN progress
* Reopened
* Resolved
* Closed
* Done

## Priority Order

p5 - Lowest priority

p4

p3

p2

p1 - Highest Priority

Incident Manager -> p1,p2

We can raise only -> p5,p4,p3

## KANBAN



# Daily how many tickets you will solve ?

Jira tickets weekly 1 or 2  
Service now incidents daily basis

# Who will rise the tickets ?

Incident Manager

# Tell me recently solved some tickets ?

Talk about Error

# What is the recent issue you have faced & How did you solve it ?

I was deploying to tomcat with Jenkins using the deploy to container plugin. i have added the user in tomcat conf/users.xml file. But during deployment i got errors saying deployment was unsuccessful. I was testing this out as new challenge i have researched into this for 2-3 hours but i got only immediate short term solution suggesting to edit a file.

<user username="admin" password="admin" roles="manager-gui,manager-script,manager-jmx,manager-status"/>

sol: . By default Tomcat does not allow access to the manager from external machines, i.e. other than localhost. I had two options remove the restriction where any machine can deploy to tomcat. Or give full access to tomcat/webapps folder to the user which i have added

# What is BCP - business continuity plan ?

Step 1: Risk Assessment.

Step 2: Business Impact Analysis (BIA)

Step 3: Business Continuity Plan Development.

Step 4: Strategy and Plan Development.

Step 5: Plan Testing & Maintenance.

HR Interview Questions

# Form 16?

No, Less than 5 Lakhs so no form 16. If required ill do self auditing and ill submit

# PF account no?

No, My organization has less than 300 employees

# Salary Divide ?

PT -> Professional Tax

PF -> Providend Fund

TDS ­> Tax Deduction

HRA -> House Rent Allowance

# What is the recent mistake you have made, how did you recover ?

# General tips:

* Start by researching the company and your interview/role
* Say Excuse me? before getting into the room/place
* Greet the HR’s with Good morning sir/madam, Nice to meet you or any other greeting. Never say Good night even if the interview is on 10PM, greet with good evening.
* Practice your answers to common interview questions.
* Never speak too slow or too fast, Be precise.
* Practice good manners and body language.
* Respond truthfully to the questions asked. Never lie.
* Plan your interview attire the night before. Follow the Proper Dress Code. *Dress to Impress !*
* Bring copies of your resume, a notebook and pen.
* Silence Your Phone
* If it’s a Yes or No question, don’t simply answer with Yes or No in single word. say Yes or No and why ? talk atleast 2 sentence.
* Try to Stay Calm, There are thousands of positions available

# Tell me about yourself ?

TLDR: Start with Name. Educational achievements, Applicable skills, Professional goals, Reason for interest in the company.

## Do’s

* Keep it within the sweet spot 2-3 minutes. Not more Not less.
* Tell about your Education in Chronological order (Past to Present), Tell your marks in percentage.
* End with a positive comment, “why you are exited to get this new position in the esteemed company”

## Don’ts

* Don’t talk about your Family, Life History, Hobbies, Fav Movies, Pets or Interest
* Don’t mention life goals (buying a car, house)
* Don’t recite your resume

Link: <https://www.indiabix.com/hr-interview/tell-me-about-yourself/>

# Why should we hire you ? (for Fresher)

### Alternative Ques:

What makes you the best candidate ?

What makes you think you have what it takes ?

## Do’s

Tell you are strong in theoritical concepts and ready to learn practical applications

Ready to give your fresh energy, enthusiam, dedication, skills, concentration for the company and self improvement

say you are a team player or perfectionist or punctual or passionate or searching for a platform to help your growth

## Don’ts

* Don’t be arrogant, Like if you don’t give me a job you will regret it.
* Don’t tell/express that you are desperate for the job

Link: <https://www.indiabix.com/hr-interview/why-should-i-hire-you/>

# What is your strength & weakness ?

Choose any 3-5 strength from the below and talk about it

## strengths :

* Communicating: Communication Skills
* Flexibility and Adaptability
* Learning agility: Quick learner
* Tolerance: Stress tolerance
* Critical thinking: Decision making skills
* Creating Ideas: Creativity
* Positive Attitude: Creating a positive work environment
* Intelligent
* Leadership
* Responsible
* Listening
* Goal oriented: results achiever, Meeting deadlines
* Self Motivated
* Delegating tasks
* Personal Management
* Organizing: organization skills
* Prioritizing tasks
* Judgment
* Computer Skills
* Honesty and Integrity
* Punctuality
* Analytical and Problem Solving Skills
* Strong Analytical abilities
* Attention to details
* Accuracy
* Interpersonal Skills
* Team Work Skills
* Initiative
* Loyalty

## Weakness :

### Alternative Ques:

What makes you angry?

This is actually a trick question to **ELIMINATE YOU, Never ever talk about your weakness**. When the HR is asking you about your weakness you should answer them about your strength which can relate to weakness. Don’t give them a reason to eliminate you.

Example: I would say “*I have High Standards, I believe that everything matters and that nothing is small stuff and sometimes not everybody has High Standards this leads to frustration but with little communication I can smooth that over*“

# Where do you see yourself in 5 years ?

### Alternative Ques:

What are your professional goals ?

## Do’s

* Talk about career growth, Understanding business and work process.
* Talk about completing Professional goals
* Talk about completing projects, learning new skills, new responsibility and new knowledge.
* Say you want to become Experienced developer who can undertake large projects

## Don’ts

* Don’t talk about Life goals
* Don’t talk about salary improvement
* Don’t ever say to HR as “I would like to see myself in your position in the next 5 years.”

# Why do you want to work in our company ?

### Alternative Ques:

Explain how you would be an asset to this organization?

## Do’s

* Start your sentence with “It would be my pleasure/honour to work in ABC company”, “It’s a great privilege to work in ABC company”, “I would be delighted/grateful to work in ABC company”.
* Do Research about the company and the job title.
* Explain you need a good platform for your personal growth as well as company’s benefit/growth.
* If you have job specific skills, explain why it would be beneficial for the company.

## Don’t’s

* Don’t be over confident in your skills be polite
* Ans within 2 or 3 sentences.
* Don’t say you are in need of money
* Don’t promise, if you give me the job ill be in a good position after few years.

# Do you have any questions for me? (at the end of interview asked by HR)

## Tips:

But first of all, thank you so much for giving me the opportunity to be iñterviewed by you it was a wonderful experience. I just wanted to ask what kind of exposure I could expect to from your organization and also I would be satisfied if you can give me suggestions on the areas that I need to focus on to survive in this prestigious organization if I get selected.

## Some other questions you can expect

Can you work under pressure?

Are you willing to relocate or travel?

How do you feel about working nights and weekends?

Who has inspired you in your life and why?

AWS

# No of AWS Regions, AZ, Edge locations ?

Regions = 25

AZ = 81

Edge Locations = 121

# Alternative ways for connecting your console ?

* AWS CLI
* Session Manager

# Some AWS CLI commands

* aws ec2 describe-instances
* aws ec2 start/stop/terminate instance id
* aws s3 ls
* aws s3 cp
* aws ec2 import

# How to backup a running instance ?

* Snapshot
* AMI
* Aws Backup

# What is AWS Outpost ?

AWS Data Center on-premise.

# Issues during login to & PC If your Linux-build server getting slow down, what will you do to check ?

* MAX CPU
* MAX FILE Usage
* PEM FILE issues
* Port no not open in NACL or Security Group

# While connecting to your EC2 instances, what are the possible connection issues one might face ?

* Connection time out: Connection may time out due to long running processes or low network speed.
* Permission denied: You may be denied permission to connect to EC2 instance if the host key is not found.
* Security Group is not check the port no.

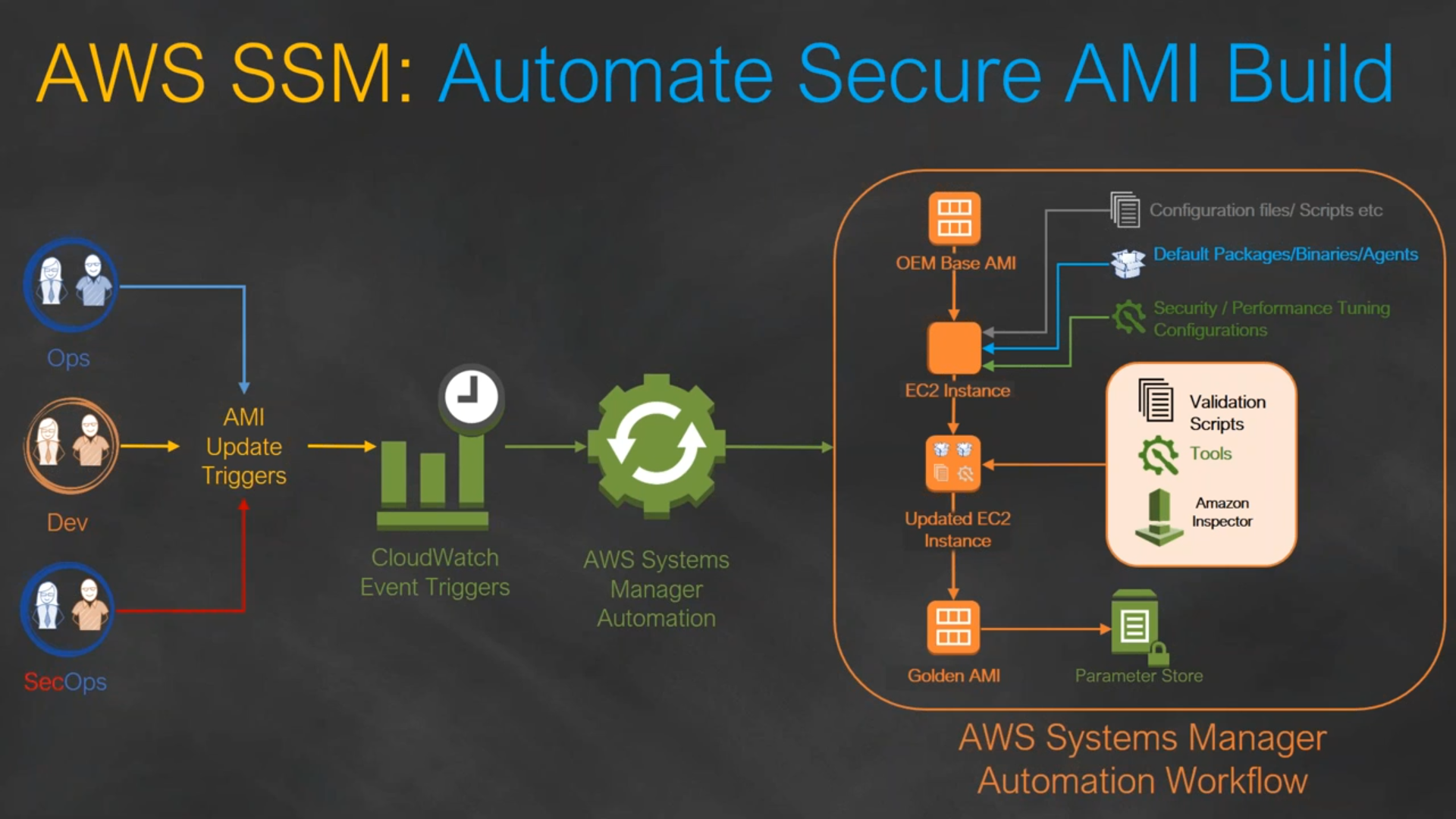
# What are the Global Services in AWS ?

* IAM
* Route53
* S3
* CloudFront

# How to automate ami update in autoscaling ?

CloudFormation or Elastic Beanstalk or CodeDeploy

# What is Systems Manager (SSM) & Golden AMI ?



Link: <https://github.com/miztiik/AWS-Demos/tree/master/How-To/setup-ami-lifecycle-management-using-ssm>

# Synchronous vs Asynchronous application ?

* A synchronous request blocks the client until operation completes
* Examples of synchronous communication are phone calls or video meetings.
* An asynchronous request doesn’t block the client i.e. browser is responsive.At that time, user can perform another operations also.
* Examples of Asynchronous communication happens when information can be exchanged independent of time. Examples of asynchronous communication are emails, online forums, and collaborative documents.

# Statefull vs Stateless Application ?

A Stateless app is an application program that does not save client data generated in one session for use in the next session with that client.

In a stateless application, different servers can be used to process different information

while in a stateful application, only one server is used to process all requests that are linked to the same state information.

A Stateful application saves data about each client session and uses that data the next time the client makes a request.

# What to do when PEM is Lost ?

We can take AMI of the instance and create a new Instance or If we need access to the particular instance withour creating a new one we should unmount the drive attach it to a different instance change the public keys in authorised keys file under .shh folder. and re mount it to the existing instance.

# Application FLow ?



Reference architecture for hosting WordPress on AWS



EC2

# How to create AMI ?

Choose a running Instance and actions create Image. It will create a new AMI. We can also use vmie to export our onpremise VM as a AMI to the cloud.

# Types of ec2 Models ?

Cost Based : ON demand, Reserved Ec2, Spot, Dedicated

Memory Based : CPU optimised, Memory Optimised, GPU OPtimised

# What is EC2, S3, EBS ?

EC2 is used to Provision VM instances

S3 is a Simple Storage Service where we can store dormant data upto 5TB per file size and virtually unlimited.

EBS is a volume which we attach to our EC2 instance. EBS volumes include SSD HDD Magnetic.

# EC2 vs ami ?

EC2 is a running server/instance provided by AWS. EC2 is created with the help of AMI

AMI image is a backup of an entire EC2 instance. Associated with an AMI image are EBS snapshots. AMI + Snapshot creates a EC2 Instance. we can create multiple EC2 instances with Single AMI.

AMI works in a single region only

# AMI vs Snapshot ?

Snapshot is a point in time backup of specific volume while AMI is is backup of the entire EC2 instance that might have multiple attached volumes

An EBS snapshot is a backup of a single EBS volume. ... An AMI image is a backup of an entire EC2 instance. Associated with an AMI image are EBS snapshots. Those EBS snapshots are the backups of the individual EBS volumes attached to the EC2 instance at the time the AMI image was created.

# EBS Volume Types ?

[Amazon EBS volume types - Amazon Elastic Compute Cloud](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volume-types.html)

**General Purpose SSD** – Standard - gp2, gp3 - 1 GiB - 16 TiB

**Provisioned IOPS SSD** - High Input/Output - io1, io2, io2 Block Express - 4 GiB - 16 TiB

**Throughput Optimized HDD** - Big Data, Data Warehouses, log processing - st1 - 125 GiB - 16 TiB

**Cold HDD** - Scenarios where the lowest storage cost is important - sc1 - 125 GiB - 16 TiB

**Magnetic** - Workloads where data is infrequently accessed - standard - 1 GiB-1 TiB

# Volume Add & Volume Create in EC2 ?

df -h -> File system

lsblk -> Shows attached Volumes

sudo mkfs -t xfs /dev/nvme1n1 -> Formatting

sudo mkdir /data -> Create a Folder

sudo mount /dev/nvme1n1 /data -> Mount folder with Disk

# Launch Template vs Launch Configuration ?

We cant edit both the LC and LT after it has been created.

Launch Configuration are used with ASG.

Launch Template are used with creating a new EC2 instance as well as ASG. LT allows you to have multiple versions of a template.

# In-Place Deployment ?

During In-place Deployment the there will be a down time the system will be offline during the deployment. Load balancer so that each instance is deregistered during its deployment and then restored to service after the deployment is complete.

# Blue Green Deployment or Update EC2 without Downtime?

Set of Server in under Blue and Green. While updating ill change the lb to point Blue while I update the Green with Code with Latest code. After Updation change the LB to point green again.

# How is a spot Instance different from an On-demand Instance ?

A Spot Instance is an instance that uses spare EC2 capacity that is available for less than the On-Demand price.

The instances are acquired through a bidding process in which the customer specifies a price per hour he is willing to pay.

Because Spot Instances enable you to request unused EC2 instances at steep discounts, you can lower your Amazon EC2 costs significantly. We should not use Spot instances for hosting our servers. Spot Requests is used to check on the orders

IAM

# Security in AWS ?

Assign policy to Users using IAM.

We can connect via CLI using SSH.

# What is inline policy/ managed policy/ customer managed policy ?

use managed policies instead of inline policies.

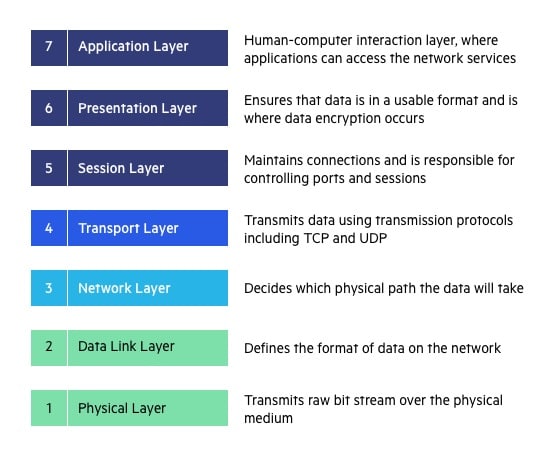
**inline policy** -> directly attached to the users/group/roles, Non-resuable

**customer managed policy** -> creates reusable object which we can apply to multiple users/groups

**aws managed policy** -> they are policies managed by aws so they will update it and manage it overtime

# Types of Load Balancer ?

1. Classic Load Balancer (4th 7th Layer of OSI)
2. Application Load Balancer (7th Layer of OSI)
3. Network Load Balancer -> (4th Layer of OSI) Used for Micro services / container based application
4. Gateway Load Balancer-> (3rd Layer of OSI)



# What is difference between application load balancer & classic load balancer & Network Load Balancer?

[Load balancer types - Amazon Elastic Container Service](https://docs.aws.amazon.com/AmazonECS/latest/developerguide/load-balancer-types.html)

**ELB** - ELB works at both layer 4 (TCP) and 7 (HTTP) and is the only load balancer that works in EC2-Classic. ELB only allows routing based on port number.

**ALB** - An Application Load Balancer (ALB) only works at layer 7 (HTTP). ALB can route requests to many ports on a single target. Plus, ALB can route requests to Lambda functions.

**NLB** - A Network Load Balancer (NLB) works at layer 4 (network layer)only and can handle both TCP and UDP, it uses static IP addresses and can be assigned Elastic IPs—not possible with ALB and ELB.

# What is gateway Loadbalancer (Newly Launched)

A Gateway Load Balancer operates at the third layer (network layer) of the Open Systems Interconnection (OSI) model, the network layer. It listens for all IP packets across all ports and forwards traffic to the target group.

# Is is possible to scale an Ec2 Instance vertically ?

To vertically scale an individual AWS EC2 instance up or down the instance has to be stopped, then the instance size changed, then restarted.

# Horizontal Scaling vs Vertical Scaling ?

Horizontal Scaling - Increasing the no of Instances using Auto Scaling Group

Vertical Scaling - Increasing the Power of Instance (Not available in Cloud)

# How session is maintained in LB ?

With sticky sessions, a load balancer assigns an identifying attribute to a user, typically by issuing a cookie or by tracking their IP details.

Then, according to the tracking ID, a load balancer can start routing all of the requests of this user to a specific server for the duration of the session.

# What is Proxy vs Reverse Proxy vs Load Balancer ?

# How to add Load Balancer to existing Instance ?

Open load balancer and edit targets and add your instances.

# Traffic manager vs Load balancer ?

Traffic manager is used to route traffic between regions. It works on a Global level

Load Balancer is used to route traffic inside region. Routes between instances Inside AZs

# What is Auto Scaling Group ?

ASG works based on Launch Template or Launch Configuration

Desired Value, Minimum Value, Maximum Value

AWS Auto Scaling is a service that automatically monitors and adjusts compute resources to maintain performance

As demand spikes, the AWS Auto Scaling service can automatically scale those resources, and, as demand drops, scale them back down.

# Which type of ELB is good for application load ?

Classic Load Balancer is likely to be the best choice if your routing and load-balancing needs can all be handled based on IP addresses and TCP ports. In contrast, the Application Load Balancer can address more complex load-balancing needs by managing traffic at the application level.

# Template Deployment ?

Deployment using AMI templates

S3

# What is 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.

Offering **99.999999999% (11 9’s)** of Durability

Max file size **5TB**

Max no of buckets **100** but can be increased.

# S3 Storage Classes

1. Standard
2. Intelligent Tiering
3. Standard-IA
4. One zone-IA
5. S3 Glacier
6. S3 Glacier Deep Archive
7. S3 Outpost.

# What is Glacier vs Glacier Deep Archive ?

S3 Glacier extremely low-cost cloud storage service for long-term backup.

Customers can reliably store large or small amounts of data for as little as $0.004 per gigabyte per month.

**Amazon S3 Glacier**: Expedited – 1-5 minutes, Standard – 3-5 hours, Bulk – 5-12 hours.

**Amazon S3 Glacier Deep Archive**: Expedited – Not available, Standard – within 12 hours, Bulk – within 48 hours

# How to secure s3 bucket/ What are the security available for users to access S3 ?

While creating bucket give Block Public Access to restrict Bucket access.

Using IAM, ACL, Bucket Policy

We can encrypt our data

Create CloudTrial Logs to check object level access and track

# S3 Encryption ?

objects are encrypted using server-side encryption with either **Amazon S3-managed keys (SSE-S3)** or AWS **KMS keys** stored in **AWS Key Management Service** (AWS KMS) (SSE-KMS).

# S3 Buck policy vs ACL ?

An S3 ACL is a sub-resource that's attached to every S3 bucket and object. It defines which AWS accounts or groups are granted access and the type of access.

S3 bucket policies and IAM policies define object-level permissions by providing those objects in the Resource element in your policy statements.

# S3 Bucket share region to region or. Account to Account ?

By enabling **Cross-origin resource sharing** (CORS) . CORS defines a way for client web applications that are loaded in one domain to interact with resources in a different domain.

We can also use Acess Control List ACL to share data between accounts.

Cross Region Replication is another way of sharing information

# Life Cycle Policy ?

Lifecycle policies allow you to automatically review objects within your S3 Buckets and have them moved to Glacier or have the objects deleted from S3. We can set life cycle rule suggesting that after 30 days move to Standard IA and after 60 days move to Glacier.

Route 53

# What is Hosted Zone ?

Hosted zone is analogous to a traditional DNS zone file

It represents a collection of records that can be managed together, belonging to a single parent domain name

All resource record sets within a hosted zone must have the hosted zone's domain name as a suffix

Public hosted zones contain records that specify how you want to route traffic on the internet.

Private hosted zones contain records that specify how you want to route traffic in an Amazon VPC.

# Types of routing policy and scenario ?

* **Simple routing policy** – Use for a single resource that performs a given function for your domain, for example, a web server that serves content for the example.com website.
* **Failover routing policy** – Use when you want to configure active-passive failover.
* **Geolocation routing policy** – Use when you want to route traffic based on the location of your users.
* **Geoproximity routing policy** – Use when you want to route traffic based on the location of your resources and, optionally, shift traffic from resources in one location to resources in another.
* **Latency routing policy** – Use when you have resources in multiple AWS Regions and you want to route traffic to the Region that provides the best latency with less round-trip time.
* **Multivalue answer routing policy** – Use when you want Route 53 to respond to DNS queries with up to eight healthy records selected at random.
* **Weighted routing policy** – Use to route traffic to multiple resources in proportions that you specify.

# Types of record set in Route 53 ?

Like a phone book, Route 53 lets you manage the IP addresses listed for your domain names in the Internet's DNS phone book. Route 53 also answers requests to translate specific domain names like into their corresponding IP addresses

A record type – Ipv4 routing to a resource

AAAA record type – Ipv6 Routing to resource

CNAME record type - A CNAME record maps DNS queries for the name of the current record, such as acme.example.com, to another domain (example.com or example.net) or subdomain (acme.example.com or zenith.example.org).

MX record type - An MX record specifies the names of your mail servers and, if you have two or more mail servers, the priority order. Each value for an MX record contains two values, priority and domain name.

NS record type - An NS record identifies the name servers for the hosted zone.

PTR record type - A PTR record maps an IP address to the corresponding domain name.

SOA record type - A start of authority (SOA) record provides information about a domain and the corresponding Amazon Route 53 hosted zone. For information about the fields in an SOA record

SQS & SNS

# Max size of message in SQS

Amazon Simple Queue Service (SQS) now has an **Extended Client Library** by JAVA that enables you to send and receive messages with payloads up to 2GB. Previously, message payloads were limited to 256KB. Using the Extended Client Library, message payloads larger than 256KB are stored in an Amazon Simple Storage Service (S3) bucket, using SQS to send and receive a reference to the payload location.

# What to connect queue from one account to another account ?

Create multiple Queue and make sure one is subscribed to another queue

VPC

# What is VPC ?

Virtual Private cloud is a isolated Network like we use in our onpremise Network.VPC lets us add subnets, associate security groups, and configure route tables and NACL. A subnet is a range of IP addresses in your VPC. We create our Instances in the Subnets

# why we require to create VPC ?

Amazon VPC enables you to build a virtual network in the AWS cloud - no VPNs, hardware, or physical datacenters required. You can define your own network space, and control how your network and the Amazon EC2 resources inside your network are exposed to the Internet.

# What is Subnet ? max no of subnets per VPC ?

A subnetwork or subnet is a logical subdivision of an IP network. The practice of dividing a network into two or more networks is called subnetting. Computers that belong to the same subnet are addressed with an identical most-significant bit-group in their IP addresses.

MAX 200 subnets per VPC

# Public vs Private Instance ?

Public Instance, Can be accessed from anywhere from the Globe. They have a Public IP address.

Private Instance Can be accessed from the resources within the VPC. They dont have a Public IP address. They will also have restriction ins the CIDR range in Security Group limiting access to certain resources.

# How to Troubleshoot VPC ?

Check CIDR range

Public and Elastic IP addresses

System and instance status checks

Security groups

VPC route table

Network ACLs

# Why do we make subnets ?

After creating a VPC with a CIDR we can create Subnet using by selecting the VPC which we want to create our VPC and giving CIDR range within the VPC range

# NACL (Network Access Control List) VS Security Group ?

Nacl are stateless i.e setting applied to Inbound rule is not applied to output rule , It is associated with Subnets. NACL has applied automatically to all the instances which are associated with an subnet.

Security Group are Stateful i.e settings applied to Inbound rule is also applied to outbound rule, It is associated with EC2 instance. Security Group is applied to an instance only when you specify a security group while launching an instance.

# What is routing table ?

A route table contains a set of rules, called routes, that are used to determine where network traffic from your subnet or gateway is directed. To put it simply, a route table tells network packets which way they need to go to get to their destination.

# What is Route Propogation in VPC ?

Route propagation allows a virtual private gateway to automatically propagate routes to the route tables. This means that you don't need to manually enter VPN routes to your route tables.

| Destination | Target |
| --- | --- |
| 10.0.0.0/16 | Local |
| 172.31.0.0/24 | vgw-11223344556677889 (propagated) |
| 172.31.0.0/24 | igw-12345678901234567 (static) |

# How you can connect a private subnet with a public subnet ?

Using Nat gateway.

# What is VPC peering ?

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses.

# Can VPC peering possible in two different region ?

## Inter-Region VPC Peering allows VPC resources like EC2 instances, RDS databases and Lambda functions running in different AWS regions to communicate with each other using private IP addresses, without requiring gateways, VPN connections or separate network appliances.

# What is transit gateway?

A transit gateway is a network transit hub that you can use to interconnect your virtual private clouds (VPCs) and on-premises networks.

# VPC peering vs VPC transit gateway

Transit Gateway solves the complexity involved with creating and managing multiple VPC peering connections at scale. While this makes TGW a good default for most network architectures, VPC peering is still a valid choice due to the following advantages it has over TGW:

1. **Lower cost** — With VPC peering you only pay for data transfer charges. Transit Gateway has an hourly charge per attachment in addition to the data transfer fees.
2. **No bandwidth limits** — With Transit Gateway, Maximum bandwidth (burst) per VPC connection is 50 Gbps. VPC peering has no aggregate bandwidth. Individual instance network performance limits and flow limits (10 Gbps within a placement group and 5 Gbps otherwise) apply to both options. Only VPC peering supports placement groups.
3. **Latency**— Unlike VPC peering, Transit Gateway is an additional hop between VPCs.
4. **Security Groups** compatibility — Security groups referencing works with intra-Region VPC peering. It does not currently work with Transit Gateway.

Within your Landing Zone setup, VPC Peering can be used in combination with the hub and spoke model enabled by Transit Gateway.

# What is NAT gateway ?

Network Address Translator Gateway is a highly available AWS managed service that makes it easy to connect to the Internet from instances within a private subnet in an Amazon Virtual Private Cloud (Amazon VPC). Public IP address is needed for connectivity to Internet, Since Private Instances does not have Public IP address we cannot connect to internet. NAT gateway solves that problem.

# NAT Gateway vs NAT Instance ?

|  |  |  |
| --- | --- | --- |
| Attribute | NAT gateway | NAT instance |
| Availability | Highly available. NAT gateways in each Availability Zone are implemented with redundancy. Create a NAT gateway in each Availability Zone to ensure zone-independent architecture. | Use a script to manage failover between instances. |
| Maintenance | Managed by AWS. You do not need to perform any maintenance. | Managed by you, for example, by installing software updates or operating system patches on the instance. |
| Bandwidth | Scale up to 45 Gbps. | Depends on the bandwidth of the instance type. |

# IGW vs Nat ?

An Internet Gateway is a logical connection between an Amazon VPC and the Internet. If a VPC does not have an Internet Gateway, then the resources in the VPC cannot be accessed from the Internet. We need a Route Table to Route Internet to our Subnets

NAT Gateway is a highly available AWS managed service that makes it easy to connect to the Internet from instances within a private subnet in an Amazon Virtual Private Cloud. You need one in each AZ since they can operate in a single AZ.

# VPC from A to B, B to C will it work for A to C ?

No

# What is VPC Endpoint ?

A VPC endpoint allows you to privately connect your VPC to supported AWS services without requiring an Internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Endpoints are virtual devices that are horizontally scaled, redundant, and highly available VPC components.

# Can we access a Endpoint without using Internet ?

VPC Endpoint which is free and lets you communicate to S3 and DynamoDB from private subnets without natting. For some AWS services, you can create an Interface VPC Endpoint which is cheaper than a NAT gateway.

# S3 to Vpc Connection ?

You can now access Amazon Simple Storage Service (Amazon S3) from your Amazon Virtual Private Cloud (Amazon VPC) using VPC endpoints. ... Additionally, you can control what buckets, requests, users, or groups are allowed through a specific VPC endpoint.

# Flow log in VPC ?

VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC. Flow log data can be published to Amazon CloudWatch Logs or Amazon S3. After you've created a flow log, you can retrieve and view its data in the chosen destination.

# How you monitor VPC ?

VPC flow logs

Cloud watch

Cloud Watch & Cloud Trial

# Cloud Trial vs Cloud Watch ?

Cloud Trial is a logging tool. We can create a new Customized trial It logs the access in AWS and stores it in a S3

Cloud Watch is a monitoring tool. It has Metrics to Monitor the resources on AWS. We can create Alarm in CW when a metric crosses the certain threshold.

# What is Cloud Watch agent ?

Collect more system-level metrics from Amazon EC2 instances across operating systems. Collect logs from Amazon EC2 instances and on-premises servers, running either Linux or Windows Server.

# What is metrics in cloudwatch ?

Metric resources are the fundamental monitoring unit in CloudWatch. A metric represents a time-ordered set of data points that are published to CloudWatch. Think of a metric as a variable to monitor, and the data points as representing the values of that variable over time.

1. *disk\_total*
2. *disk\_used\_percent*
3. *swap\_free*
4. *netstat\_total*
5. *cpu\_time\_active*

# How to conifgure Cloud Trial for Multi Account Buckets

Add Bucket Permission so files can be received from multiple accounts

<https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-set-bucket-policy-for-multiple-accounts.html>

Cloud Front

# What is Cloud Front ?

Amazon CloudFront is a content delivery network operated by Amazon Web Services. Content delivery networks provide a globally-distributed network of proxy servers that cache content, such as web videos or other bulky media, more locally to consumers, thus improving access speed for downloading the content.

# Edge Location vs Server ?

## Edge location is a physical location where a w s has its servers that are used to stores cdn caching. Edge locations are used to provide low latency access to the data.Thus it will provide low latency access to your data and end users will have better experience.

# Cloud Front default ttl, Can we incerase the ttl ?

default ttl is 1 day 86400 seconds, minimum is 0 seconds and maximum is 1 year

# How to refresh CloudFront Cache ?

CloudFront instance and go to the `Invalidations` tab where you can 'Create Invalidation'. Then click on 'Invalidate' and the invalidation will run: All Done! Your cache should be cleared and you are good to go once the run completed.

Lambda

# AWS Lambda ?

AWS Lambda is an event-driven, serverless computing platform provided by Amazon as a part of Amazon Web Services. It is a computing service that runs code in response to events and automatically manages the computing resources required by that code.

# What is purpose of using Lambda ?

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.

# WHat is the difference between Lambda vs EC2

Lambda is just a serverless tool to run our code. Lambda will be executed based on the trigger. Lambda will be charged based on the Code execution.

Where as EC2 is a VM instance which will be always running and we can install applications on top it

RDS

# What is Database ?

A Database contains multi tables. Each table has entries in the form of Rows and Columns. Multiple data bases are handled by RDBMS.

In AWS we have RDS to create hassle free database in cloud. which provides max size of 64 TiB

# Data base Architecture ?

Master in one AZ and Standy in another AZ. Creating Read Replica to create a read only copy of a Database to decrease the server load

# What is Amazon 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.

# Aurora Advantage ?

Aurora works in the form of Cluster.

Up to 5 times the throughput of MySQL and 3 times the throughput of PostgreSQL

Up to 64TB of auto-scaling SSD storage

6-way replication across three Availability Zones

Up to 15 Read Replicas with sub-10ms replica lag

Automatic monitoring and failover in less than 30 seconds

# read replica vs write replica?

Read Replica -> Creates a Read only copy of a running Database they are in Synchronous Replication.

Write Replica -> Creates a Write only copy of a running Database they are in Synchronous Replication

# Multi AZ Deployment vs Multi Region Deployment vs Read Replicas ?

|  |  |  |
| --- | --- | --- |
| Multi-AZ deployments | Multi-Region deployments | Read replicas |
| Main purpose is high availability | Main purpose is disaster recovery and local performance | Main purpose is scalability |
| Non-Aurora: synchronous replication; Aurora: asynchronous replication | Asynchronous replication | Asynchronous replication |
| Non-Aurora: only the primary instance is active; Aurora: all instances are active | All regions are accessible and can be used for reads | All read replicas are accessible and can be used for readscaling |
| Non-Aurora: automated backups are taken from standby; Aurora: automated backups are taken from shared storage layer | Automated backups can be taken in each region | No backups configured by default |
| Always span at least two Availability Zones within a single region | Each region can have a Multi-AZ deployment | Can be within an Availability Zone, Cross-AZ, or Cross-Region |

# Security in RDS to DB

\s  
SSL Cipher

# RDS vs DynamoDB ?

## Rds

* Managed relational (SQL) database
* Has several database instance types for different kinds of workloads and supports six database engines – Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database, and SQL Server.
* -128 TB for Aurora engine – 64 TB for MySQL, MariaDB, Oracle and PostgreSQL engines -16 TB for SQL Server engine.
* Can use VPC to give maximum security

## DynamoDB

* Fully managed key-value and document (NoSQL) database
* Delivers single-digit millisecond performance at any scale.
* Supports tables of virtually any size.
* Integrates with IAM

# How is Amazon RDS, Redshift & DynamoDB different ?

RDS is used to create RDBMS like MYSQL, Postgres, Aurora  
Dynamo DB is a No sql Database

Amazon Redshift is a fully-managed petabyte-scale cloud based data warehouse product designed for large scale data set storage and analysis. It is also used to perform large scale database migrations. Redshift is also used to analyze data.

# RDS vs EC2 in Mysql ?

Installing MYSQL in EC2 means we have to fully manage & maintain the Database. Where as if we choose MYSQL in RDS aws will take care of the configuration and maintanence of Database. We lose features like AutoScaling, Read Replica, Multi AZ deployment if we choose EC2 over RDS.

# Autoscaling in RDS ?

20gb – 16TB Storage. Only Storage Autoscaling is available and it increases the storage automatically as the storage gets full.

Cloud Formation

# What is CF stack ?

A stack is a collection of AWS resources that you can manage as a single unit.

All the resources in a stack are defined by the stack's AWS CloudFormation template. A stack, for instance, can include all the resources required to run a web application, such as a web server, a database, and networking rules.

# What is mapping in cloudformation template ?

* Mappings is a section of CloudFormation to help organizing parameters by named keys and corresponding values for each group
* if you want to set values based on a region, you can create a mapping that uses the region name as a key and contains the values you want to specify for each specific region.

# What are the main components of CloudFormation ?

We can use either JSON or YAML to write CloudFormation Template,

A CloudFormation template consists of 6 sections – **Description, Parameters, Mappings, Conditions, Resources and Outputs.**

# How is YAML different from JSON ?

YAML uses Double spaces as its hierarchy

JSON uses braces and brackets.

# How Infrastructure As Code processed & executes in AWS ?

CloudFormation reads a template and generates a stack, a set of resources ready to use on AWS. By using CloudFormation, you can define complex multi-resource applications and automatically deploy the resources on AWS. You can test your Infrastructure as Code by fine-tuning your configuration and repeating the process.

# CloudFormation Drift\_status ?

Manual Changes made in the Stack will be shown in Drift Status

Stack -> Update Stack to New changes made

Extras

# Elastic cache ?

Amazon ElastiCache is a fully managed in-memory data store and cache service by Amazon Web Services. The service improves the performance of web applications by retrieving information from managed in-memory caches, instead of relying entirely on slower disk-based databases.

# ECS & ECR ?

Amazon ECS is a regional service that simplifies running containers in a highly available manner across multiple Availability Zones within a Region

ECR - EC2 Container Registry , To store the Docker Images, ECS Agent

Create ECS Cluster using ECS Task and write code to add meta data for individual Instance

# ELK & Kibana ?

ELK stack gives you the ability to aggregate logs from all your systems and applications, analyze these logs, and create visualizations for application and infrastructure monitoring, faster troubleshooting, security analytics

Log stash gives the file to Kibana

Kibana is an open-source data visualization and exploration tool used for log and time-series analytics, application monitoring, and operational intelligence use cases. It offers powerful and easy-to-use features such as histograms, line graphs, pie charts, heat maps, and built-in geospatial support

# EMR

Amazon EMR is the industry-leading cloud big data platform for processing vast amounts of data using open source tools such as Apache Spark, Apache Hive, Apache HBase, Apache Flink, Apache Hudi, and Presto. Amazon EMR makes it easy to set up, operate, and scale your big data environments by automating time-consuming tasks like provisioning capacity and tuning clusters. With EMR you can run petabyte-scale analysis at less than half of the cost of traditional on-premises solutions and over 3x faster than standard Apache Spark. You can run workloads on Amazon EC2 instances, on Amazon Elastic Kubernetes Service (EKS) clusters, or on-premises using EMR on AWS Outposts.

# Inspector

It is used to run secuiryt checks on the AMI

an automated security assessment service that helps improve the security and compliance of applications deployed

Penetration Testing

# File beat

Filebeat is a lightweight shipper for forwarding and centralizing log data. Installed as an agent on your servers, Filebeat monitors the log files or locations that you specify, collects log events, and forwards them either to Elasticsearch or Logstash for indexing.

# What is AWS Security Manager ?

AWS Secrets Manager helps you protect secrets needed to access your applications, services, and IT resources. The service enables you to easily rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle.

# What is WAF ?

AWS WAF is a web application firewall that helps protect your web applications or APIs against common web exploits and bots that may affect availability, compromise security, or consume excessive resources. **cross-site request forgery (CSRF), cross-site-scripting (XSS), file inclusion, and SQL injection.**

# What is Parameter Store in System Manager ?

String, String List, SecureString -> KMS

aws ssm-get prameters by path --path /url/myapp/dev

Projects

# 3 tier Application Serverless

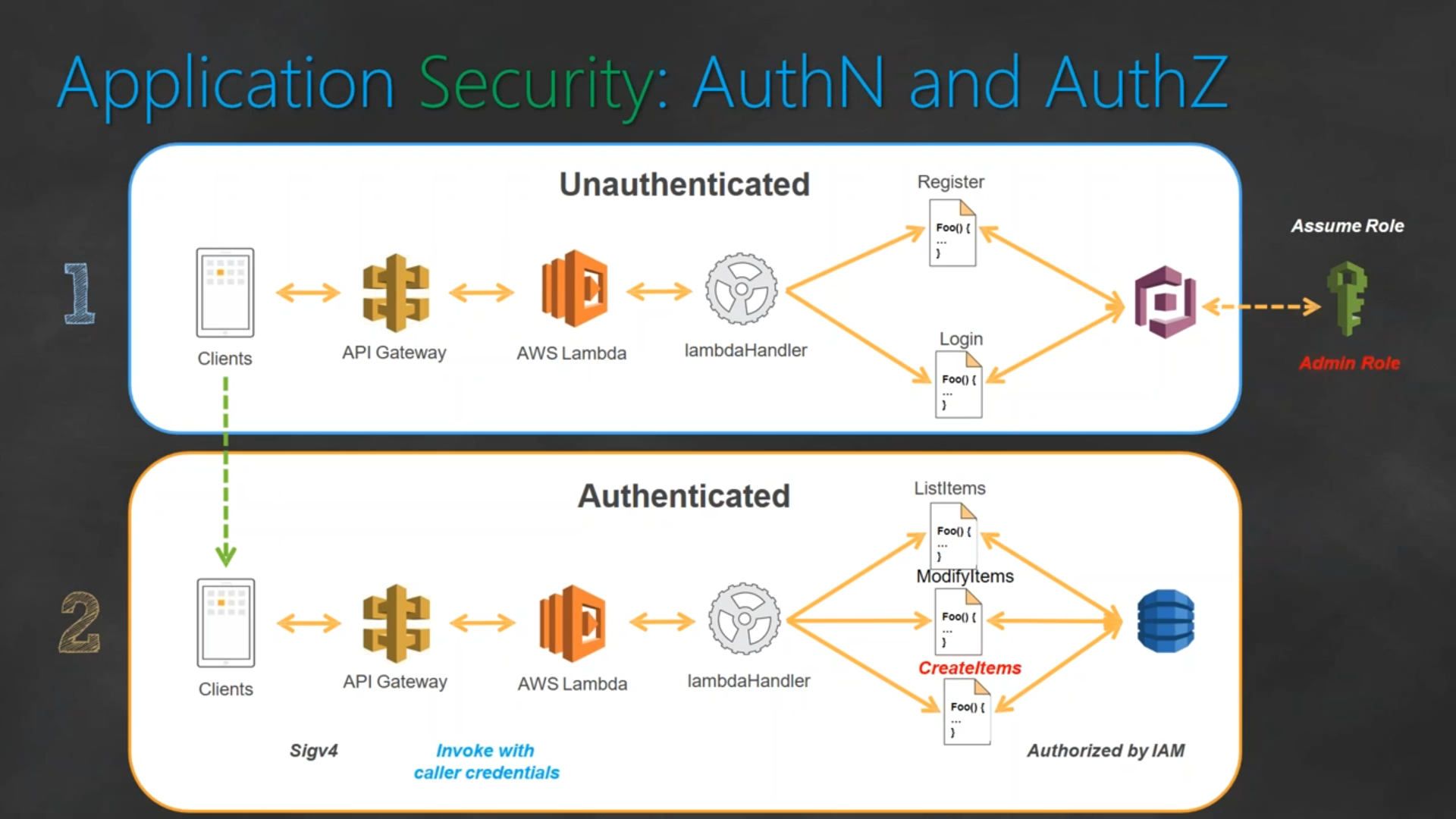
## Requirements

* 3 Tier Web - Application to built in AWS
* Use higher stack - aka- use XaaS where possible
* Enable 3rd Parties to call the API Securely
* Security in all elements of the architecture
* End-To-End Encryption is required
* AuthZ - Support users with multiple privileges

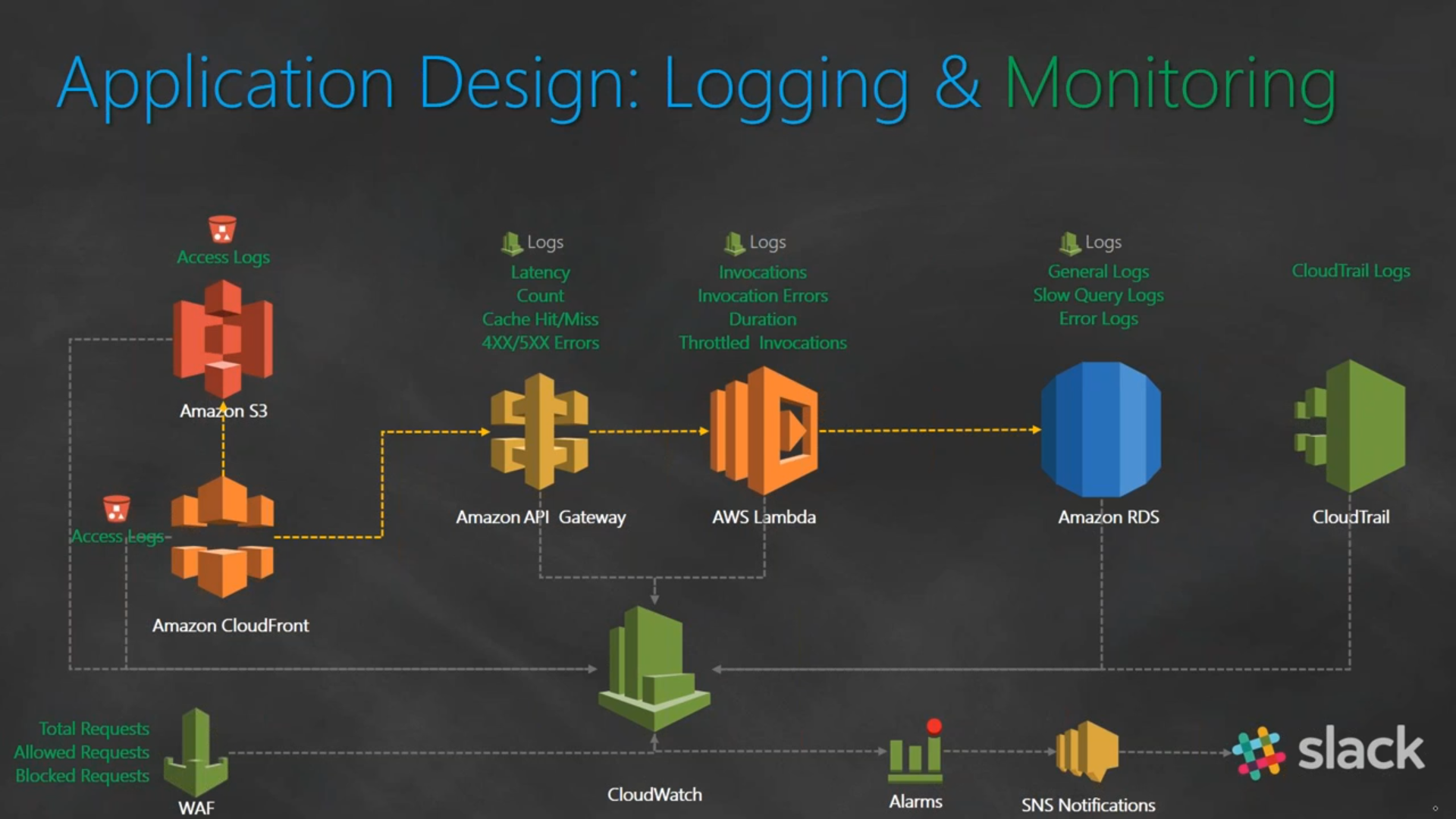
## Solution



## Authorisation for Different Users



## Logging the Users



DevOps Tools Version

|  |  |
| --- | --- |
| linux | RHEL 7.5 |
| git | 2.20 |
| jenkins | 2 |
| docker | 1.13 |
| ansible | 2.7 |
| terraform | 0.13 |
| python | 3.8 |
| App Dynamics | 4.5.13 |
| splunk | 7.3 |
| salt | Sodium: 3001 |
| kubernetes | 1.20 |
| Maven | 3.6 |
| Sonarqube | 8.3 |
| Tomcat | 9 |
| Prometheus | 2.28 |
| Nexus | 3.30 |

Git

# Git Commands

**git config --global user.name "Your Name"** - Set the name that will be attached to your commits and tags.

**git config --global user.email "you@example.com"** - Set the e-mail address that will be attached to your commits and tags.

**git config --global color.ui auto** - Enable some colorization of Git output.

**git init [project name]git clone [project url]**

**git status**

**git diff [file]**

**git diff --staged [file]**

**git checkout -- [file]**

**git add [file]**

**git commit [-m "message here"]**

**git rm [file] -** Remove file from working directory and add deletion to staging area.

**git branch [-a]** List all local branches in repository. With -a: show all branches (with remote).

**git branch [name]** Create new branch, referencing the current HEAD.

**git checkout [-b] [name]** Switch working directory to the specified branch. With -b: Git will create the specified branch if it does not exist.

**git merge [from name]** Join specified [from name] branch into your current branch (the one you are on currenlty).

**git branch -d [name]** Remove selected branch, if it is already merged into any other. -D instead of -d forces deletion

**git log [-n count]** List commit history of current branch. -n count limits list to last n commits.

**git log --oneline --graph --decorate** An overview with references labels and history graph. One commit per line.

**git log ref..** List commits that are present on current branch and not merged into ref. A ref can be e.g. a branch name or a tag name.

**git log ..ref** List commit, that are present on ref and not merged into current branch.

**git reflog** List operations (like checkouts, commits etc.) made on local repository

**git fetch [remote]** Fetch changes from the remote, but not update tracking branches.

**git fetch --prune [remote]** Remove remote refs, that were removed from the remote repository.

**git pull [remote]** Fetch changes from the remote and merge current branch with its upstream.

**git push [--tags] [remote]** Push local changes to the remote. Use --tags to push tags.

**git reset [file]** - Get file back from staging area to working directory

**git reset --soft**

**git reset --mixed**

**git reset --hard**

**git revert HEAD** - Undo the commit and create a new commit

**git branch –d [head]** - Delete the Current Branch

**git branch -d <branch>** - (Only works if it is merged with another branch)

**git branch -D <branch>** - (Forcing the delete)

**git reflog --no-abbrev** - contains all log information of the steps you have done

**git checkout** [sha]

**git checkout -b** [branchname] <sha>

**git branch --merged master** lists branches merged into master

**git branch --merged** lists branches merged into HEAD (i.e. tip of current branch)

**git branch --no-merged** lists branches that have not been merged

**git stash**

**git stash apply**

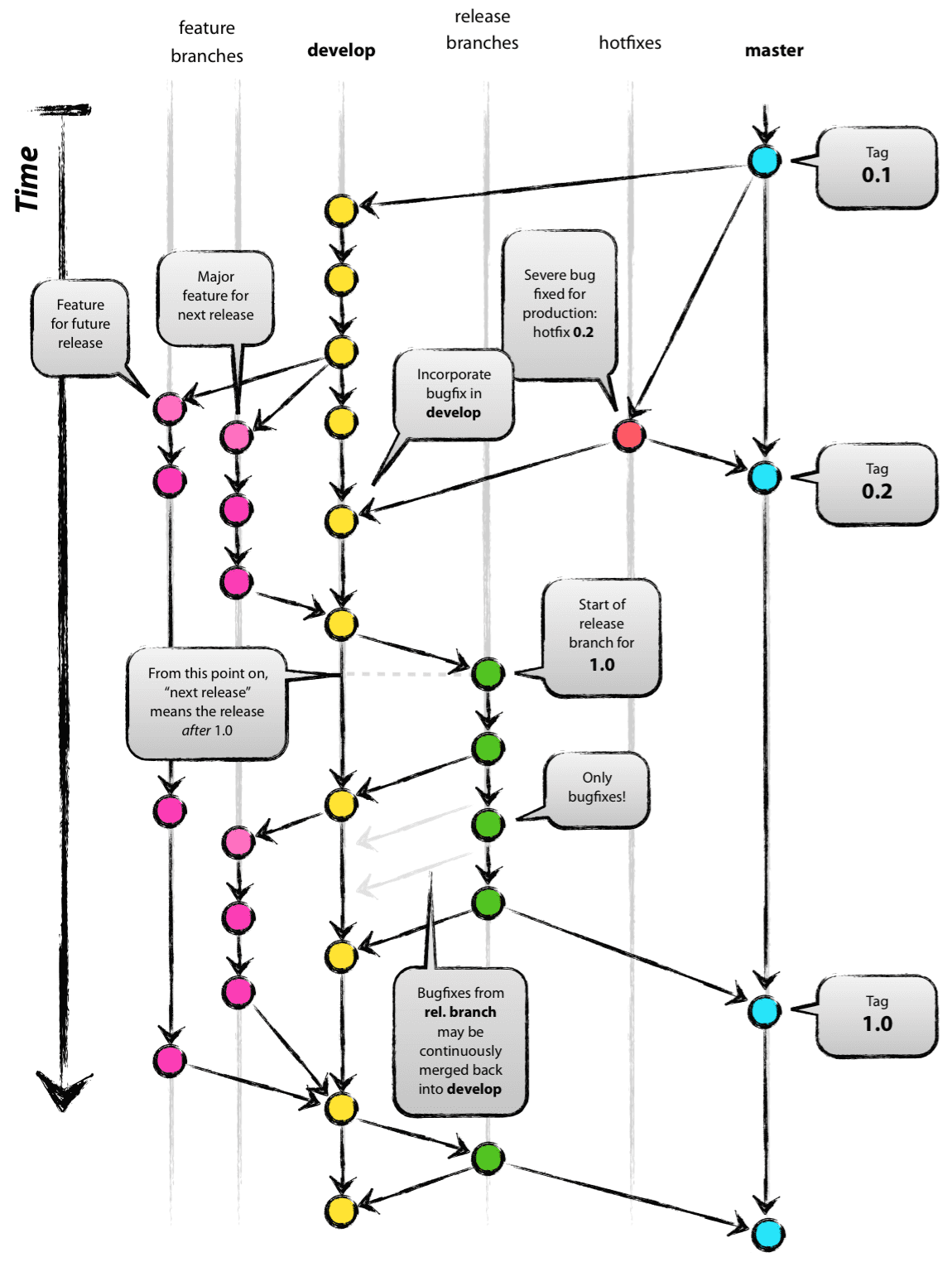
**git stash pop**

**git stash drop**

# CVCS vs DVCS ?

|  |  |
| --- | --- |
| CVCS | DVCS |
| One Central repository which is a server | Every user has a complete repository which is called as local repository |
| Every User who needs to access must be connected to the Network | We can go offline in DVCS but Network is needed to share repositories with others |
| SVN | Git, Mercurial, Bazaar |

# Which Branching Strategy used in your Project ?



# Which branch will be pushed to release?

release branch

# What is GIT?

GIT is a distributed version control system and source code management (SCM) system with an emphasis to handle small and large projects with speed and efficiency.

# What is a repository in GIT?

A repository contains a directory named .git, where git keeps all of its metadata for the repository. The content of the .git directory are private to git.

# What is Git Client?

“Git” is really just a collection of individual commands you execute in the shell (Appendix A). This interface is not appealing for everyone. Some may prefer to do Git operations via a client with a graphical interface

# What is GIT version control?

With the help of GIT version control, you can track the history of a collection of files and includes the functionality to revert the collection of files to another version. Each version captures a snapshot of the file system at a certain point of time. A collection of files and their complete history are stored in a repository.

# what is the use of SCM?

Source Code Management (SCM) and version control systems ensure all members of a team stay on top of changes to source code and related files. These tools are also crucial in coordinating parallel work on different features and the integration of the features for software releases.

# Why is git more secure?

Git is secure against modifying the content without being able to know that something went wrong. Each commit references to the parents hash. Each item in the commit has its hash as reference and the commit itself is also hashed. So changing just one item would change the complete hash of the commit itself.

# What are the advantages of using GIT ? / Why do you prefer Git ?

1. Data redundancy and replication
2. High availability
3. Only one.git directory per repository
4. Superior disk utilization and network performance
5. Collaboration friendly
6. Any sorts of projects can use GIT

# What is the difference between Git and Github?

Git is a revision control system, a tool to manage your source code history.

GitHub is a hosting service for Git repositories.

GitHub is a website where you can upload a copy of your Git repository. It is a Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.

# How do you rate GIT in terms of speed?

Git is fast. Speed and performance has been a primary design goal of the Git from the start. With Git, nearly all operations are performed locally, giving it a huge speed advantage on centralized systems that constantly have to communicate with a server somewhere.

Git was built to work on the Linux kernel, meaning that it has had to effectively handle large repositories from day one. Git is written in C, reducing the overhead of runtimes associated with higher-level languages.

# Mention some of the best graphical GIT client for LINUX?

Some of the best GIT client for LINUX is

a) Git Cola

b) Git-g

c) Smart git

d) Giggle

e) Git GUI

f) qGit

# What is Subgit? Why to use Subgit?

‘Subgit’ is a tool for a smooth, stress-free SVN to Git migration. Subgit is a solution for a company -wide migration from SVN to Git that is:

1. It is much better than git-svn
2. No requirement to change the infrastructure that is already placed
3. Allows to use all git and all sub-version features
4. Provides genuine stress –free migration experience.

# Name a few Git repository hosting services

Pikacode

Visual Studio Online

GitHub

GitEnterprise

SourceForge.net

# What is the difference between GIT and SVN?

The difference between GIT and SVN is

1. Git is less preferred for handling extremely large files or frequently changing binary files while SVN can handle multiple projects stored in the same repository.
2. GIT does not support ‘commits’ across multiple branches or tags. Subversion allows the creation of folders at any location in the repository layout.
3. Gits are unchangeable, while Subversion allows committers to treat a tag as a branch and to create multiple revisions under a tag root.

# What is GitOps?

GitOps is a paradigm or a set of practices that empowers developers to perform tasks which typically fall under the purview of IT operations. GitOps requires us to describe and observe systems with declarative specifications that eventually form the basis of continuous everything.

# What language is used in GIT?

GIT is fast, and ‘C’ language makes this possible by reducing the overhead of runtimes associated with higher languages.

# Difference between perforce and git?

Git is DVCS

perforce is CVCS

# How git instaweb is used?

‘Git Instaweb’ automatically directs a web browser and runs webserver with an interface into your local repository.

# Why GIT better than Subversion?

GIT is an open source version control system; it will allow you to run ‘versions’ of a project, which show the changes that were made to the code overtime also it allows you keep the backtrack if necessary and undo those changes. Multiple developers can checkout, and upload changes and each change can then be attributed to a specific developer.

# What is the function of ‘git config’?

The ‘git config’ command is a convenient way to set configuration options for your Git installation. Behaviour of a repository, user info, preferences etc. can be defined through this command.

# what is git init vs git init –bare ?

**git init** will create a .git folder containg all the files related to git like ref, object

**git init --bare** will not create a .git folder it will simply create all the files inside .git folder in our dir which we ran git init --bare . This command it used for setting up a remote repository in our local server

# How can you create a repository in Git?

In Git, to create a repository, create a directory for the project if it does not exist, and then run command “git init”. By running this command .git directory will be created in the project directory, the directory does not need to be empty.

# What are bare and non-bare repositories?

developers clone and fetch from the bare repository and push updates to it... if you set up a repository into which developers push changes, it should be bare. In effect, this is a special case of the more general best practice that a published repository should be bare.

# What is ‘bare repository’ in GIT?

To co-ordinate with the distributed development and developers team, especially when you are working on a project from multiple computers ‘Bare Repository’ is used. A bare repository comprises of a version history of your code.

# How do you identify a git base directory?

Folder which containes .git folder

# What is HEAD?

When working with Git, only one branch can be checked out at a time - and this is what's called the "HEAD" branch. Often, this is also referred to as the "active" or "current" branch. Git makes note of this current branch in a file located inside the Git repository, in . git/HEAD .

# What is ‘head’ in git and how many heads can be created in a repository?

A ‘head’ is simply a reference to a commit object. In every repository, there is a default head referred as “Master”. A repository can contain any number of heads.

# What is the command you can use to write a commit message?

The command that is used to write a commit message is “**git commit –m**”. The –a on the command line instructs git to commit the new content of all tracked files that have been modified. You can use “git add<file>” before git commit –a if new files need to be committed for the first time.

# What does commit object contain?

a) A set of files, representing the state of a project at a given point of time

b) Reference to parent commit objects

c) An **SHA Simple Hashing Algorithm** name, a 40 character string that uniquely identifies the commit object.

# Explain what is commit message?

Commit message is a feature of git which appears when you commit a change. Git provides you a text editor where you can enter the modifications made in commits.

# How can you fix a broken commit message?

To fix any broken commit, you will use the command “**git commit --amend**”. By running this command, you can fix the broken commit message in the editor.

# What are hooks?

Git hooks are scripts that run automatically every time a particular event occurs in a Git repository. They let you customize Git's internal behavior and trigger customizable actions at key points in the development life cycle.

# What does ‘hooks’ consist of in git?

This directory consists of Shell scripts which are activated after running the corresponding Git commands. For example, git will try to execute the post-commit script after you run a commit.

# How to restrict few filetypes from getting committed?

Using **.gitignore global** or using **Hooks**

# What is the best way to stop developers from doing an accidental commits?

By creating a Hook using shell script we can stop developers from doing commits on DEV & Master

*#!/bin/sh*

*branch="$(git rev-parse --abbrev-ref HEAD)"*

*if [ "$branch" = "dev" ]; then*

*echo "Dev Branch commit is blocked"*

*exit 1*

*fi*

*if [ "$branch" = "master" ]; then*

*echo "Master Branch commit is blocked"*

*exit 1*

*fi*

# In Git how do you revert a commit that has already been pushed and made public?

**git revert <name of bad commit>**

There can be two answers to this question and make sure that you include both because any of the below options can be used depending on the situation:

Remove or fix the bad file in a new commit and push it to the remote repository. This is the most natural way to fix an error. Once you have made necessary changes to the file, commit it to the remote repository for that I will use

git commit -m “commit message” .

Create a new commit that undoes all changes that were made in the bad commit.to do this I will use a command

# Why is it advisable to create an additional commit rather than amending an existing commit?

There are couple of reason

a) The amend operation will destroy the state that was previously saved in a commit. If it’s just the commit message being changed then that’s not an issue. But if the contents are being amended then chances of eliminating something important remains more.

b) Abusing “git commit- amend” can cause a small commit to grow and acquire unrelated changes.

# What is the purpose of branching in GIT?

The purpose of branching in GIT is that you can create your own branch and jump between those branches. It will allow you to go to your previous work keeping your recent work intact.

# To delete a branch what is the command that is used?

**git branch –d [head]**

**git branch -d <branch>** (Only works if it is merged with another branch)

**git branch -D <branch>** (Forcing the delete)

Once your development branch is merged into the main branch, you don’t need development branch.

To delete a branch use, the command.

# How to recover a Deleted Branch ?

**git reflog --no-abbrev** - contains all log information of the steps you have done

**git checkout [sha]**

**git checkout -b [branchname] [sha]**

# How will you know in GIT if a branch has been already merged into master?

**git branch --merged master** lists branches merged into master

**git branch --merged** lists branches merged into HEAD (i.e. tip of current branch)

**git branch --no-merged** lists branches that have not been merged

# What is the function of ‘git checkout’ in git?

A ‘git checkout’ command is used to update directories or specific files in your working tree with those from another branch without merging it in the whole branch.

# What is the use of ‘git log’?

To find specific commits in your project history- by author, date, content or history ‘git log’ is used.

# What is git Is-tree?

‘git Is-tree’ represents a tree object including the mode and the name of each item and the SHA-1 value of the blob or the tree.

# What is the function of ‘GIT PUSH’ in GIT?

‘GIT PUSH’ updates remote refs along with associated objects.

# What is “Staging Area” or “Index” in GIT?

Before completing the commits, it can be formatted and reviewed in an intermediate area known as ‘Staging Area’ or ‘Index’.

# What is ‘git add’ is used for?

‘git add’ adds file changes in your existing directory to your index.

# What is the function of ‘git rm’?

git rm is used to remove a file from a Git repository.

# Git Remote vs Git Add vs Git Clone ?

git remote add <alias> <URL> -> adds remote URL

git add -> adds file to staging area

git clone -> downloads an existing Git repository to your local computer.

# What is the function of ‘git diff ’ in git?

‘git diff ’ shows the changes between commits, commit and working tree etc.

# What is ‘git status’ is used for?

As ‘Git Status’ shows you the difference between the working directory and the index, it is helpful in understanding a git more comprehensively.

# What is the difference between the ‘git diff ’and ‘git status’?

'git diff ' depicts the changes between commits, commit and working tree

'git status' shows you the difference between the working directory and the index

# Git checkout ?

git checkout command lets you navigate between the branches created by git branch

# What is merge ?

merges commit from one branch to another

# Git 3 way merge ?

With a three-way merge, it can compare the two files, but it can also compare each of them against the original copy (before either of you changed it). So it can see that you removed the first line, and that your friend added the last line. And it can use that information to produce the merged version.

# Git mergeconflict how to solve ?

we can use git mergetool to resolve the conflict and make changes to our file

git reset – hard or git reset -- soft or git reset --mixed

# What is Rebase ? What is the syntax for “Rebasing” in Git?

Rebasing is the process of moving or combining a sequence of commits to a new base commit.

# git cherrypick

git cherry-pick is a powerful command that enables arbitrary Git commits to be picked by reference and appended to the current working HEAD. Cherry picking is the act of picking a commit from a branch and applying it to another.

# How can you bring a new feature in the main branch?

To bring a new feature in the main branch, you can use a command “git merge” or “git pull command”.

# What is a ‘conflict’ in git?

A ‘conflict’ arises when the commit that has to be merged has some change in one place, and the current commit also has a change at the same place. Git will not be able to predict which change should take precedence.

# How can conflict in git resolved?

To resolve the conflict in git, edit the files to fix the conflicting changes and then add the resolved files by running “git add” after that to commit the repaired merge, run “git commit”. Git remembers that you are in the middle of a merger, so it sets the parents of the commit correctly

# What is another option for merging in git?

“Rebasing” is an alternative to merging in git.

# Git fetch vs Git Pull vs Git Clone ?

**git fetch** will only update your local repository, we need to use checkout or merge command to update it to our working copy

**git pull** is a combination of **git fetch + git merge,** it will update your both local repo and make changes in working directory

**git clone** is used for just downloading exactly what is currently working on the remote server repository and saving it in your machine's folder where that project is placed

I have even heard of people running **git fetch periodically in a cron job** in the background.

# What is the difference between git pull and git fetch?

Git pull command pulls new changes or commits from a particular branch from your central repository and updates your target branch in your local repository.

Git fetch is also used for the same purpose but it works in a slightly different way. When you perform a git fetch, it pulls all new commits from the desired branch and stores it in a new branch in your local repository. If you want to reflect these changes in your target branch, git fetch must be followed with a git merge. Your target branch will only be updated after merging the target branch and fetched branch. Just to make it easy for you, remember the equation below:

Git pull = git fetch + git merge

# What is the difference between ‘git remote’ and ‘git clone’?

‘git remote add’ just creates an entry in your git config that specifies a name for a particular URL. While, ‘git clone’ creates a new git repository by copying and existing one located at the URI.

# What is the function of git clone?

The git clone command creates a copy of an existing Git repository. To get the copy of a central repository, ‘cloning’ is the most common way used by programmers.

git clone <Repo URL>

# Git restore ?

**git-restore** is a tool to revert non-commited changes. Non-commited changes are:

a) changes in your working copy,

b) content in your index (a.k.a. staging area).

# git reset –hard vs –soft vs –mixed ?

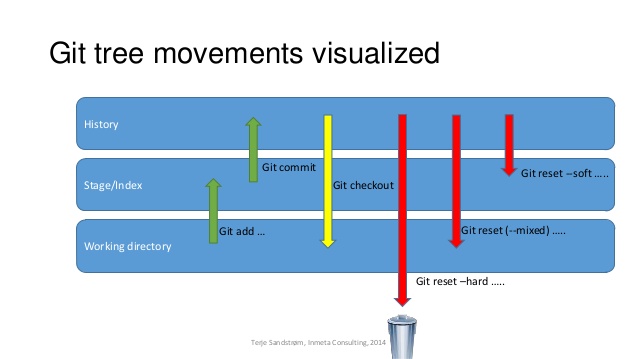
In other words, --soft is discarding last commit, --mix is discarding last commit and add, --hard is discarding last commit,add and any changes you made on the codes which is the same with git checkout HEAD

# Git revert vs git reset ?

difference between git reset and git revert is that git reset will reset the state of the branch to a previous state by dropping all the changes post the desired commit while git revert will reset to a previous state by creating new reverting commits.

**git revert HEAD** - Undo the commit and create a new commit

# What is git reset? How many types of reset is possible? What is the difference?



# how to get back to previous commit ?

git reset --hard

# What is the function of ‘git reset’?

The function of ‘Git Reset’ is to reset your index as well as the working directory to the state of your last commit.

# What is GIT stash?

GIT stash takes the current state of the working directory and index and puts in on the stack for later and gives you back a clean working directory. So in case if you are in the middle of something and need to jump over to the other job, and at the same time you don’t want to lose your current edits then you can use GIT stash.

# What is GIT stash drop?

When you are done with the stashed item or want to remove it from the list, run the git ‘stash drop’ command. It will remove the last added stash item by default, and it can also remove a specific item if you include as an argument.

# What is the function of ‘git stash apply’?

When you want to continue working where you have left your work, ‘git stash apply’ command is used to bring back the saved changes onto the working directory.

# Git Tag

Tags are ref's that point to specific points in Git history. Tagging is generally used to capture a point in history that is used for a marked version release (i.e. v1. 0.1). A tag is like a branch that doesn't change. Unlike branches, tags, after being created, have no further history of commits.

Jenkins

# Jenkins Executors ?

A Jenkins executor is one of the basic building blocks which allow a build to run on a node/agent (e.g. build server).

executor as a single “process ID”, or as the basic unit of resource that Jenkins executes on your machine to run a build.

Jenkins default 2 Executors

# What is Flyweight executors ?

Flyweight executors are unlimited and will be created automatically when needed, unlike heavyweight executors, which are limited based on their node's configuration. Every Pipeline script itself runs on the master using a flyweight executor

# Master Slave ? / What is Jenkins slaves ?

Under Nodes & Cloud -> We have a option to add new node -> select Launch agent via SSH and install the plugin

Laucnh Agent using SSH. Give SSH username and the private key and the IP address of the machine it will create the node

A Master-Slave architecture to manage distributed builds. The machine where we install Jenkins software will be Jenkins master and that run's on port 8080 by default. On the slave machine, we install a program called Agent. This agent requires JVM.

# What is Jenkins Agent ?

A Jenkins agent is an executable, residing on a node, that is tasked by the controller to run a job. Managing the different Jenkins agents is the job of the agent controller, also known as the master node. In a single Jenkins node configuration, the controller, can also act as both a Jenkin agent and run build jobs.

# how to check if node is alive ?

We can check from Jenkins Node agent.

# How to configure a cloud access in Jenkins ?

AMAZON EC2 PLUGIN is used to configure cloud access

Create a policy with the user

Manage Jenkins -> Configure System -> Add a new Cloud -> Compute Engine -> AMAZON EC2 PLUGIN

give the credentials

build tool and inexecute shell give the AWS CLI command

Give AWS cli access to

# Jenkins Shared Library

A shared library is a collection of independent Groovy scripts which you pull into your Jenkinsfile at runtime.

A shared library is a collection of groovy files (DSLs + Groovy). we will be using Github as our git repo. https://github.com/devopscube/jenkins-shared-library. You can clone this repo to get the basic structure of the shared library.

jenkins-shared-library

|\_\_\_\_vars

|\_\_\_\_src

|\_\_\_\_resources

**vars:** This directory holds all the global shared library code that can be called from a pipeline.

**src:** It is a regular java source directory. It is added to the classpath during every script compilation. Here you can add custom groovy code to extend yous shared library code. There will be scenarios where your groovy DSL’s will not be flexible enough to achieve some functionality. In this case, you can write custom groovy functions in src and call it in your shared library code.

**resources:** the libraryResource step to be used from an external library to load associated non-Groovy files

# how to use shared library in jenkins pipeline ?

<https://www.lambdatest.com/blog/use-jenkins-shared-libraries-in-a-jenkins-pipeline/>

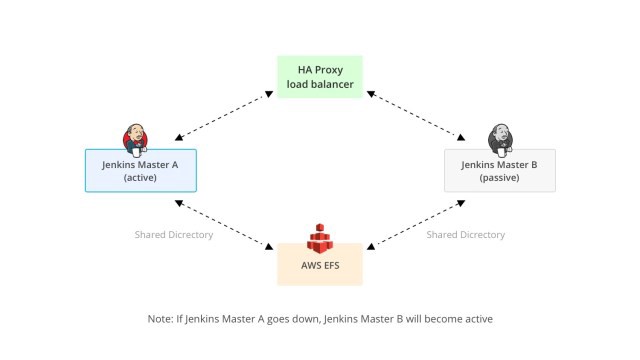
<https://www.jenkins.io/doc/book/pipeline/shared-libraries/>

# High Availability of Jenkins ?

* AWS Account with access to EC2 and EFS
* Three instances — one for HA proxy and two with Jenkins installed
* Knowledge of EFS (Elastic File System) on AWS
* Knowledge of HAProxy.

HA Proxy is configured to perform a availability check on the Primary Jenkins instance. As long as the Primary Instance is up and running, traffic from the user is relayed to the it. When HA Proxy detects a failure, it starts redirecting traffic to the Secondary Jenkins instance instead of the Primary.

* The user’s web browser sends a request to the load balancer over port 80 which is assigned to commonly used HTTP, using the fully qualified domain name such as http://jenkins.domain.com
* The Domain Name System (DNS) reaches the IP address of the domain name entered and fetches data from there.
* If a load-balancer server is unavailable, the request to that server will be timed out. If the load-balancer is available, then according to the availability of the machine, it routes requests.
* Referencing the load-balancer pool in its configuration file, HAProxy determines the application server to which the client request has to be routed. This server receiving the request is usually a part of the auto-scaling array comprising dedicated application servers.
* HAProxy forwards this request to the server port referenced in its configuration file (generally port 80).



# What is HAProxy used for ?

HAProxy is free, open source software that provides a high availability load balancer and proxy server for TCP and HTTP-based applications that spreads requests across multiple servers.

# Jenkins Backup ?

Install Backup Plugin -> Manage Jenkins -> Backup Maganer -> Give File Path & Root Directory path -> initiate backup -> after complete -> We can restore any time we want.

Will Install Plugin, or backup jenkins war file

Thin backup Plugin.

# Plugins Used in your Project ?

Git

Maven

Sonarqube

SSH Build Agent

Deploy to Container

Role based Strategy Authorisation

Delivery Pipeline view

Parameterised Trigger Plugin

Declarative Pipeline

# What is Jenkins Pipeline ?

Jenkins Pipeline is a combination of plugins that supports integration and implementation of continuous delivery pipelines.

Jenkins Pipeline (or simply "Pipeline") is a suite of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins. A continuous delivery pipeline is an automated expression of your process for getting software from version control right through to your users and customers.

# What are different types of Jenkins Pipeline ?

Scripted Pipeline

Declarative Pipeline

# What is Declarative Pipeline?

[Comprehensive Guide To Jenkins Declarative Pipeline [With Examples] (lambdatest.com)](https://www.lambdatest.com/blog/jenkins-declarative-pipeline-examples/)

*Jenkinsfile (Declarative Pipeline)*

*pipeline {*

*agent any*

*stages {*

*stage('Stage 1') {*

*steps {*

*echo 'Hello world!'*

*}*

*}*

*}*

*}*

* The key difference between Declarative pipeline and Scripted pipeline would be with respect to their syntaxes and their flexibility.
* Declarative pipeline is a relatively new feature that supports the pipeline as code concept. It makes the pipeline code easier to read and write. This code is written in a Jenkinsfile which can be checked into a source control management system such as Git.
* Whereas, the scripted pipeline is a traditional way of writing the code. In this pipeline, the Jenkinsfile is written on the Jenkins UI instance.
* Though both these pipelines are based on the groovy DSL, the scripted pipeline uses stricter groovy based syntaxes because it was the first pipeline to be built on the groovy foundation. Since this Groovy script was not typically desirable to all the users, the declarative pipeline was introduced to offer a simpler and more optioned Groovy syntax.
* The declarative pipeline is defined within a block labelled ‘pipeline’ whereas the scripted pipeline is defined within a ‘node’.

**Declarative pipeline**

Structure and syntax of the Declarative pipeline:

The Agent is where the whole pipeline runs. Example, Docker. The Agent has following parameters:

**any** - Which mean the whole pipeline will run on any available agent.

**none** - Which mean all the stages under the block will have to declared with agent separately.

**label** - this is just a label for the Jenkins environment

**docker** - this is to run the pipeline in Docker environment.

The Declarative pipeline code will looks like this:

*pipeline {*

*agent any*

*stages {*

*stage('Build') {*

*steps {*

*echo 'Building..'*

*}*

*}*

*stage('Test') {*

*steps {*

*echo 'Testing..'*

*}*

*}*

*stage('Deploy') {*

*steps {*

*echo 'Deploying....'*

*}*

*}*

*}*

*}*

# What is Scripted Pipeline ?

* The scripted pipeline is a traditional way of writing the Jenkins pipeline as code. Ideally, Scripted pipeline is written in Jenkins file on web UI of Jenkins.
* Unlike Declarative pipeline, the scripted pipeline strictly uses groovy based syntax. Since this, The scripted pipeline provides huge control over the script and can manipulate the flow of script extensively.
* This helps developers to develop advance and complex pipeline as code.

Structure and syntax of the scripted pipeline:

**Node Block:**

Node is the part of the Jenkins architecture where Node or agent node will run the part of the workload of the jobs and master node will handle the configuration of the job. So this will be defined in the first place as

*node {*

*}*

Stage Block:

Stage block can be a single stage or multiple as the task goes. And it may have common stages like

Cloning the code from SCM

Building the project

Running the Unit Test cases

Deploying the code

Other functional and performance tests.

So the stages can be written as mentioned below:

*stage {*

*}*

So, Together, the scripted pipeline can be written as mentioned below.

*node ('node-1') {*

*stage('Source') {*

*git 'https://github.com/digitalvarys/jenkins-tutorials.git''*

*}*

*stage('Compile') {*

*def gradle\_home = tool 'gradle4'*

*sh "'${gradle\_home}/bin/gradle' clean compileJava test"*

*}*

*}*

* Declarative Pipeline encourages a declarative programming model, whereas Scripted Pipelines follow a more imperative programming model.
* Declarative type imposes limitations to the user with a more strict and pre-defined structure, which would be ideal for simpler continuous delivery pipelines.
* Scripted type has very few limitations that to with respect to structure and syntax that tend to be defined by Groovy,thus making it ideal for users with more complex requirements.

# How to clone git using Jenkins ?

Installing the git plugin, and giving the credentials to git tool, Creating a new job with Git repository URL in our project we have done By adding the git clone command in scripted pipeline we can clone

# Job Trigger Cron, SCM, Remote trigger?

In Cron we can set the cron expression to trigger the job automatically at particular time.

Using remote trigger we can call the jenkins job remotely via URL and it will start the job

Using SCM tool like Git. Jenkins will clone the code on the slave or local machine and it will start executing the pipeline

# How to add password in Jenkins ?

Manage Creditianls -> Global Credentials -> Secret Text

# Run Jobs Parallely vs how to run job parallely?

The no of executors will be usefull in running the jobs parallely.

Use Multijob plugin

# Parameterized build/ parameterised job ?

A build parameter allows us to pass data into our Jenkins jobs. Using build parameters, we can pass any data we want: git branch name, secret credentials, hostnames and ports, and so on. Any Jenkins job or pipeline can be parameterized.

# UpStream vs Downstream?

UpStream and Downstream defines the order of execution of jobs. after creating a job we can set build other job in Post build actions it will automatically create Upstream and downstream project

# Job Failure --> how to revert back ? how to revert a job and start again ?

There is a plugin called as Nangination which will automatically trigger the faily in a periodic set of time.

# How to run a job in Particular Node

using label

# Jenkins Groovy Script ?

Groovy language can be used as a scripting language for the Java platform. It is almost like a super version of Java which offers Java's enterprise capabilities. It also offers many productivity features like DSL support, closures, and dynamic typing.

# How to run a groovy script in Jenkins ?

Usage. To create Groovy-based project, add new free-style project and select "Execute Groovy script" in the Build section, select previously configured Groovy installation and then type your command, or specify your script file name. In the second case path taken is relatively from the project workspace directory

# How to secure Jenkins? / If you have 200 employees in your company, how you can assign Jenkins access to these employee how you can give permission in Jenkins ?

Role based Authorisation strategy, Manage roles -> global role, item role & assign roles.

# Is Jenkins a CI tool or both CI/CD ?

Jenkins is both CI/ CD tool

# How to install Jenkins with non root access in Linux ?

we can edit the etc/sysconfig/jenkins file to edit the user permission access JENKINS\_USER

or we can use sudo to install jenkins.

# What is DSL in Jenkins ?

Jenkins job DSL is a plugin that allows us to define jobs in programmatic form with minimal effort. DSL stands for Domain Specific Language.

A Job DSL script consists of API methods provided by the Job DSL plugin; you can use these API methods to configure different aspects of a job, such as its type (freestyle versus pipeline jobs), build triggers, build parameters, post-build actions.

# How to start a pipeline when a particular group of users commit ?

* Polling ignores commits with certain messages
* Polling ignores commits in certain paths
* Polling ignores commits from certain users

# What is Multi Branch Pipeline ?

Job -> MultiBranch Pipeline -> Add Branch Source -> Add Git Project URL -> Each Branch Needs a Jenkins File

# What is Matrix Project/ Multi-Configuration Project ?

Yaml Axis Plugin -> Yaml matrix Execution Strategy

# blue ocean plugin ?

UI based plugin for remodelling the jenkins UI

# Jenkins Project Pipeline ?

*node{*

*stage('SCM Checkout'){*

*git 'https://github.com/damodaranj/my-app.git'*

*}*

*stage('Compile-Package'){*

*def mvnHome = tool name: 'maven3', type: 'maven'*

*sh "${mvnHome}/bin/mvn clean package"*

*sh 'mv target/myweb\*.war target/newapp.war'*

*}*

*stage('SonarQube Analysis') {*

*def mvnHome = tool name: 'maven3', type: 'maven'*

*withSonarQubeEnv('sonar') {*

*sh "${mvnHome}/bin/mvn sonar:sonar"*

*}*

*}*

*stage('Build Docker Imager'){*

*sh 'docker build -t saidamo/myweb:0.0.2 .'*

*}*

*stage('Docker Image Push'){*

*withCredentials([string(credentialsId: 'dockerPass', variable: 'dockerPassword')]) {*

*sh "docker login -u saidamo -p ${dockerPassword}"*

*}*

*sh 'docker push saidamo/myweb:0.0.2'*

*}*

*stage('Nexus Image Push'){*

*sh "docker login -u admin -p admin123 65.0.181.193:8083"*

*sh "docker tag saidamo/myweb:0.0.2 65.0.181.193:8083/damo:1.0.0"*

*sh 'docker push 65.0.181.193:8083/damo:1.0.0'*

*}*

*stage('Remove Previous Container'){*

*try{*

*sh 'docker rm -f tomcattest'*

*}catch(error){*

*// do nothing if there is an exception*

*}*

*stage('Docker deployment'){*

*sh 'docker run -d -p 8090:8080 --name tomcattest saidamo/myweb:0.0.2'*

*}*

*}*

*}*

Ansible

# Ansible Commands

inventory files - /etc/ansible/hosts  
configuration file - /etc/ansible/ansible.cfg (manually create this file)

ansile all -i slaves.txt -m ping

ansible all -i slaves.txt -a “uname -a” (-a action)

ansible all -i slaves.txt -m ping (Grouping [ ])

ansible all -i slaves.txt -m yum -a “name=httpd state=present” -b  
ansible all -i slaves.txt -m service -a “name=httpd state=started” -b

ansible all -i slaves.txt -m copy -a “src=./slaves.txt dest=/tmp/slaves.txt” -b

ansible-playbook -i slaves.txt first.yaml –syntax-check  
ansible-playbook -i slaves.txt first.yaml

ansible-vault encrypt vault-pass.yaml

ansible-vault view vault-pass.yaml

ansible-vault view vault-pass.yaml --ask-vault-pass

ansible-galaxy init apache

# What is Configuration Management ?

Configuration management is a process for maintaining computer systems, servers, and software in a desired, consistent state.

# What are the other tools in market other than Ansible ?

Ansible, Bcfg2, CFEngine, Chef, Otter, Puppet, Quattor, SaltStack, Terraform, Pulumi and Vagrant.

# What is Ansible ?

* Ansible is a configuration management tool
* Created by Redhat
* Work on Master Slave Control Node Managed Node
* Slave ip in Inventory file
* ADHOC commands or playbooks to install, deploy or make configurations on slave or Managed Nodes

# How Ansible works ?

* Ansible works by connecting to your nodes via SSH and pushing out small programs, called modules to them.
* Modules are used to accomplish automation tasks in Ansible.
* These programs are written to be resource models of the desired state of the system.
* Ansible then executes these modules and removes them when finished.

# the different ways other than SSH by which Ansible can connect to remote hosts ?

default ssh, chroot, lxc, and jail containers

# Ansible is idempotency ?

Ansible tasks will only change the system if there is something to do.

repeteadly running the task will not change the result

example yum module wont install again and again

# How Ansible is different from Chef & Puppet ?

Ansible platform is written on python but It supports YAML command Scripts

Puppet platform is built with Ruby and it only supports Domain Specific Language (DSL) and Embedded Ruby (ERB).

Chef supports Ruby DSL with crucial prototype programming.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metrics** | **Chef** | **Puppet** | **Ansible** | **Saltstack** |
| Availability | ✔ | ✔ | ✔ | ✔ |
| Ease of Setup | Not very easy | Not very easy | Easy | Not very easy |
| Management | Not very easy | Not very easy | Easy | Easy |
| Scalability | Highly Scalable | Highly Scalable | Highly Scalable | Highly Scalable |
| Configuration language | DSL(Ruby) | DSL(PuppetDSL) | YAML(Python) | YAML(Python) |
| Interoperability | High | High | High | High |
| Pricing (upto 100 nodes) | $13700 | $11200-$19900 | $10,000 | $15,000(approx.) |

# Where is the Ansible Configuration file located ?

/etc/ansible/ansible.cfg -> config file

/ect/ansible/hosts -> inventory file

# Where ansible logs are stored ?

By default they are disable by giving the path in ansible.cfg file they will start working

/var/lib/ansible.log --> enable in ansible.cfg file

# what is inventory file & config file ?

inventory file contains the maganed node/ slave machine Ip address

The inventory file can list individual hosts or user-defined groups of hosts.

config file contains ansible related configurations

# What are the types of Inventories ?

Static & Dynamic

# What is Dynamic Inventory & when we use it & for what ?

Python code to fetch all IP’s of running instances and update it to the inventory file

# Difference between hosts & groups ?

Each IP address is a host

we can combime multi ip into a single group

<https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html>

# Group vars vs Host vars ?

group\_vars/

group1 # here we assign variables to particular groups

group2 # ""

host\_vars/

hostname1 # if systems need specific variables, put them here

hostname2 # ""

# How to give multiple groups in hosts

hosts: web:test:group

hosts: <ip>,<ip>,<ip>

# Playbook order of execution:

1. Variable loading
2. Fact gathering
3. The pre\_tasks execution
4. Handlers notified from the pre\_tasks execution
5. Roles execution
6. Tasks execution
7. Handlers notified from roles or tasks execution
8. The post\_tasks execution
9. Handlers notified from post\_tasks execution

# What is AD Hoc commands ?

Single task command

to perform quick functions.

using the modules we can run

# What is play & playbook ?

A Playbook is a list of plays. It can contain a single play, or many.

An Ansible playbook is an organized unit of scripts that defines work for a server configuration

A Play is a list of tasks and roles that should be run. A Play can also define vars that should be used for that play.

# What are the components of Ansible ?

* Inventory file(s)
* Group vars
* Host vars
* inclue
* import
* become
* Roles
* Tasks
* remote\_user
* handler
* notify
* listen
* block
* rescue
* always

# Pre-task & Post-task ?

Tasks that can be runned before a role execution is started

- hosts: www

remote\_user: vagrant

sudo: yes

pre\_tasks:

- shell: echo 'I":" Beginning to configure web server..'

roles:

- nginx

post\_tasks:

- shell: echo 'I":" Done configuring nginx web server...'

In the preceding example, we only printed some messages using the echo command. However, we can create tasks using any of the modules available with Ansible, which can run before, or after, applying roles.

# with item vs loop

The with\_ keywords rely on Lookup Plugins - even items is a lookup.

The loop keyword is equivalent to with\_list, and is the best choice for simple loops.

The loop keyword will not accept a string as input, see Ensuring list input for loop: query vs. lookup.

Generally speaking, any use of with\_\* covered in Migrating from with\_X to loop can be updated to use loop.

Be careful when changing with\_items to loop, as with\_items performed implicit single-level flattening. You may need to use flatten(1) with loop to match the exact outcome.

# What is handlers and Notify and listen?

when task is returns status changed it will notify handler and will execute

task not changed it returns status ok and notify will not work

we can use listen to listen handlers from different role and execute another handler

<https://medium.com/@george.shuklin/listen-feature-for-handlers-in-ansible-29183524c7e1>

# when condition ?

a condition when it is satisfied the play will run, example ansible\_os\_family == ‘Debain’

# What are ansible modules ?

pre defined libraries written in python

use modules to write tasks in playbook

750 + modules

* delegate to -> run only on a particular IP
* local action -> run on control node
* line in file -> edit single line in file
* block in file -> edit chunks of line
* template
* copy
* get\_url
* apt
* yum
* yum\_repo
* shell
* service
* unarchive

# template module ?

The template module also copies a file to a remote server, but it allows you to use Jinja2 to render a template to a file dynamically. This enables you to use variables, such as Ansible facts, to customize a particular file for a specific server. ... The file uses the . j2 suffix so that you know it is a Jinja2 template.

# What is Jinja 2 template ?

Jinja is a web template engine for the Python programming language.

save file in .j2 format and using template module we can edit particular data

# How to include custom modules in Ansible ?

~/.ansible/plugins/modules/

# Difference between COPY & FILE & template modules ?

**copy** takes a file from host,"as-is",and copies it to the remote destination. backup=yes

**template** takes a file (template) from host,changes variables based on Jinja2 filtering,and copies it to the remote destination.

**file** used to touch, create dir, remove file, set permissions state=directory, state=absent

# Difference between SHELL & COMMAND modules ?

**shell** – Execute shell commands on targets

It is almost exactly like the command module but runs the command through a shell (/bin/sh) on the remote node.

**command** – Execute commands on targets

The command(s) will not be processed through the shell, so variables like $HOME and operations like "<", ">", "|", ";" and "&" will not work.

# What is Setup module ? what it does ?

task is setup and

This module is automatically called by playbooks to gather useful variables about remote hosts that can be used in playbooks. It can also be executed directly by /usr/bin/ansible to check what variables are available to a host. Ansible provides many facts about the system, automatically.

# What is variable in Ansible ?

create vars: and specify ports:2000

can include in playbook task

also while running playbook can pass var as -e

# What are different types of variables scopes?

Ansible has 3 main scopes:

* Global: this is set by config, environment variables and the command line
* Play: each play and contained structures, vars entries, include\_vars, role defaults and vars.
* Host: variables directly associated to a host, like inventory, facts or registered task outputs

# How variable precedence takes place ?

Configuration settings (lower precedence)

Command-line options

Playbook keywords

Variables (Highest Precedence)

# what is include\_var vs vars\_files ?

**vars\_files:**

variables from those files are included in the playbook

Since it is used in the start of the play

*vars\_files are read when the play starts. include\_vars are read when the play reaches the task.*

# What is gather\_facts ?

This module is automatically called by playbooks to gather useful variables about remote hosts that can be used in playbooks.

# What is ansible Vault ?

Ansible Vault can encrypt the playbook if it contains username and password

AES256 encryption

ansible-vault create vault.yml

ansible-vault encrypt encrypt\_me.txt

ansible-vault view vault.yml

ansible-vault edit vault.yml

ansible-vault decrypt vault.yml

ansible-vault rekey encrypt\_me.txt

ansible-playbook --vault-id dev@prompt site.yml

ansible-playbook --vault-id dev@dev-password --vault-id prod@prompt site.yml

ansible-vault encrypt\_string <password\_source> '<string\_to\_encrypt>' --name '<string\_name\_of\_variable>'

# If a string is encrypted in a file with a password then how to pass the password using parameter while decrypting ?

ansible-playbook --ask-vault-pass site.yml

# If a file is encrypted using password & password is stored in a file how to pass the file to decrypt the file ?

ansible-playbook --vault-password-file /path/to/my/vault-password-file site.yml

# Can we recover ansible-vault password if lost ?

Unless you brute force the password there is no way to recover it. I suggest you use a password manager to store your password and share the password database within your organization.

# tags vs skip tags

to run a specific playbook we can use tags

skip tags will skip that particular task and executes everything else

# What is Roles ?

In Ansible, the role is the primary mechanism for breaking a playbook into multiple files. ... This simplifies writing complex playbooks, and it makes them easier to reuse.

# Differentiate Playbook vs role ?

Role is primarly used for Code Reusability

Roles contain folder like vars, files, handlers, tasks, meta

ansible-galaxy command is used to create a role

in playbook we will attach role

# How to create roles ?

ansible-galaxy init <role\_name>

# How to install a Role ?

ansible-galaxy install <role\_name>

# How to install multiple roles ?

ansible-galaxy install -r requirements.yml

requirements.yaml file

# what is import role vs include role/ Difference between include & import ?

import tasks will be parsed at the beginning when you run your playbook

include tasks will be parsed at the moment Ansible hits them

# Template folder vs file folder in roles ?

**Template folder**

This folder contains the template files used by the role to create the actual configuration files.

These are then deployed by the role to the remote hosts.

They are Jinja2 template engine scripts that enable loops and other features.

**File Folder**

This folder holds all extra files that are required to achieve the role task.

These files usually get dispatched to remote hosts as part of certain tasks.

They are usually static, and they do not contain any variables to change, be copied, extracted, or compressed to the remote host.

# Role dir Structure ?

**tasks**/main.yml - the main list of tasks that the role executes.

handlers/main.yml - handlers, which may be used within or outside this role.

**library**/my\_module.py - modules, which may be used within this role (see Embedding modules and plugins in roles for more information).

**defaults**/main.yml - default variables for the role (see Using Variables for more information). These variables have the lowest priority of any variables available, and can be easily overridden by any other variable, including inventory variables.

**vars**/main.yml - other variables for the role (see Using Variables for more information).

**files**/main.yml - files that the role deploys.

**templates**/main.yml - templates that the role deploys.

**meta**/main.yml - metadata for the role, including role dependencies.

# Defaults folder in roles

defaults/main.yml is a configuration file that you can use to define default values for variables used in your role.

It allows for a centralized management of the default values of the variable of the role.

Default values are always vulnerable because they change a lot depending on the needs and policies of the user.

Having this solution allows one file to change all the values

# Difference between default & vars directory in Roles ?

If roles/x/vars/main.yml exists, Ansible adds the variables in that file to the play.

If roles/x/defaults/main.yml exists, Ansible adds the variables in that file to the play.

The defaults directory is for defining the variable defaults. The variables in default have the lowest priority thus becoming easy to override. If definition of a variable is nowhere else, the variable in defaults/main.yml will be used.

# Can we disable automatic facts gathering in Ansible ?

gather\_facts = no, use explictly setup

# How error handling can be done in Ansible ?

using block rescue always

# What is changed\_when & failed\_when in Ansible ?

**changed\_when** = mark it as changed when a condition is met

**failed\_when** = mark it as failed when a condition is met, Ram and CPU checking

# How to ignore failed commands in Ansible ?

ignore\_errors: yes

# How to control the command failure in Ansible ?

changed\_when & failed\_when

# How to debug your playbook ?

using register and debug

# What is block/ rescue/ always in Ansible ?

we can group multiple tasks togethor in blocks

**block** -> try for errors

**rescue** -> work when block has error

**always** -> always gets executed

# What is register in Ansible?

Ansible register is a way to capture the output from task execution and store it in a variable.

cmd – The command that ran to generate the output.

**stdout** – The output of the command.

stderr – The error output of the command.

start – The date and time when the command began executing.

end – The date and time when the command finished executing.

delta – The time taken to run the command. This is the difference between the end and the start properties.

stdout\_lines – An array containing each output line of the command. Same as stdout, but stdout separates the lines using a newline (\n) characters instead of arrays.

stderr\_lines – An array containing each error output line of the command. Same as stderr, but stderr separates the lines using newlines (\n) characters instead of arrays.

# What is debug in Ansible

This module prints statements during execution and can be useful for debugging variables or expressions without necessarily halting the playbook. msg, var

# what is –check & --diff in ansible ? / What is Dry Run in Ansible & how to do that ?

**--check** Check mode is just a simulation test run

**--diff** shows the before and after changes

**--check --diff** shows what would have been the changes

# What is Privilege Escalation in Ansible ?

Ansible uses existing privilege escalation systems to execute tasks with root privileges or with another user’s permissions.

Because this feature allows you to ‘become’ another user, different from the user that logged into the machine (remote user), we call it become.

The become keyword leverages existing privilege escalation tools like sudo, su, pfexec, doas, pbrun, dzdo, ksu, runas, machinectl and others.

**become** set to yes to activate privilege escalation.

**become\_user** set to user with desired privileges — the user you become, NOT the user you login as. Does NOT imply become: yes, to allow it to be set at host level. Default value is root.

# What is lookup in Ansible playbook ?

You can use lookup plugins to access data from outside sources (files, databases, key/value stores, APIs, and other services) within your playbooks.

# How you can run your all tasks at once ?

async : 45 maximum time i give to the task to complete

poll : 0 doesnot wait for the task to get complete simultaneously runs all tasks

setting poll to 0 will create erros when using yum module as it has dependency

# loop -> syncronize module

A wrapper around rsync to make common tasks in your playbooks quick and easy.

# Installing Tomcat ?

- hosts: all

  remote\_user : ec2-user

  become: yes

  tasks:

  - name: install the latest version of apache using ansible

    yum:

      name: httpd

      state: present

  - name: starting apache

    service:

      name: httpd

      state: started

  - name: copying files

    copy:

      src: /index.html

      dest: /var/www/html/index.html

      mode: '0777'

## Creating EC2

- hosts: localhost

  remote\_user: ec2-user

  become: yes

  tasks:

    - name: Creating EC2

      ec2:

        key\_name: LaptopKey

        instance\_type: t2.micro

        image: ami-0bcf5425cdc1d8a85

        region: "ap-south-1"

        count: 1

        vpc\_subnet\_id: subnet-ea6a06a6

        assign\_public\_ip: yes

# Installing Jenkins

- hosts: jenkins

  remote\_user : ec2-user

  become: yes

  vars:

    ports: 9000

  tasks:

    - name: Installing jenkins

      yum\_repository:

        name: jenkins

        description: jenkins YUM repo

        baseurl: http://pkg.jenkins.io/redhat-stable

        gpgkey: http://pkg.jenkins.io/redhat-stable/jenkins.io.key

    - name: Installing java and Jenkins

      yum:

        name: "{{ item }}"

        state: present

      loop:

        - java

        - jenkins

    - name: going to change port no

      lineinfile:

        path: /etc/sysconfig/jenkins

        regexp: '^JENKINS\_PORT='

        line: JENKINS\_PORT={{ports}}

      notify: restart jenkins

    - name:

      service:

        name: jenkins

        state: started

  handlers:

  - name: restart jenkins

    service:

      name: jenkins

      state: restarted

# Using Ansible Loop

- hosts: all

  remote\_user : ec2-user

  become: yes

  tasks:

    - name: install the latest version of apache using ansible

      yum:

        name: "{{ item }}"

        state: present

      loop:

        - php

        - mysql

        - unzip

        - http\_present

# Installing Tomcat

- hosts: tomcat

  remote\_user : ec2-user

  become: yes

  vars:

    tomcat\_port: 9090

  tasks:

  - name: yum update

    yum:

      name: "\*"

      state: latest

  - name: installing java

    yum:

      name: java-1.8.0-openjdk

      state: present

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

  - name: installing java

    apt:

      name: java-1.8.0-openjdk

      state: present

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

  - name: Download tomcat

    get\_url:

      url: https://mirrors.estointernet.in/apache/tomcat/tomcat-9/v9.0.46/bin/apache-tomcat-9.0.46.tar.gz

      dest: /opt

      mode: '777'

  - name: Extract apache.tar

    unarchive:

      src: /opt/apache-tomcat-9.0.46.tar.gz

      dest: /opt

      remote\_src: yes

      mode: '777'

  - name: Template a file to /etc/file.conf

    template:

      src: /home/ec2-user/server.xml.j2

      dest: /opt/apache-tomcat-9.0.46/conf/server.xml

  - name: stop tomcat

    shell: nohup /opt/apache-tomcat-9.0.46/bin/shutdown.sh &

  - name: start tomcat

    shell: nohup /opt/apache-tomcat-9.0.46/bin/startup.sh &

  - name: copying files

    copy:

      src: /home/ec2-user/sample.war

      dest: /opt/apache-tomcat-9.0.46/webapps/sample.war

      mode: '0777'

Docker

# Docker commands

docker version

docker info

docker images

docker ps

docker ps -a

docker stop <container id/name>

docker start <container id/name>

docker pause <container id/name>

docker unpause <container id/name>

docker kill <container id/name>

docker exec -it <container id/name> <shell name>

docker rm <container id/name>

docker rm -f <container id/name>

docker rmi <image id/name>

docker container prune

docker image prune

docker logs <container id/name>

docker top <container id/name>

docker stats <container id/name>

docker inspect <container id/name>

docker port

docker scan hello-world

docker scan --file Dockerfile docker-scan:e2e

docker scan --file Dockerfile --exclude-base docker-scan:e2e

docker scan --json hello-world

docker scan --dependency-tree debian:buster

docker scan --severity=medium docker-scan:e2e

docker build -t <image name>:<tag> <Dockerfile path>

docker run -itd

--name <container name>

-p "<new port>:<port>" <image>

-e MYSQL\_ROOT\_PASSWORD=<your passwd> <image name>

--link <container name>:mysql -p "<new port>:<port>" <image name>

-v "<location in container>" <image name>

-v "<local path>:<container path>" <image>

--volumes-from <cont name you want to get data from> <image name>

--mount source=<volume name>,destination=<container location> <image name>

--network <your network> <image name>

docker volume ls

docker volume create <volume name>

docker inspect <volume name>

docker login

docker logout

docker tag <new image name> <dockerid>/<name>

docker pull <image name>

docker push <dockerid>/<name>

docker commit <container id> <new image name>

docker save -o <path/filename.tar> <new image name>

docker load -i filename.tar

docker cp <src-path> <container>:<dest-path>

docker cp <container>:<src-path> <local-dest-path>

docker network ls

docker network create --driver bridge <new network name>

docker-compose up -d

docker-compose ps

# Whats is docker ?

* Platform as a service products that use OS-level virtualization to deliver software in packages called containers.
* Containers are isolated from one another and bundle their own software, libraries and configuration files
* they can communicate with each other through well-defined channels.

# Docker Container and VM – What is the difference ?

* containers provide a way to virtualize an OS so that multiple workloads can run on a single OS instance. They are microservices
* With VMs, the hardware is being virtualized to run multiple OS instances. They are monolithic application

# what is virtualisation ?

virtualisation is the act of creating a virtual version of something, including virtual computer hardware platforms, storage devices, and computer network resources.

# what is micro services ?

Microservices - also known as the microservice architecture - is an architectural style that structures an application as a collection of services that are. Highly maintainable and testable. Loosely coupled. Independently deployable. Organized around business capabilities.

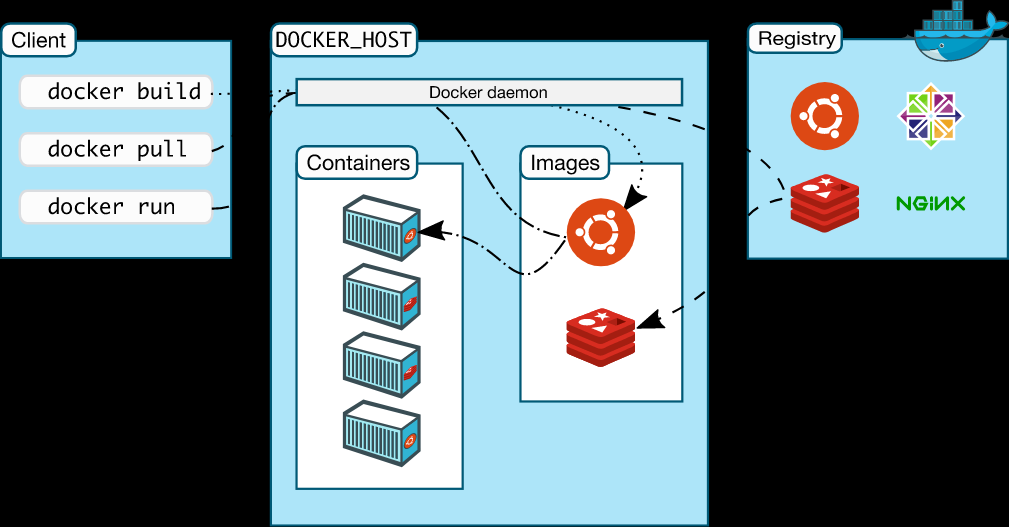
# Why should i Go Docker in my project ?

Docker containers are process-isolated and don't require a hardware hypervisor.

This means Docker containers are much smaller and require far fewer resources than a VM. Docker is fast. Very fast.

To convert monolith application to microservices we can Docker

# Docker Architecture ?



# What are registries ?

* A registry is a storage holding Docker images
* Docker images are available in different tagged versions in registry
* We interact with a registry by using docker push and pull commands.
* It is part of Docker Architecture

# What is Docker sock

docker group in linux that is being created by most docker installations. It’s purpose is to allow non-root users access to docker. You simply add a user to the docker-group using

gpasswd -a USER docker

and the USER can run docker commands freely. This works because the docker group is set to be the owner of the /var/run/docker.sock socket used for the communication with the daemon.

Be careful - this is very powerful and gives the user indirect access to the whole host machine since the docker daemon still runs as root!

# What is Docker File ?

* Dockerfile is a text file that contains a list of commands (instructions), which describes how a Docker image is built.
* The commands tells Docker to build the image by following the content (instructions) inside the Dockerfile.

# How to build DockerFile with different Name ?

$ docker build -f dockerfiles/Dockerfile.debug -t myapp\_debug .

$ docker build -f dockerfiles/Dockerfile.prod -t myapp\_prod .

# What is Docker Image ?

* A Docker image is a file used to execute code in a Docker container.
* Docker images act as a set of instructions to build a Docker container, like a template.
* An image is comparable to a AMI in AWS virtual machine (VM) environments.

# What is Docker Container ?

* Docker images are used to create a container.
* Containers are isolated from one another and bundle their own software, libraries and configuration files
* IT is a running application.
* They are mutable

# How to reduce the Docker image size ?

* Utilize the Multi-Stage Builds Feature in **Docker**.
* Use Small Base Image Like Apline Or Busybox
* Minimise Layers Used In Dockerfile
* Use Dockerignore Similar To Gitignore
* Don't Install Debug Tools Like Vim/Curl
* Use Docker Squash Tool or Docker compress while building the image
* Avoid Adding Unnecessary Layers to **Reduce Docker** Image **Size**.

# What is Docker Squash ?

* docker image build --squash
* will merge multiple layers in Docker file into a single layer/ Squash newly built layers into a single new layer

# What is Docker compress ?

* docker image build --compress
* Compress the build context using gzip

# What are image layers ?

A Docker image consists of several layers. Each layer corresponds to certain instructions in your Dockerfile . The following instructions create a layer: RUN , COPY , ADD . The other instructions will create intermediate layers and do not influence the size of your image.

# What is multi stage builds in docker ?

multi-stage builds, you use multiple FROM statements in your Dockerfile.

Each FROM instruction can use a different base, and each of them begins a new stage of the build.

You can selectively copy artifacts from one stage to another, leaving behind everything you don't want in the final image.

# syntax=docker/dockerfile:1

FROM golang:1.16

WORKDIR /go/src/github.com/alexellis/href-counter/

RUN go get -d -v golang.org/x/net/html

COPY app.go .

RUN CGO\_ENABLED=0 GOOS=linux go build -a -installsuffix cgo -o app .

FROM alpine:latest

RUN apk --no-cache add ca-certificates

WORKDIR /root/

COPY --from=0 /go/src/github.com/alexellis/href-counter/app .

CMD ["./app"]

# Difference between DockerFile vs DockerCompose ?

* Dockerfile is a text file that contains a list of commands (instructions), which describes how a Docker image is built.
* The commands tells Docker to build the image by following the content (instructions) inside the Dockerfile. **docker build -t <imagename>**
* Compose is a tool for defining and running multi-container Docker applications.
* With Compose, you use a YAML file to configure your application's services. with a single command, you create and start all the services from your configuration. **docker-compose up**

# Docker File Syntax ?

FROM

LABEL

WORKDIR

ENV

ARGS

ADD

COPY

RUN

ENTRY POINT

CMD

EXPOSE

# Difference between ADD vs Copy ?’

* Copy can send files from Local to Container
* Where are ADD can get data from url and can Extract tar files
* COPY is Same as 'ADD', but without the tar and remote URL handling.

# CMD vs EntryPoint What is the difference ?

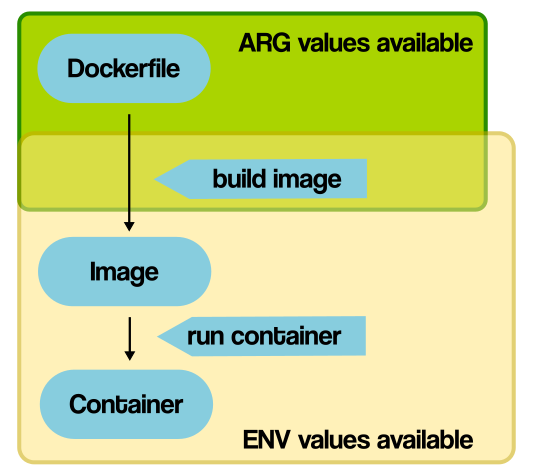
* ENTRYPOINT command and parameters will not be overwritten from command line.
* CMD sets default command for parameters, which can be overwritten from command line when docker container runs.
* All command line arguments will be added after ENTRYPOINT parameters.

# Difference between ENV and ARG in Docker file ?

ENV is for future running containers.ARG for building your Docker image.

You can’t change ENV directly during the build!

ARG values are not available after the image is built.



# What is Docker Build Context

* When we try to build a docker image, we need to send the files to the docker server.
* These files are basically the build context.
* These files are archived into a .tar file by the docker client and then they are uploaded to the docker server.

# What is .dockerignore file ?

* Dockerignore files allows you to mention a list of files and/or directories which you might want to ignore while building the image.
* This would definitely reduce the size of the image and also help to speed up the docker build process.

# To copy a file from the local file system to a container

docker cp <src-path> <container>:<dest-path>

kubectl cp <src-path> <your-pod-name>:<dest-path>

# To copy a file from the container to the local file system

docker cp <container>:<src-path> <local-dest-path>

kubectl cp <your-pod-name>:<src-path> <local-dest-path>

# Docker file we give expose and we give -p while running docker what is the difference ?

* We expose ports using the EXPOSE keyword in the Dockerfile or the --expose flag to docker run.
* Exposing ports is a way of documenting which ports are used, but does not actually map or open any ports.
* We publish ports using the --publish or --publish-all flag to docker run .

# Difference between docker commands: up, run & start ?

* docker compose up is used to run a docker-compose.yaml file
* docker run is used to create a container
* docker start is used to start a stopped container

# Docker kill, docker pause ,Docker stop

* docker kill subcommand kills one or more containers (SIGKILL)
* docker stop will stop the container which can be resumed later (SIGTERM).
* docker stop release the memory used after the container is stopped.
* docker pause would still keep memory portion while the container is paused> This memory is used when the container is resumed.

# Why should i go Docker Volume ?

* Docker volumes are file systems mounted on Docker containers to preserve data generated by the running container.
* The volumes are stored on the host, independent of the container life cycle.
* This allows users to back up data and share file systems between containers easily.

# How to share docker data in multiple volume ?

* -v "<local path>:<container path>" <image>
* --volumes-from <cont name you want to get data from> <image name>
* --mount source=<volume name>,destination=<container location> <image name>

# Docker attach vs Docker exec?

* docker attach will leave the docker in exited status after coming out from the container
* docker exec will let the container run in background even after coming out from the container

## Docker detach ?

you can detach from a container and leave it running using the **CTRL-p CTRL-q** key sequence.

# how to push Docker image in Docker HUB ?

docker login

docker tag <new image name> <dockerid>/<name>

docker push <dockerid>/<name>

# How to add volumes to running container ?

commit the container into a new image

add the volume using docker run -v "<local path>:<container path>" <image>

# Can I limit the mem/ CPU utilization for a docker in my machine ?

docker run -it --memory=”[memory\_limit]” [docker\_image]

docker run -it --cpus=”1.0” ubuntu

# How to Link Docker Containers ?

--link is old legacy way of linking containers

now we can use docker networking to link between containers. Create a new network and create containers on them they can communicate between each other

# Docker link vs depends on ?

depends\_on expresses start order (and implicitly image pulling order), which was a good side effect of links.

--link is also deprecated and should be replaced by a custom network.

# How to communicate between 2 containers in a different network

* Solution a) Connect one container into the other network overlay (this may not meet the constraint you have).
* Solution b) Create a third network and plug both containers into this network

# what is difference between -v and mount

-v is used to mount a local dir to a dir inside container they will synced with each other

if we have a seprate volume created by docker volume command we can use mount command to mount the volume

# how you will confirm there is no security vulnerability in public image ?

create new snyk account which can scan for vulnerabilites

docker scan hello-world

docker scan --file Dockerfile docker-scan:e2e

docker scan --file Dockerfile --exclude-base docker-scan:e2e

docker scan --json hello-world

docker scan --dependency-tree debian:buster

docker scan --severity=medium docker-scan:e2e

# how to monitor docker container ?

Prometheus

# how will you handle or store your private images in your env

Private artifactory

Jfrog (needs lisence) & sonatype Nexus

Create new instance yum install java

Download Nexus and ./nexus start -> give initial admin password and create new password

Create respository -> Docker Hosted -> give name, http 8083, give Docker v1 API access

install docker -> create /etc/docker/deamon.json -> insecure registry give publicip:8083

also do the same in Docker machine where jenkins is running the docker and dp the same /etc/docker/deamon.json

# what is the difference between bridge network and custom bridge network

* User-defined bridges provide automatic DNS resolution between containers.
* User-defined bridges provide better isolation.
* Containers can be attached and detached from user-defined networks on the fly.
* Each user-defined network creates a configurable bridge.
* Linked containers on the default bridge network share environment variables.

# Where the image layes can be found in which directory ?

The /var/lib/docker/aufs directory points to three other directories: diff , layers and mnt .

Image layers and their contents are stored in the diff directory.

# What are the 3 different directories in /var/lib/docker/aufs ?

diff/ : Differences introduced in the writable container layer, such as new or modified files.

layers/ : Metadata about the writable container layer's parent layers.

mnt/ : A mount point for each running container's unified filesystem, exactly as it appears from within the container.

# What is overlayFS ?

* OverlayFS is a modern union filesystem that is similar to AUFS, but faster and with a simpler implementation.
* Docker provides two storage drivers for OverlayFS: the original overlay , and the newer and more stable overlay2.

# How can we check the content of each layer ?

/var/lib/docker/aufs/diff -

/var/lib/docker/aufs/layers -

# How to check the layers stacked with image ?

/var/lib/docker/aufs/layers -

# What is Union Mount & AUFS ?

AUFS is a union filesystem, which means that it layers multiple directories on a single Linux host and presents them as a single directory. These directories are called branches in AUFS terminology, and layers in Docker terminology. The unification process is referred to as a union mount.

# Why use Union mount system for Docker ?

avoid duplicating a complete set of files each time you run an image as a new container

isolate changes to a container filesystem in its own layer, allowing for that same container to be restarted from a known content (since the layer with the changes will have been dismissed when the container is removed)

If you didn't have UnionFS, an 200MB image run 5 times as 5 separates containers would mean 1GB of disk space.

# How to create a bridge in container ?

docker network create --driver bridge <new network name>

docker run -itd --network <your network> <image name>

# How a container gets an internal IP ?

By default, the container is assigned an IP address for every Docker network it connects to. And each network is created with a default subnet mask, using it as a pool later on to give away the IP addresses. Usually Docker uses the default 172.17. 0.0/16 subnet for container networking.

# How kernel isolates to run the container and how resources managed by the kernel ?

Docker makes use of kernel **namespaces** to provide the isolated workspace called the container . When you run a container, Docker creates a set of namespaces for that container. These namespaces provide a layer of isolation.

# What is namespace and cgroups ?

cgroups limits the resources which a process or set of processes can use these resources could be CPU,Memory,Network I/O or access to filesystem

namespace restrict the visibility of group of processes to the rest of the system.

# What is docker-compose and docker-swarm ?

Docker Swarm is used to scale your web app across one or more servers

Where as Docker-compose will simply run your web app on a single Docker host.

# Can we run more than one process in a container ?

It's ok to have multiple processes, but to get the most benefit out of Docker, avoid one container being responsible for multiple aspects of your overall application. You can connect multiple containers using user-defined networks and shared volumes.

Terraform

# Commands

**terraform fmt** *#format as per HCL*

**terraform validate** *#check for syntax*

**terraform init** *#initialize directory, pull down providers*

**terraform init -get-plugins=false** *#initialize directory, do not download plugins*

**terraform init -verify-plugins=false** #initialize directory, do not verify plugins for Hashicorp signature

**terraform apply --auto-approve** #apply changes without being prompted to enter “yes”

**terraform destroy --auto-approve** #destroy/cleanup deployment without being prompted for “yes”

**terraform plan -out plan.out** #output the deployment plan to plan.out

**terraform apply plan.out** #use the plan.out plan file to deploy infrastructure

**terraform plan -destroy** #outputs a destroy plan

**terraform apply -target=aws\_instance.my\_ec2** #only apply/deploy changes to the targeted resource

**terraform apply -var my\_region\_variable=us-east-**1 #pass a variable via command-line while applying a configuration

**terraform apply -lock=true** #lock the state file so it can’t be modified by any other Terraform apply or modification action(possible only where backend allows locking)

**terraform apply refresh=false** # do not reconcile state file with real-world resources(helpful with large complex deployments for saving deployment time)

**terraform apply --parallelism=5** #number of simultaneous resource operations

**terraform refresh** #reconcile the state in Terraform state file with real-world resources

**terraform providers** #get information about providers used in current configuration

**terraform workspace new mynewworkspace** #create a new workspace

**terraform workspace select default** #change to the selected workspace

**terraform workspace list** #list out all workspaces

**terraform state show aws\_instance.my\_ec2** #show details stored in Terraform state for the resource

**terraform state pull > terraform.tfstate** #download and output terraform state to a file

**terraform state mv aws\_iam\_role.my\_ssm\_role module.custom\_module** #move a resource tracked via state to different module

**terraform state replace-provider hashicorp/aws registry.custom.com/aws** #replace an existing provider with another

**terraform state list** #list out all the resources tracked via the current state file

**terraform state rm aws\_instance.myinstace** #unmanage a resource, delete it from Terraform state file

**terraform import aws\_instance.new\_ec2\_instance i-abcd1234** #import EC2 instance with id i-abcd1234 into the Terraform resource named “new\_ec2\_instance” of type “aws\_instance”

**terraform import 'aws\_instance.new\_ec2\_instance[0]' i-abcd1234** #same as above, imports a real-world resource into an instance of Terraform resource

**terraform output** #list all outputs as stated in code

**terraform output instance\_public\_ip** # list out a specific declared output

**terraform output -json** #list all outputs in JSON format

# Terraform Best Practices

Run terraform command with var-file.

Enable version control on terraform state files bucket.

Manage S3 backend for tfstate files.

Manage multiple Terraform modules and environments easily with Terragrunt.

Retrieve state meta data from a remote backend.

Turn on debug when you need do troubleshooting.

# Terraform order of Resource Creation ?

Terraform uses this dependency information to determine the correct order in which to create the different resources. To do so, it creates a dependency graph of all of the resources defined by the configuration

# What is terraform ?

Terraform is an open-source infrastructure as code software It is used to create infrastructure in various Cloud service providers. Terraform is written in HCL (HashiCorp Configuration Language.

# What version are you using ?

We are using terraform version v 0.13

# What is terraform locals ?

Terraform locals are named values that you can refer to in your configuration. You can use local values to simplify your Terraform configuration and avoid repetition. Local values (locals) can also help you write more readable configuration by using meaningful names rather than hard-coding values.

locals {

service\_name = "forum"

owner = "Community Team"

}

# Where will you store your Statefile in terraform ?

**S3**

# terraform how to delete particular resource

taint

# Terraform provider vs provisioner ?

Provider development teams often prioritize features based on interest, so opening an issue is a way to record your interest in the feature.

Provisioners are used to execute scripts on a local or remote machine as part of resource creation or destruction. Provisioners can be used to bootstrap a resource, cleanup before destroy, run configuration management, etc.

## Provisioner

Provisioners can be used to model specific actions on the local machine or on a remote machine in order to prepare servers or other infrastructure objects for service.

**file** The file provisioner is used to copy files or directories from the machine executing Terraform to the newly created resource. The file provisioner supports both ssh and winrm

**local-exec** The local-exec provisioner invokes a local executable after a resource is created.

**remote-exec** The remote-exec provisioner invokes a script on a remote resource after it is created.

# How to update changes in state file

terraform refresh -> update changes happened in the infrastructure

terraform import -> Add infrastructure into the state file so Terraform can manage it

# Depends on in Terraform ?

Explicit & Implicit.

Using depends on on the resource is Explicit

Adding Nat to Route table is Implicit

In both cases the 1st resource will be created after the next

# Terraform state file is deleted if again terraform apply ?

will re-create the entire infra again

# backend.tf in terraform ?

Each Terraform configuration can specify a backend, which defines where and how operations are performed, where state snapshots are stored, etc.

local backend

remote backend

# terraform fmt

The terraform fmt command is used to rewrite Terraform configuration files to a canonical format and style. This command applies a subset of the Terraform language style conventions, along with other minor adjustments for readability.

# terraform modules --> Reuse code for different modules

Root module -> Out folder which contains tf files

Chile module -> inside folder which contains even more tf files

Organize configuration

Encapsulate configuration

Re-use configuration

Provide consistency and ensure best practices

Modules can either be loaded from the local filesystem, or a remote source. Terraform supports a variety of remote sources, including the Terraform Registry, most version control systems, HTTP URLs, and Terraform Cloud or Terraform Enterprise private module registries.

# What is Terraform statefile ?

Terraform must store state about your managed infrastructure and configuration. This state is used by Terraform to map real world resources to your configuration, keep track of metadata, and to improve performance for large infrastructures.

# terraform datasource/ workspace/ resource

**Terraform Data sources** - Data sources allow data to be fetched or computed for use elsewhere in Terraform configuration.

**Terraform Workspace** - Workspaces are how Terraform Cloud organizes infrastructure. If a large organisation creates a infrastructure. workspaces are used to restrict access.

**Terraform Resource** - Resources are the most important element in the Terraform language. Each resource block describes one or more infrastructure objects, such as virtual networks, compute instances,

# Terraform count ?

count is a meta-argument defined by the Terraform language. It can be used with modules and with every resource type. The count meta-argument accepts a whole number, and creates that many instances of the resource or module.

# terraform null resource

triggers (Map of String) A map of arbitrary strings that, when changed, will force the null resource to be replaced, re-running any associated provisioners.

# how to log statefile

Terraform has detailed logs which can be enabled by setting the **TF\_LOG** environment variable to any value. This will cause detailed logs to appear on stderr.

# Terragrunt

Terragrunt is a thin wrapper that provides extra tools for keeping your configurations DRY, working with multiple Terraform modules, and managing remote state.

APPD

Apm Tools: Appdynamics, Newrelic, Dynatrace, Eg Innovations

1. Mttr: Mean Time To Resolution – RCA (Root Cause Analysis)
2. Risk Of Outage (Maintenance) – Functionality – Upgrade/ Changes
3. Analysis – Data & Metrics

Performance: Load/Request, Cpu/Memory, Network, Database – (Data/ Metrics/Values)

# Components of AppD

## AGENT

Launguages we support - Java, .Net, Php, NodeJS, C++, Python, Go

Need to Download PACKAGE

Types of Agent:

1. APP AGENT – Application Details/ Request/ Load/ Response Time
2. Machine Agent – Infra Details/ CPU/ Memory/ Network I/O
3. Webserver Agent – Static Webpage
4. DataBase Agent – Oracle DB/ MYSQL DB
5. End User Agent

Installed path LINUX:

/opt/appdynamics/version(20.1)/app\_agent/

/opt/appdynamics/version(20.1)/machine\_agent/

## Controller – Central Managent

Agent (EC2) – (APP/Machine) --> Firewall/port# (Httpd 8090/ Http 8181) --> Controller (User Interface)

Controller – 100 Application – 10k Applications

Types of Controller:

SAAS – Appdynamics Vendor – Maintain – URL – Rent – 100- Application (Http 80/ Https 443)

ON PREMISES – Controller Built – Maintain – 1000+ Applications (Http 8090/ Https 8181)

Event Services --> Storage/ Processing Unit

## CONTROLLER.INFO.XML

/opt/appdynamics/version(20.1)/app\_agent/conf/controller.info.xml

IP address – port#

Application Name

Server (Tier Name)

Account Access Key

Account Name

## Details given to us after admin instllation

Url

Username

password

TimeFrame – Last 30 min

Group of Nodes – Tier (CSK)

EC2 Instance – Node (11 Nodes)

Youtube -> Refresh -> Calls/Load

Baseline

Request Time (1000ms – 1 second)

Error – Webbased Error – Application Error – Error/Exceptions

Less than 1000ms – Normal

Above 1000ms – Slow

Above 2000ms – Very Slow

infinity - Stall

no page - Error

Troubleshoot – Errors – Slow Calls & Errors – Export Grid Data

Tiers & Nodes – Servers – Shows all Metrics

Alert & Respond

1. Health Rule – Stability of the Application – 24\*7
2. Action – depending on the health rule take action – Email, Mobile, Ticket, p4, p5 Jira
3. Policy – Action + Health

Create Health Rule – User data from last 30 min – Wait time after Validation – Affected Entities – Conditions – Threshold 2000 Errors – 30 min – Crash/Down

Health Conditions

1. Critical – Red – 1500 Errors – Email – 500 Errors – Reaction Time
2. Warning – Yellow – 1000 Errors
3. Normal – Green – 500 Errors
4. Unknown – Grey

Action – Set Actions

## Dashboards

Predefiend Dashboard

Splunk Version

▪ Splunk Web Port: 8000

▪ Splunk Management Port: 8089

▪ Splunk Network port: 514

▪ Splunk Index Replication Port: 8080

▪ Splunk Indexing Port: 9997

▪ KV store: 8191

# Logrotate ?

logrotate is designed to ease administration of systems that generate large numbers of log files. It allows automatic rotation, compression, removal, and mailing of log files. Each log file may be handled daily, weekly, monthly, or when it grows too large. Normally, logrotate is run as a daily cron job.

# Monitoring tools

1. Application Monitoring
2. Server Monitoring
3. Network Monitoring
4. Log Monitoring

Log = Entry/Event

Application – EC2 (Server) – Business Impact

Amazon/ Flipkart – Login/ Logout

## Types of Logs

Application Log – Login/ Success/Password

Server Log – Login/ Command

Network Log

Retention Period : EC2 Log – Time – 30 Days/90 Days – Zip/Compressed

# Log Path

## Linux

/opt/<app>/<log\_folder>/application.log

/opt/<app>/<log\_folder>/application\_13062021.log

## Windows

C:/Application/Logs/Logs

# Sample Log using Keywords

INFO

DEBUG

ERROR

## Splunk: Enterprise Product – 60 Days Trial

# Installation

## Master

Dedicated Server

User Interface

**Splunk Enterprise Product** – 500mb/day (Free tier)

Download Enterprise Product using wget

/opt/splunk/bin -> ./splunk start

<ip address>:8000

## Slave

Ec2 Machine

Application Hosted

**Splunk Forwarder**

1. Universal Forwarder
2. Heavy weight Forwarder

Download Splunk Forwarder using wget

/opt/splunk/bin -> ./splunk start –accept-license

## Components of Splunk

1. Search Head – Entering our Queries
   1. Presets
   2. Relative
   3. Real-time
   4. Date Range
   5. Date & Time Range
2. Indexers – Index of Splunk
3. Forwarders – Slave – Log – Send to Master

# Configuration

## Master – UP & Running

Time & Location

Master Dedicated Listener Port no – 9997

/opt/splunk/bin => ./splunk enable listen 997

## Slave – Up & Running

Need Master IP Address & Port No

./splunk add forward-server <master-ip>:9997

Place demo logs in Slave machine using Win-scp

./splunk add monitor /var/log/syslig -index main

## Any path can be logs folder

/var/log/

/var/log/\* -> Wildcard entry

/opt/Application/logs

index = “main”, Host = “ip address”, Source = “folder location”, Sourcetype = “file name”

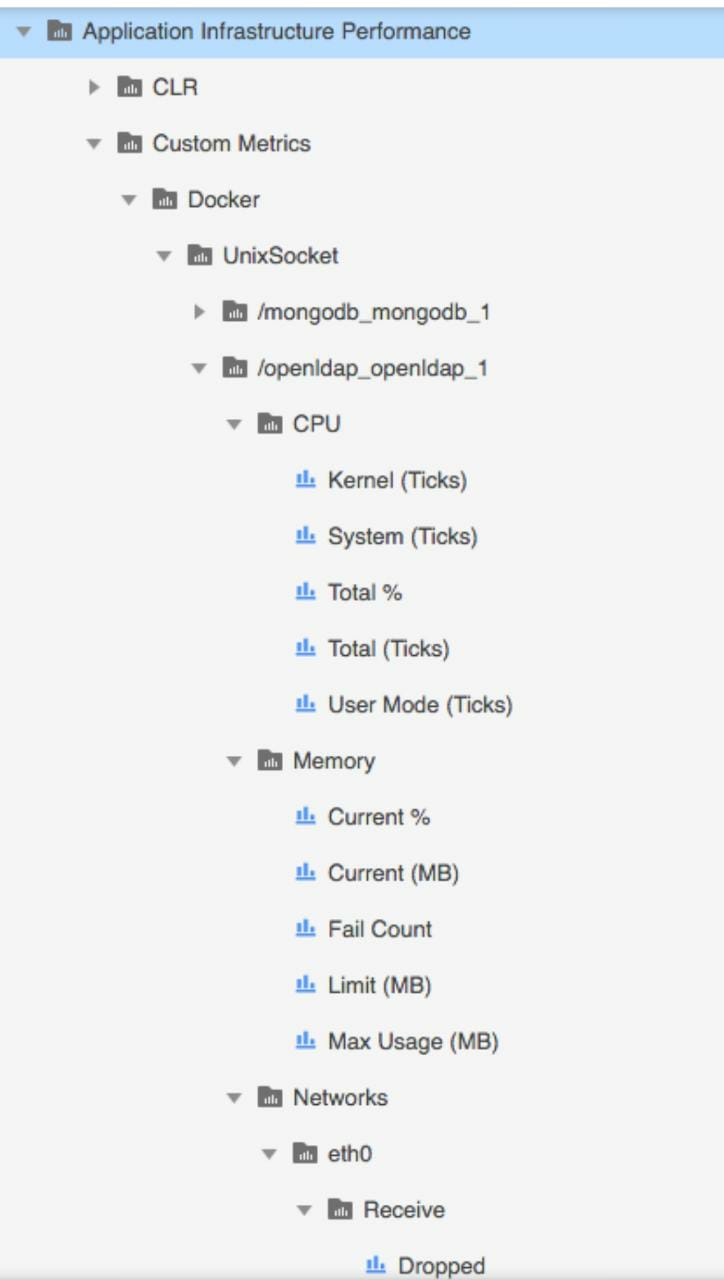
# Search using keywords

# Types of Dashboard

1. Dynamic form-based dashboards.
2. Static real-time dashboards.
3. Dashboards as scheduled reports.

## Types of panel

"A Splunk index is a repository for Splunk data." Data that has not been previously added to Splunk is referred to as raw data. When the data is added to Splunk, it indexes the data (uses the data to update its indexes), creating event data. Individual units of this data are called events.



Apache

## Where will you edit the tomcat port no

conf -> server.xml file -> connector port

## Where will you get the apache httpd error files ?

access\_log & error.log file we can get the tomcat access.

# AppD

## What is APM ?

Application performance management (APM) also known as Application Performance Monitoring allowing monitoring tools for IT Support and Developers to monitor their backend application architecture to resolve performance issues and bottlenecks in a timely manner.

## What is AppDynamics ?

The AppDynamics APM Platform enables you to monitor and manage your entire application, to view, analyze the request/traffic through your network, backend databases and application servers. AppDynamics APM gives you a single view across your application landscape

## What are the Languages worked on AppDynamics ?

AppDynamics works with popular programming languages such as Java, .NET, Node.js, PHP, Python, C/C++

## HOW APPDYNAMICS WORK?

* AppDynamics app agents need to connect to an AppDynamics Controller to retrieve configuration data and send back information about the monitored environment
* This page provides general information about the connections between the agents and Controller
* The connection between the agent and Controller is a ONE-WAY connection initiated by the agent

## What are AGENTS ?

* Agents are plug-ins or extensions that monitor the performance of your application code, runtime, and behavior.
* The Controller receives metrics from Agents and sends them instructions. Once deployed, Agents immediately monitor every line of code.
* This allows AppDynamics to trace every transaction from start to finish.

## What is CONTROLLER?

The Controller helps Monitor, Troubleshoot and Analyze your entire application from backend infrastructure to the end user—in one simple interface. Agents collect data from a monitored environment and send it to the Controller. The Controller UI is where you can view, understand, and analyze the data

## What are the Port numbers for a Controller ?

* For a SaaS Controller, use 80 for HTTP or 443 for HTTPS
* For an on-premises Controller, use 8090 for HTTP or 8181 for HTTPS

## What is the CPU Consumption for AppDynamics?

0% to 2% additional CPU consumption

## What is the Heap Size (-Xmx) to accommodate the AppD Agent ?

Maximum heap size (-Xmx): 100 MB in addition to the amount required by the application

## What is the PermGen space to accommodate the AppD Agent ?

Maximum PermGen (permanent generation) heap size (-XX:MaxPermSize): 20 MB in addition to the amount required by the application.

## What are the Dashboard Layouts available ?

* GRID
* ABSOLUTE

## How the connection between Agent and Controller happens ?

The connection between the agent and Controller is a one-way connection initiated by the agent.

## What is a Business Transaction?

* A Business Transaction is made up of all the required services within your environment that are called upon to fulfill and deliver a response to a user-initiated request.
* These are typically things like login, search, checkout, etc
* A business transaction represents the data processing flow for a request, most often a user request
* AppDynamics automatically discovers Business Transactions

## What is the Default limit for Business Transaction in AppD?

* Business Application Limits: Each application is limited to 200 registered business transactions.
* App Server Agent Limits: Each agent is limited to 50 registered business transactions.

## How will you configure Agents for SSL ?

Set these SSL-related properties:

* Set controller-ssl-enabled to true.
* Set the controller-port to the correct value for either on-premises or SaaS Controller.

## What is the important pre-requirement to install the Agent for an JVM ?

* All files should be readable by the user under which the JVM runs. The user must have write privileges to the conf and logs directories in the Java Agent home.
* To achieve this is to install the agent as the same user that owns the JVM or as an administrator on the host machine

## What are the details seen in a simple CONTROLLER.INFO.XML file ?

The following shows a controller-info.xml file with sample configuration values:

<controller-info>

    <controller-host>192.168.1.20</controller-host>

    <controller-port>8090</controller-port>

    <application-name>ACMEOnline</application-name>

    <tier-name>InventoryTier</tier-name>

    <node-name>Inventory1</node-node>

</controller-info>

## How will you Verify Java Agent Installation?

“Started AppDynamics Java Agent Successfully” should be seen after installation in agent log in <agent\_home>/logs.

## What are the ACTIONS can be executed in Alert & Respond section ?

Actions are

* Send an Email
* Take Thread Dump
* Create or Update a JIRA Ticket
* Make an HTTP Request
* Run a Script or Executable on problematic nodes

## What are the Report Type options available for AppD?

* Application Health
* Custom Dashboard
* Controller Audit
* Home Screen
* All Application Summary

## What is a Transaction Snapshots?

A transaction snapshot gives you a cross-tier view of the processing flow for a single invocation of a transaction.

## What is Tier Response Time?

The total response time for the call as measured at the calling tier. This includes the processing time on the called tier as well as on any tiers and backends it calls in turn.

## What is Call Drill Downs?

A call drill down contains details for that business transaction execution on a particular tier. It takes you to the code-level information for the transaction.

## What is a Call Graph?

A call graph in a transaction snapshot shows you business transaction processing information on a particular tier that participated on the business transaction. A call graph lists the methods in a call stack and provides information about each call

## How do you enable Server Visibility?

Enable <sim-enabled> to TRUE in controller-info.xml under Machine Agent folder

## How will you enable .NET Compatibility Mode on the agent ?

Set the following options in <machine\_agent\_home>/conf/controller-info.xml:

<dotnet-compatibility-mode>true</dotnet-compatibility-mode>

## What are the Two types of AppDynamics Browser Monitoring?

* Browser Real User Monitoring (Browser RUM)
* Browser Synthetic Monitoring

## What is EUM in AppD ?

AppDynamics End User Monitoring (EUM) gives you visibility on the performance of your application from the viewpoint of the end user.

## What are the Parameters analyzed using EUM in AppD?

* What your slowest web/Ajax requests are, and where the problem may lie.
* What your slowest mobile and IoT network requests are, and where the problem may lie.
* How application server performance impacts the performance of your web and mobile traffic
* How performance varies by location, Client type, device, browser and browser version, application and application version, operating system version, device

## What are the Two files JavaScript Agent consists?

The JavaScript Agent consists of two files: adrum.js and adrum-ext.js.

## What is Browser Synthetic Monitoring?

Browser Synthetic Monitoring uses geographically distributed Synthetic Agents to continuously test key user workflows in your application. This allows you to monitor the correctness and performance of multi-step flows independently of the user-generated load.

## What are the two ways of Synthetic Monitoring?

Two types are Scheduled Jobs & On-Demand Snapshots

# Splunk

# AWS Project session

SSL Certificate using AWS Certificate Manager

User Services - EC2,S3,ELB,Route53,RDS,IAM,ACM,CloudWatch

Certificate Authority

ACM -> Get Certificate -> DNS Validation

Route 53 -> Create Hosted Zone -> Map the Name Server

ACM -> Get Certificate -> DNS Validation -> Create Record Set FROM ACM

RDS -> Create two Databases

Create Iam Role

Create Ec2 Machine -> Use Script

Create LoadBalancer -> Add EC2 instances

add wp-config.php in /var/www/html in prod machine

add wp-config.php in /var/www/html in dev machine

Add record set for greens cloud and dr.greens cloud

sync using s3 by cron tab

adding new parameters for the CloudWatch Services

Scratch

Maintence

Deployment

Work Order

Change Req

Patchiching activity

Support – Defects / Incidents

# DEVOPS Project session

*// Added for demo*

*node{*

*stage('SCM Checkout'){*

*git 'https://github.com/javahometech/my-app'*

*}*

*stage('Compile-Package'){*

*// Get maven home path*

*def mvnHome = tool name: 'maven-3', type: 'maven'*

*sh "${mvnHome}/bin/mvn package"*

*}*

*stage('SonarQube Analysis') {*

*def mvnHome = tool name: 'maven-3', type: 'maven'*

*withSonarQubeEnv('sonar-6') {*

*sh "${mvnHome}/bin/mvn sonar:sonar"*

*}*

*}*

*stage('Email Notification'){*

*mail bcc: '', body: '''Hi Welcome to jenkins email alerts*

*Thanks*

*Hari''', cc: '', from: '', replyTo: '', subject: 'Jenkins Job', to: 'hari.kammana@gmail.com'*

*}*

*stage('Slack Notification'){*

*sendSlackNotifications(message='Welcome to Jenkins, Slack!',msgColor='good')*

*}*

*def sendSlackNotifications(String message, String msgColor){*

*slackSend baseUrl: 'https://hooks.slack.com/services/',*

*channel: '#jenkins-pipeline-demo',*

*color: msgColor,*

*message: message,*

*teamDomain: 'javahomecloud',*

*tokenCredentialId: 'slack-demo'*

*}*

*}*

Git -> Source Code Management

Jenkins -> Integration Tool

Maven -> BUILD TOOL

Docker -> Container Tool

Tomcat -> Server for Java

Sonarqube -> Code Quality Check

Nexus

Prometheus

Cadvisor - Agent

AWS

## Steps

Create Ec2 Instance with t2.medium

Git install

jenkins install

maven 3.6.3 install

Docker install

POM.xml file -> Project Object Model, Code dependencies, Application.

A Project Object Model or POM is the fundamental unit of work in Maven. It is an XML file that contains information about the project and configuration details used by Maven to build the project. It contains default values for most projects.

## Life cycle of Maven

Validate -> Compile -> test -> Package

Maven will generate war file in target/sample.war file

Write DockerFile for Tomcat app

webapps/sample.war file location for deployment

## Maven Architecture

Going to use Scripted Pipeline using groovy script with stages

**Maven Plugin Integration Plugin**

use maven tool under tools in jenkins and give name & path

Create a project with Pipeline add the groovy script,

Stage 1 : SCM Checkout

Stage 2 : Maven Build to get newapp.war file

Stage 3 : Build Docker Image

Stage 4 : Docker Image Push

## How to add password in Jenkins ?

Manage Creditianls -> Global Credentials -> Secret Text

Stage 5 : Docker Container run

Sonarqube

# Sonarqube installation

Needs 3GB RAM minimum

JAVA is pre-requisite

You can follow the below-given commands.

Create RDS DB server

CREATE DATABASE sonar CHARACTER SET utf8 COLLATE utf8\_general\_ci;

CREATE USER 'sonar' IDENTIFIED BY 'Sonar@1234';

GRANT ALL ON sonar.\* TO 'sonar'@'%' IDENTIFIED BY 'Sonar@1234';

GRANT ALL ON sonar.\* TO 'sonar'@'localhost' IDENTIFIED BY 'Sonar@1234';

FLUSH PRIVILEGES;

install sonarqube in instance

add database, user group in DB

install sonarqube

conf -> wrapper.conf -> give java jdk path & dont edit anything

sonar.properties file -> edit the mysql config, port no = 9000, sonar.web.context=/sonar

chown -R ec2-user:ec2-user /opt/sonarqube-6.7.0

/sonarqube/bin/linux-64bit/sonar sh start – To start sonar.

/sonarqube/bin/linux-64bit/sonar.sh stop – To stop sonar.Page 20 of 56

/sonarqube/bin/linux-64bit/sonar.sh restart – To restart sonar.

/sonarqube/bin/linux-64bit/sonar.sh console – To see the output.

create a token in sonar for communicating from jenkins to sonarqube (can also create new token and change password in account)

# Sonar with Jenkins Maven

Open Jenkins and add install sonarqube scanner

Add sonar token in jenkins credentials Configure system -> Add sonarqube token and Sonarqube URL with port no

Add the new script and it will execute with sonarqube

First, you have to download sonar-scanner plugin (sonarqube in old versions).

# In Jenkins

Global tool configuration

In sonar installations section. Click add sonarscanner. Type a name. Give Sonarqube home path

Go to configure system, In sonarqube servers section. Type name. Sonar server url. Sonar version. Sonar account username and password if it asks (by default, admin).

# In Maven

Go to settings.xml,

Under pluginsGroups section.

Below ‘→’ mark, remove the </pluginGroups> line and paste this. <pluginGroup>org.sonarsource.scanner.maven</pluginGroup> </pluginGroups>

In profiles section, • Below ‘→’ mark. Remove </profiles> line and paste this

<profiles>

<profile>

<id>sonar</id>

<activation>

<activeByDefault>true</activeByDefault>

</activation>

<properties>Page 22 of 56

<!-- Optional URL to server. Default value is http://localhost:9000 -->

<sonar.host.url>

http://192.168.10.32:9000

</sonar.host.url>

</properties>

</profile>

</profiles>

It will download the plugin from given website above in plugin group section and it will detect the sonar from the given url.

After copying this code in settings.xml file, run any maven cmds to see whether we did correct (or) not.

To test sonar in linux, go to your project pom.xml dir and run mvn sonar:sonar….It will test the code and give you the errors in the

code in sonar GUI dashboard along with your project name.

Login to your sonar GUI dashboard with your username and password, there you can see your project name. Inside the project

name you can see all your code along with errors if there are any.

After configuring sonar in Jenkins, while creating a project,

In build Environment, Select Prepare sonarqube scanner environment.

In post build actions, Select sonar analsys with maven and save the project.

Now, after build is completed, it starts sonar analsys and push the code errors and bugs to sonar dashboard.

# Quality Gate

You can use Jenkins' Quality Gates plugin for it.

It will provide you 'Quality Gates' as a post-build option. You should then just fill in your project key from SonarQube. Remember, it will fail your Jenkins job both in cases of warning or failure on your quality gate.

Sonar way

waitForQualityGate() -> func for checking quality gate

## Sonar qube quality gate Metrices

* complexity /class/file/function
* Documentation Comment lines Comment % Public API Public Documented API
* Duplication blocks/ files/ lines
* Blocker Issues/ Critical Issues/ False Positive Issues/ Info Issues/ Major Issues/ Minor Issues
* Classes Directories Files

## Metric Line

* measure:
  + Coverage on new code
  + Blocker issue
  + New Critical Issue
  + Technical Debt Ratio on New Code
* comparison operator
* error value

# Can Sonarqube be started as root

Sonarqube cannot be started as root -> SonarQube starts an Elasticsearch process, and the same account that is running

SonarQube itself will be used for the Elasticsearch process.

Since Elasticsearch cannot be run as root , that means SonarQube can't be either.

ElasticSearch will not require root privilege to run the server. So you need to create one user named elasticsearch and then try to run the ElasticSearch program.

Nexus Artifactory

Private artifactory

Jfrog (needs lisence) & sonatype Nexus

Create new instance yum install java

Download Nexus and ./nexus start -> give initial admin password and create new password

Create respository -> Docker Hosted -> give name, http 8083, give Docker v1 API access

install docker -> create /etc/docker/deamon.json -> insecure registry give publicip:8083

also do the same in Docker machine where jenkins is running the docker and dp the same /etc/docker/deamon.json

# Github

repo -> settings -> webhook -> give jenkins URL

jenkins add Github hooktrigger for GitSCM polling -> give repo url

# Prometheus

install prometheus in docker container

needs promethues.yml file

monitoring docker container using Cadvisor(insatall slave)

promethues.yml file -> give machine ip

container\_memory\_usage metrics

Day 1

<https://kubernetes.io/docs/reference/kubectl/cheatsheet/>

**Kubernetes VERSION 1.18**

# Kubernetes Agenda for Day 1:

- Microservices vs Monolithic

- Why Kubernetes

- History of K8s

- Knowledge Pre-requisites for K8s

- K8s Architecture

- K8s Objects

# Monolithic vs Microservice ?

## Monolithic Architecture

Every enterprise application has a similar kind of layered architecture:

1. **Presentation**: The user interface.

2. **Business logic**: The application’s internal business logic.

3. **Database access**: Almost all applications need to access DB, either SQL or NoSQL.

4. **Application integration**: Quite often, the application needs integration with other applications. This is usually achieved via web service calls (SOAP or REST), or via messaging.

Even though applications have clear, logically modular architecture, but usually, most of the application is packaged and deployed as a monolith. There are actually some advantages to doing this.

*Advantages of Monolithic Architecture:*

* Simple Development
* Simple Testing
* Simple Deployment
* Simple Scaleup

*Disadvantages of Monolithic Architecture:*

* Flexibility
* Reliability
* Speed in Development
* Building complex apps
* Continuous deployment

Because of all the above drawbacks of monolithic applications, microservice architecture is becoming more and more popular day by day. So what is microservice-based architecture?

## Microservice Architecture:

In short, the microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, usually via RESTful web services or messaging.

How Microservice Architecture Tackles the Drawbacks of Monolithic Architecture??

1. **Flexibility:** Microservices architecture is quite flexible. Different microservices can be developed in different technologies. Since a microservice is smaller, the code base is quite less, so it’s not that difficult to upgrade the technology stack versions. Also, we can incrementally adopt a newer technology without much difficulty.

2. **Reliability:** Microservices architecture can be very reliable. If one feature goes down, the entire application doesn’t go down. We can fix the issue in the corresponding microservice and immediately deploy it.

3. **Development speed:** Development is pretty fast in microservices architecture. Since the volume of code is much less for a microservice, it’s not difficult for new team members to understand and modify the code. They become productive right from the start. Code quality is maintained well. A microservice takes much less time to start up. All these factors considerably increase developers' productivity.

4. **Building complex applications:** With microservice architecture, it's easy to build complex applications. If the features of the application are analyzed properly, we can break it down into independent components which can be deployed independently. Then, even the independent components can be further broken down into small independent tasks which can be deployed independently as a microservice. Deciding the boundaries of a microservice can be quite challenging. It’s actually an evolutionary process, but once we decide on a microservice, it’s easy to develop, as there are no limitation in technologies.

5. **Scalability:** Scalability is a major advantage in microservice architecture. Each microservice can be scaled individually. Since individual microservices are much smaller in size, caching becomes very effective.

6. **Continuous deployment:** Continuous deployment becomes easier. In order to update one component, we have to redeploy only that particular microservice.

# Why Kubernetes ?

Why Kubernetes?

* A software tool to run Dockerized application in the Cluster of Nodes
* Used to Monitor & efficiently managing the environment of the Enterprise Application
* Also been called as Orchestrational Engine
* It take cares about the Stability of the Application, in order to brings up the functional units automatically.

## Use case for understanding in Deep:

Example case - Swiggy or Flipcat Online Applications.

Microservices applications involved:

1. Search App
2. Pricing App
3. Cart App
4. Order App
5. Payment App

Note: Differs in the number of VMs or Containers involved.

## History of K8s?

* It is an open-source container-orchestration system for automating application deployment & management, which was originally designed by Google in 2014.
* Now it is maintained by CNCF [Cloud Native Computing Foundation] .
* Written with **GO language**.

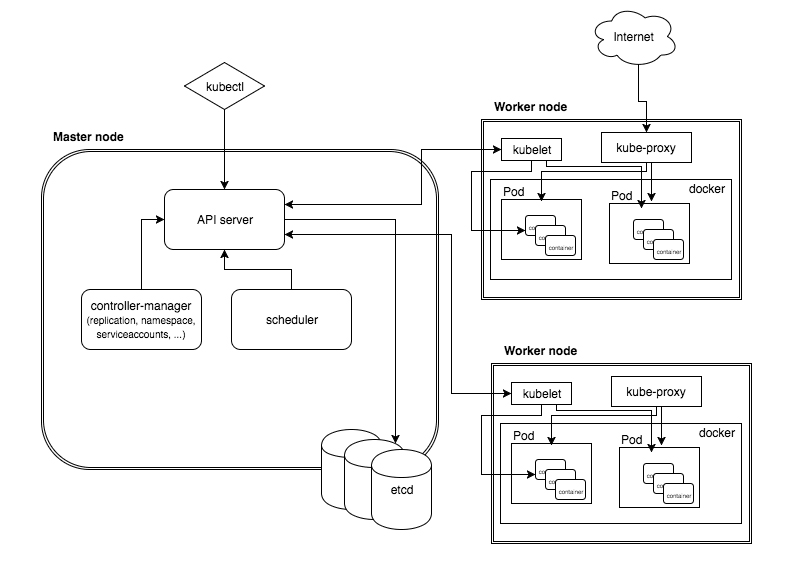
## Knowledge Pre-requisites for K8s

Below are the Key points which we need to be familiar,

* Basics structures of Json/Yaml
* Concepts about API & Rest API
* Content of an Enterprise Application - EAR
* Understanding of Architecture flow.

# Kubernetes Architecture

It's a management Architecture, that comprises various managable components into it. Which gives an Orchestration system



## Master Node/ Control Plane

1. **API server** - Its a front end, to communicate & maintain configurations of Cluster
2. **Controller Manager** - Controls the number of POD replicas & creations.
3. **Scheduler** - Schedules the Pod activities.
4. **etcd** - Distributed key value db. All the details of the cluster will be stored & kept here.

## Worker Node

1. **Kubelet** - It is an agent/connectivity, to Monitor/ manage the containers in the Pod. **CAdviser**
2. **Kubeproxy** - Bridge between App\_user & Application. It is a logical endpoint.
3. **POD** - Grouped containers, functional units in K8S.
4. **Containers** - Container runtime, K8S suppports Docker, Mesos, OpenVZ, Marathon.

## Kubectl

1. Its a tool used to communicate with the Kubernetes cluster.
2. It connected with the APIServer for the communication
3. ~/.kube/config file contains the cluster information and shares to kubectl
4. kubectl command format,

*kubectl <Operation\_command> <Type\_command> <Name\_command>*

- **Operations**:- Get, Create, Delete, Describe, Logs, etc

- **Type**:- Pods, Deployments, Jobs, Namespace, etc

- **Name**:- Seach-pod, Cart-Deployment, Ui-Service, mail-jobs, etc

*eg:- kubectl get pod test-pod*

Day 2

# Ways of Installation Kubernetes:

* Kubeadm
* **kops – aws**
* kubespray
* miniqube

We manage

cloud provider managed -> EKS, AKS, GKE

# Kubeadm Method

## Docker Install

1. Curl get gpg key
2. add Docker repo
3. repo update
4. install docker
5. hold docker to specific version
6. check docker is running using systemctl command

## Kubernetes Install

1. Curl get gpg key
2. add Kubernetes key
3. repo update
4. install kubectl, kubelet, kubeadm
5. Create a new network
6. Run the commands said by the Kubernetes in Master
7. Run the command in Node machine using Token (23 hours lifetime)
8. Create a new cluster network **Flannel Network**

## Create New Token

kubeadm token create --print-join-command (Creates a new Node)

# Kops method

1. Create new Ubuntu Machine
2. Download kops binary, change permissions, move to binaries file
3. Download kubectl binary, change permissions, move to binaries file
4. Create IAM User, aws configure in kops machine
5. Export accesskey, Secret accesskey
6. Clustertype: **Single Node Cluster**, MultiNode CLuster
7. Create S3 bucket to store statefile
8. ssh-keygen
9. Create **Gossip-based Cluster** & give bucketname (Wait 10 min)
10. Automatically Creates 1 master, 2 worker, load Balancer will be Created & ASG will be created, S3 contains statefiles
11. You can edit master and worker by running commands
12. We will run the commands from KOPS machine & ssh into to master & slaves
13. ASG change the desired capacity to 0
14. If you want can also install using Ansible

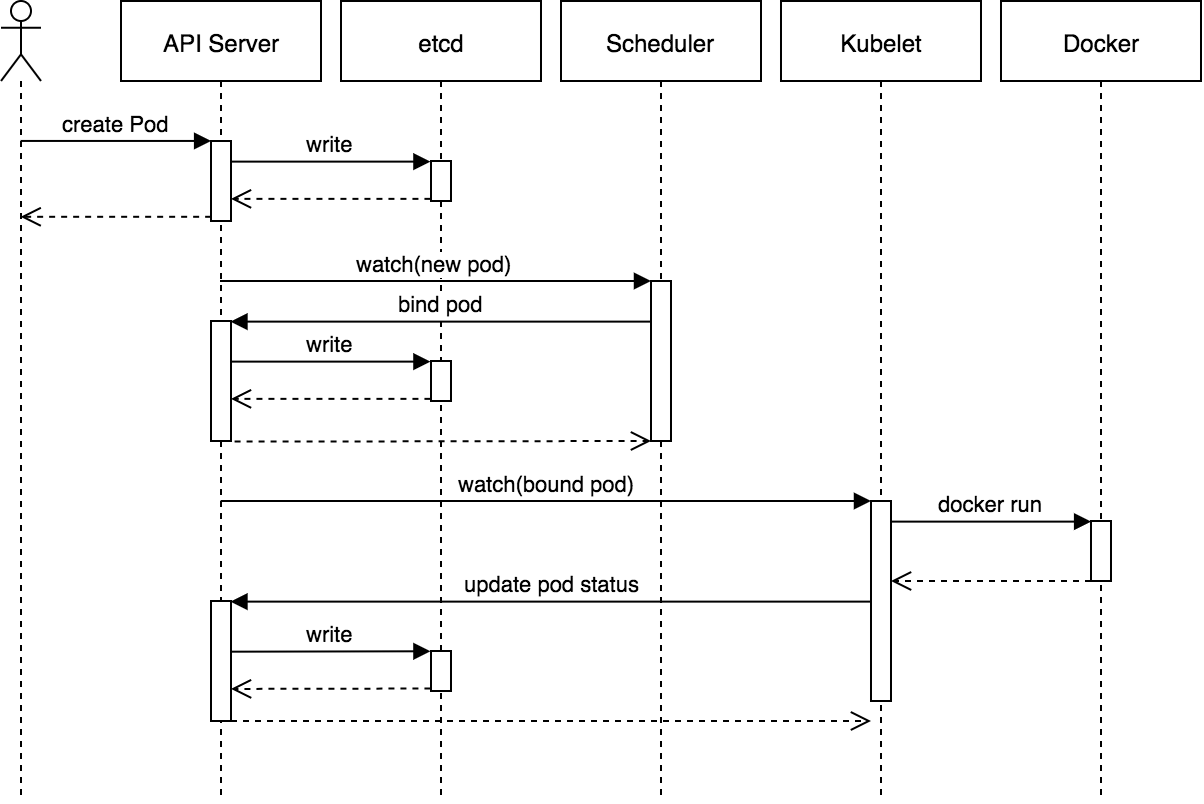
Day 3

# Cluster

Logical grouping of the Pods across Nodes

# Pod in Detail

* An abstraction layer that represents the group of containers that are deployed together on the same Host/Node.
* It is used to manage the applications in the Kubernetes.
* A wrap layer for the deployed applications.
* In one Pod multiple containers can also been ran.



# Life Cycle of Pod

- pod.yml - description of the pod details

- API server - will gets the pod.yml through the client [kubectl]

## Status of POD

1. **Pending** - After issuing the command to create pod, K8S will accept it & starts working with Scheduler allocate the resource.
2. **Running** - As per the Scheduler suggestion, the node will be allocated and container starts to creates by pulling the image from DockerHub.
3. **Succeeded** - Successfuly container getting terminated.
4. **Failed** - Due to some reason, Application getting crashed and not started.
5. **CrashLoopBackOff** - Its a Retry status to restart the containers after a failure.

# Networking in POD

Application to Application communication inside K8S can be by following methods,

* Container to Container communication --> Network **Namespace**
* Container from one Pod to Another Pod from same Node --> Network **Bridge using Ethernet**
* Container from One Node to Another Node --> Network **Routing Table**

# Deployment of yaml file

*apiVersion: apps/v1*

*kind: Deployment*

*metadata:*

*name: nginx-deployment*

*labels:*

*app: nginx*

*spec:*

*replicas: 3*

*selector:*

*matchLabels:*

*app: nginx*

*template:*

*metadata:*

*labels:*

*app: nginx*

*spec:*

*containers:*

*- name: nginx*

*image: nginx:1.14.2*

*ports:*

*- containerPort: 80*

# Api Version

Kind apiVersion

CertificateSigningRequest certificates.k8s.io/v1beta1

ClusterRoleBinding rbac.authorization.k8s.io/v1

ClusterRole rbac.authorization.k8s.io/v1

ComponentStatus v1

ConfigMap v1

ControllerRevision apps/v1

CronJob batch/v1beta1

DaemonSet extensions/v1beta1

Deployment extensions/v1beta1

Endpoints v1

Event v1

HorizontalPodAutoscaler autoscaling/v1

Ingress extensions/v1beta1

Job batch/v1

LimitRange v1

Namespace v1

NetworkPolicy extensions/v1beta1

Node v1

PersistentVolumeClaim v1

PersistentVolume v1

PodDisruptionBudget policy/v1beta1

Pod v1

PodSecurityPolicy extensions/v1beta1

PodTemplate v1

ReplicaSet extensions/v1beta1

ReplicationController v1

ResourceQuota v1

RoleBinding rbac.authorization.k8s.io/v1

Role rbac.authorization.k8s.io/v1

Secret v1

ServiceAccount v1

Service v1

StatefulSet apps/v1

# Replica Set (similar to autoscaling)

Manager for Pods & Containers

Desired Number == Current Number

label

Does not create about Nodes

Application Supporting pods are handled by Replica set

# Deamon Set

Node supporting pods are handled by Deamon Set

Depends on Node

Node level Log manager is Deamon set & fowards to Splunk

# Replication Controller

# Deployment model

1. Recreate POD (All update)
2. Ramped/ Roll Update (One by One)
3. canary
4. blue/ green
5. a/b testing

<https://blog.container-solutions.com/kubernetes-deployment-strategies>

Day 4

# Kubernetes Namespace

*kubectl apply*

*kubectl get*

*kubectl describe*

*kubectl delete*

kubectl get namespace

1. default
2. kube-public
3. kube-system
4. kube-node-lease

kubectl get pods -n kube-system

1. All pods related to Kubernetes Architecture

kubectl create ns greens -> Creates new NameSpace

kubectl get namespace

# Create new pod using doc refrence

<https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.21/>

**vi pods.yaml**

apiVersion: v1

kind: Pod

metadata:

name: pod-example

spec:

containers:

- name: ubuntu

image: ubuntu:trusty

command: ["echo"]

args: ["Hello World"]

kubectl apply -f pod.yaml

kubectl get pods

**vi replicaset.yaml**

apiVersion: apps/v1

kind: ReplicaSet

metadata:

# Unique key of the ReplicaSet instance

name: replicaset-example

spec:

# 3 Pods should exist at all times.

replicas: 3

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

# Run the nginx image

- name: nginx

image: nginx:1.14

kubectl apply -f replicaset.yaml

kubectl get pods

kubectl delete pod <pod name>

kubectl get pods

kubectl get rs

kubectl delete rs replicaset-example

**vi deployment.yaml**

apiVersion: apps/v1

kind: Deployment

metadata:

# Unique key of the Deployment instance

name: deployment-example

spec:

# 3 Pods should exist at all times.

replicas: 3

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

# Apply this label to pods and default

# the Deployment label selector to this value

app: nginx

spec:

containers:

- name: nginx

# Run this image

image: nginx:1.14

kubectl apply -f deployment.yaml

kubectl get pods

kubectl get deployment

kubectl describe pod <deplyment pod>

if editing the old deployment.yaml file and re-running with error does not delete the old pod

if editing the old deployment.yaml file and re-running without error it deletes the old pod

kubectl get pods -o wide

kubectl exec -it <pod> /bin/bash

# kind: pod vs deployment

1. pods will not have replica set, deployment will have replica set
2. if pod failed everything fails, deployment will have backup pod if failed

# What is Deamon set ?

Docker image: Prometheus Node Exporter

apiVersion: apps/v1

kind: DaemonSet

metadata:

name: mynode-exporter

labels:

app: mynode-exporter

spec:

selector:

matchLabels:

app: mynode-exporter

template:

metadata:

labels:

app: mynode-exporter

spec:

containers:

- name: mynode-exporter

image: prom/node-exporter:v0.18.1

ports:

- containerPort: 9100

hostPort: 9100

protocol: TCP

kubectl apply -f deamonset.yaml

kubectl get daemonset

Day 5

# Services

A Service allows you to dynamically access a group of replica pods

Traffic -> Proxy -> Service -> Pods

* **NodePort** Services (External) range: 30,000 – 32,767
* **ClusterIP** (Internal) Pod to Pod communication (Network Related connectivity is Netword Bridge)
* **LoadBalancer** (External Cloud Load Balancer)
* **Ingress** for efficient Load Balancer (Inbuilt Ingress Load Balancer)

# Config Map

EAR --> WAR + JAR + CFG Files

property file

XML files

JDBC files

Config file Internal

Config file External --> Config Map

# Secret Map

Used for Storing UserId & Password

# Volumes

Volumes in yaml file

pod.spec.volumes

pod.spec.containers.volumeMount

NonPersistent Volume -> Empty Dir

Persistent Volume -> HostPath

Persistent Volume Claim

Day 6

# Node port range: - 30,000 – 32,767

kind service -> metadata -> Selector

kind pod -> metadata -> contains labels (Should be same as key: value in Selector)

kubectl apply -f nodeport.yaml

kubectl get svc

kubectl get pods -o wide

Give all traffic on Security Group from AWS

# Cluster IP

Default is Cluster IP, All Pods are created in Cluster IP

kubectl apply -f clusterip.yaml

kubectl get svc

kubectl get pods

kubectl exec -it <pod-name> sh

# Load Balancer

kubectl apply -f loadbalancer.yaml (only works in KOPS method)

We need to specify Cloud Provider in Kubdeadm

# Ingress Controller

nginx ingress Controller is used for Load Balancing

We can user with multiple domain based routing & Path based Routing

Route 53 -> Load Balancer -> Rules -> Services -> Selecter -> Deployment/ Pod (Label)

RBAC -> Role Based Access Controller

Day 7

# GCP

Using Google Cloud Platform

Kubernetes Engine -> Cluster -> Bootstrapping a Cluster

Automatically VMs will be created

# ENV

Creating a yaml file for Env variable

kubectl apply -f env.yaml

kubectl get pods

lubectl get pods -l purpose=demonstare-envar

kubectl exec <pod name> --printenv

# Config Map

Config Maps allow you to decouple configuration artificats from image content to keep containerized applications portable

Imperative way -> launching directly in CLI

Delarative way -> writing in yaml way

kubectl create configmap –help

kubectl create configmap <map-name> <data-source>

in yaml file -> envfrom -> config map

kubectl create configmap get-config --from-file=<dir path>

kubectl get configmap

kubectl describe <pod name>

# Secrets

We can store ENV Variables in ConfigMap but we cannot store Username & passwords

in yaml file -> envfrom -> Secret Ref

# Volumes

in yaml file -> spec -> Volumes -> hostpath -> path= /data -> type= directory

in yaml file -> spec -> Volumes -> hostpath -> path= /data -> type= DirectoryorCreate

in yaml file -> spec -> Volumes -> hostpath -> path= /data -> type= FileorCreate

Kubernetes 3rd party Volumes

* aws elastic block storage
* Elastic File system

Create a Persistent Volume –> Retained, Recycled, or Deleted

PV = Created by Admin

PVC = Created by Users

Claim by the PODS

## Storage Class

For Dynamic Storage we can use Storage Class

Create Storage Class -> PVC -> Attach it to your pods

# HPA -> Hortizonal Pod Autoscaling

kubectl autoscale deployment <pod-name> --cpu-percent=50 --min=1 --max=10

kubectl get hpa

# Monitoring

* Grafana
* ELK
* Prometheus

kubectl top pods

kubectl top nods

kubectl logs <cont name>

Grafana -> Data Source -> Test DB source

Kubernetes

## What is config map ?

How is Kubernetes related to Docker?

What is Container Orchestration?

What are the features of Kubernetes?

How does Kubernetes simplify containerized Deployment?

What is Google Container Engine?

What is Heapster?

What is Minikube?

What is Kubectl?

What is the syntax for the Kube-proxy command?

What is the syntax for the Kubectl command?

What is Kubelet?

What are the different components of Kubernetes Architecture?

What do you understand by Kube-proxy?

What is the Kubernetes controller manager?

What are the different types of controller manager running on the master node?

What is ETCD?

What do you understand by load balancer in Kubernetes?

Write a command to create and fetch the deployment.

What is Ingress network?

What do you understand by Cloud controller manager?

What are the different types of cloud controller manager?

What is a Headless Service?

What are federated clusters?

What is a pod?

What is the difference between a daemonset, a deployment, and a replication controller?

What is a sidecar container, and what would you use it for?

How can you separate resources?

What are K8s?

What is a node in Kubernetes?

What does the node status contain?

What process runs on Kubernetes Master Node?

What is the job of the kube-scheduler?

What is a cluster of containers in Kubernetes?

What is a Namespace in Kubernetes?

Name the initial namespaces from which Kubernetes starts?

What are the different services within Kubernetes?

What is ClusterIP?

What is NodePort?

What is Kube-proxy?

How can you get a static IP for a Kubernetes load balancer?

What is the difference between config map and secret?

If a node is tainted, is there a way to still schedule the pods to that node?

Can we use many claims out of a persistent volume?

What is Kube-proxy?

What are the tools that are used for container monitoring?

What are the important components of node status?

What is minikube?

Explain Prometheus in Kubernetes.

List tools for container orchestration.

Define Stateful sets in Kubernetes.

Explain Replica set.

Why uses Kube-apiserver?

Explain the types of Kubernetes pods.

What are the labels in Kubernetes?

What is Sematext Docker Agent?

Define OpenShift.

What is ContainerCreating pod?

What do you mean by Persistent Volume Claim?

What will happen while adding new API to Kubernetes?

How do you make changes in the API?

What is kubectl drain?

Define Autoscaling in Kubernetes.

What is the “Master”?

What is a Swarm in Docker?

What is Kubernetes Log?

What are the types of Kubernetes Volume?

Explain PVC.

What is the Kubernetes Network Policy?

What are minions in Kubernetes cluster?

Kubernetes cluster data is stored in which of the following?

Which of them is a Kubernetes Controller?

Which of the following are core Kubernetes objects?

The Kubernetes Network proxy runs on which node?

What are the responsibilities of a node controller?

What are the responsibilities of Replication Controller?

How to define a service without a selector?

What did the 1.8 version of Kubernetes introduce?

The handler invoked by Kubelet to check if a container’s IP address is open or not is?

kubernets master slave architecture & purpose

What method in kubernetes

WHat is pod

what is deployment

what is deamon set

statefull set

kubernets service node port cluster ip ingress controller

config map secret

pv pvc

what to monitor kubernets using prometheus and grafana

ELK

what to do if pod fails

kubernets resource limit

deamonset vs replicaset