**DevOps**

What is DevOps?

Dev and Op team work together to automate everything

Software Development Life Cycle: Waterfall, Agile, DevOps

Requirement Gatherings

Planning

Designing

Development

Testing

Deployment & Maintenance

**DevOps Lifecycle:**

Code – Build – Test –Analysis – Delivery – Ops Changes – Software testing – Deploy – Prod live

A blue screen with text and icons

Description automatically generated

**Continuous Integration:**

Repository – Build – War/Jar/Zip – servers – Testing – production Servers

A diagram of a program

Description automatically generated

The goal is to detect errors before it grows.

**Continuous Delivery:**

**A blue screen with text and icons

Description automatically generated**

**Tools:**

[**https://github.com/devopshydclub/vprofile-project/tree/prereqs**](https://github.com/devopshydclub/vprofile-project/tree/prereqs)

Github sign up

Godaddy domain purchased

Link github with sonarCloud

Signup to aws

**Aws:**

Create user: IAM user (itadmin)

Activate billing preferences

Set alarms for billing

Certificate manager : once you create certificate, provide certificate cname and value to DNS server (manage domain in GoDaddy)

Login to the IAM user with the URL provided in the IAM dashboard.

Certificate in AWS.

A screenshot of a computer

Description automatically generated

**Virtualization:**

VMware: One computer runs multiple OS.

Baby computers within the physical computer

Each computer runs in isolated Environment (OS)

Host OS : Physical computer

Guest OS: Baby computers

Snapshot: VM backup files

Hypervisor: Tools that create VM’s.

**Set up virtual machine:** Linux VMs – 64bit.

1. uncheck windows VM platform … (windows features)

**Manually:**

1. Oracle virtual machine
2. New VMs—centos/Ubuntu/64bit
3. Download iso files and add them to settings.
4. Configure network for adapter 2. Wi-Fi adapter/attached to bridge.
5. Install
6. Connect with ssh (ssh username@ip)
7. A screenshot of a computer

   Description automatically generated

**Automate :**

Vagrant automates VM life cycle

Link <https://app.vagrantup.com/eurolinux-vagrant/boxes/centos-stream-9>

* Vagrant init <name of distros> // generates vagrant file.
* Vagrant up (start the VM).
* Vagrant ssh // connect to the VM.
* Vagrant status/global-status.
* Vagrant halt (power off).
* Vagrant reload (reload changes in vagrantFile)
* Vagrant destroy (delete VM)

A computer screen shot of a diagram

Description automatically generated

**Linux Fundamentals :**

/etc : configurations

/var, /srv server data

Touch file(1..10}.txt : creates 10 files

Redirections

Filtering

Pipe

Users and groups

adduser ansible

addgroup devops

usermod -aG devops ansible

/etc/group, /etc/passwd

File permissions:

File rw-:USER ---:GROUP ---:OTHERS

-rw-------. 1 root root 2027 Dec 18 13:20 anaconda-ks.cfg

drwxr-xr-x. 2 root root 24 Dec 23 00:20 devopsdir

-rw-------. 1 root root 1388 Dec 18 13:20 original-ks.cfg

**Software management:**

Yum

apt

Wget

Create bash script to install packages

**Services**

Systemctl start/restart/reload/enable httpd

**Process**

Top

Ps aux

Ps -ef // parent and child processes

ps -ef | grep httpd | grep -v 'grep'

kill <processID>

**archiving**

tar -czvf folder\_timestamp.tar.gz folder //compress

tar -xzvf path //extract

zip -r folder\_time.zip path

unzip folder\_time.zip

**Vagrant:**

Vagrant file contains configurations about the vm you want to create such us:

Ram, cpu cores. Networking etc.

**Provisioning**

Script to install dependencies or create folder in vagrant File

Vagrant reload --provision

**Server management**

Set up website:

1. Create VM with vagrant
2. Change hostname in /etc/hostname and exit
3. Install httpd/centos or apache2/ubuntu
4. Install dependencies such us wget, zip, unzip, vim etc.
5. Systemctl Start/enable httpd
6. Brows ip address
7. Put index.html in /var/www/html or template from <https://www.tooplate.com/view/2135-mini-finance>
8. Use wget with zip https file found in network browser
9. Systemctl restart httpd

Automate the process by writing all the previous commands in vagrantFile shell section

A computer screen shot of a black screen

Description automatically generated

**Wordpress**

* Create vm for wordpress(edit vagrantfile)
* Change hostname and logout
* apply this: <https://ubuntu.com/tutorials/install-and-configure-wordpress#9-thats-all>
* access ip address in browser

**Automate the process: IAC : Infrastructure as code**

**Infrastructure as code** (**IaC**) is the process of managing and provisioning infrastructure (networks, virtual machines, load balancers, and connection topology) through CODE/Config Files.

Paste the commands into the vagrant file shell section (see the script for the commands)

**MULTIVMS:** In a single vagrant file you can configure multiple vms , this is useful in projects where many servers are required.

**Variables – Python Data Structure – JSON – YAML**

Variables: in bash -> skill=”DevOps” to print -> echo $skill

Python DS: Integer, list, tuple, string, Dictionary

**JSON – YAML: Devops configs use these formats**

**Project: VProfile Project set up (locally)**

* Multi tier web application stack
* Setup on local machine
* Baseline for Upcoming Projects
* VM automation

**Architecture of project services**

NGINX: SECURITY (load balancing, API gateway)

TOMCAT: SPRING WEB SERVER (java web application service)

RABBITMQ: MESSAGING BROKER

MEMCACHED: MEMORY CASHING SYSTEM (DB)

MYSQL: DB MANAGEMNET SYSTEM

**Architecture of Automated Setup**

Vagrant

VirtualBox

Git Bash

See the pdf file to manually create all the vms (services)

Vagrant up: will make all the VMS

Ssh to all of them and check /etc/hosts to see if all of the ips are associated with vm names

**Set up all services:** See the pdf ([VprofileProjectSetupWindowsAndMacIntel.pdf](file:///C:\Users\yassi\School\Personal_Projects\DevOps\DevOps\vagrant-vms\VProfile\VprofileProjectSetupWindowsAndMacIntel.pdf))

**Automate the process:**

Automating the process of set up vprofile website involves the same steps in the manual method except that the vagrantFile will contain scrppts in the provision section that automates the creation of the 5 services(Mysql, Memcache, RabbitMq, Tomcat, and ngnix). See this:

**https://github.com/Yibhir0/DevOps/tree/main/vagrant-vms**

**Networking:**

OSI Model: Common language of networking

Application: allow access to network resources

Presentation: translate and encrypt

Session: manage sessions

Transport: process to delivery

Network: Packets, logical addressing(packets)

Data Link: Physical addressing (frames)

Physical: bits

Services are service that one layer pass to another.

Protocols rules that one layer uses to exchange data with another layer

Interface is communication between layers

Distance between network distance:

LAN : local area network -🡪

WAN : Wide Area Network

MAN : Metropolitan area Network

CAN : Campus Area Network

Switches: Connect multiple devices together

Routers connect multiple networks.

**Protocols:**

Reliable TCP: FTP, HTTP, HTTPS

Unreliable UDP: DNS, DHCP. It’s faster.

**Commands:**

Ipconfig to check ip address

Netstat -antp to check open ports

Dig url to see the DNS resolution of your url

Nslookup

Arp shows mapping of ip/name and mac adress

Route , shows tables of the gateway

Theses commands help you troubleshoot when connection is not established between 2 networks

**Containers:** Isolate processes in different computer (apche2, nginx, mongodb)

**Since they are very small we can archive them :** image which we can easily ship

we install docker through vagrantFile

docker run hello-world

docker images

docker ps 🡪 running images

docker ps -a

docker run --name web01 -d -p 9080:80 nginx (9080: outside port, 80: inside port)

**Microservices: Fast to develop, deploy, easier to scale**

S1. Notifications: python

S2. Chat : Node

S3. User Interface : java

Api gateway to connect all of them

Docker compose build all the microservices

**Ssh key:**

ssh-keygen

ssh-copy-id devops@web01

**AWS:**

Cloud computing: access virtual resources through the internet

**Services: IaaS, PaaS, SaaS**

**SysOps?**

What is EC2? Elastic compute cloud: services to provision virtual machines

AMI : Required to launch an instance which is a virtual server

Instance Type : hardware

Elastic Block Store: storage for your instance

Tags:

Security group : virtual firewall

Key:

**CLOUD WATCH:** monitoring performance of the environment

Metrics for services: alarms for notifications of violations of resources.

You can create an alarm for cpu metric that gets triggered when violatining certain criteria.

Logs

Events

**Amazon Elastic File System**

Used in containers, analytics, machine learning, dynamic websites images, database backups.

Create security group for efs

Add the efs command in /etc/fstap

Mount -vaf

Auto scaling : launch configs of number of instances to minimize costs

By making snapshots. The instances are launched from the snapshot. Don’t change the instance manually through command line. If you want to change configurations, you change the snapshot.

Makes sure storage is outside these instances (like in efs).

**What is S3?**

Object storage: bucket to upload files, images and access them through the internet.

I need to acquire more knowledge on this inshallah.

**RDS: Relational Data Base Service**

Administration of data base: Installs, patching, monitoring, performance tuning, backups, scaling, security, hardware upgrades, storage.