**Toolstack :**

**AWS**

**Ansible**

**Jenkins**

**GIT – upto devops**

**Nagios—monitoring tools**

**Scripting**

**Basic Shell scripting**

**AWS**

**Cloud Formation templates**

--------07/08/17------------

---------Day 1--------------

We created linux machine

We connected to linux machine in the cloud

We installed apache webserver

We created a sample webpage

We are able to access it

Note: Linux is case sensitive use all small letters in commands

https://www.tecmint.com/useful-linux-commands-for-newbies/

cd - change directory

pwd- present working directory

ls -list

cd .. one path back

ls -ll

Mkdir — to make new directory

Ex: mkir nameoffolder

Touch— to create new file

Ex: touch filename

Vim — text editor

http://www.radford.edu/~mhtay/CPSC120/VIM\_Editor\_Commands.htm

yum install nameofthesoftware for all linux platforms

apt-get install namoofsoftware for ubuntu

Jira — ticket assigning tool in companies

--------07/09/17----------

--------Day2--------------

GIT :

Dailynotes:

https://github.com/raghu4a4/dailynotes.git

**GIT Cheat Sheet :**

https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf

GIT Download URL

<https://git-scm.com/downloads>

type : git –version

main commands in GIT:

git config --global user.name “username”

git config --global user.email [you@example.com](mailto:you@example.com)

git status

git add . – for adding files

git commit -m "sample file"

git push – to push into remote git repository

git pull – from remote repository to local repository

--MAVEN --

Prerequisites for Maven installation:

Install JRE/JDK:

Install Java for mac -- <https://www.java.com/en/download/mac_download.jsp>

Set JAVA\_HOME

<https://www.mkyong.com/java/how-to-set-java_home-environment-variable-on-mac-os-x/>

Download Maven

<https://maven.apache.org/download.cgi>

**Set Maven PATH**

export M2\_HOME="/Users/johndoe/Development/apache-maven-3.0.5"

export PATH=${PATH}:${M2\_HOME}/bin

**Sample Maven Project:**

https://github.com/raghu4a4/time-tracker.git

Maven POM expalanation:

POM – project object model

<https://maven.apache.org/guides/introduction/introduction-to-the-pom.html>

**HW:**

Difference between webserver and application server

Ex for webserver: apache or httpd, windows IIS

Ex for application server: apache-tomcat, websphere, weblogic

Maven Lifecycle

Install Maven and set home path

**----Day3------**

Deploy created war file into tomcat app server

**Tomcat:**

Download URL:

<http://tomcat.apache.org/download-90.cgi>

start the server in bin – ./startup.sh

**Add role authentication:**

In tomcat-users.xml

<role rolename="manager-gui"/>

<user username="admin" password="admin" roles="manager-gui"/>

Host the webapplication in webapps folder:

Restart tomcat

<http://localhost:8080/time-tracker-web-0.3.1-SNAPSHOT/>

Change the source code and rebuild and redeploy the server

Check the changes in the home page

**Jenkins Installation:**

[**https://jenkins.io/download/**](https://jenkins.io/download/)

**----Class 4--------**

**Jenkins workspace in Mac:**

/Users/Shared/Jenkins/Home/workspace

Jenkins Plugins walkthrough

Jenkins slave configurations

Workspace – shared workspace

Upstream and downstream jobs

Step1: I want to run Job A

Step2: I want to run Job B

Parametrized builds:

<https://wiki.jenkins.io/display/JENKINS/Building+a+software+project>

Log history

Jenkins system variables

Running shell commands in Jenkins

**Jenkins Installation on Ubuntu:**

[**https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu-16-04**](https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu-16-04)

**---------Class 5 -----------**

**----Jenkins----**

[**http://ec2-34-229-15-192.compute-1.amazonaws.com:8080/**](http://ec2-34-229-15-192.compute-1.amazonaws.com/)

**Timestamper:**

[**https://wiki.jenkins.io/display/JENKINS/Timestamper**](https://wiki.jenkins.io/display/JENKINS/Timestamper)

**Thinbackup**

[**https://wiki.jenkins.io/display/JENKINS/thinBackup**](https://wiki.jenkins.io/display/JENKINS/thinBackup)

**Mitlijob plugin**

**Publish over ssh plugin**

**Jenkins autotrigger by webhooks plugin**

**-----------Class6-----------**

Deplyed the artifact in tomcat using Jenkins

Build step🡪 Execute Shell-> cp /sourcefolder /destinationfolder

Sourcefolder🡪 Workspace location

Destinationfolder🡪 tomat webapps folder

Ex:

cp /var/lib/jenkins/workspace/SampleDeploy/web/target/time-tracker-web-0.3.1-SNAPSHOT.war /home/ubuntu/apache-tomcat-8.5.16/webapps/

Note: please change the tomcat folder ownership and permissions to Jenkins for allowing to deploy the application

Chown –R Jenkins:Jenkins apache-tomcat

Chmod 755 –R 754 apache-tomcat

**-----------Class 7-------------------**

**Many ways accessing AWS:**

**Console – UI**

**AWS CLI – Command Line Interface**

**Cloud Formation Templates**

**3rd Party Tools like – Ansible, Chef, Puppet, terraform etc.**

**Python BOTO – scripting using Python**

**List of Topics:**

**IAM**

**Creating a user and providing aws cli access**

Writing Policies and attaching for s3, ec2 etc

Create Users, Roles, policies, groups

AWS CLI access etc

**EC2**

**ELB, ASG, SG, Elastic Ip, Keys, Instance Types,**

**AZ-** Availability zones

Regions

<https://aws.amazon.com/about-aws/global-infrastructure/>

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/resources.html>

**-------- Class 8 ---07/24/2017 -----------------**

**EC2: create AMI**

**restore AMI**

**S3**

**Objects**

**Versioning**

**Policies**

**Permissions**

**Policy creation sample for user training**

**Sample S3 Policy Document:**

{

"Id": "Policy1500943915901",

"Version": "2012-10-17",

"Statement": [

{

"Sid": "Stmt1500943914393",

"Action": [

"s3:GetObject",

"s3:PutObject"

],

"Effect": "Allow",

"Resource": "arn:aws:s3:::devops07242017/\*",

"Principal": "\*"

}

]

}

**s3 commands:**

aws s3 ls –list buckets

aws s3 ls bucket\_name – list objects in that bucket

aws s3 cp sourceobject s3://destinationbucket/

aws s3 cp s3://destinationbucket/filename destinationfolder

aws s3 rsync

For Volume Operations:

<https://devopscube.com/mount-ebs-volume-ec2-instance/>

**Integrate with Jenkins job to run this command**

**-------- Class 9 --------**

**SNS Queues**

**SQS Queues – rabbitmq,**

**Ec2 Alarms**

**LoadBalancer – create classic loadbalancer**

**Auto scaling groups – step 1 launch configuration and step2 : autoscaling groups**

**---------Class 10--------------**

**one real time example on s3,ec2,Iam,policies- practice**

**Problem :**

**I want to access an s3 Bucket with only particular EC2 instance using IAM Role based approach, that should not be accessible from other EC2 instances.**

**Solution:**

1. **Create s3 first without any policies**
2. **Create an IAM policy with Resource as s3 bucket**
3. **Create an IAM Role attaching that policy created**
4. **Create Ec2 instance using that IAM role**
5. **Create s3 bucket with resource as same bucket name and principle as IAM Role.**

**RolePolicy:**

**{**

**"Version": "2012-10-17",**

**"Statement": [**

**{**

**"Effect": "Allow",**

**"Action": "\*",**

**"Resource": [**

**"arn:aws:s3:::{{ S3\_bucket\_Name }}/\*"**

**]**

**}**

**]**

**}**

**s3 policy:**

**{**

**"Version": "2012-10-17",**

**"Id": "Policy1495728848930",**

**"Statement": [**

**{**

**"Sid": "Stmt1495728846942",**

**"Effect": "Allow",**

**"Principal": {**

**"AWS": "arn:aws:iam::584516218565:role/{{ CDSStackName }}"**

**},**

**"Action": "s3:\*",**

**"Resource": "arn:aws:s3:::{{ S3\_bucket\_Name }}/\*"**

**}**

**]**

**}**