AWS AT A GLANCE

Types of services :

1.IAAS- Infrastructure services

1.Computing - VM EC2

Create VM->connect VM ->Edit

Create: Choose AMI-> Instance type->configure->add storage->add tags-.>configure

security Group->Review

Connect: 1.EC2 connect : direct

2.Putty : pem file to PPK file generate -> key and public IP

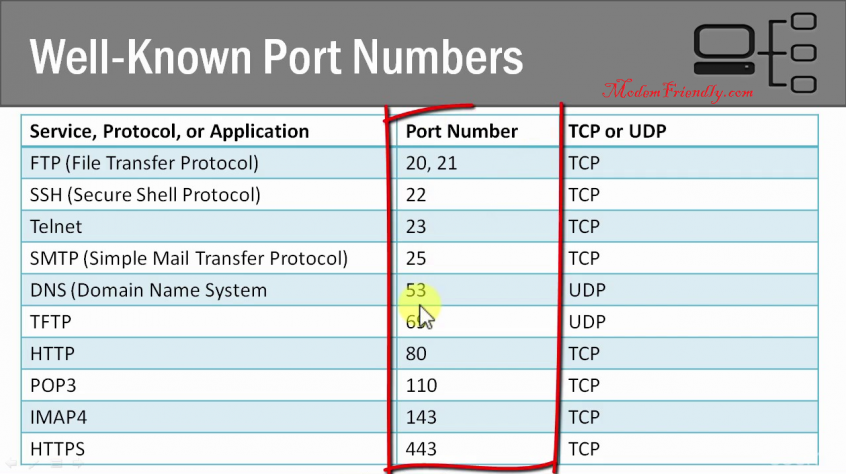
Address

3. SSH connect : using ssh -i System name ip address

Edit : in bound rules to manage the security levels.IP 0.0.0.0 is to access to all .Slect

Specific is not to allow.

Popular ports in network:



Connect to RDP Client : Generate password using pem file -> use name administrator

Password: generated password in cmd prompt

2. Storage -

1. Block storage : Laptop storage -EC2 VM- EBS

2. Object Storage : Google Drive -S3

3. Network Storage : EFS

* EBS create : EC2 -> EBS -> create volume ->attach volume -> verify attached vol.df- kh or ($ lsblk)-> if the commands show /dev/xvdf: data“,

it means your volume is empty.

$  sudo file -s /dev/xvdf

Format the vol to file system($ sudo mkfs -t ext4 /dev/xvdf)

Create a Dir to new volume ($ sudo mkdir /newvolume)

Mount the new volume

($ sudo mount /dev/xvdf /newvolume/)

Cd /newvolume -🡪df –h (should show empty)

* Object Storage: S3

Create a bucket

3. Network : VPC

VPC: Subnet public (web) /private (DB/APP)

Route table

NAT Gateway – provide internet to Pvt Subnet

Create a VPC

2. PASS: Platform

1. Database : RDS

2. Dev Tools

IDE -> GitHub –Maven

Cloud9-> Code commit🡪 Code star->Code build-> Code aircraft->code deploy -> code pipeline

**Code Commit:**

**GIT HUB**

|  |
| --- |
| git-scm.com |
|  |  | Windows - exe |
|  |  | git bash ----> basic Linux cmd in windows + git + ssh |
|  |  | Linux |
|  |  | apt-get install git -y |
|  |  | yum install git -y |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | Step 2 - Create git repo [ONE TIME] |
|  |  |  |
|  |  | $ git init |
|  |  |  |
|  |  |  |
|  |  | Step 3 - Decide which files you want to version| write code |
|  |  |  |
|  |  | -------------------ONE TIME----------------------- |
|  |  | git config user.name "Rajesh Kumar" |
|  |  | git config user.email "devops@rajeshkumar.xyz" |
|  |  | git config –list |
|  |  | ------------------------------------------- |
|  |  | Step 4 - Add these files to Git [ Add ur files from workspace to git staging ] |
|  |  | $ git add index.html |
|  |  |  |
|  |  | Step 5 - Commit files [ From git staging to repo area ] |
|  |  | $ git commit -m"This is my commit" |
|  |  |  |
|  |  | Step 6 - How to see commited history |
|  |  | $ git log |
|  |  |  |
|  |  | Step - 7 How to see what was commited? |
|  |  | $ git show f8cfdfa0c96adaffec9e16bf0257ca1fd85d1b1f |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | 487 git init |
|  |  | 488 git config user.name "Rajesh" |
|  |  | 489 git config user.email "devops@rajeshkumar.xyz" |
|  |  | 490 git add --all |
|  |  | 491 git commit -m"adding" |
|  |  | 492 git remote add origin git@github.com:devops-school-com/devopssupport.in.git |
|  |  | 493 git push origin master |
|  |  | 494 git pull origin master |
|  |  | 495 git status |
|  |  | 496 git add --all |
|  |  | 497 git commit -m"adding" |
|  |  | 498 git push origin master |
|  |  | 499 git pull origin master |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**SONARQUBE**

1. **It’s a Quality management tool**
2. **Code Quality Management tool (peers code review , static code review)**
3. **Test Quality management tool (code coverage , test coverage)**

**Static code review: code reviewed by code /rules/tool /sca**

1. **Sonar port 9000**
2. **Ip no:192.168.1.11:9000**

**JENIKINS**

1. Jenkins which facilitates to automate the continuous integration and delivery in automate the building ,testing and deployment
2. It provides 1000+ plug-ins which helps for CI
3. Port NO:

**KUBERNETS**

**(6443 port)**

**Master**

**----------------------------**

**- API Server -> POD -> Container -> Docker Image -> Google Registry**

**- ETCD -> POD -> Container -> Docker Image -> Google Registry**

**- Controller Mgr -> POD -> Container -> Docker Image -> Google Registry**

**- Schedular -> POD -> Container -> Docker Image -> Google Registry**

**- Kubelet**

**- Docker**

**- Kubeclt**

**- Kube Proxy -> POD -> Container -> Docker Image -> Google Registry**

**Worker**

**----------------------------**

**- Kubelet**

**- Docker**

**- Kubeclt**

**- Kube Proxy -> POD -> Container -> Docker Image -> Google Registry**

Kubelet: Instantiate the POD

Kubeadm init : initiates Master’s cluster

Kubeadm join : join to the cluster of Worker

Kublet->APiservr->etcd-> scheding manger-> Apiserver->dcoker container->Pod->Apiserver-> Etcd.

PODS have unique IP address and pod will contain different container