DevOps

→ Devops is a framework that automates the process of development and operation.

→ DevOps is a technology that integrates various tools. Using these tools the development team can continuously update the application (continuous integration) and at the same time the deployment team can deploy the application to cloud resources known as Continuous Deployment

→ Using a CI/CD pipeline the process of software development and deployment can be faster and less costly.

→ A CI/CD pipeline has 5 basic stages

Stage-1 → Automated creation of Resources - Terraform (Tools)
Stage-2 → Automated configuration of these resources - Ansible (Tools)
Stage-3 → Containerization of an application - Docker (Tools)
Stage-4 → Container orchestration - Kubernetes (Tools)
Stage-5 → Resource Monitoring - CloudWatch (Tools)

→ To combine all these tools we need to create a pipeline script. The script can be written in Bash or Python.

→ The process of resource creation and deployment is mostly done in a public provider cloud.

* All our resources will be created and deployed by AWS.

Linux - Basic Comands -

Linux Directory Structure -

Top level Directories -

- /bin - binary or executable files.
- /etc - contains system configuration files.
- /home - home directory. It is default Current Directory.
- /opt - contains optional files or third party software
- /tmp - temporary files / space, typically cleared on reboot.
- /usr - User related program.
- /var - Log files (It contains all log files.)

Linux Terminals-

Commands - ls, pwd

e.g. ls
ls -a
ls -alh
ls -lah

#Today's Self Practice (03/7/23)

Q1. Trace the history of Ubuntu operating system.

Q2. Pratice all variants of 'ls' commands.

Q3. Do a man ls, man pwd, man whoami, man cd, man mkdir, man rm

Navigation commands -

example

cd - change directory. (cd absmath)

mkdir - make directory. (mkdir absmath)

rm - remove (rm. a.txt)

* rm-r → Recursive remove

↳ It will remove all files and folders inside a directory

04/7/23

Command to create a file of 0KB.

→ touch file.txt

mkdir dil
cd dil
ls
touch file.txt

→ A blank file can be created by using the touch command.
→ To edit the file we need to open it in an editor such as Vi or nano

Syntax - vi file.txt → to save esc, colon key, wq
or nano file txt → ctrl+O then ctrl+X to exit.

-> vi and nano are used to edit text file.

-> cat command is used to see contents of a file.

curl and wget

/sudo apt-get install curl - to install

- curl is a network downloader that can download files in the terminal.
- Mostly curl is used as a browser to check webpages.
- wget is a network downloader that can be used to download ~~website~~ files from website.

systemctl

- systemctl stands for system control that can be used to start an application, stop an application, restart an application and check status of an application.

```
/sudo systemctl status nginx
/sudo systemctl stop nginx
/sudo systemctl start nginx
```

Kill

- Kill command sends a ~~sytle~~ sig - kill that terminates the process.
- To kill a process we need to find its pid
- ps command is used to list all processes and pids.

```
Kill -9 id
```

cp

- The cp command is used to copy a file from source directory to destination directory.

```
cp src-file Dest-file
```

mv

- mv stands for move
- It has 2 basic uses it can move a file from source to destination.
- It can rename a file.

Today's Self Pratice (4/7/23)

Q1. Check all the option of GREP using man command.

Q2. Try to find various use cases where GREP can be used

~~Q3. Error permission Q20 using Open Redirect~~

GREP

GREP is a command line tool that can be used to search for text or regular expressions within a file.

Syntax - grep wordyouwanttosearch filename - (search)
grep -i 'bar' filename (case-insensitive search)
grep -R 'httpd' (search in all the subdirectories in linux)

find (search for files) -

find . -name abi.txt (search for files in current directory)
find /home -name *.jpg (Look for all .jpg files in the /home and directories below it)

find /home -user randomperson -mtime 6 -iname ".db".

- The above command will find all files present in /home folder and has edited by the user called randomperson.
- The mtime option specifies the modification type.
- Here all files will be count that has 6 days old.
- The command will filter out with name with ".db" extension only.

Self practice

Q3. Using wget download any text file from any downloadable link. Inside the txt. file search for string 2023. used find command to find all the log files that are atleast 2 days old.

Q4. Use grep command to recursively find "and" in all folders and subfolders of the home directory. (grep -r "and" *)

Q5. use chmod to change permission of the file so that only user can read write execute.

LINUX PERMISSIONS :

- Every file and directory has three kind of permissions.
  1. Read
  2. write
  3. Execute
- Again there 3 types of permissions are present for three types of users.

  For users, for groups and for others.

  For user - 3 permissions
  For groups - 3 permissions
  For others - 3 permissions
  } Total 9 permissions.

- The permissions of a file can be changed using the chmod command.
- The numeric mode define permissions in decimal format.

Number	permission type	symbol
0	no permission	---
1	Execute	--x
2	write	-w-
3	execute + write	-wx
4	Read	r--
5	Read + execute	r-x
6	Read + write	rw-
7	Read + write + Execute	rwx

eg. If I want to give read - permission for user read-write group read write execute - others.

ans → sudo chmod 467 b.txt

vi editor

dd - deleting a single line
yy - copying in a single line.

press ? - stand.
to add new line - shift + O
O - new line after existing line

Redirection

cat a1.txt (input) > a2.txt (output) → Redirects a1.txt to a2.txt

cat a1.txt >> a2.txt → appends
— output, input

BASH

→ A bash script is a plain-text file which contains a series of commands.

→ Anything you can run normally on the command line can be put into the script and act same.
Similarly commands written in the script can be separately executed on the command prompt.

Shebang line -

```
#!/bin/bash
```

#! is referred to as shebang and rest of the line is the path to the interpreter specifying the location of bash shell in our operating system.
Under the unix-like OS, when a script with a shebang runs as a program, the program loader parses the rest of the lines with the first line as an interpreter directive. So shebang denotes an interpreter to execute the script lines, and it is known as the path directive for the execution of different kinds of scripts like bash, Python etc.

* Redirection

```
cat [file1] > [file2]
# content of file1 will be written to file2

cat [file1] < [file2]
# content of file 2 will be written in file1

cat [file1] >> [file2]
# content of file will be written to end of file 2

cat > [file name]
    → take input from keyboard (default input file) and
      write to the file.
```

BASH SCRIPTING -

- A Bash Script is a wraper that can automate a sequence of terminal commands. Any command that can be executed in the terminal can also be executed through a Bash script. Variables can be declared, string manipulation can be done, relational and logical comparisons can be done and also system commands can be executed through a Bash Script.
- Every Bash script starts with a dedicated line called Shebang line.
- The shebang line instructs the compiler that this is a script that must be run through the Bash Binary present in /bin folder.
- The script has to have executable permissions for the user.
- To make the script executable by writing chmod +x filename.sh

Q. Write a bash script to install MySQL server, start MySQL and enable my SQL server.

How to know if the previous command executed succesfully or not.

- The exit status of the previous command is stored in the special variable called $?

So by echo $? if the output is zero, then the previous command executed succesfully. Otherwise if it is a non zero value then previous command is field.

Arithmatic operators

Arithmatic operators can be used inside the expression commands.
The expr commands should be preceded by ~~backtick~~ forward apostrophe (`)

```
a=10
b=20
val=`expr $a+$b`
echo "a+b: $val"
```

The if command is used to test a condition and then command is used to take action. and clause should end with a 'fi'

1- then operator in shell

```
if [ $a == $b ]
then
   echo "a is equal to b"
fi

if [ $a != $b ]
then
   echo "a is not equal to b"
fi
```

Test Command

A test command is used to check value of a statement.
It returns boolean value true or false

- -eq → check equality
- -geq → greater than equal to
- -leq → less than equal to
- -s → used to check if the file exists in the folder or not.

```
if test -s $_____
              filename
```

Input/Output Redirection :-

In some situation it is required that the output of a command should be redirected to a file. Similarly it is also required that some commands take input from a file.

```
ls > a.txt   { store all the files showed in ls in a.txt
date > a.txt { store date in a.txt
echo this is line 2 >> a.txt.
                  ↳ append redirection

cat < a.txt { input redirection.
```

BASH SCRIPTING : STRINGS

① Cut Command - To cut by using hyphen (-) as the delimiter

cut -d'-' -f (column-no) <filename>

② pipe command - The pipe command is used to transfer the output of one command as input to another command.

cat a.txt | grep line

✓ ③ AWK - AWK is a scripting tool that is used to process data and create sql type inputs

∴ To find duplicates in a row

awk -F ',' '{if ($2==$3) {print $1","$2","$3} else {print "no duplicates"}}' answers.txt

✓ ④ sed - sed stands for string editor.

sed is a commandline tool that is used to search and replace some part of line or an entire line in text

```
sed '/old-word/new-word/' filename.txt
```

Cloud Computing :-

Cloud Computing is a computing framework that provides computing resources such as processing power, virtual machines, networks, VPCs, subnets etc to the end user on a pay-as-you-go basis

Cloud computing provides computing resources as utility just like a metered gas connection or water connection where I pay as much he uses

Advantages of Cloud Computing

- low investment cost
- faster setup of backend
- less management troubles

There are 2 types of Cloud Computing models

- private
- public
- Hybrid

A public cloud is a cloud that can be accessed by any individual whereas a private cloud is a cloud i.e. restricted to a certain organisation.

Cloud Deployment Models -

Infrastructure-as-a-service (IAAS) - If the end user only requesting for cloud hardware then the deployment is called infrastructure as a service. If the end user requests for hardware and some associated platform then it is known as platform as a service. (PAAS)

~~If the end-user request~~

Software-as-a-service (SAAS) - If the end user request for infrastructure, platform & application software then it is known as SAAS.

Role of Virtualization in cloud computing -

Virtualisation is a technique where a physical system can be isolated into several virtual system each running a different operating system.

When an end-users requests for resources, the cloud provider virtualises the hardware, & provides a chunk of resources to end users.

The common hardware is used by multiple virtual systems and they interact with the hardware using a hypervisor.

ex. of hypervisor - KVm
Virtual box
VMware

This type of virtualisation is called hardware level virtualisation

Hardware level virtualisation allows a cloud provider to maximise profit. by running multiple VMs on a single hardware

AWS

Cloud Provider :-

A cloud provider is an entity that has physical hardwares and computing resources and on request it provides those resources to end user. The most popular cloud provider is amazon AWS. Below AWS stands Azure and GCP (Google cloud Platform).

Basic terminologies

Instance

In AWS an instance is a virtual Machine that has RAM and storage and can run an operating system.

AWS provides instances through the EC2 (Elasting compute cloud) service.

~~An instance~~

By default AWS provides 8GB EBS volume with every instance.

Bucket

A Bucket is a object storage

Bucket is provided through a S3 service.

VPC

VPC is a private cloud that you can create inside AWS so that your resources are inaccessible from outside.

cloud watch

Cloud watch is the AWS monitoring service that can monitor the resources

Subnets

Subnets or subnetworks are smaller networks are used to divide a bigger network into smaller network for manageability and easier debugging.

Internet gateway

An internet gateway allows a subnet to access internets. Basically it is a router to which system connect. Internet gateway acts as an access point and allows the connected system to access internet.

NAT Gateway

Internet gateway cannot provide internet to ~~private~~ private subnet, it can only provide to public subnet. NAT gateway is used when a private gateway needs internet.

IAM (Identity and Access Management)

- An AWS account has one root user and ~~an aws~~ several other subordinates users. The subordinate users are IAM users.
- The IAM users has access to those services to which a root user grant them access. An user can provided access to certain or all services through IAM policies.
- An user can attached to inline policies or he may be attached to some predefined policies.

User group - An user group is an collection of IAM users. The group has same permission for all members.

As a best practice, the industry reccomends not to deploy any resources from the root account rather the root account should always be used to create and grant permission to IAM user.

Multi factor Authentication

It is a secure way of ensuring security to IAM user.

AWS Command line interface binds the terminal with IAM user account since Devops requires scripting we have to access all the aws services only from the command line interface.

aws configure.
ap-south-1

Q. write a bash script to create an IAM user and attach ec2 full access
S3 "
IAM "
VPC "
Administrator Access

Q. IAM cli commands

Elastic Compute Cloud :-

- It is an AWS service that allows a user to request for virtual machines. The system from which we create the virtual machines is known as the control node.
- The virtual machines are isolated virtual system running on real physical hardware.
- In AWS a virtual machine is also known as instance.
- AWS allows us to use prebuilt images known as "ami."
- Access to the instances are secured by assymetric key cryptography.
- By default an elastic block storage of 8GB is provided to every instance.
- Every instance have a public IP and a private IP. Any webpage or server hosted by instance can be accessed using its public IP.

AMI (amazon machine image) :-

AMI is a prebuilt operating system bundle provided by amazon. Every virtual machine requires an operating system. Instead of installing OS manually we can use a ami of our choice.

Key pair :-

Every AWS instance can only be accessed through its private key we need to have private key in our control node in order to SSH from control node to VMs.

Instance type :-

It defines the physical characteristics of an instance like its memory size, storage.
For a free tier account only t2.micro instance should by used.

Volumes :-

An EC2 instance can be attached to a dedicated mount volume. This volume is a elastic block storage (EBS) volume which is different from an S3 bucket.

Spot instances :-

A spot instance is an instance that is created on demand. These are spare instances that are provided by AWS on an discount. Therefore by using spot instances we can reduce the cost of our AWS account.

Security Group

By default all ports and protocols are not opened in an EC2 instance. A security group is a set of rules that defines which port should open and what protocol it should allow.

Every ~~EC2~~ EC2 instance must be attached to a security group.

If no security group is specified, AWS attaches the instances at the default security group.

SSH

SSH or the secure shell protocol is a cryptographic network protocol for remote login into a system. SSH uses the private key of the instance to remote login the instance

Syntax — ssh -i keyfile.

WEB SERVERS :-

- Mostly we want our instances to be part of the client server architecture. A server is a system or a software that listens to client requests. Once a server ~~requests~~ recieves a request from user it validates the user. If the user identity is validated then the server ~~sends~~ sends a response for the request.
- There are different types of servers.
- When a server is used to serve static or dynamic webpages, it is known as web server.
- Web servers store information i.e. retrieved using HTTP or HTTPS protocol.
- Proxy servers - A proxy server stands in between a client & a server. A proxy server adds extra layer of security and is also used for load balancing.
- FTP servers - FTP servers transfer files between client and server. The file transfer protocol is used to upload & download files

Application Servers - Application Servers hosts a running application. User connect through virtual private connection or virtual server connection.
Application servers are highly scalable and available.
web servers can be effectively hosted using apache, nginx.

- NGINX is a open source webserver which is highly available and can handle a large number of requests.
- orginally designed as a webserver.
- NGINX is also used for Reverse proxy, caching, load balancing, Content delivery Network.
- It can also perform as a proxy server for email because it can handle a large no of requests. it is very popular as a load Balancer.

Q. What is local host?

A- local host or loop back address 127.0.0.1 is the address of the local computer which is used for debugging purposes.
- loop back address or local host is not accessible by the internet.
- when we host a web server before we publish it to internet we need to test if the web page is running or not.
- loop back address is used to test the working of the webserver.

when we get unable to connect that means no webserver is running.

To go inside html file of nginx.

```
cd /var
cd /www
cd html
sudo vi index.nginx-debian.html → To edit the contents of html.
```

copy the public ip from instance page.

http://< >
public ip.

Q. write a bash script to create an ec2 ubuntu instance, the script should find the public ip address of the instance. the script should nginx inside the instance. finally the script should change the index page and your name to it.

Bootstraping a VM with user data :-

- Sometimes it is required, that some operation must occur in the VM even before the VM starts. Such a scenario can be handled by bootstraping user data to the VM

Advanced details → user data (write script)
or import script file
then launch instance

AWS-S3 :-

S3 - Simple storage Service

- S3 is an object storage provided by aws.
- In S3 storage takes place in logical partitions called buckets. The files that are stored in buckets ~~and buckets you~~ any type of buckets can be uploaded and updated.
- S3 bucket name should be globally unique. i.e. we cannot assign a bucket name i.e. being used by someone else.
- S3 is different from EBS and EFS
  EBS - Elastic Block storage
  EFS - Elastic file storage.
- S3 is an object storage while EBS is Block storage.
- we cannot install a block software like operating system in S3 bucket. For that we want EBS.

- There are different types of S3 storage & can be used in diff scenario

S3 standard
S3 intelligent Tiering
S3 standard-IA
S3 onezone-IA
S3 Glacier instant Retrieval
S3 glacier flexible Retrieval
S3 Glacier Deep archive
S3 outposts.

For eg when we have data that is infrequently used we can stored them in glacier type bucket.

- Every bucket is protected by an ACL (Access control List)
It controls ownership of the objects present inside bucket.
The ownership defines who can access the objects.

Bucket Versioning :-
It is the technique where multiple versions of an object can exist in a bucket.

Encryption
Server side Encryption (SSE-S3)
AWS Key Management Savice Keys (SSE-KMS)
Dual layer server side Encryption (DSSE-KMS)

- Every object uploaded to a bucket is accessible through an object end point, which is a object URL that can be opened.
- To delete a bucket all objects inside it must be deleted first.

s3-cp - used to copy an object from local system to a bucket.
s3-mv - " " move
s3-rm - " " remove

Q. Write a bash script to create a bucket and upload a object into the bucket.

S3 best practices -

1. Use IAM console to control access to your resources.
2. Add MFA for users
3. Restrict access to the bucket by allowing trusted host and trusted network
4. Review the security group rules regularly and ensure the concept of least privileged.
5. Disable password based login
6. Try avoiding creating resource from root account.

Q1. Write a bash script to create an s3 bucket. upload a website into the EC2 instance. change the index page using awk sed and redirections. The same script will also create a website on s3 bucket. At the end the script will return the public IP of EC2 instance and the bucket.

Q2. Write a bash script to take backup of a folder.

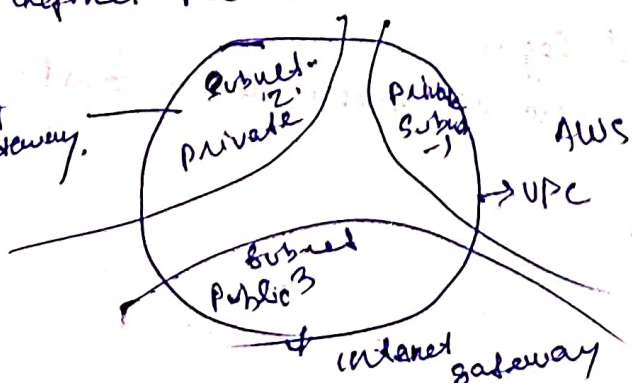
Q3. write a bash script to create a dashboard that prints system information. - memory
ram
Disk etc. . .

VPC (Virtual Private Cloud) :-

- Public cloud such as aws are insecure therefore aws provides a service called as virtual private cloud (VPC). A VPC is private cloud inside a public cloud provider.
- Resources inside a VPC are isolated from other public resources. This helps in providing security. This provides security and prevents the resources from unauthorised access.
- VPC enables us to launch our resources in a private network.
- This private network is again divided into subnetworks called subnets.
- A subnet is a subnetwork i.e. used to accomodate more hosts.

Homework
- what is CIDR
- what is subnet & subnetmask for a network 10.0.0.1 create 2 subnet with a proper subnetmask.

- A subnet can be private or public. It must be noted that a subnet can not access internet unless it is connected to a gateway.
- A public subnet is connected to a internet gateway. and a private subnet is connected to a NAT Gateway.
- The gateway acts as an access point and are always connected to a routing table.
- The routing table defines the inbound and outbound rules for the particular gateway.



VPC Peering :-

2nd -Hype -online J

Todays Self Assignment

Q1. Write a bash script to create a VPC & subnet inside the VPC create a internet gateway and attached IG to VPC create a route table & attach a route table to subnet & the VPC The script launches an EC2 instance inside the VPC, ssh into the ec2 instance & install nginx.

Q2. create a VPC & launch instance into it create another VPC & launch another instance in it, Try to ping from instance-1 to instance-2 using VPC peering

Q3. Write a bash script to create 2 EC2 instances, copy authorize key of 1 instance to another then try to ssh from 1 instance to another.

Application and Resource Monitoring :-

- Often resources that are created need to be monitored for critical events cloud-watch is an aws service that provides application & resource monitoring
- AWS resources are attached to cloudwatch API. one or more matrix are used to monitor the resource.
- Matrix are data related to system performance such as CPU load percentage network load percentage etc.
These matrix provide the state of the system at the current state where monitoring & resources predefined matrix can be used to alarm user defined matrices.

An alarm is a trigger that happens when a matrix crosses a certain threshold.

An alarm constantly watches a single matrix and notifies a user when a abnormality occurs.

Dashboard

Cloudwatch Dashboard is a visualisation platform where alarms/matrix can be visualised or analized.

SNS (Simple Notification Service)

when ~~the alarm~~ matrix crosses a predifined threshold then the alarm triggers an action.

SNS is a service that can be integrated with an alarm and anytime an alarm goes off the notification is sent to the user via email.

AWS config -

- AWS config is a monitoring tool that is used to access, audit and evaluate the configuration of your aws resources.
- AWS config looks for ~~any~~ changes in configuration of an instance and notifies the user once it detects changes in the configuration.
- There is a similar service known as CloudTrail.
- Cloud Trail can notify who made the change and when was change but AWS config can notify what was changed.

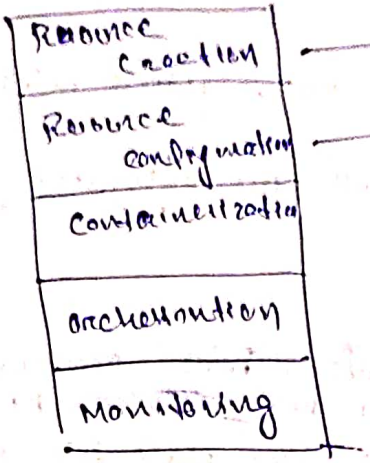
VPC flowlog

VPC flowlog is a feature that enables you to capture information about the IP traffic going to and from the gateways connected to VPC.

VPC flowlog data can be published to cloudwatch log, an S3 bucket or Kinesis data firehose.

After ~~create~~ you create a flowlog, you can retrieve & view the flow log records in the log group, bucket or delivery configuration.

A typical Devops job has 5 stages



Automated Resource Creation

- In Devops a script requires automated creation of cloud resources. Being written from a script there needs to be some code that can create the infrastructure. This process of creating infrastructure from code is known as IAC (Infrastructure as a code). IAC helps us to automate the process of resource creation by writing simple codes.
- Typically every Devops job requires some resources. such resources can be created by using an IAC tool. such as aws cloud formation or terraform
- cloud formation and terraform do the same job but terraform is not limited to any provider.

TERRAFORM

- Terraform is an open source infrastructure as a code tool i.e. used in a collaborating environment. Terraform has features like a state file that can be securily distributed among the collaborating members.
- Terraform providers provide the specific API for the particular provider.
- Terraform modules are an efficient way of reusing code to create resources. Terraform provider specified commands that make call to the provider API.
- The provider API again runs the command that create the actual resource.
- The system in which terraform is installed is known as the control node.
- The control node has the capabilities of creating resources on any provider.

Terraform Work flow :

1st Stage - Write

- Writing is the stage where the required infrastructure is written in a file using terraform blocks.
- The file is mostly saved with the extension .tf

2nd Stage - Init

- After writing the code "terraform init" command is used. This initializes the backend and downloads the providers API.
- Backend is the place where the state file is stored. It can be local system, S3 bucket, GitHub Repo etc etc.

3rd Stage - plan

Terraform plan command is used after the init command.

- During the plan stage, the code is mapped onto real world infrastructure
- This mapping is saved in the file know as the state file
- The plan process verifies if our code has any errors.

4th Stage - Apply

- The terraform Apply command actually creates the designated resources in the provider's end.
- It is the step that creates Real World Infrastructure in the providers end.

5th Stage - Destroy

- Destroy command destroys all the resources created using the current state file.

- The provider block gives us the information about the provider and the API version. Here we can force terraform to download a certain version of API.
- Data block is used to retrieve some data & store it in a variable.
- It is the resource block that defines actual AWS resources

- The Data block searches for an AMI id i.e. of ubuntu 20.04 type supports virtualization and is published by ubuntu and is owned by Canonical.

Q. write a batch script to create an ec2 instance, s3 bucket, security group, key pair, vpc, subnet, using terraform

Terraform Modules :

- Terraform modules are collection of .terraform files. Modules are always used to create ~~extra~~ reusable infrastructure of code.
- A module basically contains provider.tf file, main.tf file, outputs.tf file and variables.tf file.
- provider.tf contains the provider config & api versions.
- main.tf contains the data blocks and resource block.
- variables.tf contains the input variables that can be defined, there can be input variable of type bodean, list etc.
- output.tf contains the output variables that we wish to get output from terraform.
- A module can be stored remotely or locally.
- when a module is stored locally, can be used by specifying the path of the module.
- Similarly, a module can also be stored in s3 bucket or Github Repository.
- Modules are mostly written to provide reusability. Once a module is written, it can be used over & over again.

~~A module~~

- just as in the script we need some output variable
- while creating a resource it is possible to specify some pre-conditions- Resources in terraform will be created only if pre conditions are fulfilled e.g. precondition given checks ami id is for a 32 bit or 64 bit. only if it is 32 bit or 64 bit then resources are created.
- Similarly it is also possible to specify timeout condition such as create, delete with specific time interval. for eg aws database instance create time-out for create, update or delete operations.

Q1. Write a bash script to find & stop all running instances

Q2. " " " to create two VPC & Merge them into 1 VPC

Q3. " " " to find the no of objects in S3 bucket and total size of all objects in S3 bucket

Q4. " " " to detach the EBS volume from an EC2 instance and attach it to another instance (This is the solution when key.pem is lost)

TERRAFORM PROVISIONERS

- Sometimes it is required that some action is taken when the instance is created such as a software is installed or a file is sent to newly created VM.
- Instead of running a bash script locally, we can directly sent a configuration file to newly created VM.
- Provisioners are used to model specific action on a local machine or a remote machine.
- Most commonly there are 3 types of provisioners.
  - The File provisioner
  - Local-exec provisioner
  - remote-exec provisioner

- The file provisioner copies files or directories from the control node to the managed node.
- The local-exec provisioner runs some local commands which executed in the terminal of control node but it effects the newly created instance
- The remote-exec provisioner invokes a script in the remote system after it is created.
This can be used to run a configuration tool such as ansible, chef & puppet.

Q. write a bash script to create an aws instance using terraform then
i) Use remote exec to install nginx in the instance
ii) Use file provisioner to send a script to a newly created EC2 instance & the script will install nginx.

* When terraform creates an instance it attaches an instance to default security group. So check default security group has port 22 is anywhere.

DOCKER

Containerization using Docker :-

- Every software or application needs its set of dependencies for different operating system.
- Containerization is an efficient way of packing the application & its dependencies into a single wrapper.
- The advantage is that this wrapper can be executed on any OS or hardware specification.
- Containerization is a deployment process that bundles an application scored with all files & libraries.
- Software developers create an application and then pack this application into a container using several containerization tools.

Container Orchestration -

- Containerization is a process of creating a container but container orchestration is an automated way of managing the set of containers.
- Orchestration is required when the application is deployed as microservices

Create - Docker, Elastic container service
Orchestration - Kubernetes, OpenShift.

DOCKER -

Docker is an open source containerization tool. It can be used to

- create a container
- pull a container
- edit a container

Containers vs VMs

- A VM works on the concept of hardware level virtualisation.
- Each VM has its own OS, however, containers works on the concept of OS level virtualisation.
- All containers share the same OS.

Docker has 4 basic modules -

- Docker Engine
- Docker CLI
- Docker Desktop
- Containerd

- Containerd is the run time environment that supports os level virtualisation.
- Docker Engine is Docker daemon that listens to new containers
- Docker CLI is the command line interface.
- Docker Desktop is the GUI interface.

Docker Image -

- An image is the read only package that contains application & dependent libraries.
- A container is basically a running image.
- Application programers create an application, pack it into an image, upload it to any repository.
- An user who wants to run the application must pull the image as a repository & run the image as a container.
- Images can exist without containers. But containers cannot exist without image
- Every container runs several layers of an image but only the topmost layer of a container is writable.

* Docker hub is a free image repository that contains prebuilt images & user defined images. After creating an image it can be uploaded to docker hub & then pulled in any OS.

Docker Commands :-

Docker pull :- It is used to pull an image from docker-hub. The pull command has several options that let us download or pull a particular tag.

Docker run :- The Docker run command runs the required image. If the image doesn't exists it pull the image then runs it.

To ssh into a pseudo terminal for a container, Run must be used with -it

eg docker run --name -it Debian.

Docker PS :- It is used to list the running containers.

127.0.0.1

Date
Page No.

sudo docker run -it nginx

Date: 2/8/2023

A docker image can be created from an existing image by tagging the image or a docker image is created from a docker file. Docker file has no extensions

Command - A docker file starts with the "FROM" directory. It downloads an existing image. The 'work dir' specifies the working directory for the container. The copy directory copies the content of the app into the docker file

Run command issues the command to start the app.

Cmd or entry point is one of the many instructions which is used to configure the executables that will run when the container initializes.

cmd sets default parameters that can be overwritten but entry point sets default parameters that can not be overwritten.

By default docker run command does not open any ports. To open a port docker run command should be used with '-p' option.

Date

Page No.

Create a docker file to containerize an application. Run the image and map the port ip to port 3000. Follow the next steps given in the documentation to add a persistant into the container. So, that even if the container is refreshed the to-do list is existing.

Container Orchestration

- Orchestration is the process which automatically provisions, deploys, scales and manages containers.
- Features like auto scaling & replica set ensures that the user does not have to worry about managing the containers.
- Containers are volatile because of OS level virtualization.
- When there are large no of containers, it may so happen that a container might fail.
- It is the responsibility of orchestration tool to make sure that the failed container is up & running.
- Orchestration also provides additional features such as load balancing, ingress only network, multi-tier networking, name spacing & sub clustering
- Docker swarm is an orchestration tool that can manage a group of VMs as though they belong to a single cluster. The activities of the cluster are managed by swarm manager.
  All the other cluster members must join the swarm manager.
- Docker swarm lets you ~~out~~ connect containers to multiple host just like Kubernetes.

KUBERNETES

- Kubernetes (K8S) is an open source container orchestration system that is used for automating software deployment through containers, their scaling and management.
- Kubernetes orchestrates a group of resources by creating a cluster out of them.
- Every Kubernetes cluster must have at least one master node.
- The master node is responsible for managing and monitoring the cluster. All other nodes are known as the slaves.
- After creating the master nodes a control plane must be initiated in it.
- All the slaves must then join the master node.
- Once the slaves join the master nodes a cluster is formed. And now the master node can monitor the health application, running status, log files, errors, etc. of each slave nodes.
- Therefore, this clustered approach allows us to automate the process of orchestration because master node is always the control node.
- Application are not run in master node rather it is done by slave nodes.

Kubernetes Architecture -

Control Plane Components :-

The control plane components centralised decisions about the cluster for e.g. scheduling as well as detecting & responding to cluster events. for eg. failure of a container.

API Server -

1. Kube-apiserver - The api server is a component of Kubernetes control plane that exposes the Kubernetes api.
   - The control plane can manage the cluster by interacting with the API
   - The API contains pre-defined functions that are used to manage a cluster.

2. Kube-scheduler - control plane component that watches for newly created pod and assigns a node to a POD.

* Each slave node is known as a node. Inside every node there is a logical partition called POD. Each POD runs 1 or more containers & there can be multiple PODs in a single node.

3. Kube-controller-manger - It is the control plane component that runs the controller process.
   - A controller process is a continuous running process that listens to changes in a cluster.
   - Node controller is responsible for notifying the control plane about failed nodes Job controller - It controls the PODs and replica sets if the POD fails job controller make a new one.

4. Cloud-controller-manger - This component lets you to link your cluster to the cloud provider API
   It separates the components that interact with cloud platforms from components that interact with the cluster only.

5. etcd
   - It is a highly available key values store.
   - consistent & reliable.
   - works as a backup store to store cluster data.

Node Components :-

1- **Kubelet** - It ensures that containers are not running barely rather
It is always running on a POD.
Kubelet takes a series of POD specs and ensures the containers running described in POD specs are healthy & running.

2. **Kube-proxy** - Kube proxy is the networking service that runs on every slave node.
Kubeproxy maintains network rules that allow communication inside the cluster or outside.
~~The container runtime is~~

Container Runtime -
It is the software i.e. responsible for creating an environment inside slave node that can run container.
Kubernetes supports runtime such as container D or CRI-D.

Kubernetes Services -
When a POD is running in an node definitely application is running there.
A Kubernetes service is a mechanism of exposing the application running inside pod to the outside node.

There are 3 types of services.

1) **Cluster IP** - It is a Kubernetes service that allows a network application running on a POD to be exposed outside the cluster. Using Cluster IP any other slave can access the application POD.

2) **Node Port** - It is a Kubernetes service that allows an application running on a POD to be exposed outside the cluster. For a particular node a port is opened. Through this port network outside the cluster can access the application running on a POD.

3) **Load Balancing** - Load Balancing in Kubernetes lets one to distribute the traffic across replicated servers.

Replica Set

- The Replica set defines the no of copies of a POD that must be running at all instances of time.
- If a POD fails then it is the replica set that ensures that the POD is up & running.