DevOps Practitioner training:

**(Day 1: Feb 15)**

**Basics:**

* Computer - Hard Disk/volume, Processor, RAM
* Hard drive is not considered a computing service.
* Hard drive is a storage device.
* OS is an interface software between the user to system.

**Server:**

* Business data - How much sold, quantity, profit, cost of products.
* Server which is present in one location and accessible from many locations is called Data center.
* Multiple people are accessing my server in different locations.
* Computer - Single person alone using the system.
* Server - Serving multiple users for multiple services.
* Each rack has a labelling and rack number to organize the server.
* Server - storage capacity, services, availability, database.

Eg: Big bazaar, retail business etc.

**Data Centre:**

* A data center is a building to place a server.
* Server can be anywhere
* Data center is a building space, it contains multiple servers and Databases.

**Virtualization:**

Each server is virtualized to multiple virtual machines (one server acts as three servers).

Each server is divided into 3 servers but physically it is one server. The concept is called virtualization.

In the server backend, we have a storage device.

Every virtual server has one Public IP.

-------------------------------------------------------

**Infrastructure:**

* Hardware devices are present Physically.
* Infrastructure - [Hardware, Processor, Memory, networking and storage, the infrastructure is used to run my OS]
* OS - Linux, Red Hat, Ubuntu, Cent OS, Amazon Linux, Windows, Mac OS.
* Application - Apache HTTP, Apache tomcat, Jenkins, MS Office.

**(Day 2 - 16 Feb)**

**Cloud computing basics:**

**- On premises vs cloud**

On-premise software is installed locally, on your business' computers and servers, where cloud software is hosted on the vendor's server and accessed by a web browser. On-premise infrastructure, or private clouds, are cloud computing environments that are reserved for use by one organization.

**- Cloud service models**

The cloud service models are Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS)

**Software as a Service (SaaS)**

Software as a service (SaaS) is a software distribution model in which a cloud provider hosts applications and makes them available to end users over the internet. An example for SAAS is email.

**Platform as a Service (PaaS)**

Platform as a service (PaaS) is a cloud computing model where a third-party provider delivers hardware and software tools to users over the internet. The example for PAAS is AWS Elastic Beanstalk, Windows Azure.

**Infrastructure as a Service (IaaS)**

Cloud-based services, pay-as-you-go for services such as storage, networking, and virtualization. The example for IAAS is Amazon Web Services (AWS), Cisco Meta pod, Microsoft Azure.

**AWS:**

- How to create the AWS account.

- Overview of AWS

- AWS Global infrastructure

- Regions and availability zones.

[26, Data centers are the availability zones]

We need to choose which region server to work in AWS.

- Regions in AWS

- EC2 service to create the virtual server and launch the instance.

- Comparison between the On-premises server and Cloud server.

**(Day 3 - 17 Feb)**

**AWS - EC2 service**

- AMI Instance [It shows the latest verison]

- EC2 instance creation - Linux server

- How to create the new key pair

- Using putty to connect the instance.

- AWS pricing calculator

- Create the windows server in AWS

- How to create a permanent IP for instance.

- How to change the instance type to another because of the performance issue.

1. Copy the IP address from AWS and go to the mobaxterm.

2. Go to the session and click on SSH and paste the IP address and name the specified username as ec2\_user and go for advaned setting and click on use private key (for making key pair)

3. Commands

* sudo su -- for changing from home directory to root user.
* cd /opt -- going into the directory (opt is a default directory)
* touch devops -- for creating empty files using touch.
* nano devops -- going into the file add same content onto it
* ctrl+x --> Yes(Y) ((for existing))
* cat devops -- for showing what content is present in the file.
* ll -- listing the files and directories present
* mkdir snega -- for creating directory
* cp devops snega -- copying files from devops to snega
* cat snega -- showing content present in the file
* cat /opt -- It will show opt is a directory
* rm -rf snega -- Deleting directory snega
* re devops -- will remove the file devops.

**(Day 4 - 18 Feb)**

**How Linux is organized**

User

|

Shell (C shell, K shell, Borne shell)

|

Kernel

|

Hardware

**Linux**

Linux is an open-source operating system. An operating system is software that directly manages a system's hardware and resources, like CPU, memory, and storage. Linux has good performance. It is much quicker, faster and smoother even on the older hardware.

**kernel**

Kernel is for OS and able to communicate with hardware (convert the command and give it to the hardware).

**User**

It is going to interact with shell

**Shell**

It is used to give commands from users.

**Linux commands**

* whoami -- find out who are you
* Groups or id -- used to find out what group you belong to
* users, who, w -- find out who is logged in
* su -- to start a new shell as a different user
* history -- last executed commands.
* last -- who logged into system.

**(Day 5 - 21 Feb)**

Apache http server

**Server**

A server serving multiple services

**Commands**

* Service sshd status – used to check the server status
* Service sshd stop – used to stop the server service.
* Service sshd start – used to start the server service.

**Other services:**

* 22 Port – ssh – service – Port: Login and secure file transfer
* 80 http – web application
* 443 https – web application (secure)
* 53 named – Domain Name resolution
* 21 FTP – File transfer

**Assignment Questions**

* Create an ec2 instance
* Install Apache webserver
* Setup a webpage where it displays “Welcome to Aspire Systems” message
* Access application from webserver

**Solution for Assignment:**

Instance --> virtual machine

[In AWS we are going to create instance and add the security group and rules because we are going to work on http server so we are going to add particular group in instance.

http --> port range --> 80

Ssh --> port range --> 22

Session --> SSH --> copy and paste the ip --> specify username(ec2-user) --> Advanced SSH setting --> use private key

**Commands will be used:**

1. sudo su
2. Yum install httpd -- commands for installing software
3. service httpd start --> for starting the particular server
4. service httpd status --> for knowing about the status of particular instance
5. Cd /car/www/html --> for accessing the http path
6. Vi index.html --> we are creating a file (index will be filename)
7. (Will click I in keyboard) --> because we are going to insert mode, if we go to insert mode only, we can type content
8. Esc --> escape will be used to come out of the file
9. :wq --> for save and exit
10. http://(ip address) --> copy the address from aws instance (which instance is used in mobaxterm) we are using this command for showing the content present in the file.

**(Day 6 - 22 Feb)**

* create two users (user 1 and user 2) and setup password
* Enable password-based authentication
* Login with your credentials
* Create file under user1 and grant the root privileges
* Create a group devops and add the users under this group
* Give the total count of users on the server

**(Day 7 – 23 Feb)**

**Assignment - EBS, Snapshot and AMI**

* Create an ec2 Instance and Install Apache webserver
* Create an EBS volume and mount to the directory
* Create a snapshot and restore the volume
* Create a Golden Image
* Launch the new ec2 Instance using Golden Image

**Working on Assignment:**

Create an instance

Go for volume --> Add volume

Go for Actions --> Attach volume

Choose Instance

Now the volume is attached to a particular instance.

Allocate --> Action --> Allocate IP

Click ins & ot

Copy the elastic IP

Go to Mobaxterm --> session --> SSH --> paste ip --> ec2-user

**Using commands**

1.sudo su

For root user

(Now we are going to install httpd server

2. yum install httpd

3. service httpd start

4.service httpd status

5. Go to Mobaxterm and use the commands

6. vi index.html

7. Press I

(Write any content)

8. ESC --> :wq

9.ls

For listing the file

10. lsblk

For listing the block files

11. mkfs –t ext4 /dev/xvdf

This command is for formatting (By using this we can format the files)

12. mount /dev/xvdf /var/www/html

Mounting the files

13. df -h

Use to list the disc files information

14. vi index.html

Creating the file (Index will be the file name)

(Press I and type the content in that file and give esc)

15. umount /var/www/html

For unmounting the file

16. df –h

Checking whether it is unmounted or not

17. lsblk

18. mount /dev/xvdf/var/www/html

19. df -h

20. ls

For listing the file

**(Snapshot will be created for backup)**

21.umount /var/www/html

22. df –h

Go to the snapshot --> Action --> Create volume from snapshot **(Snapshot for backup)**

23. Mount /dev/xvdf/var/www/html

For storing purposes

**Go to create Golden image**

Instance --> Actions --> Image and templates --> Create Image (Give image name) --> Image --> AMI (for checking image is created or not)  
  
Go and terminate the instance

Create instance --> AMI --> Create new instance

Copy and paste the Ip in mobaxterm and use the commands

1. sudo su
2. service httpd start
3. service httpd status
4. ls -l

Go and put the IP(AWS) in a new tab, you can read the file content.

**(Day8 – 24Feb)**

1. Cloud watch alarms
2. Know to use SMS (Simple Notification Service)
3. Stress command used to put load on the server
4. EPEL – Extra Packages for Enterprise Linux

Assignment – Cloud Watch and SNS

* Create an ec2 instance and Install Apache webserver
* Create a Cloud watch alarm
* Create an SNS topic and enable the e-mail notification
* Login back the EC2 instance and apply stress commands
* Now verify email alert

**(Day9 – 25Feb)**

What is S3

1. Computing resources
2. Handling I/O
3. Estimating EBS volume size
4. S3 versioning

**Assignment:**

* Create an ec2 instance and S3 bucket in root account
* Upload a text file in the S3 bucket and enable versioning
* Create an IAM User Account by giving the S3 and EC2 full access
* Login the IAM User Account Console and access the S3 bucket and EC2
* Login through AWS CLI and access the S3 Bucket

Commands for Assignment:

1. sudo su

2. yum install httpd

3.service httpd start

4. service httpd status

5. yum install git

6.pwd

7. git clone <https://dptrealtime@bitbucket.org/dptrealtime/html-web-app.git>

**(Day10 – 28Feb)**

1. What is System Manager?

Computer and information systems managers supervise organizations computer-related activities. They plan, install, and monitor software and hardware to meet overall information technology needs.

2. Networking basics

* Network
* IP
* Subnet
* IP classes
* Port
* Protocol
* Service
* Daemon
* Private IP
* Public Ip
* Privat3e Subnet
* Public Subnet

**Assignment:**

1. Create an IAM user account from the root user.
2. Create an IAM role with full permission for EC2 instance and assign to the IAM user account
3. Launch an EC2 instance from IAM user account
4. Login to the EC2 instance and verify the amazon-ssm-agent status

(The main theme is to use that in root user and IAM user (i.e)., we can able ro see the content of the file which we had been created in the root user can be able to see in IAM user) --> by giving full access.

**Solution:**

Iam (search) --> User --> roles --> Create role in that click AWS service and EC2 --> Next --> EC2 full accesss (search) and select it --> next --> Create role

EC2 --> Actions --> Security --> Modify IAM role

Click the Name of the instance and save.

Go to Mobaxterm and use the commands

1. sudo su
2. aws –version

Go to AWS

Go to IAM user and create an instance

Sign out and sign in as root user

Mobaxterm and use the command

1. Service amazon-ssm-agent status.

**(Day11 – 01 March)**

**Networking**

Build your own Data center

1. How many Networks?
2. What is network range
3. How many servers are in each Network
4. How many Public Networks
5. How many Private Networks
6. Server connectivity

**VPC**

1. VPC
2. Internet Gateway
3. Public Subnet
4. Private Subnet
5. Public Route Table

(i) Local

(ii) IGW

6. Private Routable

(i) Local

**Assignment –VPC Creation – Public and Private Server**

* Create a VPC along with Public and Private Subnet
* Create an ec2 instance in Public Subnet
* Create an ec2 instance in Private Subnet
* Login the Private ec2 instance through public ec2 instance
* Create a NAT Gateway to provide Internet access to the private EC2 instance
* Install the Apache Webserver in the Private ec2 instance

**Solution:**

AWS – VPC – Your vpc – Create vpc – Name – Give IP (192.168.0.0/16) -Create vpc

Internet Gateway – Create Internet Gateway – Name tag (specify name) - Create Internet Gateway – Attach a VPC – Available VPC – Attach Internet Gateway

Subnet – Create subnet – VPC Id and give IPv4 CIDR block (192.168.1.0/24) - Create Subnet – Subnet name (private Subnet) - IPv4 CIDR block (192.168.2.0/24) - Create Subnet

Route table – create route table – name,vpc (create private and public) - create

Route table – create route table –name (public) - create – click on public – edit route (Destination (0.0.0/0) Target (internet gateway))

Save changes

Go to subnet association

Edit subnet association

Security Group – Create Security group –Give name and description – Add rule (SSH and Anywhere)

EC2 --> Launch instance

3.Configure instance detail (Don’t use default)

Subnet --> public demo (Click)

Auto assign --> enable

6.Configure security group

Select an existing security group

Copy the public IP and open mobaxterm

Step 1: Upload the key (yamini.pem)

Step 2: Click 4 (read means 4 and write means 5)

Step 3: chmod 400 yamini.pem

Step 4: ssh -I yamini.pem ec2-user (ip address) [{use the private ip address}]

Step 5: ping [www.google.com](http://www.google.com) {we can find that the net is communicating with NAT)

**(Day12 – 02 March)**

1. What is Auto scaling

2. ASG Architecture

**Assignment:**

* Create a VPC along with Subnet and Private Subnet
* Create an Auto Scaling Group
* Create an ELB
* Access the Web Browser through ELB DNS Name
* Provide Load Balancer DNS Name for Validation

Solution:

VPC --> Create VPC --> Give name --> IPv4 CIDR (enter the Ip (ex: 100.101.0.0/16) --> Create VPC --> Internet Gateway --> Add Name --> Create IG

{want to create four subnets 2 private and 2 public)

Subnet --> Give VPC Id (what you have created) --> Give subnet name (ex: public subnet 01) --> Give Availability zone --> IPv4 CIDR block (100.101.0.0/24) and (100.101.1.0/24)

{create 4 subnets in different names of two public and 2 private but in different availability zone).

Create two route table (one public and one private)

Route table --> Give name and select VPC

Internet Gateway --> Actions --> Attach to VPC --> Select the VPC --> Attach Internet Gateway

Route table --> (public route) --> Edit route --> Add route --> 0.0.0.0/0(Anywhere) --> IG

Route table --> (private route) --> Edit route --> Add route --> 0.0.0.0/0(Anywhere) --> IG

Edit --> Add --> 0.0.0.0/0v--> NAT

(Create NAT)

NAT Gateway --> create NAT --> Subnet (give public subnet which you have been created) --> Connectivity type (public) --> Allocate elastic Ip --> Create NAT

Go to Security group --> create security group --> add inbound rule --> custom tpc (all traffic) --> source (Anywhere) --> create security group

Ec2 --> give launch configuration --> launch configuration --> AMI (copy and paste the AMI id)

Choose instance type --> choose --> Additional configuration (give s3 access) --> security group --> select an existing security group --> Key pair (Choose an existing pair) --> gove key --> create --> ec2 --> actions --> create auto scaling group --> next --> choose vpc --> choose vpc --> Availability zone (private) --> next --> give seconds --> next --> Tag (give name and value) --> group size --> Target tracking sealing policy --> next

Load balancer --> create load balancer --> network load balancer --> create --> name it

--> network mapping --> (select created VPC) --> subnet (give the public subnet created region) --> next --> target group name --> VPC --> next --> Create target group --> get back to load balancer (refresh) listener (Attach the target group)

--> Create load balancer --> Auto scaling --> Auto Scaling group --> Click already created auto scaling --> edit --> load balancers (click on application group)

**(Day13 – 04 March)**

**Assignment:**

1. Install the vs-studio code
2. Generate the AWS access key and secret key
3. Download the terraform for WIN
4. Declare the environmental variables
5. Create and main.tf file in the IDE and upload the code
6. Now start applying the terraform code with the sequence of commands

**(Day14 – 07 March)**

**Assignment:**

* Create a VPC through terraform
* Create an ec2 instance through terraform associating with existing VPC
* Create a new module template for launching the ec2 instance
* Create the ec2 instance through terraform module

**(Day15 – 08 March)**

**Git:**

It is a local repository accessible by a specific developer

**Assignment:**

* Install git Bash in your respective systems
* Create a Bitbucket Account
* Create a New project
* Create a New repository under the New project
* Configure the Username and email id in your git bash
* Push the terraform VPC code from local repository to the remote repository

**Solution:**

Open windows power shell and use the command for installing git

1. Winget install –idmGit.Git -e –source winget

Create a folder library

In visual studio code open new terminal

Open the folder which is created in local desktop and create a file vpc.tf

1. git init
2. ls –la
3. git config user.name yamini
4. git config user.email [yamini.dayanithi@aspiresys.com](mailto:yamini.dayanithi@aspiresys.com)

Give project name, repository name and create the repository

Create the file in VSC

output “vpc\_id” {

Value = aws\_vpc.my vpc.id

}

And save it.

Login to bit bucket --> create repository --> clone --> copy the address

5. git remote add origin (paste the address)

6. git pull orgin master

Create a file

Eg : vpc.tf

provider “aws” {

region = “us-east-1"

}

1. git add.
2. git status
3. git commit –m “Provider block added”
4. git push originmaster