Introduction

19/12

DevOps

Development + Operations

If we develope DevOps we want cloud services.

1. AWS - Amazon Web services
2. Azure
3. GCP - Google Cloud Provider

* Why we need cloud service?

* Tools in ~~AWS~~ DevOps

1) AWS, EC2-VPC, RDS → Cloud providing - Deployment or execut°
2) Ansible/chef → Configurat° mngt. tool - Host to chng whole sys.
3) Docker → Container technology - container
4) Kubernetes → Orchestrat° - To handle absence of new ver°
5) Jenkins → CICD - code integrat°
6) Ifrog → Artifact Repository → Store the backup appl°
7) Sonarqube → Code Analyzer - code checked (Error find)
8) Nagios → Monitoring tool → To check (status) sys. monitoring
9) Vagrant → Virtual Envi. tool → To create server clon
10) Terrerfarm → Cloud Format° - Using code create server
11) Git → version control tool - store code from Developers

(1) → (11) → (7) → (5) → (1) → (3) → (4) → (2) → (6) → (8) → (9) → (10)

20/12

Product° Server
Client site

Development Server
System or S/w

1) To create an account

AWS account creat° ← google

Create a a/c in aws
→ sign in to the console
↳ create a new account

Choosing Region

1) Govt policy
2) Latency
3) To find out the services are included in region
4) Pricing (when we r providing server)

* Services (Right side of page)
EC2 → Elastic Cloud Computer
↳ Instances - Stored in oregon
↳ Launch instances → Right side
↳ name server.
↳ choose O.S.
↳ Ubuntu, 18.04 (free tier

Instance type
↳ t2.micro (1 vCPU 1 GiB) RAM

* Key pair (login)
↳ Create a new key pair
▷ Key pair name
↳ Put name - like myOregon_key
Private key file format

-

* N/w Settings ① ②
↳ SSH traffic Anywhere ①
AND

✓ Edit ② SSH
↳ Add security group
↳ Type info → All traffic
↳ Source Typ → Anywhere

* Configure Storage (Hard Disk)

1 x [8] GiB [gp2] No. of instances → 1

Launch ~~Create~~ Instance (Click)

Generated instance ID
Click on ID hyperlink to open Instance

* Connect a server use 'git'

Search in google git
↳
Download for windows
↳ 32 or 64 bit for download

Next to a key file you hv to right click & get the git bash

Right click on local & click git bash

Select the server created & click on connect.
↳ ~~EC2~~

Click on SSH client
↳ Copy the link.

SSH copied link past on git ~~bash~~ bash
do paste & enter.
→ ~~dss~~ yes (write) ↵ enter.

We will get IP address.

Terminal
Ubuntu → apt-get update → window sh
Linux yum
Ubuntu apt
Permission denied

→ sudo -i → root user
↳ apt-get update
Updating....

21/12

* copy public ip & paste in browser

LAMP Server

Linux Apache MySQL PHP

OS Web server DB Assembly lang / browser

* How to install &

Search Google -> install lamp on ubuntu 18.04

OR

Bash file

- apt-get install tasksel ↵
  ↳ Y/N ↵ → Y

Search - Google - install stask shell using LAMP

steps are there ....

code ✓ sudo tasksel install lamp-server [install to lamp]

Do copy in bash file

sudo tasksel install lamp-server

↳ processing (installing)

Install phonix -> install PHP

Check PHP installed or not &

sudo nano /var/www/html/info.php.

copy & pase in git bash.

You get blank nano notepad to print php code then select

```
<& php
  phpinfo();
&>
```

↳ ctrl→x, press y & enter

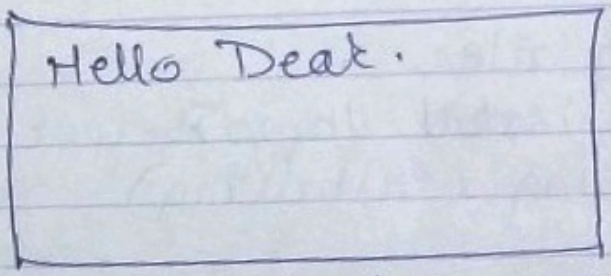
Put in url public ip d

Public ip/info.php

OR

~~chang nano /var/w~~

→ do info.php



You will get browser

Hello Dear

Clone

To create a clone for another server

Server → click Act^n → Image & templates → create image.

Create Image page

Image name → myImageVertex
All data are opt^{nal} don't click anything

Last do click on Create Image

New EC2 Experience

On left side chose → AMI's

AMI → Amazon Machine Image
wait for few minutes

Select → Launce instance

Name — myImageVertex // clone
↳ Choose Ubuntu O.S.
Browse more AMI
↳ MY AMIs
shows you AMI no.
↳ Select

Choose Key previous one

New security

→ All traffic
→ Anywhere

Storage

8 GB

No. of instances → change 2

→ Launch Instance.

Change Name of clone

When we are create clone we can't change it.

* AMI's
  ↳ Actions
    ↳ Deregister AMI

But the data is stored in AMI then it is stored in Snapshot.

left side
* Elastic Block Store
  ↳ Snapshots
    ↳ Actions → Delete snapshot

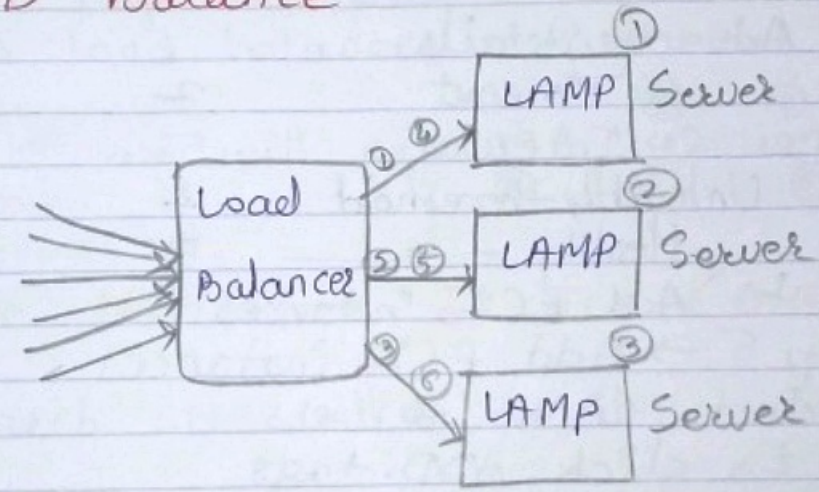
22/12

IP address

Main Server - 199
1st clone - 142
2nd clone - 227

to client
We can't give any key to use server i.e. problem so....
(we are created so many servers we has to provide provide proper service to client).

LOAD Balance



Left side click LOAD Balancing

↳ Create Load balancer
↳ Last line Classical Load Balancer
↳ Click on Create

Step 1 → Define Load Balancer → Not assign security grp
Step 2 → Assign security groups
click ↳ create a new security group

↳ give name for sequrity group.
→ SLB_LB

1> Custom TCP — Anywhere ~~Source~~ Ano

2> All traffic — Anywhere

↳ Next configure Security setting
↳ Next

Step 3 → Configure Security Setting
↳ click next

Step 4 - Configure Health Check

Advance details

Timeout	2
Interval	5
Unhealthy threshold	2
Healthy —11—	5

↳ Add EC2 instances

Step 5 → Add EC2 instances
Select LAMP servers
↳ click ADD. tags

Step - 6 → Add tags
↳ Review & create (click)

Step 7 → Review
↳ Create

Successfully create Load Balancer

To chk LB use private key.

In Load Balancer → click on myLB

✓ Instances

Check status of the all instances.
status must be inService

Click Descript^n

DNS name → do copy & paste URL ↵

Refresh page the public key will be change means load balancer adjust the load on server.

1st we will get ① 199, ② 142, ③ 127
④ ⑤ ⑥

Simple Notificat Service (SNS)

Search SNS click on it

choose name → mySNS
↳ Next Step

1) Create topic
→ Standard ①
↳ Create topic (click)

↳ Create Subscript^n

Subscript

Protocol
↳ choose Email

Endpoint
↳ write email_id

↳ Create Subscript (~~creat~~ click)

go to email address
check inbox or spam
click on AWS link

Finall we get confirm in Subscript

~~Cloud Watch~~

SNS → dashboard

Subcript?

Cloud Watch (to report the all amazon services in server)

Left → Alarms
↳ inalarms
↳ Create alarm (click)
↳ Select metric (click)

Copy instance id & paste in

Metrics(1) - text box.

Click on instanc

Select per instance metrics
↳ select myvertex

Why & when ?

select all files -A
git add -A

vi mygitfile

1> git add mygitfile

2> git commit -m "this is 1st file" -> comment

3> git push

24/12

Auto Scaling

→ Launch Configurat°

→ Create a Launch config..

→ Name - myautoconfig
↳ AMI → Existing AMI
↳ Instance typ → Choose → t2.micro

↳ Security Group
↳ create or ....
↳ Select existing
↳ LB

Rules ~~custom +~~
↳ ① SSH type
② All traffic

Source type
Anywhere

↳ Keypair
☑ Select acknowledge

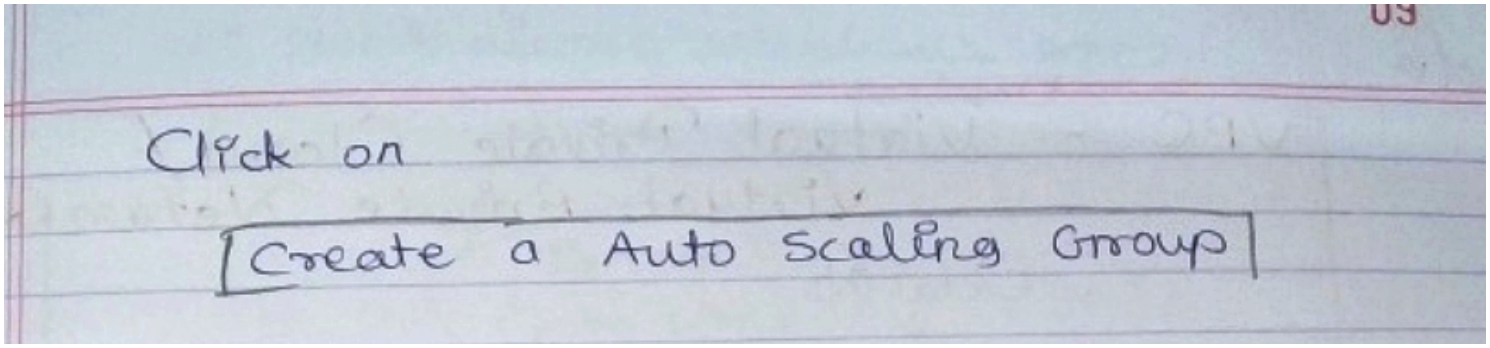
→ Create a launch config (click)
Once we create launch config we can't change it, we want change use
Copy to launch template

Create Auto scaling use
Click on create Auto Scaling Group

1st Name
↳ My Auto Config.
↳ Click switch to launch config.
Select Previous Auto config name → Click Next.
2nd N/w
↳ VPC –
↳ Subnet click all

3rd
↳ Attach to an existing load (choose)
→ Attach to existing
↳ choose from classic load bal
↳ Existing myCLB
Health check
↳ EC2
↳ 100 second
Step 4 ↳ Scaling Policies
② ⊙ target tracking scaling policy
① ○
○
○ Maximum → set 5

5th Add Notifi
SNS topic
→ choose existing
→ Next
6th → Next 7th → Next


09
Click on
Create a Auto Scaling Group

25/12

Net_Id Host_Id

CIDR - ~~Classless~~ inter domain Route No.
Classless

VPC – Virtual Private Cloud / Virtual Private Network

Octal No.

CIDR 192 168.1.95

8 bits ↗ ↑ Router

255.255.255.0

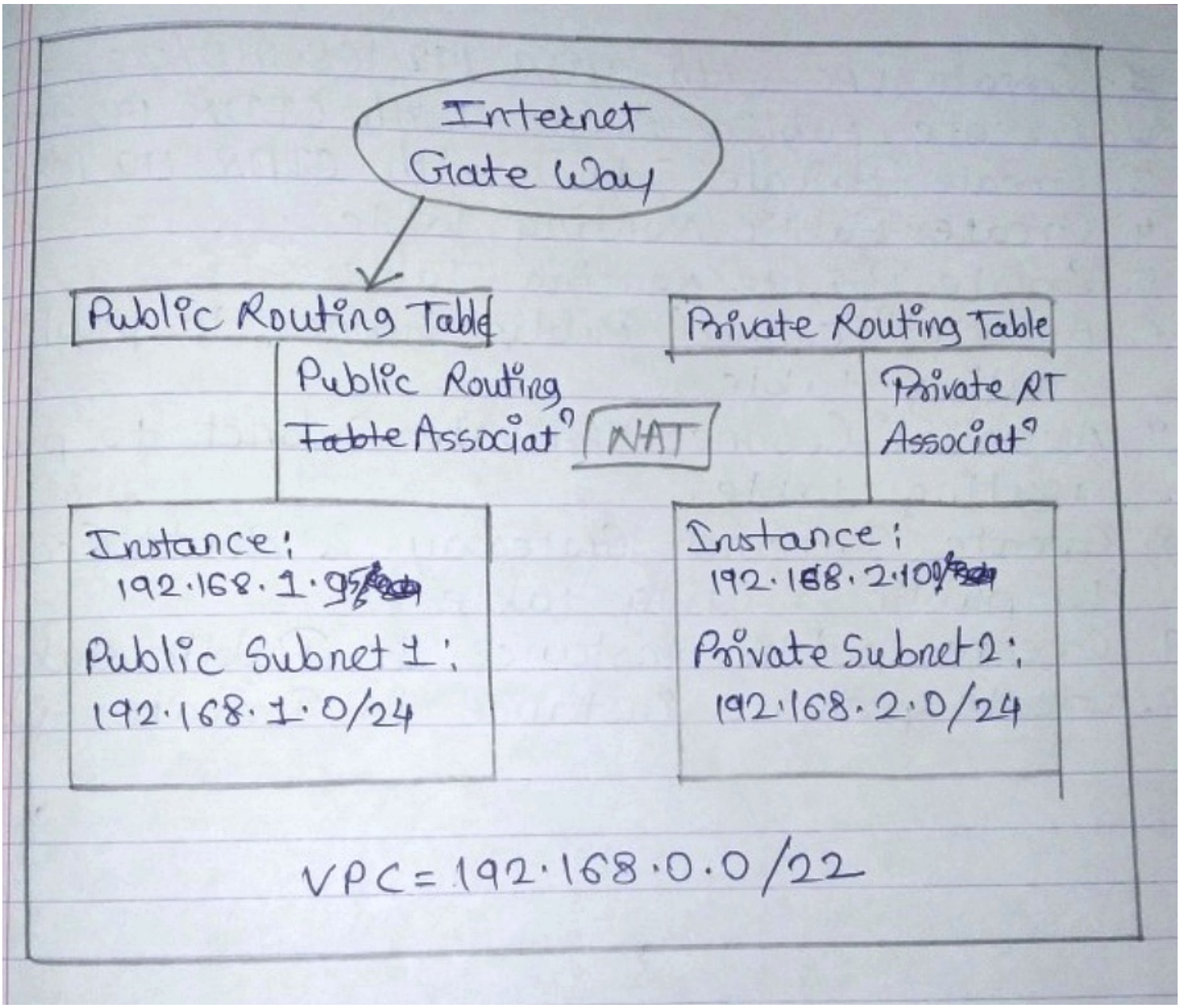
Subnet 1
192.168.1.0/24

Subnet 2
192.168.2.0/24

Subnet 3
192.168.3.0/24

Subnet 4
192.168.4.0/24

VPC = 192.168.0.0/22


Internet Gate Way
Public Routing Table
Private Routing Table
Public Routing Table Associat°
NAT
Private RT Associat°
Instance;
192.168.1.95
Public Subnet 1:
192.168.1.0/24
Instance;
192.168.2.10
Private Subnet 2:
192.168.2.0/24
VPC = 192.168.0.0/22

Steps

1. Create VPC with CIDR 192.168.0.0/22
2. Create Public Subnet with CIDR 192.168.1.0/24
3. Create Private Subnet with CIDR 192.168.2.0/24
4. Create Public Routing Table
5. Create Private Routing Table
6. Associat (Connect) Public Subnet to public routing table.
7. Associat (Connect) Private Subnet to public routing table.
8. Create Internet Gateway & Route (connect) to public routing table
9. Create public instance in Public subnet
10. Create private instance in Private subnet.

VPC Implementation

Search VPC ↵

Select VPC's your
↳ Create VPC
↳ Name - myVPC
↳ IPV4 → 192.168.0.0/22

[Create VPC] ↵

Subnet

↳ Create Subnet → click
↳ VPC IP → ~~PublicSubnet~~ Select myVPC

↳ Subnet settings
↳ Subnet name :- PublicSubnet

↳ IPv4 cidr code :- 192.168.1.0/24

[Create Subnet]

Create Route Table

RT Setting
↳ Name :- MyPublicRT
↳ Select VPC :- myVPC

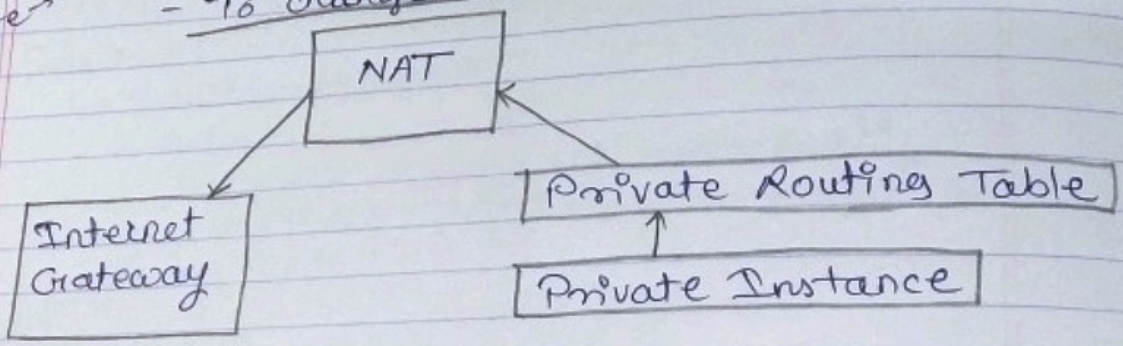
[Create RT]

Private
⇒ cannot connect directly
⇒ It is not connected to IGW

24/12

NAT - N/w Access Translat' (Gateway)
- To change or hide IP address.

Update →



It is chargable

It is used to change or update private database or data through public.

Create NAT

↳ Name → MyNAT
↳ Subnet → Private
↳ Elastic IP
↳ Allocate Elastic IP
(Click) Autogenerated
↳ Create NAT (Click)

Parmanent IP

Routing :
Select
↳ ~~Click~~ Private RT
↳ Select Routes
↳ Edit Routes
↳ Add route
↳ Target NAT Gateway (created)
↳ 0.0.0.0/0
↳ Save changes (Click)

To create a file in Public instance with a name of privatekey.pem ~~boo~~ file created by us. using vi editor

PublicInstance bash file

:

:$: vi myPrivateKey.pem ↵

Open a myPrivateKey.pem open with notepad & copy key all (starting to end without missing any line).

Open vi file & paste it.

To exit vi editor; Escape : wq!

$: ls -la

:

-rw-r--r-- - - - - - myPrivateKey.pem

:

:Here is all read permiss" we can't change or update data inprivate so.. To change change access permission use chmod.

chmod ~~or +x~~ 600 myPrivateKey.pem

OR

chmod +x myPrivateKey.pem

~~#~~ ~~or~~ To connect a server of private subnet

ssh -i myPrivateKey.pem ubuntu@192.168.2.199 ↵

server type — Private ~~key~~ IP of private instance

When I execute previous command then I will enter private subnet through public bash file.

Public IP - 192.168.1.246 → Public Subnet
Private - 192.168.2.250 — Private ——

I entered in 250 private IP in bash like :-

ubuntu ~~root~~@ ip-192.168.2.250 : ~ #

↳ sudo -i
↳ apt-get update
↳ ping ~~ww~~ google.com

↳ exit
$ ↳ exit
Public IP again Public ~~IP~~ Subnet

Deleting Process of VPC N/w (Whole)
Instance → NAT → Elastic IP → VPC

AWS Format version to search google

AWS Template format Version → select 1st link 14
& select copy JSON Template & paste in { } json file

08/12

Visual Studio Code

↳ Download for Windows

Open VSCode

→ Create folder → MyVPC

Open-VSCode
↳ File → Open folder
→ Select

Cloud format to create infrastructure as code
Json & YAML format file

↳ Install extens^n
↳ Search Cloud format
↳ Click & install

File
↳ Right click on folder file & create new file

↳ edit name myVPC.json

Parameters → Region
Resources → VPC, Elastic, Subnet, Routers
Outputs → Dupw Data

Creating VPC in Provising — Visual code

29

```
{
"AWSTemplateFormatVersion": "2010-09-09",
"Parameters" : {
},
"Resources" : {
    "myVPC" {
        "Type" : "AWS::EC2::VPC",
        "Properties" : {
            "CidrBlock" : "192.168.0.0/22",
                } // close of properties
            }// close of myVPC
};// close of resources
"Outputs" : {
}
},// close of main
```

type VPC & Enter, change name

Cloudformation

↳ Create Stack (click)

Step 1
↳ Template is ready

Specify template
↳ ⊙ upload the template file
Choose file
↳ Select file
↳ Stored in S3 bucket

→ Next

Step 2
stack Name
↳ MyStack

→ Next

Step 3 →
Don't select anything

→ Next
Review Data

→ Submit

3/1/23 RDS 17

RDS & Dynamodb basics

Search RDS [Public]

Select RDS
↳ Create a new database
→ Standard create
Engine opt°
→ mySQL select
↳ version - anyone
Templates
→ Product - Live envi
→ Dev/Test - Test. envi
↳ free tire - choose
Availa
→ ⊙ single DB instance

Settings
↳ mydatabaseserver name [DB instance]
↳ admin
↳ Password [mydata]
↳ confirm pwd
~~Allocate storage - 200~~
instance config.
⊙ stadard

- don't change

Storage -
as it is

Connectivity
◉ Don't connect VPC

Public access
✓◉ Yes
○ No.

↳ choose existing

↳ choose mySG or default

DB authentication
↳ previous password [mydata]

Additional config.

↳ myDB — DBase name

↳ [Create database]

SQLyog

Seakh in google SQL yog download

gitup → SQLyog 13.2.0 → download

mySQL Port address - 3306

New
↳ mySQLcomm

Select mydatabaseserver (click)

(Copy) endpoint address

(Paste) in mySQLyog Host

Port - 3306

Click [Test Connect]

If not connected then

{ Security group

Select inbound security rules
↳ inbound rules
↳ Edit inbound rules
Add
↳ Type - All traffic
Source type → Anywhere IPV4

[Save rules] }

SQLyog

[Save] ↵

It is used to connect ~~private~~ public instance through database using MySQL workbench to install DB
~~through which instance hv a IGWay conn?~~
Private RDS

Select RDS
↳ Create a new database
↳ Standard create
Engine opt°
↳ mySQL select
↳ version - anyone
Templates
↳ Product° - Live envi.
↳ Dev/Test - Test or develop. envi.
↳ Free tier - choose
Availability
↳ ⊙ Single DB instance

Settings
↳ myPrivateDB name [DB instance]
↳ admin
↳ pwd [mydata]
↳ comfirm pwd

Instance config
↳ ⊙ standard
:
- dont change

Storage → as it is

Connectivity → If you hv VPC connect o.w.
choose default VPC & check VPC IP add.

Public Access
- ○ Yes You want a public connectivity
- ⊙ No You want private connectivity or subnet
  - ↳ choose existing
  - ↳ choose mySG or default

DB authenticat°
- ↳ previous pwd [mydata]

Add°al config
- ↳ myDBstud → Database Name

[Create Database]

Instance Create

We hv to create a instance which is connected to the VPC which has IGWay conn°.

Create Instance → Eto [Create Instance Name] → choose [Ubuntu OS], 18.04 → t2.micro → Choose Key → No Settings → Choose VPC if exists (chck IPadd)
- Security Grp rule → SSH & Anywhere
- Add → All traffic & Anywhere

Config. storage → 8 Gb...

[Launch Instance]

MySQL workbench

To connect RDS & public instance which has a Gateway.

Search → Install → go community edition
~~OR~~ (if)
MS visual c++ 2019 redistribut°

mySQL workbench → click (+)

Conn° Name → my Conn°
Conn° method → Stand. TCP/IP ssh

Parameter

Instance data
- SSH Hostname : → ~~Ext~~ EC2 instance public IP do paste
- SSH Username : Ubuntu
- SSH Pswd : –
- SSH Key : Choose from create instance.

SQL data
- MySQL HostName : copy endpoint of RDS ~~of~~
- Server Port : 3306
- User Name : admin
- Pwd : ~~Store in vlt~~ → mydata

OR

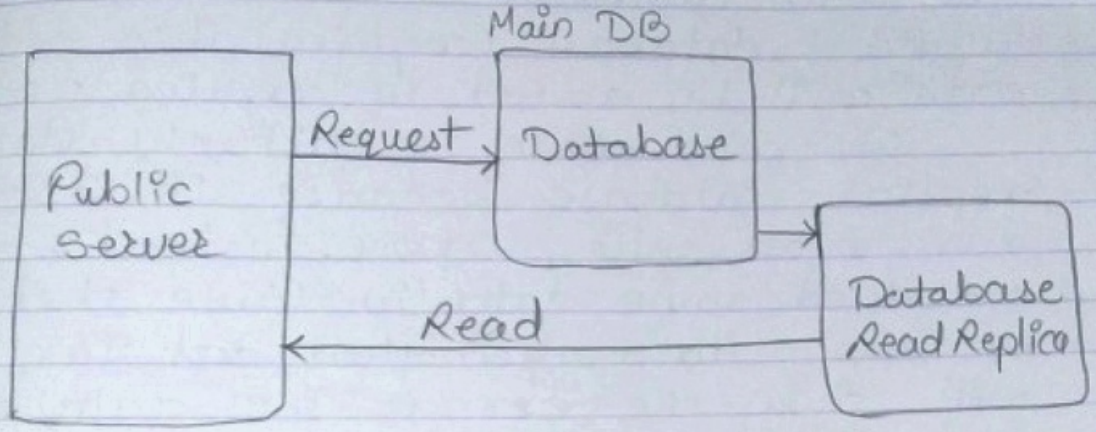
Schemas

Right Click on ~~Mystudent~~ Tables

~~Select tables~~ Select Create Table

Table Name: Mystud schema: Mystudent

Read Replica or Backup Server



RDS → Database → Action → create read replica

Setting

Replica Source :- myPrivateDB → existing or which DB we have to do replica

DB instance identifier

myPrivateDBreplica

Storage

Create Read Replica

We created a database in MyPrivateDB the main database is here when we create a read database (replica) then all updated data will be added in replica read database.

We can't enter or edit the data in replica database because it is created for read only purpose, instead of that we added some data in table then when we apply data then that ~~sqps~~ time system will show the error i.e. database is for read-only.

Note :-

~~If the crash or data~~ ~~database~~ ~~was currupt~~ ;

In some case the main database was currupt or deleted then the replica takes the main database position means the main database is removed or deleted & replica takes place of main database. Then that read only replica became read & write database like main database. Then we can Add, delete data in that replica main database.

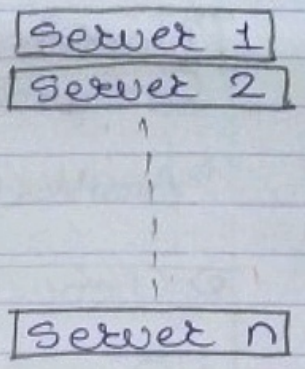
We can create a multiple replicas but its payable. When we create multiple replicas then when main database crashesh then FCFS algo. applied & 1st created replica became main database.

Configuration Management Tool

Manual configurat° processes

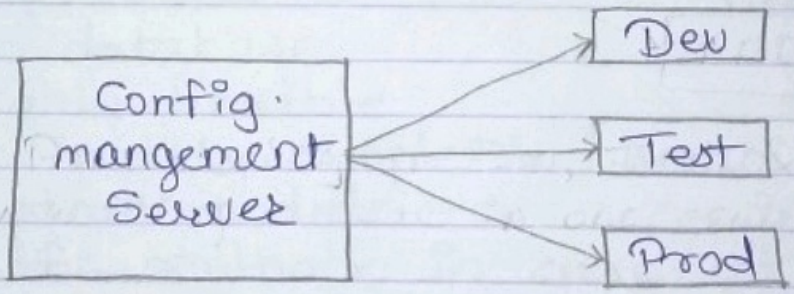
replaced with

automated processes



Tools of CM#

1) Ansible - push model
2) Chef - pull model
3) Puppet - pull model

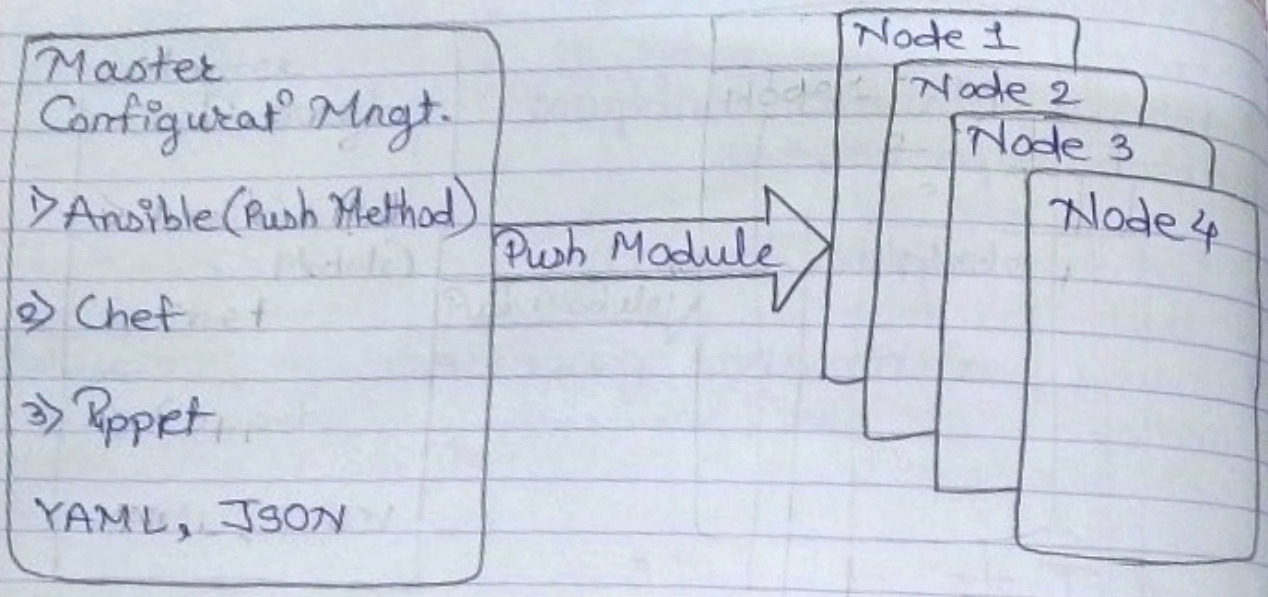


Provision of Dev, Test & Prod envi. by writing a code in one centralized locat°.

[Provisiong means to provide a database, servers, diff. applicat° etc).

Master ⟶ Server → node
no. of nodes

To handle multiple servers or nodes, we use Master.



In master we have diff. tools
1) Ansible 2) Chef 3) Puppet
use YAML

Ansible

Introduct° to Ancible

Ansible is a simple It automat° tool that makes your applications and systems easier to deploy.

It supports in configurat° mngt.

It uses playbooks (yaml) to create Infrastructure as code for deploy appl^rs in various envi. (dev, test, prod).

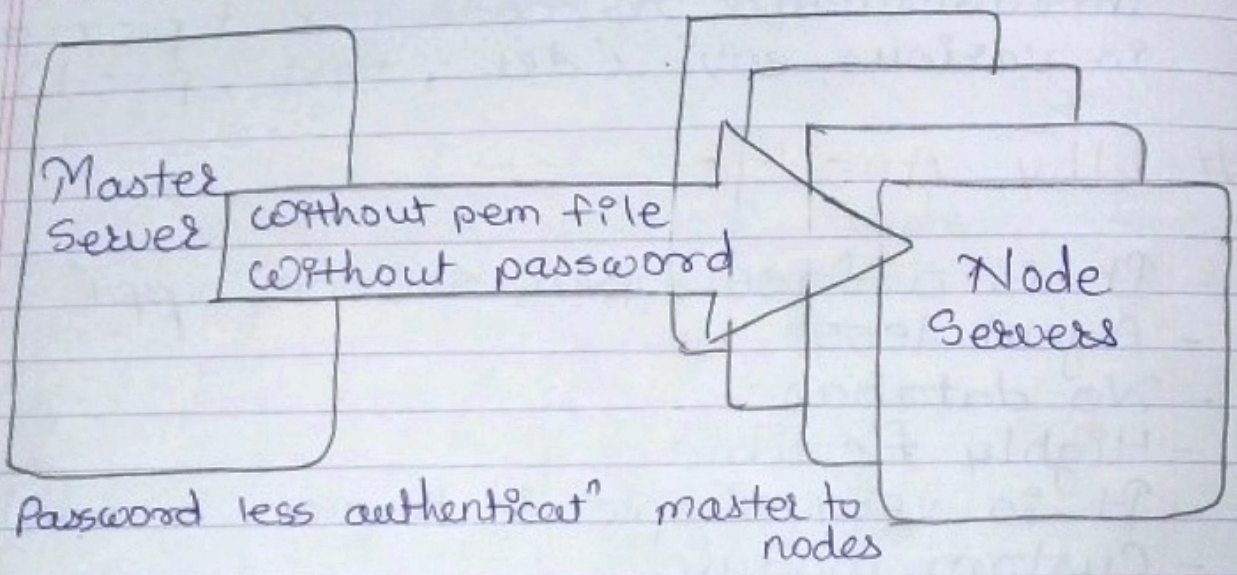
Why Ansible

- It is a free open source appl^
- Agent-less
- No database
- Highly flexible
- It is very simple & human readable
- Custom module
- Confi. rollback in case of error.
- Self documenting

What Ansible does?

- Provisioning
- Change Management - to chnge version
- Automation
- Orchestration - to tell master to change or updated
- Config. of servers - you can install diff. packages
- Appl° deployment

Passwordless authenticat°



Password less authenticat° master to nodes

Stage 1 : Master server config.

1) Create aws ec2 server
2) Create user
   # adduser malu } bashfile

3) Make user as sudoer
   # visudo
   malu ALL=(ALL) NOPASSWD: ALL
   Ctrl x, press y, press enter

4) Change password authenticat° yes
vi /etc/ssh/sshd_config
Passwd authentication (yes) Press i
~~ente~~ type yes
esc : wq!

5) Restart ssh service
$ service ssh restart

Stage 2 : Node Server Configuration

6) Create aws ec2 server
7) Create user
adduser malu

8) Make user as sudoer
visudo
malu ALL=(ALL) NOPASSWD: ALL
ctr + x, press y, press enter

9) Change password authenticat° yes
vi /etc/ssh/sshd_config
Passwd authentication yes
esc : wq!

10) Restart ssh service
$ service ssh restart

Stage 3 : connect from master to nodes without pem and passwd.

ON MASTER AS SUDOER (maha) USER

switch user

11) # su maha
12) #$cd
13) #$pwd - Present working directory. home/maha
14) #$ssh-keygen dont use any key
15) $ ssh-copy-id <private ip nodes>
16) $ ssh <private ip nodes> → exit

we once

We can't connect from root user we have to connect master user or node

To check ansible version

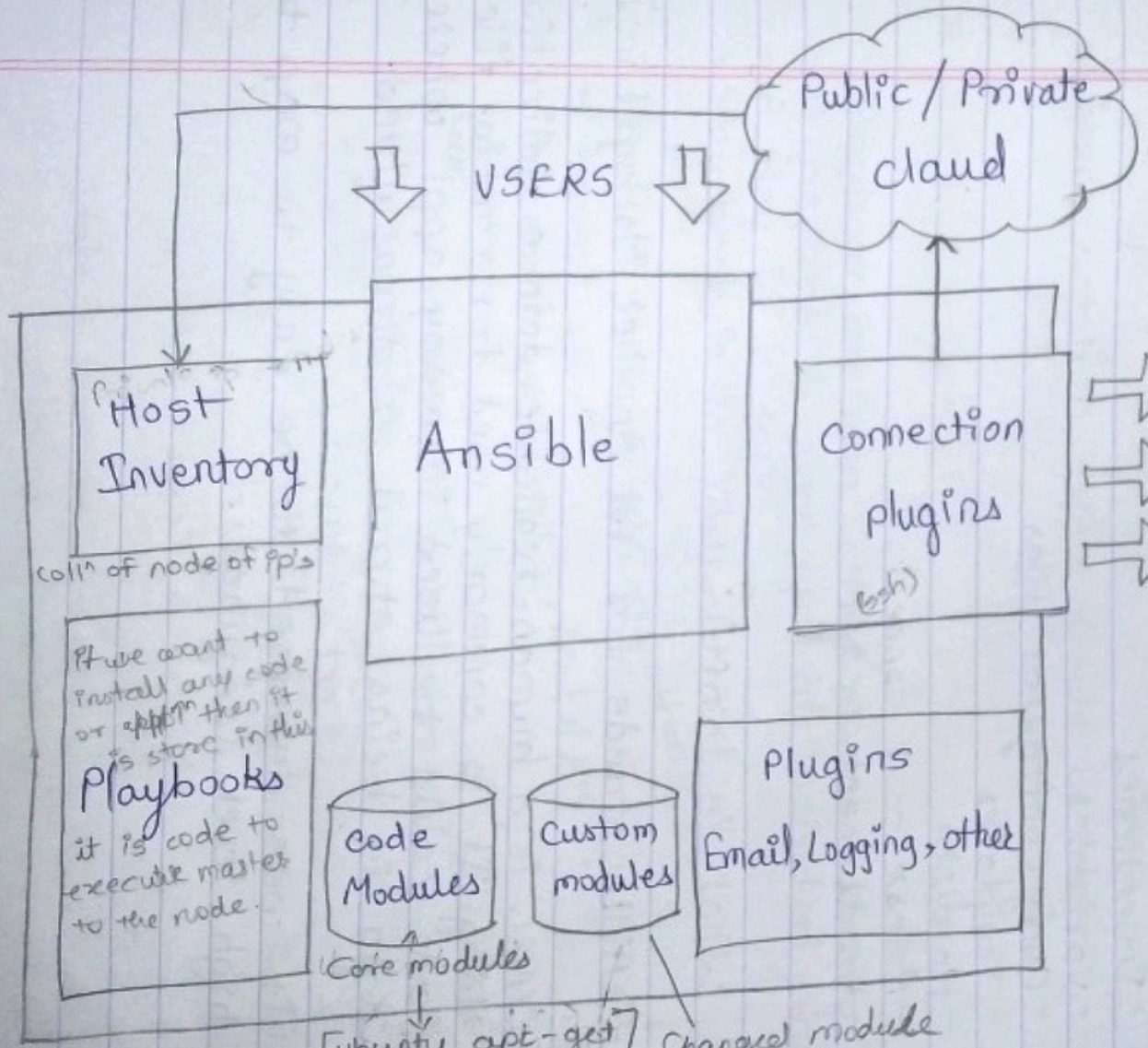
ansible --version

node

to connect public or private to use ssh conn

User → Ansible using public or private to use ssh conn° to connect node

Architecture of Ansible



What are all core components? (Agenda of Ansible)

- Ad-hoc commands
- Playbooks
- Modules
- facts
- Inventories
- Variables
- ~~Config~~ Configurat° files
- Templates
- Handlers
- Roles
- Vault

What is YAML?

- YAML stands for Yet Another Markup Language
- YAML is a human-readable data serialization lang. It is commonly used for config. files, but could be used in many appl^ns where data is being stored or transmitted.
- It was designed to be easy to map to high level languages.

Rules of YAML file

1) .yaml or yml

2) --- - start with 3 hyper

3) Key: values - every value in a key value pair

4) Space notation - space should be proper

Ansible Playbooks

- Playbook are your instruct manual. master to node
- A playbook is made up with individual plays
- A play is a task
- Play book is in YAML format.

```
---
- hosts: all - allow all private ip address
  become: yes - install any kind of packages
  tasks:
      -name: install tree  package name
       apt:
       name: tree
       state: present means install
```

It is ubuntu to install

state: present - install
absent - unsistall

To create inventory file using vi cmd

$ vi myhost
↳ do copy & paste of private ip of node.

To write playbook

$ vi mysample.yaml
↳ edit file using yml code & press i to insert text.

```
---
- hosts: all
  become: yes
  tasks:
    - name: I want to install apache2
      apt:
        name: apache2
        state: present
```

Every task should start with -

state: present / installed / latest
absent / remove

To check master is communicated with node?

$ ansible all -i myhost -m ping
(-i: inventory, -m: argument)

We get success output
If we have wrong ip address in myhost file then this stat will show you error.

Yellow - for change
Red - for error
green - for already installed

Execute Playbook

$ ansible-playbook -i myhosts [mysample.yml]

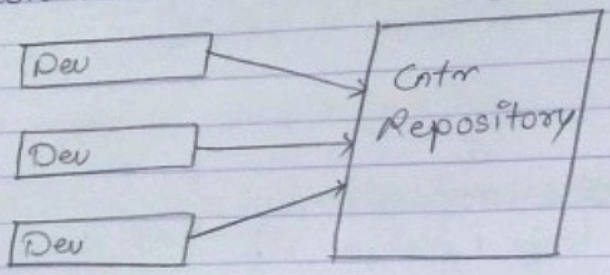
(-i : inventory, myhosts : ip address, mysample.yml : playbook name)

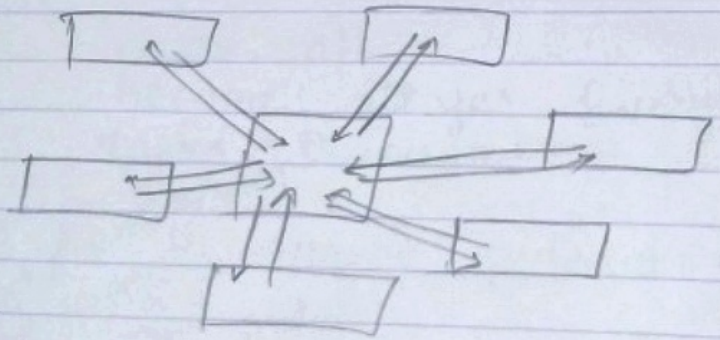
github is a code repository.

12/8/23

Git

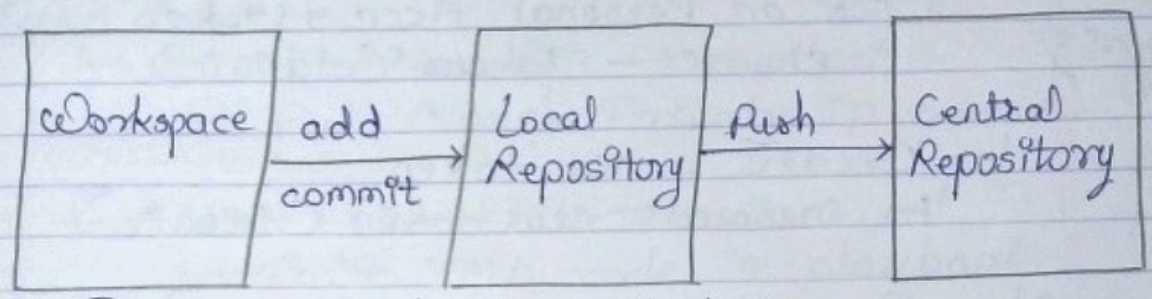
It is version control system or distributed ~~cont~~ version control sys.





Types of Git or stages of Git

1) Workspace – working space
2) Local Repository – which file we hv to send in main we add init
3) Central Repository



Distributed Version Control Sys

Google → Github – GitHub site click

2nd # Create new repository

↳ create repository name ; myDevOps6pm

Descript' : optional

✓ ○ Public . It shows all data to world
○ Private is it private but payable

Create Repository

1st → to generate a token

Go to profile → settings

left side → Developer settings - click

Click on Personal Access token
choose - Tokens (classic)

↳ Create a new token
↳ Generate new token (classic)

New Personal Access token (Access)

Note (name) :- mytoken
Expirat° → choose expirat° → No expirat°

Select scopes : → Select all scopes

[Generate token]

* Save token ip in any notepad file.

Create a normal folder → MyProjFile

Open folder right click open git bash

*Do copy & paste one by one line execute in git bash.
If will create a git files three command.

Writing a code in VSCode & push git into repository

Visual Studio Code

Install YAML attachment in VSCode.

Open folder

↳ RClick → New file → myHost
↳ Put Node's Private Ip → save } inventory file

↳ RClick → New file → mySam.yml
↳ Write YML code in playbook

Save ↵

Send YML file in github

↳ Create new repository github

:

- ✓ Create Repository
- ✓ Create Token

↳ Select Terminal in VS code → To comm^ server & node.

↳ Select each line in repository & do paste in VSCode terminal.

↳ then
```
git add -A
git commit -m "Next"
git push
```

-rf → remove forcefully directory.

then in Mashter bash file

→ su mala
→ cd
→ pwd
→ ls
→ check path ✗
then
do clone

To store or save file in github to server then click code & copy the HTTP url link & do paste in server Master bash file.

↳ $ git clone [http url of repository of github]

[MyNwRepository → Code → HTTPS → copy code]

↳ check ls → You will get MyNwAnsible Repository folder

↳ (do) cd MyNwAnsible

↳ check ls
README.md

You will get only one file in MyNwAnsible Repository folder then you have to execute 3 cmds in VS code terminal.

VScode terminal

↳ git add -A
↳ git commit -m "next"
↳ git pull

then in Master Bash file

↳ malu ... ~/MyNwRepository $ ls ↵

You will get now 3 files
1) myHost
2) mySam.yml
3) README.md

then we will execute the ansible code

malu ... ~/MyNwRepository $ ansible all -i myHost -m ping

↳ shows success of execut°

—||— $ ansible-playbook -i myHost mysam.yml↵

↳ Shows you gathering facts & installation of apache2

16/1/23

Idem potens → To gather data or to skip existing one

Ansible facts

In Master bash file to check server family
It may be ubuntu, redhat

ansible all -i myHost -m setup -a "filter=*family*"

↳ Debian family

To check the family of server node when creating of instance we choose O.S type so the O.S type is diff then we can find the family of O.S & change only for Debian, so it is used.

Put the code in mySam.yml last line after state

apt

—

—

when: ansible_os_family == "Debian"

Create new instance

→ nodeRed_Demo

Choose OS Amazon.aws (Linux)

Copy public ip of nodeRedDemo put data in bash file of new node

↳ Yes to continue

↳ ~~yr~~ yum update

↳ sudo -i

↳ & yum update

Masterbash

master user मध्ये असताना new node add करता येते
root user वरून new node add करता येत नाही.

↳ adduser malu
↳ type passwd malu
↳ enter pass
↳ visudo
↳ malu ALL = (ALL) NOPASSWD: ALL
↳ vi /etc/ssh/sshd_config
↳ yes
↳ service sshd restart
AWS O.S. linux use sshd

Connect master to node

Already we are in root user of malu

* Don't run ssh-keygen
If you execute ↑ disconnect all node from master

open master bash file su malu → cd → ssh-copy-id <Private ip of node>
↳ yes
↳ put password
↓
No. of key(s) add: 1

↳ ssh <private ip of node>
↳ o/p means Master successfully executed in node

Do copy of private ip of new RedHat OS node into VScode myHost file 2nd ip.

↳ To send data to git use VScode terminal
↳ git add -A

Only 3 - add, commit & push are used in VSCode terminal

↳ git commit -m "RedHAT"

↳ git push

↳ Go to master bash file

↳ malu - - - - $ cd chck path

↳ cd myAnsible

↳ git pull

To check master is connected to REDHAT OS node

↳ ansible all -i myHost -m ping

To check family of node

ansible all -i myHost -m setup -a "filter = *family*" ↵

↳ we will get family of REDHAT OS.

↓

Shows you "Ansible_os_family": "Redhat"

* To execute playbook mySam.yml file in ubuntu statement

ansible-playbook -i myHost mySam.yml ↵

When we r executing this stat cmd we will get o/p → skipped - Redhat ip address of node
ok - ubuntu ip address of node

Sat 4/23

Ansible facts

- Ansible facts is a way of getting the info of Hosts (nodes).
- 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 hosts. Ansible provides many facts about the sys, automatically.
- This module is also supported for windows targets.
- It's not always required.

$ ansible all -i myhosts -m setup -a "filter = *family*"

I want to install webserver on both ubuntu & redhat by same playbook.

Ansible Modules

- Ansible with a no. of modules.
- Modules are discrete units of code that can be used from the cmd line or in a playbook task.
- Ansible executes each module, usually on the remote target node, & collect

return values.
- Each module supports taking arguments.

Nearly all modules take

key = value arguments, space delimited.

- Some modules take no arguments, space & the command / shell modules simply take the string of the command you want to run. ~~yum~~

yum
apt
service
Get_url
etc
And we can write own module.

Custom Inventories

- Static — fixe
- Dynamic — changable

Static Inventory

[mywebservers]
public add (If diff. N/w or diff. cloud)
public add (if same n/w)
public DNS name (if diff. n/w or diff. cloud)
public DNS name (if same n/w)
too example.com

[mydbservers]
one example.com
two example.com

Inventories are 2 types

1) Default inventories
2) Custom inventories

Installing Java JDK

Open master server & copy ssh id & past in bash file i.e. run master ~~file~~ bash file.

```
: sudo -i
: su maku
: cd
: ls
: cd MyNwRepository
```

In VSCode
↳ Create new file
↳ myjavafile.yml ↵

Code in next page

myjavafile.yml

```
---
- hosts: all
  become: yes
  tasks:
    - name: I want to install java
            jdk 8 on Ubuntu server
      apt:
        name: openjdk-8-jdk
        state: present
      when: ansible_os_family == "Debian"

    - name: I want to install java jdk 8
            on RedHat server
      yum:
        name: java-1.8.0-openjdk
        state: present
      when: ansible_os_family == "RedHat"
```

Search in google to install open java jdk 8 on ubuntu & RedHat.

1st in Ubuntu

→ openjdk-8-jdk

2nd in RedHat

→ java-1.8.0-openjdk

} installing links for java in playbook.

Write these name links in myjavafile.yml link path.

Save the file;

In terminal again do 3 steps

```
$ git add -A
$ git commit -m "jdk"
$ git push
```

In master bash file

```
~ MyNwRepository $ git pull ↵
```

Check — $ ls

```
... $ ansible-playbook -i myHost ~~myjavafile.yml~~
        myjavafile.yml ↵
```

Execution of above statement we can see

Installing

Ubuntu — openjdk-8-jdk

RedHat — java-1.8.0-openjdk

Means

We can install specific packages on specific server using families.

24/1/23

Playbook Variables

↳ create new file
myplaybookvars.yml

myplaybookvars.yml

```
---
- hosts: myweb
  become: yes
  vars:
        myvar1 : tree
        myvar2 : git
        myvar3 : wget
  tasks:
     - name : i want to display variable value
       debug:
            msg : "{{ myvar1 }}"
```

Terminal

↳ git add -A
git commit -m "vars"
git push

Master Bash file

malu ....ν,

Playbook variables installing no. of packages in only one task

```
---
- hosts: mywebserver
  become: yes
  vars:
      myvar1: tree - file structure shows in tree format
      myvar2: git
      myvar3: wget - installing the s/ws directly from internet
  tasks:
    - names: i want to display variable value
      debug:
          msg : "{{ myvar1 }}"


    - names: i want to install tree, git, wgit
      apt:
          name: "{{ item }}"
          state: present
      when: ansible_os_family == "Debian"
      loop:
      - "{{ var myvar1 }}"
      - "{{ myvar2 }}"
      - "{{ myvar3 }}"
```

Linux Commands

Master bash file - [Server]

$ hostname -f

VSCode

↳ New file — mylinuxCmd.yml

- hosts: all/mywebser/individual ip 172.31.20.33
  become: yes
  tasks:
    - name: i want to ~~dir~~ execute linux cmd
      command: hostname -f
      register: myResult

tasks to execute

    - name: i want to display linux output
      debug:
        msg: "{{ myResult }}"

O/p

It shows the ip & zone of user hostname
we will get stderr = " "
&
stdout -
shows - path of hostname -f cmd o/p

Create new playbook in VScode
↳ mydebug.yml ↲

```
---
- hosts:
  become: yes
  tasks:
      - name: i want
```

```
    - name: I want to install tree
      apt:
          name: tree
          state: present
      when: myResult.stderr == " "
                               space

    - name: I want to install wget
      apt:
          name: wget
          states: present
      when: myResult.stderr != " "
```

O/P

skipped - wget

we changes in mydebug.yml hostname error then shows fatal error.

Based on 1st task result i.e (myResult) 2 tasks we can execute

Tags

Create new playbook
↳ mytagos.yml ←

```
---
- hosts: ip address
  become: yes
  tasks:
      -name: I want to install tree
       apt:
           name: tree
           state: present
       tags: myTree
      -name
       apt:
           name: wget
           state: present
       tags: myWget
      - name:
       apt:
           name: git
           state: present
       tags: myGit
```

In Master bash file

```
git add -A
git commit -m "
git push
```

Master bash file
↳ git pull

$ ansible-playbook -i myHost mytags.yml
--tags "myTree"

It will shows you only one task ~~mytr~~ i.e. Tree at a time if you want to execute all then

$ ansible-playbook -i myHost mytags.yml
--tags

[If we want to skip one task then]

$ ansible-playbook -i myHost mytags.yml
--skip-tags "myTree"

Ignore for ex. [name: wget]

If there is any error in playbook then o/p shows the error & failed=1.
Then ~~we~~ ansible have a facility to ignore the specific task.

```
- name: I want to install wget
  apt:
      name: wget
      state: present
  tags: myGet
  ignore_errors: yes
```

then we will get o/p error with ignore keyword → failed=0

Ignore doesn't shows errors in o/p

Group Variables

Create new file in VS code

→ @ group_vars - same as it is

↳ R Click on group_vars & create file

↳ When you give a file name use as it is a group name.

like [mywebserver] file name should be same as of group name.

Group file uses =

group_vars uses :

Host variables / node variable

↳ create new folder → host_vars

→ create new instance put ip in myHost inventory file

Click on host_vars → click new file private & file name must be a new node ip address.

↳ put variable name with jdk 8 path

[In ansible host variable has highest priority than the group variables].

Open jdk 8

ansible-playboo -i myHost -v myjavafile.yml

verboss - to show full details / basic

-vvv - full details

To chck java version

private node bash file - java -version

Ansible Roles

Role can be thought of as a playbook spit into multiple files.

The multiple files create as follow folders - ↵

- files:
- handlers
- meta:
- templates:
- tasks:
- vars:

Master bash file

malu--~ m

$ ansible-galaxy init myWebRole ↵

(init = initialize; myWebRole = name of role)

We are not installed tree in master node then we have to install tree in master server

(root permissions)

malu $ sudo apt-get install tree ↵

To view in folder (tree) structure use

$ tree myWebRole ↵

if you are using ls in myWebRole the it shows only list.

To Ansible galaxy role create role-based structure in ansible master but to edit or make any changes in a role we have to take that file to local laptop (for ex. sys. VScode) we use git for that.

Master bash file

make - - - $ git add -A
$ git commit -m "Role"

Same prodcedure for 1st time
email id of ~~use email id of Repository [Github]~~
owes
~~& token id which is stored in notpad.~~
& ~~email~~ id name of Repository [Github] ↵

$ git push

⟶ id — Repository id name
- token of repository which is saved ↵

In VScode terminal

$ git pull ↵

O/p created myWebRole folder in myNewRepository

Create new ~~folder~~ file in myNewRepository.

↳ myWebRolefile.yml

```
---
- hosts: mywebserver
  become: yes
  roles:
    [- { role: myWebRole, when: ansible_os_family == "Debian" }]
```

In myWebRole select tasks → main.yml

```
---
- name: I want to install apache2
  apt:
    name: apache2
    state: present
```

In terminal

```
$ git add -A
$ git commit -m "RoleTask"
$ git push
```

Master bash file

```
$ git pull
$ ansible-playbook -i myHost myWebRolefile.yml
```

When we executing this file Debian O.S or servers are installed apache2 & RedHat O.S. skipp this yml file.

In Master ~~bash file~~

Search in google ansible vault ↵

Ansible Vault link ⇐

↳ To create datafile

from google

ansible-vault encrypt myWebRole/vars/main.yml ↵

New vault password: mulu123 - don't forget

Confirm new - - - -

Encrypt? Successful

unzip for - xzvf

30/1/23

Manual Tomcat Installation

Search Google

→ tomcat installat° in ubuntu 18.04 ↵

Select [DigitalOcean]

Install Apache Tomcat 9 on Ubuntu 18.04

Create a new instance for install tomcat

→ name - TomcatServer
- Ubuntu - 18.04
- T2.micro
- Give Key
- Security group - Add
  All traffic
  Anywhere
- Chk storage - 8
- No. of instance

Launce Instance

Connect TomcatServer instance to the server
Copy ssh path &
Open in bash file ↵

- sudo -i
- apt-get update

then Do copy paste all step one by one in the site of Digital Ocean link

Step-1 - Install Java

① sudo apt update ↵

② sudo apt install default-jdk

↪ yes to continue

Step 2 - Create Tomcat User

① sudo groupadd tomcat

② ----

Step 3 - Install Tomcat

①

To check container details → docker inspect myc1

23/2/2023

Docker Network

root ~ # docker network ls — [N/w present in root] [list of n/w available] ↵

[Shows N/w list] - delete existing container

Create container

docker run --name myc1 -d -p 8081:8080 [imageName]:1 ↵
container
~~image~~ created
same for myc2 ↵

2 container created myc1 & myc2.

~~In container myc1~~

To check container details → docker inspect myc1

✓ myc1 ip address 172.17.0.2 [store in NPad]
✓ myc2 —"— 172.17.0.3

To go inside the container
docker exec -it myc1 /bin/bash

... /webapps # cd ↵
... # ping 172.17.0.3 ↵ ping from myc1 to myc2 using ip only ~~myc2~~
shows pinging process. ctrl z → stop process

container communicate with bridge

N/w topology for bridge

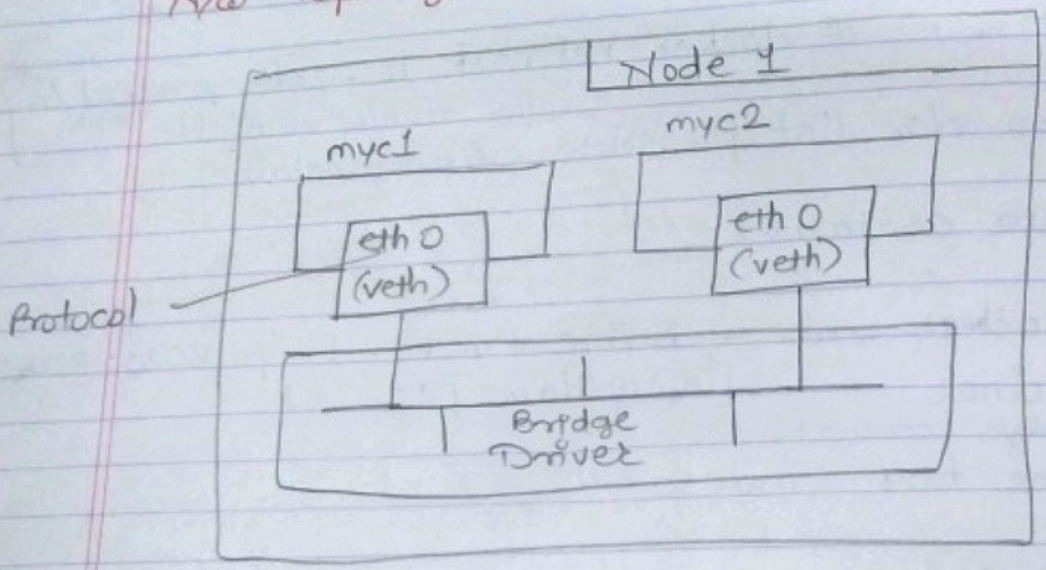


fig :-

Create ur own n/w

bridge

docker network create -d bridge --subnet 192.168.1.0/24 maliNw

created n/w

docker network ls ↵

shows list of n/w

Delete existing container

docker rm $(docker ps -a -q) -f ↵

nginx – by default image name

⊛ Create new container using created n/w

docker image
copy img name

docker run --name myc1 (-d) detach --network maluN/w [imageName] : 1 ↵
shows process of creating container
Created container

docker run --name myc2 -d --network maluN/w [imageName] : 1 ↵
2nd container created
docker inspect myc1 ↵ you will get ip address
✓myc1 → 192.168.1.2 → 192.168.1.(1) # takes bridge by default
✓myc2 → 192.168.1.3

To ~~insert~~ inside the container ~~or~~ (in)
docker exec -it myc1 /bin/bash ↵
~~root@ webapps # cd ↵~~
root@ #
If there are show any error regarding ping not exist then install ping.

then
in myc1
✓# ping 192.168.1.3 ↵
shows pinging process
✓# ping myc2 ↵
shows pinging process

If we have existing n/w we can't ping using container name. But we use our own n/w then we can ping with ip as well as container name.

24/2/23

~~⑧ Docker C Group~~

Docker files

CMD & Entrypoint

vi Dockerfile

FROM ubuntu:14.04 (base image)
MAINTAINER malu
RUN apt-get update
} These 3 lines executed while image is ~~creat~~ executed

[CMD ["echo", "This is my 1st container"]]

Only CMD comand is created in container, when cmd is executed then container stoped.

When we don't want to give any container or ditach then —

docker build -t myImage:1 . ↵

Run myImage

docker run myImage:1 ↵ stop execut ctrl c

CMD ["ping", "google.com"]

CMD cmd can be changed in run time.
like—
docker run myImage:1 ls ↵

It shows all list of files in myImage:1. But ls cmd is replaced in Dockerfile CMD command ["ping", "google.com"] to ls.

If you executed #docker run myImage:1 then you will get updated CMD cmd in Docker file.

So, ~~overcome this pro~~ if you want to don't change ping then use ENTRYPOINT in Dockerfile.

vi Dockerfile

```
FROM ubuntu:14.04
MAINTAINER nitin
RUN apt-get update
ENTRYPOINT ["ping"]
CMD ["google.com"]
```

~~Run Docker Image~~

If you change Dockerfile then you have to create new image.

* # docker build -t myImage:2
* # docker run myImage:2

It will show you pinging process of google

* # docker run myImage:2 ls — ping: unknown host ls

this cmd is not executed because in this cmd is not executed because in Dockerfile we wrote Entrypoint is ping,

docker rmi $(docker images -q) -f

so only ping cmands are executed like yahoo.com, rediff.com etc. only at run time.

like :—

docker run myImage:2 yahoo.com ↵

Shows you pinging process of yahoo.

But the above cmd is executed only run time.

docker run myImage2 ↵

It shows process of pinging google.com which is CMD in Dockerfile.

25/2/23

Docker 'c' Group / control group

* Create 2 container

cont 1 # docker run --name myc1 -d nginx

Cont. 2 # docker run --name myc2 -d nginx

docker stats ↵

Shows - container id, name, CPU%, mem usage, Limit, memory...

	Mem Usage / Limit
myC1	2.261 MiB / 974.1 MiB
myC2	8.074 MiB / 974.1 MiB
	1.8 GB ← 1948.2 / 1024 MB

* When we create container by default its allocate memory limit is 474.1 MiB for myc1. same as myc2 then the total of both memory space is approx 1.8 GB.

But, when we create instance then that time we take only 1 GB RAM. So, when we create container there are no any limital for memory storage for container.

To control that storage or memory limit/restrictions, we use Docker 'c' / control group.

We can manage docker resources with Cgroup
~~- Memory~~
- Memory
- CPU
- Disk I/O (Volumes)
- Network
- etc

* By default, a container has no resources constraints (restrictions) & can use as much of a given resources as the host's kernel scheduler allows.

* But, we can control (restrictions) memory, CPU, disk, network for a container by cgroupos.
while Create a container by "docker run" command.

Memory

There are 2 basic types of memory.

* RAM is first type of memory:

Random Access Memory (RAM), is used to store data and very speed I/O, RAM is volatile memory, the data stored in RAM is lost if the server is turned off.
Hard drives are magnetic media used for long-term storage of data and programs.

Magnetic media is non-volatile; the data stored on a disk remains even when power off.
RAM vs HHD (ROM).

* Swap space is the second type of memory

The primary funct° of swap space is to substitute hard disk space for RAM memory when real RAM fills-up and more space is needed.
The total amount of memory in a Linux computer is the RAM plus swap space and is referred to as virtual memory.

Total memory = ram + swap

Total memory = 2 GB ram + 4 GB swap from HHD
= 6 GB

(X) After execut° of docker stats ← (74 page)

→ # vi /etc/default/grub ↵

update-grub ↵
#

- add the foll° two key-value pairs :

Edit

vi file

GRUB_CMDLINE_LINUX = "cgroup_enable=
memoryswapaccount = 1" ↵

update-grub ↵
reboot ↵ wait for few minutes

when system reboot sys was logged out, wait for few minutes.
Do same from 1st means copy ~~xxx from~~
ssh-path from instance do copy in bash file.

:
:

#sudo -i
#cd myDocker
#
...myDocker# docker ps -a

Shows container list but when system rebooted then all containers also automatically stopped it.
How to start container we will learn later.

We can control memory by below options

1) -m or --memory=
it represents. The maximum amt of memory the container can use, min 4mb.

2) --memory-swap*
it represents, the total amt of memory and swap also

3) --memory-reservation
it request additional memory, after hitting in to memory limit, to preventing service outages.

Expl :- By default, containers can use as much memory and swap memory as need.

docker run myimage:1

After reboot →

check

docker images ↵

shows list —

then

→ # docker run --name myCont1 -d -m 300M nginx (myimg1) ↵

docker stats ↵

Shows list

→ Cont. Id Name CPU% MemUsage/Limit Mem% ----

---- myCont1 0.00% 5.594MiB/300MiB

docker run --name myCont2 -d -m 300M --memory-swap 1G nginx (myimg1) ↵

docker inspect myCont2 ↵

Shows all data of myCont2 in detail.

Check
Memory - 3145728800/1024 → Convert into KB's
~~Memory-swap~~
307200/1024 → convert into MB's
You get 300 MB

docker run --name myCont3 -d -m 300M
--memory-reservation 100M nginx ↵

docker inspect myCont3 ↵

Check - memory - - -

memoryReservat° -
memory-swap -

26/2/23

CPU

for CPU manage in docker Grub cmd or vi file also used.
for ex → vi /etc/defaults/grub ↵

Execute cmds for CPU

~~# docker run --~~

--cpus=<value> :

Specify how much of the available CPU resources a container can use. For instance, if the host machines has two CPUs and you set --cpus="1.5".
Specifies the percentage of available CPU resources a container can use
Exp:
docker run --name c1 --cpus=.5 -m 300M -d ~~-p 8080~~ nginx ↵

--cpuset-cpus :

Limit the specific CPUs or cores a container can use. On a 4 core host machine we can specify 0-3. We can specify a single core or even multiple cores.

Exp :-

docker run --name myCont4 --cpuset-cpus=0 -m 500M -d nginx ↵

docker stats ↵

... name	CPU%
myCont4	.50

docker inspect myCont4 ↵

Chk - "Memory" - 314572800,
"NanoCpus" - 500000000,

26/2/24

Docker GUI

Docker GUI means we are ~~are~~ using docker in git-bash terminal but in docker Portainer.io provide us a docker GUI for ~~develo~~ DevOps engineers to do or create all tools in docker using GUI.

For that we have to execute below cmd in bash file.

#
docker run -d -p 9000:9000 --restart always -v /var/run/docker.sock:/var/run/docker.sock -v /opt/portainer:/data portainer/portainer ↵

Then goto AWS console
↳ EC2
↳ Instance
↳ Select instance
↳ copy public ip
↳ paste in browser & write port no.
↳ .....:9000 ↵

Then site will open of Portainer...

New Portainer Installat°
Username : admin — bydefault
Password : Jagmalu@2024 ↵
confirmpwd ↵
Successfully Create User

When login successfully

Click [Home]

↳ Shows you right side link click it
↳ Scroll down click image
↳ Build a new image
↳ name: mazing
↳ write Dockerfile code
⋮
↳ Finally click Build Image

Shows you Successfully build Image

Check bash file, the mazing is created or not

docker images ↵
Shows you.—
- - - mazing - - - -

Deploying Java file (1)

Start - instance
Open git bash
→ del all images - do
docker rm $(docker images -q) -f↵
check docker images
del all container
docker rm $(docker ps -a -q) -f ↵
check docker ps - a

Change vi Dockerfile

FROM tomcat:8.5.37-jre8
MAINTAINER malu
RUN apt-get update
COPY
wait !wq b ↵

... myDocker # apt-get install wget ↵
existing shows u msg.

— go in class room link in whatsapp
click comment of 18th feb do click
open that web site - scroll down
U will get SampleWebApp.war file
select that file right click on it copy link address

.... myDocker # wget [do paste here] ↵
——"—— # ls ↵
shows Dockerfile SampleWebApp.war
Copy war file name & past in vi Dockerfile

Vi Dockerfile

```
COPY SampleWebApp.war /usr/local/tomcat/
                      webapps ↵

EXPOSE 8080
CMD ["catalina.sh", "run"]
```

SampleWebApp.war file we are downloaded by using wget we are use copy cmd.
That file are in war file but we want that file in container locat" i.e. /usr/local/tomat/webapps.
Tomcat exposes in 8080
If we want continuously run then use catalina.sh in CMD

Then Create image in bash

```
# docker build -t myimg:1.↵

# docker run --name myc1 -d -p
        8081:8080 myimg:1 ↵
```

Go to instance do copy of public ip paste in browser url
for ex. 18.118.102.254 : 8081 port not given by us↵

You can see Apache tomcat page & also check
--- :8081/SampleWebApp/ ↵
--- Deploy and Test

S3 bucket use for our own war files.
war file stored in S3 bucket.

chnge vi Dockerfile

```
FROM tomcat:8.5.37-jre8
MAINTAINER mahi
RUN apt-get install
ADD [path of S3 bucket file] /usr/local/tomcat/webapps
EXPOSE 8080
CMD ["catalina.sh", "run"]
```

docker images

docker build -t myjavapro:1 .↵
(it is created our own to deploy the java appl')

docker run --name myc2 -d -p 8082:8080
myjavapro:1

~~do copy~~

docker images ↵
myc2 → myjavapro:1
myc1 → myimg:1

myc2 include S3 bucket Add file so in that mahaLogin page is there so.

Go to url ----:8082/mahaLogin-5.0/login ↵
you will get website
of mahalogin

Docker Cmds

* pt create docker instance

Open in git bash

```
# sudo -i

# mkdir myDocker
# vi Dockerfile
```

Dockerfile (vi)

FROM
MAINTAINER
RUN
ADD / COPY WORKDIR
EXPOSE
ENTRYPOINT
CMD

These cmds are most important as par requirements.

Create image (docker)

```
# docker build -t myimg:1 . ↵
```

image name should be small letters.

```
# docker images ↵
```

check how many images in server

Delete docker images

```
# docker rmi myimg:1 -f ↵ for single image

# docker rmi $(docker images -q) -f ↵
```

delete all images.

Create docker container

docker run --name myc1 -d myimg:1 ↵
This statement of cmd used for practicing or we don't want to deploy any site.

docker run --name myc2 -d -p 8081:8080 myimg:1 ↵

Here,
-d - is ditached
-p - port no' is allocated to the container
This is statement of cmd used for deploying web site.

Delete docker container

docker rm $(docker ps -a -q) -f ↵
for all containers in the instance of git bash file.

docker rm containername myc1 -f ↵
Delete myc1 container or we can use container id also.
like
docker rm container id -f ↵

Insert in container (specific)

docker exec -it myc1 /bin/bash

do exit for out

Docker container list

docker ps -a

Shows all containers in server

CPU

docker run --name cpumyc1 --cpus =.5 -m 300M -d nginx ↵

docker run --name c2 --cpuset-cpus=0 -m 500M -d nginx ↵

docker inspect c2 ↵

Check nanocpus → . . . .

Docker GUI's

Class room cmd copy paste

Dtance copy publicip: 9000

Docker Volumes

docker run --name myc1 -v/maluvolu -d -p 8081:8080 myimg:1 ↵

docker ps -a

docker volume ls

docker inspect myc1

docker exec -it myc1 /bin/bash

cd ↵

df -h to check volume create or not

cd # cd /maluvolume

maluvolume # ls
——"—— # touch f1 f2
ls
exit ↵

myDocker # cd /var

... var # ls
shows lib # ~~cd lib~~
cd lib
:
var/lib # ls
shows docker

var/lib # cd docker/ ↵

ls
shows volumes

var/lib/docker/volumes # ls

Select our volumn name copy

#ls
cd paste-it ↵.. # cd _data /

ls
shows my files
~~f1~~ f2

Docker C Group

- Memory

Create 2 containers

docker run --name myc1 -d nginx ↵

docker run --name myc2 -d nginx ↵

docker stats ↵
Shows all Memory usage

:

vi /etc/default/grub ↵

- Edit → GRUB_CMDLINE_LINUX = ...

update grub

reboot

Create again a new container

docker run --name mycc1 -d -m 300M nginx ↵

docker inspect mycc1 ↵

docker run --name mycc2 -d -m 300M --memory-swap 1G nginx ↵

docker run --name mycc3 -d -m 300M --memory-reservation 100M nginx ↵

Hub.Docker.com

docker tag myding:1 maludockerhub/myimg28feb:1 ↵

docker images ↵

docker login ↵ id & pwd

docker push mydockerhub/myimg28feb:1 ↵

Refresh Hub.docker.com -> Repository.

Create own Network

docker network create -d bridge --subnet ~~mynet 198.1~~ 192.168.1.0/24 mynet ↵

docker network ls ↵

docker build -t myimage:1 .

docker run --name mync1 -d --network mynet myimage:1 ↵

docker run --name myc2 -d --network mynet myimage:1 ↵

docker exec -it mync1 /bin/bash ↵
~~chk ip address~~ # cd # ping 192.168.1.2 ↵

docker exec -it mync2 /bin/bash ↵
chk ip address
cd
ping 192.188.1.3 ↵