AWS& DEVOPS MATERIALS

……………………………………………………………………………………………………………………………………………………………….

[**Jenkins 1**](https://docs.google.com/document/d/1JWE-otQzSllJ6qg5Ln2cW_xWbyPo8UjZ-LjIbeWQ_g8/edit#heading=h.34uaue2a9mg4)

[What is Continuous Integration (CI) ? 2](https://docs.google.com/document/d/1JWE-otQzSllJ6qg5Ln2cW_xWbyPo8UjZ-LjIbeWQ_g8/edit#heading=h.cw94rht3cnie)

[Install Jenkins 2](https://docs.google.com/document/d/1JWE-otQzSllJ6qg5Ln2cW_xWbyPo8UjZ-LjIbeWQ_g8/edit#heading=h.2b8dolw6d5p5)

[Configure Jenkins 2](https://docs.google.com/document/d/1JWE-otQzSllJ6qg5Ln2cW_xWbyPo8UjZ-LjIbeWQ_g8/edit#heading=h.zh9yoxygimf)

[**Jenkins Examples 3**](https://docs.google.com/document/d/1JWE-otQzSllJ6qg5Ln2cW_xWbyPo8UjZ-LjIbeWQ_g8/edit#heading=h.fry9x96z6qm9)

[Create Jenkins job to clone the project and build a war file. 3](https://docs.google.com/document/d/1JWE-otQzSllJ6qg5Ln2cW_xWbyPo8UjZ-LjIbeWQ_g8/edit#heading=h.dhpwjt1bwtf4)

[Create Jenkins job to build a war file and deploy to tomcat server. 3](https://docs.google.com/document/d/1JWE-otQzSllJ6qg5Ln2cW_xWbyPo8UjZ-LjIbeWQ_g8/edit#heading=h.2wi0khw52d2y)

[Trigger the above job automatically if there are new changes in git 4](https://docs.google.com/document/d/1JWE-otQzSllJ6qg5Ln2cW_xWbyPo8UjZ-LjIbeWQ_g8/edit#heading=h.edfk11sycyu1)

[Send automated email to the dev team if build fails 4](https://docs.google.com/document/d/1JWE-otQzSllJ6qg5Ln2cW_xWbyPo8UjZ-LjIbeWQ_g8/edit#heading=h.7mpqr7fvvrct)

[**Discarding Old builds 5**](https://docs.google.com/document/d/1JWE-otQzSllJ6qg5Ln2cW_xWbyPo8UjZ-LjIbeWQ_g8/edit#heading=h.i0ao6nrfolry)

[**Archiving artifacts 5**](https://docs.google.com/document/d/1JWE-otQzSllJ6qg5Ln2cW_xWbyPo8UjZ-LjIbeWQ_g8/edit#heading=h.9yd1kiaboz43)

[**Walking through jenkins directory structure 5**](https://docs.google.com/document/d/1JWE-otQzSllJ6qg5Ln2cW_xWbyPo8UjZ-LjIbeWQ_g8/edit#heading=h.eal677hpxgvj)

[**Managing users in jenkins 6**](https://docs.google.com/document/d/1JWE-otQzSllJ6qg5Ln2cW_xWbyPo8UjZ-LjIbeWQ_g8/edit#heading=h.dnxk8qjb4yyc)

[Configuring granular permissions for jenkins users 6](https://docs.google.com/document/d/1JWE-otQzSllJ6qg5Ln2cW_xWbyPo8UjZ-LjIbeWQ_g8/edit#heading=h.y3clyc3yjgz5)

[Security Realm 6](https://docs.google.com/document/d/1JWE-otQzSllJ6qg5Ln2cW_xWbyPo8UjZ-LjIbeWQ_g8/edit#heading=h.gqif37gnm3hn)

[**Master Slave Configuration 6**](https://docs.google.com/document/d/1JWE-otQzSllJ6qg5Ln2cW_xWbyPo8UjZ-LjIbeWQ_g8/edit#heading=h.uvn8zg7p5lr)

[Adding a slave to the master 6](https://docs.google.com/document/d/1JWE-otQzSllJ6qg5Ln2cW_xWbyPo8UjZ-LjIbeWQ_g8/edit#heading=h.3728dw5j33gc)

[Configure a job to run on the slave 7](https://docs.google.com/document/d/1JWE-otQzSllJ6qg5Ln2cW_xWbyPo8UjZ-LjIbeWQ_g8/edit#heading=h.f90ebfxryynd)

Jenkins

Jenkins is popularly know from **Continuous Integration** (**CI**) tool. However we can automate any kind of task using jenkins.

Jenkins is open source, i.e. free of cost.

Jenkins is web based tool implemented in java.

Previously jenkins was called as **Hudson,** initially this tool was maintained by Oracle, few developers had differences with Oracle and they came out and continued this tool with the name Jenkins.

What is **Continuous Integration** (**CI**) ?

Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day.

Each check-in is then verified by an automated build, allowing teams to detect problems early.

By integrating regularly, fixing defects are easy and quick.

Install Jenkins

Prerequisite: Jenkins requires jdk8

1. Install jdk8

sudo yum install java-1.8.0-openjdk-devel -y

If your machine has multiple versions , we need to update default version to java8.

*sudo update-alternatives --config java*

*Choose the number beside java8*

*sudo update-alternatives --config javac*

*Choose the number beside java8*

1. Install Jenkins

sudo wget -O /etc/yum.repos.d/jenkins.repo <https://pkg.jenkins.io/redhat/jenkins.repo>

*sudo rpm --import* [*https://pkg.jenkins.io/redhat/jenkins.io.key*](https://pkg.jenkins.io/redhat/jenkins.io.key)

*sudo yum install jenkins -y*

*sudo service jenkins start*

Enable jenkins on reboot (jenkins auto starts when we reboot VM)

*Sudo chkconfig jenkins on*

Configure Jenkins

Before using jenkins we need to configure it by installing plugins & creating one admin user

1. Access jenkins via web browser

<http://ip-address:8080>

1. Unlock jenkins
   1. Take initial admin password

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

* 1. Enter initial admin password and click continue

1. Choose install suggested plugins
2. Create First Admin User, fill the details and click **Save & Finish**
3. Click start using jenkins

Jenkins Examples

1. Create Jenkins job to clone the project and build a war file.

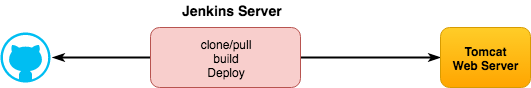
In jenkins, Job represents a task

* Click new item
* Enter item name as “**FirstJob**”
* Select **Freestyle project** and click **Ok**
* Under source code management select git
* Enter git URL   [**https://github.com/javahometech/myweb**](https://github.com/javahometech/myweb)
* Under build section add “top level maven targets”
* Under Goals enter    ***clean package***
* Save the job.
* Run the job by clicking **Build Now**

1. Create Jenkins job to build a war file and deploy to tomcat server.

For deployments we are going to use jenkins plugin, this plugin deploys the war using tomcat's manager application.

**Note:** Tomcat manager application must be enabled



* Install Deploy to container plugin
  + Manage Jenkins → Manage Plugins → Available →
  + Search for [Deploy to container Plugin](https://plugins.jenkins.io/deploy) and select → install without restart
  + Go back to jenkins home
* Click new item
* Enter item name as “**BuildAndDeploy**”
* Select **Freestyle project** and click **Ok**
* Under source code management select git
* Enter git URL   [**https://github.com/javahometech/myweb**](https://github.com/javahometech/myweb)
* Under build section add “top level maven targets”
* Under Goals enter    ***clean package***
* Under post build actions
  + - Select deploy war/ear to the container
    - War/Ear file  → target/myweb.war
    - Context Path → javahome
    - Add container → tomcat8
  + Credentials
    - Add → jenkins → select username with password
    - Enter tomcat managers username and password
    - For Id field enter relevant value for example tomcat-manager and ADD
  + Select the credentials
  + Add tomcat url for example ([***http://172.31.2.228:8080/***](http://172.31.2.228:8080/))
  + Save the job
* Run the job by clicking **Build Now**

1. Trigger the above job automatically if there are new changes in git

* Select the job “**BuildAndDeploy**” → click configure
* Select build triggers
  + Select GitHub hook trigger for GITScm polling, this tells jenkins if changes found in git then trigger this job
  + Select Poll SCM, this is the schedule give to jenkins to check for updates in SCM
  + Under schedule put five stars →  \* \* \* \* \*
  + Save the job
  + To test this job do some changes in git, it automatically triggers this job

1. Send automated email to the dev team if build fails
   1. In Order to send and receive emails we need to configure email servers in jenkins.

Companies has their own email servers, we have to use those servers to trigger

Emails, in our example we do not have our own server, so we are going to use

Gmails SMTP server.

* 1. Configure gmail SMTP server in jenkins
     1. jenkins → Manage Jenkins → Configure System
     2. Under **E-mail Notification**
        1. SMTP server  → **smtp.gmail.com**
        2. Select Use SMTP Authentication
        3. Put your gmail id
        4. Put your gmail password
        5. Use SSL select
        6. SMTP port  465
        7. Save the configuration
  2. Configure jenkins job to trigger email if build fails
     1. Open your job → Configure → Post Build Actions → Add post-build action → email notification
     2. Under Recipients put teams mail id and save the configuration

Discarding Old builds

Discarding old builds are important to free up some space on the hard disk

* Select Job → Configure → select discard old builds
* Enter Days to keep builds as 5
* Max # of builds to keep 5
* Save the configuration

Archiving artifacts

When build is triggered current artifact is overwritten with previous artifact, in order to retain previous build artifacts we have to enable archiving artifacts.

Open your job → Configure → Post Build Actions → Add post-build actions → archive artifacts

Files to archive → target/myweb.war

Walking through jenkins directory structure

ssh into the jenkins server

cd /var/lib/jenkins

Folders under jenkins installation

**Workspace:** This is the folder where jenkins keeps the project specific files

**Job:** This contains configuration details of the job

**Plugins:** This folder contains all installed plugins

**Nodes:** the information about jenkins slaves is kept under this folder

**Users:** The information about the jenkins users is kept under this folder

**userContent:** The files kept under this folder is served via http

Managing users in jenkins

This allows us to add/modify/delete users in jenkins

**Adding new user:** jenkins →Manage Jenkins → Manage Users → Create User

Fill the user details and click create user.

Configuring granular permissions for jenkins users

Jenkins has matrix based security option to configure granular permissions to the users in the jenkins

Jenkins → Manage Jenkins → Configure Global Security → under Authorization select matrix-based security.

Security Realm

This all about where jenkins maintains user credentials and their roles

1. Jenkins own database (default option)
2. LDAP (Lightweight Directory Access Protocol)

Maintaining user credentials and their roles in LDAP give more performance than maintaining in DB.

By integrating jenkins with LDAP we can grant access to the users already present in LDAP.

1. Unix user/group database

Master Slave Configuration

Master slave in jenkins is similar to load balancing, we use this for the following use cases.

* Improve Jenkins performance
* We can also use separate slave for different workload type, for example

Slave-1 → Java

Slave-2 → .Net

Slave-3 → Terraform

Slave-4 → Chef

**Master** is the machine on which we installed and configure jenkins

**Slave** is a machine added under the master, Slaves are controlled by master based on the job configuration.

Adding a slave to the master

* Jenkins → Manage Jenkins → Manage Nodes → New Node
* Node Name → Slave-1
* Select Permanent Agent  and click OK
* # of executors → 1 (*one executor represents one thread, to run multiple jobs concurrently specify bigger value*)
* Remote root directory →  /users/ec2-user/jenkins (*An agent needs to have a directory dedicated to Jenkins. Specify the path to this directory on the agent*)
* Labels → “JavaJobs”, labels are used to link a job to run on a specific slave
* Usage
  + Use as much as possible
  + Only build jobs with label expressions matching this job (*We are using this option for this demo*)
* Launch Method, Launch slave via ssh
* Host → IP address of the slave
* Create new credentials for ssh into slave, and select this credentials
* Host Key Verification Strategy → Non verifying verification strategy
* Availability → keep this agent online as much as possible and click save

***Note: On slave java must exists, apart from this, for example if our job requires git, maven etc we also need to install those tools as well.***

Jenkins Pipeline jobs (Jenkins 2.x)

Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

These jobs use Jenkins DSL as a syntax and groovy is the scripting language

Create Pipeline Job

* Jenkins → New item → Name = “Pipeline-Demo”
* Select Pipeline and click Ok

SonarQube

SonarQube is an open source automated static code analysis tool, It is an web application, which takes our code as an input, and produces the quality report as an output.

On this tool we also can publish Junit test coverage reports.

We can set up quality gates in this tool and we can fail the build if code quality is not met.

The ultimate use case of SonarQube is to improve the quality of the code and eliminate the defects at development stage.

Installing SonarQube on a linux machine

SonarQube is implemented using java language, to run this tool java needs to be installed on the server.

1. Make sure java8 is installed
2. cd /opt
3. Download SonarQube using the below command

           wget <https://sonarsource.bintray.com/Distribution/sonarqube/sonarqube-6.5.zip>

1. Unzip the file

sudo unzip  sonarqube-6.5.zip

1. Start sonar server

          cd /opt/sonarqube-6.5/bin/linux-x86-64

sudo ./sonar.sh start

**Note:** If there is issue starting sonar we can check the logs,

        under (/opt/sonarqube-6.5/logs)

Default port number of sonar is 9000

Access the sonar form webbrowser

<http://public-ip-address:9000>

Default username and password is admin/admin

Publishing java code from jenkins to sonar

1. For this to work, mvn needs java8
2. Configure sonar server detail sin maven settings.xml

ssh into jenkins server

1. Open maven settings.xml
   1. To find where maven is installed type (mvn -version)
   2. By default maven is installed at (/usr/share/apache-maven)
   3. Open setting.xml under /usr/share/apache-maven/conf

<profiles>

    <profile>

      <id>sonar</id>

      <activation>

        <activeByDefault>true</activeByDefault>

      </activation>

      <properties>

        <sonar.host.url>http://172.31.0.73:9000</sonar.host.url>

     </properties>

  </profile>

</profiles>

Create a new job in jenkins

Under build select maven

Under goals → package sonar:sonar

Publishing Junit coverage report to sonar

Under maven goal specify   org.jacoco:jacoco-maven-plugin:prepare-agent

Setting quality gates in sonar and failing build if code quality is not met.

Create quality gates in SonarQube

1. Create two gates
   1. If blockers greater than zero then error
   2. If junit coverage is less than 85% then error
2. In jenkins
   1. Install plugin “[Quality Gates Plugin](https://wiki.jenkins-ci.org/display/JENKINS/Quality+Gates+Plugin)”
   2. Configure sonar details to the quality gates plugin
   3. Manage Jenkins → Configure System
   4. Add Sonar Instance, and provide sonar details
      1. Name, URL, username, password
   5. Configure the job
      1. Post build actions add quality gate and add the project key

…………………………………………………………………………………………………………………….

MAVEN :

# Apache Maven 

[**Apache 1**](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.7ilj4m7k0ojo)

[**Apache Maven 2**](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.b0qkdest6ndd)

[What is Junit? 2](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.tcfs30ro3sjr)

[What is integration testing? 2](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.lzp2vwstmpg2)

[What is war/ear/jar ? 2](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.1lldfiwlz8dj)

[What is dependency management? 3](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.8je1y4a7i0kl)

[Dependency: 3](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.e6ukdwv6tn51)

[Transitive Dependency: 3](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.dhtyzwz05ua)

[Dependency Management: 3](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.e9reapwo4ko0)

[**Maven repositories 3**](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.h5ejf41rkb5h)

[**Maven pom.xml 3**](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.7vbl00sjjxm1)

[Pom.xml attributes 4](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.bspcurxl4v31)

[Maven plugins 5](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.oshbk38b9ky)

[Maven plugin examples 5](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.c54n9073v2jv)

[**Installing and configuring Maven on Linux 6**](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.wkosypj5lhft)

[Skipping test cases in maven(Interview Question) 6](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.bcxxfq9flkz4)

[**Snapshot And Release versions 6**](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.wos4d0kck1n8)

[Snapshot Version 6](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.2k2dgroyospi)

[Release Version 7](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.q5ghdd9kvp3y)

[Maven settings.xml 7](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.plw9neuo1xtd)

[There are two locations where a settings.xml file may live: 7](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.e3q4jggrq6kd)

[The Maven install: ${maven.home}/conf/settings.xml 7](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.e3q4jggrq6kd)

[A user’s install: ${user.home}/.m2/settings.xml 7](https://docs.google.com/document/d/1p3exZd3P_ynecHcoKSI9hsAxJPLM5DBVq2ixmfRxm1M/edit#heading=h.abic5lrbp59l)

# Apache

Apache is a company well known for open source products.

Few products from apache

* Maven
* Ant
* Hadoop
* Tomcat
* Elastic search
* etc...

# Apache Maven

Maven is a powerful **build & dependency** management tool for Java software projects.

Maven is primarily known for build and dependency management however it can do more than that.

What is a build tool?

Build tools typically automates the following activities

* Organise project specific files
* Compile the source code
* Generates documentation for the source code
* Runs Junit and integration test cases
* Create software deployable package (war/ear/jar)

## What is Junit?

Junit is a framework for automating unit testing, whenever we add new features to the software we need to retested with all functionalities,  developers write Junit test cases.

## What is integration testing?

Integration test cases are written by QA team, integrations testing is testing end to end flow

QA teams use tools like selenium, QAT for automating integration testing. This can be integrated with build tool like maven.

## What is war/ear/jar ?

War stands **W**eb**Ar**chive, it a format to package web applications

Ear stands for **E**nterprise **Ar**chive, this format is for EJB based applications

Jar stands for **J**ava **Ar**chive

## What is dependency management?

### Dependency:

We our project wants to use a framework, frameworks come as a jar file, this jar is our project dependency.

### Transitive Dependency:

A dependency on which our dependency depends on

https://docs.google.com/drawings/d/s4Nj7g628JinhnppsAvuuMA/image?w=309&h=38&rev=42&ac=1

**a.jar** is our dependency and **b.jar** is our transitive dependency

### Dependency Management:

Maven automatically manages dependencies and transitive dependencies by downloading them from maven repositories.

# Maven repositories

Is the server where maven maintains all the dependencies with different versions of it.

Maven mainly deals with three different repositories

* **Central repository**, maintained over internet
  + Dependencies which are publicly accessible to everyone
* **Remote repository**, maintained within organization
  + Organization specific dependencies which should not be public is maintained here.
* **Local repository**, maintained in our local machine
  + When we run build first time maven downloads dependencies from central, remote and next time onwards picks dependencies from local.

# Maven pom.xml

Is a configuration file used by maven to perform its tasks, Maven looks for pom.xml in the current directory when we run maven commands.

POM stands for Project Object Model

Sample pom.xml

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>in.javahome</groupId>

<artifactId>myweb</artifactId>

<packaging>war</packaging>

<version>0.7.0</version>

<name>myweb Maven Webapp</name>

<url>http://maven.apache.org</url>

<!-- This is comment in XML-->

  <dependencies>

  <dependency>

  <groupId>javax.servlet</groupId>

  <artifactId>javax.servlet-api</artifactId>

  <version>3.0.1</version>

    </dependency>

                      <dependency>

   <groupId>org.springframework</groupId>

   <artifactId>spring-core</artifactId>

   <version>4.3.10.RELEASE</version>

</dependency>

          </dependencies>

 </project>

## Pom.xml attributes

groupId: This reflects the client information for whom we are developing this project. Technically we can put any value in this, but we follow the following convention.

By convention it should be company name or reverse domain name of the company as follows

<groupId>in.javahome</groupId>

<groupId>net.citi</groupId>

<groupId>icici</groupId>

artifactId: This represents project name, some examples

         <artifactId>online-shopping</artifactId>

         <artifactId>online-banking</artifactId>

        <artifactId>order-tracking-system</artifactId>

packaging: This represents the software package format that needs to be created by maven, few examples

         <packaging>war</packaging>

<packaging>jar</packaging>

<packaging>ear</packaging>

version: current version of the project

<version>1.7.2</version>

Version numbers also has conventions, in our example

1 is Major version

7 is Minor version

2 is a patch or bug fix

## Maven plugins

Plugin gives additional functionalities to the tool

### Maven plugin examples

* Compiler plugin (Specify a specific version of a compiler to use)
* Docker plugin (Used to create docker images)
* Jetty server plugin (configuring a web server to deploy over code on jetty web server)
* Maven sonatype nexus plugin (Nexus is a Maven remote repository)

Maven Build lifecycle (Interview Question)

* Validate,  validates *pom.xml* and downloads dependencies
* Compile, Compiles the source code
* Test, runs Junit test cases
* Package, creates a software package (war, ear, jar)
* Verify, verifying integration test cases if configured
* Install, copy the package to the local repository
* Deploy, upload the package to remote repository

**Note:** Here validate, compile, test, package, verify, install, deploy are called as maven goals

Examples using maven commands

* Create  a package using maven

1. Clone the project from git

git clone <https://github.com/javahometech/myweb>

1. cd myweb
2. mvn package

* Install war file to the local repository
  + mvn install

   - Install war file to the local & remote repository

   - mvn install deploy

* Maven target directory

Target directory is created by maven to organise project out files

# Installing and configuring Maven on Linux

Maven depends on java, make sure java is installed.

*sudo wget http://repos.fedorapeople.org/repos/dchen/apache-maven/epel-apache-maven.repo -O /etc/yum.repos.d/epel-apache-maven.repo*

*sudo sed -i s/\$releasever/6/g /etc/yum.repos.d/epel-apache-maven.repo*

*sudo yum install -y apache-maven*

*Verifying maven installation*

*mvn --version*

## Skipping test cases in maven(Interview Question)

*mvn package -DskipTests*

# Snapshot And Release versions

## Snapshot Version

Snapshot version is an version which is currently under development.

If our project depends on snapshot version, every time we build maven get latest copy from central/remote repository.

**Note:** If a version is ending with -SNAPSHOT then it is snapshot version otherwise it is RELEASE version.

Example:

<dependency>

   <groupId>org.springframework</groupId>

   <artifactId>spring-core</artifactId>

   <version>*4.3.10-SNAPSHOT*</version>

</dependency>

## Release Version

A version whose development is completed is called as release version.

**Note:** If a version is not ending with -SNAPSHOT then it is RELEASE version.

<dependency>

   <groupId>org.springframework</groupId>

   <artifactId>spring-core</artifactId>

   <version>*4.3.10*</version>

</dependency>

## Maven settings.xml

This file contains certain information about maven settings, for example

* Local repository path
* Remote repository credentials
* Details about central repositories
* Proxy settings, etc

## There are two locations where a settings.xml file may live:

## The Maven install: ${maven.home}/conf/settings.xml

## A user’s install: ${user.home}/.m2/settings.xml

………………………………………………………………………………………………………………………………….

MAVEN



[**What is Git? 2**](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.qlqrr6xr2zho)

[**Getting git server to manage our projects 2**](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.82tyihh50ip)

[**Create account in github.com and signin 2**](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.ca5isqbdud)

[**Creating new repository in github.com 2**](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.n0la4rcqj9i0)

[**Git client 3**](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.2wm9pdnyg2x7)

[**Install Git bash 3**](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.qbn8j2fq1y1a)

[**Getting code from remote git server 3**](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.1cb8dfwk0aff)

[**Configure git client with email and username 3**](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.z44iltw95jjh)

[**Modifying files and pushing those changes to local and remote repositories 4**](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.uu6ejdh0u8fz)

[Working Area: 4](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.x4uzi6fpn9ph)

[Staging/Index Area: 4](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.dbhmi8c6pywy)

[Checking commit history of this branch 5](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.7oanzd6aqt12)

[Checking commit history of a file 5](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.80p59hpm8ixz)

[Pushing local commits to remote repository 5](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.5p0x56vsql4t)

[Git get specific version of a file 5](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.6d3qoes4xz69)

[**Resolving git push conflicts 5**](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.jglykxe195n6)

[**Resolving git push conflicts using pull 5**](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.v4nv1y8iuybx)

[**Git Fetch 5**](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.y5ucgk3adhww)

[**Git Branch 6**](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.3a4jd44dh8rm)

[Creating a git branch 6](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.n2bkm6e9hhdb)

[Switching a branch 6](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.96ud4jgy4mkf)

[Merging changes in our branch to main branch 6](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.u19ej8m89fww)

[**Listing git branches 7**](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.8bp7z27hc469)

[**Deleting git branches 7**](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.f1nwk6ufl1nh)

[**Git force delete 7**](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.8asl5wdlfxz0)

[**Git merging strategies 7**](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.d1yh2i5xe3zs)

[Fast forward merge 8](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.pcz2n22uyim9)

[Checking list of files modified in a specific commit 12](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.gr0hoxctln82)

[Checking difference between two commits 12](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.fjmdggsi9in3)

[Git branching strategy 13](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.2oncavuvoyvj)

[Git tag 13](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.xkgxo4z29y7q)

[Git init 14](https://docs.google.com/document/d/1Y6S-w3dQjAHqtTCG22Fl9dxBJcZP_q9paTHiZHElMK8/edit#heading=h.372l0g5vx3be)

## What is Git?

* Git is a central repository using which we can manage our project source code
* Git is also called it a version controlling system
* It maintains all modifications happening to a specific file
  + Because of versions troubleshooting and fixing bugs is easy
  + If something goes wrong in current version we can rollback to previous version
* Records who modified, when it is modified(timestamp) and why it is modified
* Git is distributed version controlling system
* Git is fast when it is compared with other version controlling tools
* Multiple developers can easily collaborate and work on same project
* It also works as backing up our project code

## Getting git server to manage our projects

* Down,Install and configure git server on our own machine
* Create account online with
  + GitHub
  + Bitbucket
  + Gitlab
  + CodeCommit(AWS)

## Create account in [github.com](http://github.com) and signin

## Creating new repository in github.com

 What is repository in git?

* In git repository represents a project.

 Steps to create repository

* New Repository → Enter repository name → select public → select README file → Create Repository.

## Git client

For interaction between our local machine and remote git server we need git client.

Git supports both GUI (Graphical User Interface) & CLI (Command Line Interface)

Available git clients

* Git bash
* Source tree
* Tortoisegit
* Git extension
* Smartgit
* Atom
* Etc…

## Install Git bash

Download git bash

Double click the exe file and install with all default selections

In Order to open git bash right click → open git bash here

## Getting code from remote git server

  From your git bash

*git clone* [*https://github.com/javahometech/devops-123.git*](https://github.com/javahometech/devops-123.git)

*cd devops-123*

 Git clone clones the remote copy into our local machine

**Note:** The local copy is called as local repository

## Configure git client with email and username

This information is used by git to record our commits

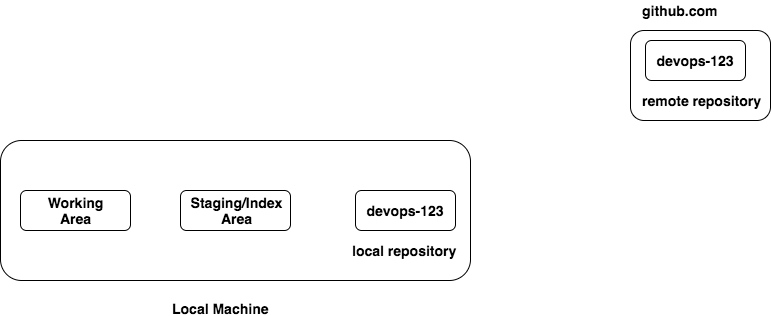
*git config --global user.name “Hari”*

*git config --global user.email “*[*hari@javahome.in*](mailto:hari@javahome.in)*”*

**Note:** --global tells git to use the same information for all the repositories i manage on my local machine.

## Modifying files and pushing those changes to local and remote repositories

* Open README.md file
* Add some content into this file and save it
* Staging a file
  + git add README.md



### Working Area:

Any modifications we do to the local repository those modification are kept under working area.

### Staging/Index Area:

This area is to stage the files we wanna commit to local & remote repository

*git add README.md*

*git add \**

*git add \*.java*

*git add \*.sh*

*git status*

  This provides information about our local repository (working area, stating area & local repository)

**Commit:** When we do commit, it picks the files present in staging and commits to local repository.

*git commit -m 'Learning git tool'*

**Note:** Before we perform commit git name and email must be configured

### Checking commit history of this branch

*git log*

### Checking commit history of a file

*git log README.md*

### Pushing local commits to remote repository

*git push origin master*

*origin → is the alias name for remote repository URL*

*master → is a branch to push our changes into*

### Git get specific version of a file

To move back to older versions of a file

*git checkout <commit-hash-id> <file-name>*

## Resolving git push conflicts

If remote contains a work which is not present in the local then git push is rejected.

To solve this problem we need to pull the remote changes and merge with local changes and push it back.

## Resolving git push conflicts using pull

*git pull origin master*

git pull pulls remote changes to local and merges with local changes by adding a new commit

## Git Fetch

Gets remote changes to the local without merging.

*git fetch origin master*

For merging

*git merge*

## Git Branch

Branch is used to work on a specific task (enhancement, bug, new feature)

Branch provides isolation, i.e. other work will not impact my work

**Master branch**: every git repository comes with a default branch which is called as master branch.

* No one should directly work on master
* Master must contain only well tested code
* In real world most of the guys will not have permissions to push to master

If there is a new task to work on, then we should create a branch and work on it.

### Creating a git branch

*git branch <branch-name>*

This creates a new branch from current branch

### Switching a branch

*git checkout <branch-name>*

### Merging changes in our branch to main branch

We can do this in couple of ways

* By using merge command

           We want changes in ‘master’ so first checkout master and run merge command

*git checkout master*

*git merge <branch-name>*

* Create a pull request

Pull Request

Pull request enables team mates to review and comment on the changes before merging to main branch, we also can see how many file are modified, we also can compare modified file with their old version.

Creating pull request

Push local branch to remote

*git push origin <branch-name>*

From github.com create a pull request

## Listing git branches

*git branch*

*Displays all local branches*

*git branch -r*

*Displays all local branches in remote*

*git branch -a*

*Displays all local plus remote branches*

## Deleting git branches

After merging changes to main branch we can go ahead and delete the branch used for our implementation.

*git branch -d <branch-name>*

## Git force delete

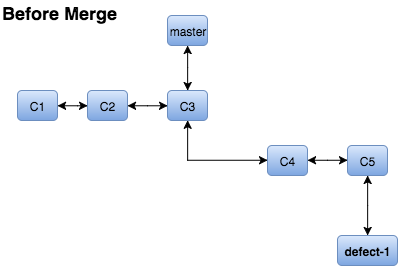
*git branch -D <branch-name>*

Note: if branch is not fully merged we git does not allow us to delete a branch, we can use force delete in such use cases.

## Git merging strategies

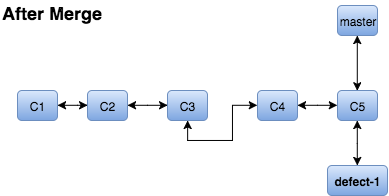
* Fast Forward merge
* Recursive/ Three way merge
* Rebase merge

### Fast forward merge



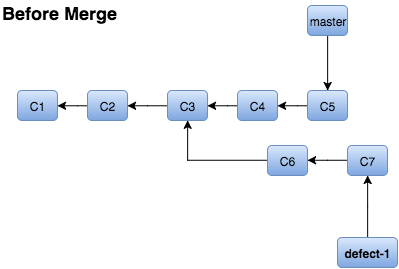
We created defect-1 from master@C3

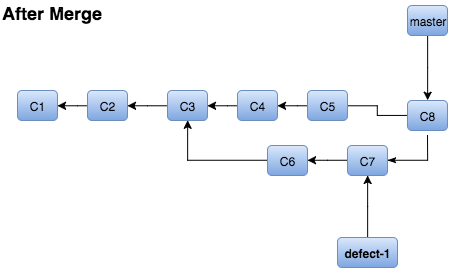
**After defect-1 is created there are no commits on the master**



Fast forward merge move the pointer of master from C3 to C5

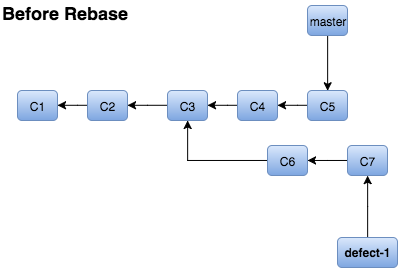
Recursive/ Three way merge

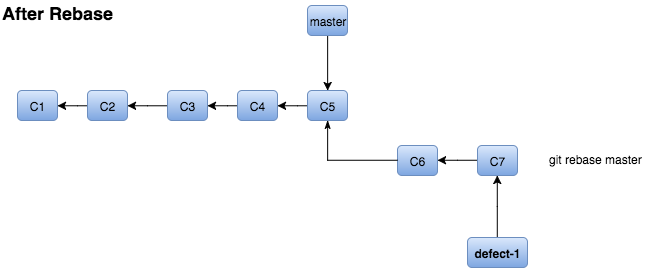




Git created a new commit(C8) for recursive merge.

Rebase Merge





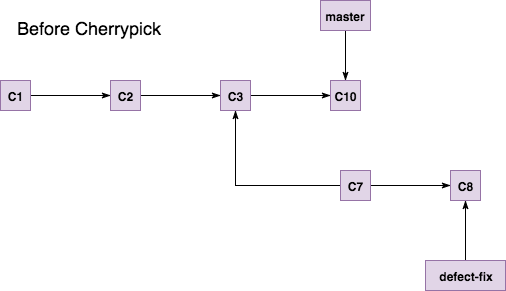
While rebasing there might be conflicts, if so we should resolve them before completing the rebase.

After rebase it enables fast forward merge

Doing rebase on the commits pushed to remote is not recommended.

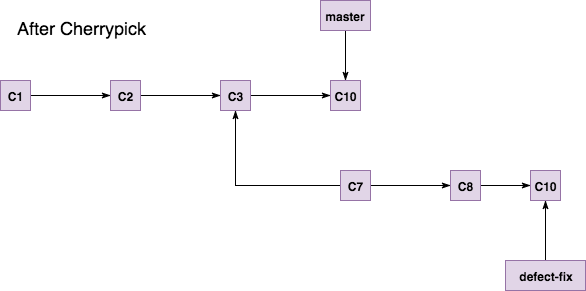
Git cherry pick

It picks a commit from different branch and applies it to current branch.



I wrongly committed C10 into master, now i need changes part of C10 in to my ‘defect-fix’

The solution is run **git cherry-pick <commit-id>** from ‘defect-fix’



Undoing commits

* Reset
* Revert

Git reset:

* Removes the commit from the history

Don't use reset if your commit is pushed to the remote, it causes  problems

git reset HEAD~1

Git revert:

It doesn't remove the commit from the history, instead it reverts changes to the files and makes a new commit.

**Note:** If commits are already pushed to remote we can't use reset instead we should go with revert

### Checking list of files modified in a specific commit

*git show <commit-id>  --name-only --pretty=""*

### Checking difference between two commits

*git diff <commit-id-1> <commit-id-1>*

Display merged branches

*git branch --merged*

Display non merged branches

*git branch --no-merged*

Git stash

git stash temporarily shelves (or stashes) changes you've made to your working copy so you can work on something else, and then come back and reapply them later on

*git stash save*

*Stashes all the changes in working/index area and keeps working/index area clean*

*git stash list*

*To display list of stashes*

*git stash pop*

*It moves changes stashed to working/index area*

*git stash apply*

*It copies changes stashed to working/index area, entry still remains in the stash*

*git stash apply stash@{0}*

*To apply a specific stash(stash@{0} is stash id)*

### Git branching strategy

**Master branch:**

 Master must contain well tested code, application release happens from this branch by

 creating a release ‘tag’.

 In real world no one directly work on master.

**Hotfixes:**

 This branch is used for fixing production defects.

  Hotfixes branch is created from master

**Develop branch:**

 This branch belongs to a specific team, code integration of this team members are

 done on this branch

Develop branch is created from master

**Feature branch:**

 Belongs to a specific developer, where his feature in implemented, after completion of

 a feature changes are merged into his develop branch.

 Feature branch is created from develop branch

**Release branch:**

 This branch is to integrate changes done by multiple teams under their develop branch

### Git tag

In git tag is a pointer pointing to a specific commit, tag is similar to branch but, branch is used for development/bugfix, whereas tag is used for releasing software.

On a branch we can do commits, but on a tag we cannot perform commits.

Creating a tag

*git tag VERSION-1.0.0*

Pushing a tag to remote

*git push origin VERSION-1.0.0*

Delete a tag in local

*git tag -d VERSION-1.0.0*

Display tags

*git tag -d VERSION-1.0.0*

### Git init

   Converts local folder as a git repository.

* Create a folder with ‘gitdemo’
* *cd gitdemo*
* *git init*
* *Add file to this repository*
* *git add \**
* *git commit -m "first commit"*
* Create repository in github.com
* *git remote add origin* [*https://github.com/javahometech/gitdemo.git*](https://github.com/javahometech/gitdemo.git)
* *git push origin master*

### Getting previous version of a file

git checkout fcb83763c4a one.sh

### How to check the files modified between two commits

git diff 541f3204a fcb83763c --name-only

…………………………………………………………………………………………………………………………………..

**Apache Tomcat** 

[**Apache Tomcat 1**](https://docs.google.com/document/d/1Eyp_s_QdUvE-GqXF5Myt3X6JRV4uL3dsb09HBwcnTpY/edit#heading=h.np9speqs3s50)

[**Apache 1**](https://docs.google.com/document/d/1Eyp_s_QdUvE-GqXF5Myt3X6JRV4uL3dsb09HBwcnTpY/edit#heading=h.7ilj4m7k0ojo)

[**Apache Tomcat 1**](https://docs.google.com/document/d/1Eyp_s_QdUvE-GqXF5Myt3X6JRV4uL3dsb09HBwcnTpY/edit#heading=h.b0qkdest6ndd)

What is web application?

Applications accessed via web browsers is called as web application

Technologies used for developing web applications

 - Java

 - .Net

 - Python

 - PHP

 - Node JS (MEAN)

 - Ruby on rails

 - HTML/CSS/Javascript

 - Cold Fusion

What is the runtime for Web applications?

Web servers are the runtime for running web applications

 - Apache (PHP/HTML)

 - Nginx (PHP/HTML)

 - Apache Tomcat  (Java)

 - IIS (.NET)

 - Weblogic (Java)

 - JBoss (Java)

 - Sun glassfish (Java)

 - Pramati Application Server (Java)

 - Websphere (Java)

Apache Tomcat

Is a web server used for deploying java based web applications

Why do we use web servers?

Web servers come with built in features, like

* http/https implementations
* Multi threading capabilities, using thread we can handle multiple requests concurrently
* Application lifecycle management
* Etc..

App servers

Web server features + transaction management + Directory services + EJB containers etc..

* Webshphere
* Weblogic
* JBoss
* etc...

Installing Tomcat on linux machine

* Java is the dependency for tomcat to work
  + Installing java
  + sudo yum install java-1.8.0-openjdk-devel -y
* cd /opt
* Download tomcat

sudo wget <http://www-us.apache.org/dist/tomcat/tomcat-8/v8.5.20/bin/apache-tomcat-8.5.20.tar.gz>

* Untar the file

sudo tar -xf [*apache-tomcat-8.5.20.tar.gz*](http://www-us.apache.org/dist/tomcat/tomcat-8/v8.5.20/bin/apache-tomcat-8.5.20.tar.gz)

* Rename the file

sudo mv apache-tomcat-8.5.20 tomcat8

* Change the ownership of tomcat to ec2-user
  + sudo chown -R ec2-user:ec2-user tomcat8
* Starting tomcat server

           ${TOMCAT\_HOME}/bin/startup.sh

Note: default port of tomcat is **8080**

* Stopping tomcat server

${TOMCAT\_HOME}/bin/shutdown.sh

Deploying web application on the tomcat server

Deployment can be done in many ways

1. Place the war file under ${TOMCAT\_HOME}/webapps

**webapps** folder is called as tomcat's deployment directory

1. Upload war file via tomcat manager application

Troubleshooting tomcat specific issues

Troubleshooting is done with help of tomcat log files

${TOMCAT\_HOME}/logs

Tomcat manager application

Tomcat manager is the web application, using this we manage applications running on tomcat

By default access is restricted to this application.

* Remove the restriction to manager application

vi /opt/tomcat8/webapps/manager/META-INF/context.xml

Remove the *<Value> …….</Value>* from this file.

* Add user to access tomcat manager application

vi /opt/tomcat8/conf/tomcat-users.xml

*<role rolename="manager-gui"/>*

*<role rolename="manager-script"/>*

*<user username="javahome" password="javahome" roles="manager-gui,manager-script"/>*

……………………………………………………………………………………………………………………………………………………..



[**VPC - Virtual Private Network 4**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.37j3b6ik82ea)

[Creating VPC 4](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.x0awzygdp8xi)

[**Subnet 4**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.ycfglmo526av)

[Creating Subnet 4](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.nexc7kf4ulc)

[**EC2 (Elastic Compute Cloud) 4**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.c6gkire6dy97)

[Launch EC2 Instance 5](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.rcu43lwdl4j1)

[SSH Key Pair 5](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.ra2atfj13auu)

[Connecting to remote EC2(VM) instance from our network/laptop 6](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.rapn3dwgebxo)

[Accessing EC2 using Internet Gateway 6](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.3wal21cgf258)

[Login to the remote Linux server 6](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.qafoq564v9sf)

[**Route Tables 7**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.a3gkth4oxx9z)

[**Public IP And Elastic IP 7**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.qpmp049mjnv1)

[Public IP 7](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.nm8x61fyuz94)

[Elastic IP 7](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.m3kp7vsj18fw)

[Allocating EIP and associating with EC2 8](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.koiv0ujevil)

[**Public And Private Subnets 8**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.tdyy7dvp1vwc)

[Public Subnet 8](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.lyikppxlhk5c)

[Private Subnet 8](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.9a61suqv0ush)

[Default VPC 8](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.be42yob3638q)

[How to give internet access to private subnet 8](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.vmhjq8y974td)

[NAT Instance 9](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.8932nbv4bgwe)

[NAT Implementation 9](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.2fm0rksbohi1)

[Testing The configuration we did 10](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.ihmtjo7nvih4)

[Difference between NAT Instance & NAT Gateway 10](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.qn5uc3yio9bd)

[NAT Instance: 10](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.6l952e6grck8)

[NAT Gateway: 10](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.rbf1n5ynd15j)

[Configuring new Users in EC2 11](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.wvr23h36ot0n)

[**VPC Wizards 11**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.z6tzrwum85bb)

[Securing VPC 11](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.t4kzt1bts4cq)

[Security Groups and its Characteristics 11](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.h4fl5vtfq3pn)

[Security Group Characteristics 12](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.nm3p6u3lc2tg)

[Network ACLs and its Characteristics 12](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.32otyupfyefw)

[How Network ACL rules are evaluated? 13](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.riaest62jc32)

[**VPC Peering 13**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.vab5i98j9kxq)

[VPC Peering Demo 13](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.8qvil259rjg4)

[**EC2 Userdata 14**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.qlp3hrfkuzif)

[**EC2 Termination protection 14**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.94zvpebs6lly)

[**Custom AMI (Amazon Machine Image) 14**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.wcwpifwclrpy)

[EC2 Instance resizing 15](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.362bladwmrla)

[**EBS Volumes 15**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.vu1ivvoxlad2)

[Creating EBS Volume 16](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.eipy7boq4ud)

[Resizing EBS volume 16](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.ibiiiix8bkah)

[Adding new volume and mounting it to EC2 16](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.nukzve3wgktg)

[Taking EBS snapshots 16](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.j1eg55by79yk)

[Restoring Volumes from snapshots 16](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.gniz28r5sguj)

[EBS Volume Types 17](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.da64uss1vhqu)

[General Purpose SSD (Solid State Disk) 17](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.9k1416h74ln7)

[Provisioned IOPS 17](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.vst8hwmzu936)

[Throughput Optimised HDD 17](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.lsa34jcf502e)

[Cold HDD 18](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.3xp3xrbgym52)

[Magnetic 18](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.7zofdz7w6u4a)

[Instance Store 18](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.my476nabd91b)

[Launching EC2 with instance store 18](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.ag2ymy4ig779)

[Instance Types 18](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.ey0mhfuhyeeu)

[**EC2 Instances (Purchase Options) 18**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.w1jebgo7c2nw)

[On-Demand Instances 18](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.4p7c35sow4jm)

[Reserved Instances 19](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.yu2kjccm5wf1)

[Spot requests 19](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.aciwf7bimlh8)

[Dedicated Hosts 19](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.b4tk6nl7a5b0)

[Scheduled Instances 19](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.qv2dijh24cng)

[**Elastic Load Balancers (ELB) 19**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.aevqbahanydh)

[ELB Characteristics 20](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.khgv8gtxkedr)

[Configuring ELB (Classic Load Balancer) 20](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.pzihyelkro1e)

[Scaling our application 20](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.wh82t5qobkw6)

[**Auto Scaling 20**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.dxqaxud258yf)

[Terminology 21](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.h3aquono3s30)

[**Identity and Access Management - IAM 21**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.ydgvwkxzgz3w)

[Examples: 22](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.190etgy1wyw2)

[Create IAM user and grant admin permissions 22](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.btxwfhouk02y)

[Create Groups 22](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.t43x2ls2rlff)

[Adding Users to the Group 22](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.tc6o5jsoawuu)

[MFA Multi Factor Authentication 22](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.87jxfzg7ka24)

[Activating MFA (Multi Factor Authentication) 22](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.fhml7t7vlfxh)

[AWS IAM Policies 23](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.886eaubq9cem)

[AWS CLI (Command Line Interface) 26](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.b39c4innp3bt)

[Installing and configuring AWS CLI 26](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.sxlq811svft6)

[AWS CLI Example 26](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.hauu8oh6w3a8)

[IAM Roles 27](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.uong9a9e5mi2)

[Create a Role 27](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.4crbmkxe8tmc)

[Use cases for roles 27](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.ckhuucrd8eir)

[IAM Identity federation 27](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.vc0g5yk1uh7z)

[IAM integration with Identity providers 27](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.40efzwhe2f93)

[**AWS CloudWatch 28**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.re6cr1881fbk)

[Monitoring: 28](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.a95n3ry2qzzg)

[Alarms: 28](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.dgxn39vojij4)

[Cloudwatch Logs: 28](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.mpfbiaobnp8z)

[Cloud watch, Scheduling Tasks 29](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.5pfts8tmxmqi)

[**AWS Lambda (AWS Development) 29**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.ydsi4xmwd1ru)

[**Databases 29**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.4lwnygev8xbu)

[RDS (Relational Database Service) 29](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.uoqaq3c8wk90)

[RDS Multi-AZ Deployments 30](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.850p2lsxcuf6)

[Amazon Read Replicas 31](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.66n1gpqfw20s)

[Working With Backups 31](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.1lenzd9n8wvx)

[The Backup Window 32](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.g7x43hqm91o8)

[Restore From a DB Backups 32](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.e4irxtor1jx)

[Redshift 32](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.syqlhrp9zknh)

[Elasticache 33](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.x6e3vy9k5p44)

[**S3 - Versioning 33**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.xnjuwyw15paw)

[**S3 - Lifecycle management 34**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.biu7fxllaox5)

[**S3 - logging 34**](https://docs.google.com/document/d/1r9kWpCnVkLxs7hvfgKcWCnWkTamEEEDAbx4Xnmp_xU4/edit#heading=h.wqibont9wb1z)

# VPC - Virtual Private Network

* VPC is virtually isolated private network in the cloud
* VPC is region specific
* As of today max we can create 5 VPCs/Region, however we can increase the limit by writing email to the support team
* VPC is free of cost

## Creating VPC

AWS Console → VPC → Your VPCs → Create VPC

Name tag → “WiproVPC”

CIDR → 173.35.0.0/16

And create VPC

# Subnet

* Is a smaller network inside VPC.
* Subnet is availability zone specific
* We can create as many subnets as possible
* Subnets are free of cost

## Creating Subnet

AWS Console → VPC → Subnets → Create Subnet

Name tag → “BangloreSubnet”

VPC → “WiproVPC”

Availability Zone → No preference

CIDR → 173.35.1.0/24

# EC2 (Elastic Compute Cloud)

EC2 is a virtual server in the cloud which has, CPU, RAM, OS and some networking performance.

**Use case:** My client wants to run website on the cloud

* Create VPC and Subnet
* Get Virtual Server from the cloud
  + 1 CPU
  + 1 GB RAM
  + Linux OS
  + 10 GB HDD
* Install Apache web server on the VM
* Deploy the website code on Apache server

## Launch EC2 Instance

AWS Console → EC2 → Launch Instance

1. Choose AMI, AMI is a template containing operating system and few pre installed softwares. Choose (**Amazon Linux)**
2. Choose Instance Type → t2.micro
3. Configure Instance Details
   1. Number of instances → 1
   2. Network → “WiproVPC”
   3. Choose any subnet
   4. Auto asign public ip → enable
4. Add Storage
   1. Storage is nothing but hard disk to the VM
   2. Choose 10 GB
5. Tags
   1. Tags allocates names to the resources and it's also easy to track their bills
   2. Create a tag Key → Name  Value → “WiproWebsite”
6. Configure Security Group
   1. It is a virtual firewall to the EC2 instance
      1. It should allow http
      2. It should allow ssh only from wipro network
      3. Create a new security group
      4. Name → “wiprowebsite-sg”
      5. Choose ssh & http source→ anywhere
      6. Review Instance Launch → Launch
      7. Select key pair
         1. Create new keypair
         2. Key pair name → hari
         3. Download key pair
         4. Launch Instances

## SSH Key Pair

This generates public and private keys, public key is deployed in EC2 and private key is downloaded by us.

For remote server authentication we can use

1. Username with password
2. Username with private key (More secured)

## Connecting to remote EC2(VM) instance from our network/laptop

The problem is, our laptop is in different network and EC2 in VPC is in different network and there is no network connectivity between them, to establish network connectivity we have the following options

1. Via internet, using AWS Internet Gateway
2. Configure VPN between our network and VPC (This uses internet)
3. Direct Connect (A dedicated physical cable between our network and AWS)

### Accessing EC2 using Internet Gateway

1. Create internet gateway, Internet Gateway is a virtual router between VPC and internet.
   1. AWS Console → VPC → Internet Gateways → Create Internet Gateway
   2. In the Name tag enter → “WiproIGW”
   3. By default the state of Internet Gateway is detached
   4. At a given point of time VPC can be attached with only one Internet Gateway
2. Attach the IGW created in previous step to ***WiproVPC***
   1. Select IGW and click Attach to VPC and select WiproVPC
   2. Attaching IGW to the VPC does not give access for our subnets to internet, to grant internet access to the subnets we need to modify the route table of the subnet.
3. In Order to grant internet access to the subnet, we need to add a route to the internet in its route table.
   1. From VP dashboard, select Route Tables
   2. Under route tables select route table associated with our VPC(WiproVPC).
   3. Select routes
   4. Click Edit
   5. Add another route
      1. In the target → select internet gateway
      2. In destination 0.0.0.0/0  (Anywhere)

## Login to the remote Linux server

* For connecting to linux remote servers we use SSH (Secured Shell)
* For connecting to Windows remote servers we use RDP  (Remote Desktop Protocol)

To connect from Windows

* Putty
* Poweshell
* Git bash (Install this)
* Open the git bash
  + ssh -i <path to private key>  ec2-user@<public-ip>
  + ssh -i ~/Downloads/hari.pem ec2-user@13.126.3.189

# Route Tables

Route table is an virtual router in the cloud, route table contains set of routes. Routers are used in networking to connect different networks to communicate.

* Every subnet in VPC is associated with a “*Route Table*”, this enables network communication with internet and communication within VPC with other subnets.
* At the time of creating VPC, a default “Route Table” is implicitly created by AWS, this is also called as Main route table.
* When we create a subnet, this subnet is implicitly associated with Main route table.
* When a new route table is created, it comes with an implicit route, this implicit route enables the subnets associated with this route table to communicate with other subnets in the same VPC.
* We can associate multiple subnets to the same route table.
* A subnet can be associated with only one route table.
* We also can create custom route tables
* We cannot delete main route table
* We can make other route table as Main route table

# Public IP And Elastic IP

Public IP/ Elastic IP  are used to expose the server to internet.

## Public IP

Public Ip is dynamic public, If we stop the server the public IP is released back to public IP pool, when we start the server AWS picks a new public IP from the pool.

## Elastic IP

Elastic IP is a static ip, even if we stop the server IP remains with us.

We can disassociate EIP from the instance and can associate with other instance.

**Note:** *If EIP is associated* ***with non running EC2 instances*** *then we incur minimal charges.*

## Allocating EIP and associating with EC2

VPC → Elastic Ips → Allocate new address

Select EIP → Actions → Associate Address → Select EC2 Instance

# Public And Private Subnets

## Public Subnet

A subnet which is directly exposed to internet is called as public subnet. We make subnets public by adding Internet Gateway.

We use public subnets for internet facing servers like web servers.

## Private Subnet

A subnet which is not directly exposed to internet is called as private subnet.

For example, database servers, build servers, any server which should not be exposed to internet should be part of private subnet.

## Default VPC

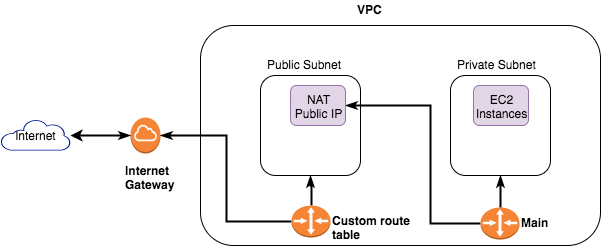
* Default VPC is ready to use VPC
* Default VPC is present in all the AWS regions
* For each availability zone has an subnet
* All subnets in default VPC are public
* Public IP is auto enables for all the subnets
* We can delete default VPC, however it's not recommended
* If we Accidentally deleted default VPC we cannot recreate on our own, we have to create a support ticket to recreate it.

## How to give internet access to private subnet

* We can use either **NAT Instance** or **Nat Gateway,**
* This provides only outbound access and no inbound access
* Internet Gateway give both inbound and outbound access
* NAT Instance and Nat Gateway are more secure than Internet Gateways.
* In case of IGW every machine requires public ip which is a limited resource over internet
* In case of NAT only one public IP is required

### NAT Instance

NAT stands for **N**etwork **A**ddress **T**ranslation, NAt instance is EC2 instance configured with NAT settings, it is used to grant internet access to the private subnets, It allows only outbound traffic and restricts  inbound traffic.



* NAT must be launched in public subnet with public IP
* Modify route table of private subnet and add route to NAT instance
* Internet traffic flow for private subnet

1. An internet traffic is initiated by EC2 in private subnet.
2. This traffic is routed to NAT instance present in public subnet
3. NAT forwards the traffic to internet via Internet Gateway
4. Internet responds back to NAT instance
5. NAT forwards the response back to EC2

* Disable **Source/destination** check for NAT instance
* In AWS all EC2 instance are enabled with source/destination check i.e. EC2 must be the source and destination for the traffic otherwise the traffic is blocked.
* In case of NAT it is not the source/destination of the traffic, so we need to disable source/destination check.

### NAT Implementation

* Create VPC → NATDemoVPC
* Create PublicSubnet (With Custom route table)
* Create PrivateSubnet (With Main route table)
* Launch NAT instance in public subnet

1. EC2 → Launch Instance
2. Select **Community AMIs**
3. Search with nat and select one from the result
4. Select t2.micro
5. Select NATDemoVPC
6. Select Public Subnet
7. Enable public IP
8. Under tags Key → Name and Value → Nat Instance
9. For security group allow all traffic from anywhere
10. Review and launch
11. Select key pair and launch
12. Disable source and destination check for NAT.
    1. Select Nat Instance
    2. Actions → Networking → Change Source/Dest. Check
    3. And click disable

* Select route table of private subnet
* Add the route to the Nat Instance
  + Destination → 0.0.0.0/0
  + Target → Nat Instance
  + Save

### Testing The configuration we did

* Launch EC2 in private subnet
* ssh into this EC2
* ping google.com

## Difference between NAT Instance & NAT Gateway

### NAT Instance:

* It is not Highly available, For making NAT Instance HA, we have to maintain multiple NAT instances and write a script which detects the failure and automatically failovers to the standby NAT instance.
* We have to maintain the NAT instance
* If we face performance issues we need to replace small instance type with big instance type.

### NAT Gateway:

* Highly Available, If primary component fails it auto failovers to the standby component.
* Highly scalable, it automatically resizes based on the demand.
* It's completely managed by AWS

**Note: NAT is not available under free tier (Charges 2000/Month approx)**

## Configuring new Users in EC2

1. Launch EC2 instance
2. Ask the new user(for ex: krishna) to generate private and public keys on his laptop
   1. Open ssh tools (Git Bash, Putty, etc)
   2. ssh-keygen -f krishna
3. Create new user in ec2 instance with name krishna
   1. sudo useradd  krishna
4. Configure public key of the krishna under his user in EC2
   1. su krishna
   2. cd /home/krishna/
   3. mkdir .ssh
   4. **chmod 700 .ssh**
   5. cd .ssh
   6. Open authorized\_keys and paste public key of krishna in this file

# VPC Wizards

Wizards simplifies our VPC, IGW and NAT configurations.

Currently we have 4 types of wizards

1. VPC with public subnet
2. VPC with public and private subnet
3. VPC with Public and Private Subnets and Hardware VPN Access
4. VPC with a Private Subnet Only and Hardware VPN Access

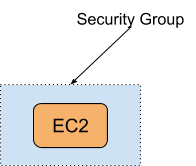
## Securing VPC

We secure VPCs in two ways

1. Security Groups
2. Network ACLs (Access Control List)

## Security Groups and its Characteristics

Security group is a virtual firewall to our EC2 instance, using security groups we secure EC2 instances. Every security group has number of rules.



## Security Group Characteristics

1. Every instance must have at least one security group
2. Same security group can be associated with multiple EC2 instances
3. Every EC2 can have max 5 security groups/eni
4. Security groups has both inbound and outbound rules
5. *Security groups are stateful*
   1. If a traffic is initiated from internet, this traffic is validated by inbound rules of security groups, no matter what is the outbound rule the traffics leaves out.
   2. If a traffic is initiated from EC2, this traffic is validated by outbound rules of security groups, no matter what is the inbound rules the traffics comes in.
6. When we update a rule, it takes effect almost immediately(1-2 seconds delay is expected)
7. Security groups does not have explicit Allow/Deny, Rules we add is to allow and others are implicitly denied.
8. Security group Source
   1. My IP → This traffic is allowed only from My IP
   2. Custom
      1. CIDIR (For example company network)
      2. We also can use a specific IP
      3. We also can use security group in the custom field, i.e. all EC2 instances with this security group can access this instance

## Network ACLs and its Characteristics

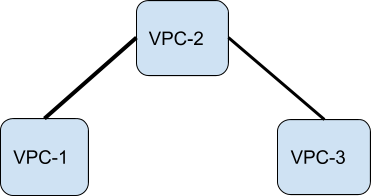
1. Network ACL is firewall acts at subnet level
2. When VPC is created NACL is implicitly created, this is called as default NACL
3. ***Default NACL allows all inbound and outbound traffic***
4. Any subnet created is implicitly associated with default NACL
5. Multiple subnets can be associated with one NACL
6. One subnet can be associated with only one NACL
7. Every rule can be explicitly allowed or denied
8. ***Network ACLs are stateless,*** i.e. inbound traffic is controlled by inbound and outbound traffic is controlled by outbound rule.
9. We have options to block an ip address/network using NACL

### How Network ACL rules are evaluated?

* Rules are evaluated in ascending order
* When matching rule is found rule evaluation stops by applying (Allow/Deny)

# VPC Peering

* Using this we establish network connectivity between VPCs
* VPC peering can be done within same or different accounts
* Both VPCs must be in same region
* CIDR notations of the VPCs should not collide
* For connecting VPCs in different regions we can either use VPN or Direct connect.
* Transitive peering is not supported



For example we have peering connection between VPC-1 → VPC-2 and VPC-2 → VPC-3, transitive peering is not supported, i.e VPC-1 cannot communicate with VPC-3

## VPC Peering Demo

1. Create two VPCs
2. At least one subnet in each VPC
3. VPC → Peering Connections → Create Peering Connection
4. Name → Peering Demo
5. Select VPCs
6. Create
7. Select peering connection → Actions → Accept
8. Modify routing tables in both the VPC

# EC2 Userdata

Using this option we can run scripts at EC2 launch time. There are many use cases for this, for example we wanna configure our servers with chef/puppet we need chef/puppet agents on our machines this can be achieved using userdata.

Example: Using user data

1. Install apache server
2. Start and enable apache server
3. Deploy a sample html file on the apache server

Launch EC2 and at step 3 under user data paste this script

#! /bin/bash

yum install httpd -y

service httpd start

chkconfig httpd on

echo "<h1> User data example </h1>" > /var/www/html/index.html

Note: ***Do not explicitly mention sudo, all the scripts in user data runs internally using sudo***.

# EC2 Termination protection

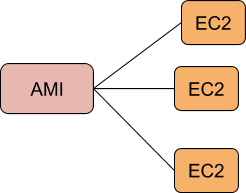
This option disables accidental termination of EC2 instances.

This option can be enabled while launching EC2 or on an existing EC2.

Select EC2 instance → Actions → Change termination protection → Enable/Disable

# Custom AMI (Amazon Machine Image)

AMI is template which contains, Operating System and pre installed softwares



We want a custom AMI which contains

* Apache Installed, Started, Enabled
* Our Application installed no Apache server
* By default AMI created by us is private
* We also can share with other AWS accounts (We can find under AMIs)
* We also can make it public (We can find under **Community AMIs)**
* When we share this image, this image is accessible from the same region.

Creating Custom AMI

* Launch EC2 and install what all you need in the image
* From above EC2 create an image

Launching EC2 instances from the AMI

## EC2 Instance resizing

Instance resizing is a way to scale up or scale down our EC2 instances.

Note: *We must stop the instance before resizing*.

Select Instance → Actions → Instance Settings → Change Instance Type

# EBS Volumes

EBS stands for Elastic Block Store, EBS is Highly Available and highly durable block level storage. By default AWS maintains data across 2 different physical components.

Block level storage is used for File systems, and we also can install and run software applications.

EBS characteristics

1. EBS is persistent store, i.e. permanent storage.
2. To attach a volume both volume and EC2 must be in the same AZ
3. We can add multiple volumes to the instance.
4. We can migrate a volume in one AZ to another AZ by taking a snapshot
5. We can detach a volume from a EC2 and attach to other instance
6. We can take snapshots (backups), and we also can restore from snapshot.
7. Snapshots are stored in S3 (Simple Storage Service)
8. A volume cannot be attached to more than one instance at a time.
9. Volumes support incremental snapshots, i.e. whatever is modified only that is backed up
10. We can resize volumes, but we cannot downsize.
11. EBS works only as a sub service of EC2
12. We also can encrypt EBS volumes

## Creating EBS Volume

We can create volumes in multiple ways

* At the time of launching ec2 instance.
* We also can create a volume from volumes from the EC2 dashboard
* We can create volumes from snapshots

### Resizing EBS volume

EC2 → Volumes → Select Volume → Modify → changes the values and save

Note: We can increase the size, but we cannot decrease it.

### Adding new volume and mounting it to EC2

* Create a new volume
* Attach the volume to EC2
* Login to EC2
  + lsblk (this command displays all the volumes)
  + Create a file system

sudo mkfs -t ext4 /dev/xvdf

Note: the command to check if filesystem is already created

**sudo file -s /dev/xvdf**

* + Create a mount point

sudo mkdir /xyz

* + Mount new volume

           sudo mount /dev/xvdf /xyz

* + Make the above changes persistent

Open  sudo vi /etc/fstab

Add the following entry to the file and save

/dev/xvdf   /xyz ext4    defaults 0 0

### Taking EBS snapshots

EC2 → Volumes → Select the volume → Actions → Create snapshot

### Restoring Volumes from snapshots

EC2 → Snapshots → Select the snapshot → Actions → Create volume

While restoring we can choose

* Different volume type
* Different availability zone
* Different size

## EBS Volume Types

1. General Purpose SSD (Solid State Disk)
2. Provisioned IOPS SSD
3. Throughput optimized HDD
4. Cold HDD
5. Magnetic

### General Purpose SSD (Solid State Disk)

These volume type supports large set of workloads.

Performance:

* IOPS (Input Output Operations/Second)
* Minimum 100 and max 10000 IPOS
* This volume types can perform above baseline when required, by spending credit points, EBS volumes accumulates credit points when it is idle.
* This volume type is good fit for (Dev, Testing, Staging environments)
* Oracle, Sybase, DB2, MySQL, mongodb etc..
* We also can use it for production

### Provisioned IOPS

This volume type provides extreme performance, it designed for machine critical applications.

* Max IOPS, 20000/volume
* We can get upto 65000 IOPS by attaching multiple volumes
* By configuring RAID we can increase IOPS beyond 65000
* Oracle, Sybase, DB2, MySQL, mongodb etc..
* IOPS is the performance criteria not the throughput
* Random reads and writes
* Good choice for production workloads, not recommended for lower environments(dev, testing, staging) in order to save the cost.
* IO size is small

### Throughput Optimised HDD

* This volume type is a good fit for processing big data.
* Like Data warehousing, Big Data, Informatica etc…
* Throughput and sequential reads are dominant performances attributes
* Gives faster access to data
* Use this type where we access data more frequently

### Cold HDD

* Works similar to Throughput Optimized HDD,  storage cost is cheaper
* But this type has additional retrieval charges
* Good option for infrequently accessed data
* Gives faster access to data

### Magnetic

Magnetic is an old generation storage type, we should always consider other types except if our application is specially optimized for this type.

## Instance Store

Is a temporary store, data stored on this store is lost when we poweroff the Instance

Storage cost is very very cheap compared with EBS

Use this types to store temporary data.

### Launching EC2 with instance store

EC2 → Launch Instance → Community AMIs → filter by root device type(Instance store)

### Instance Types

# EC2 Instances (Purchase Options)

* On-Demand Instances
* Reserved Instances
* Spot requests
* Dedicated Hosts
* Scheduled Instances

## On-Demand Instances

* These instances are charged per hour basis
* Even if we use this server for one minute,  it is counted as one hour.
* When instance is not running, there are no charges
* Even if instance is stopped, bill counts for EBS volumes attached to the instance.
* Use this instance type when we need instances for short duration (1 day, 1 week, one month)

## Reserved Instances

We can reserve instances either one or three years, the advantage is we can save upto 70% of the cost when compared with On-Demand instances.

One reserved purchase is done, there is no option to terminate the contract. However we have option to sell it over AWS marketplace.

Payment options

* All upfront
* Partial upfront
* No upfront

## Spot requests

Amazon EC2 Spot instances allow you to bid on spare Amazon EC2 computing capacity for up to 90% off the On-Demand price

## Dedicated Hosts

A Dedicated Host is a physical EC2 server dedicated for your use. Dedicated Hosts can help you reduce costs by allowing you to use your existing server-bound software licenses, including Windows Server, SQL Server, and SUSE Linux Enterprise Server (subject to your license terms), and can also help you meet compliance requirements

## Scheduled Instances

We can buy instances based on our schedule, i.e. we need servers daily, weekly, monthly only in the dedicated hours for example 2-3 hours/day, then this type is the good fit.

# Elastic Load Balancers (ELB)

Load balancers are used to Improve application performance and we also can make our application high availability.



## ELB Characteristics

* ELB is highly available and highly scalable load balancing service from AWS
* AWS maintains redundant copies of ELB, if any AZ fails it automatically failovers to the ELB in different AZ.
* ELB is region specific, i.e. it can load balance instance present in same region.
* ELB supports both external (Internet facing) and internal(not available over internet)
* ELB does health checks on instances and routes traffic to healthy instances, if it finds any instance unhealthy that instance is taken out of rotation, if ELB finds unhealthy instances are healthy, it brings them automatically into rotation.
* ELB, Supports SSL termination.
* ELB can be secured using Security Groups
* ELB types
  + Classic Load Balancer (Old Generation Load Balancer)
  + Application Load Balancer(Specially designed for microservices/ Docker)
  + Network Load Balancer

## Configuring ELB (Classic Load Balancer)

For this demo we need two EC2 instances in two AZs.

This instance must have Apache + A sample html file deployed on apache

## Scaling our application

We have two types of scaling

* Vertical scaling(Increasing instance size)
* Horizontal scaling (Adding additional servers to the environment)

# Auto Scaling

Auto Scaling makes sure our environment has desired capacity of EC2 instances when it is needed.

Auto scaling supports only horizontal scaling.

If auto scaling finds unhealthy instances it is replaced with new instances by terminating the old instance.

Note: Auto scaling service is free, however instances launched by auto scaling is chargeable.

### Terminology

**Auto scaling Group:**

It is a logical group of EC2 instances which participates in the auto scaling.

This group contains

**Minimum** - Minimum number of instances this group must contain always

**Maximum** - Maximum number of instance that can participate in auto scaling.

**Desired:** Is decided at runtime, based on the scaling policies

**Note:** We can use auto scaling with fixed group size

**Launch Configuration:**

Is a templated used by auto scaling for launching new instances.

**Launch configuration contains**

1. AMI, This has to be our custom image, which contains
   1. App dependencies + Web server + Latest application code
2. Instance Type
3. EBS volume
4. Security Group
5. IAM role
6. ssh keypair

We cannot update launch configuration, we only can configure a new one and attach to ASG

Creating Auto Scaling Group

1. Configure Launch configuration with above details
2. Group Name → demo-auto-scaling-group
3. Group size → 2
4. Select VPC and subnets
5. Advanced Details
   1. Select ELB
   2. Health check type is ELB
   3. Select use scaling policies to adjust the capacity of this group

# Identity and Access Management - IAM

Using IAM

* Manage users
* Manage groups
* Manage permissions (user level & Service level)
* Managing Roles
* We can federate with, AD, LDAP, OID or any  other SAML implementations
* We also can integrate with identity providers like (Amazon, Facebook, Google, LinkedIn, etc)
* We also can grant cross-account access
* We can enable MFA (Multi Factor Authentication)

## Examples:

### Create IAM user and grant admin permissions

IAM Dashboard → Users → Add User

Select AWS access type → AWS Management Console access

Select custom password

Select, Require password reset

Attach existing policies directly

### Create Groups

  Groups → Create New Group → Select a policy and create.

### Adding Users to the Group

Create User → Select the group and add it

### MFA Multi Factor Authentication

This provides additional level of security to our account, it is always recommended to enable this for all the users.

We can link two types of tokens

1. Hard token (RSA device)
2. Soft token
3. The following apps are supported as a soft token.
   1. Android, iphone, blackberry supports
      1. Google Authenticator
   2. Windows phone supports
      1. Authenticator

### Activating MFA (Multi Factor Authentication)

First we need the install the app on the mobile

Open the app and click BEGIN SETUP

From AWS IAM Dashboard

Select → Activate MFA on your root account  → Manage MFA → A virtual MFA device → Next → Next → From mobile app scan the QR codes and enter the two numbers one after another, number displayed on the device.

### AWS IAM Policies

Policy is a JSON document, which talks about the permissions

Policy Syntax

{  
  "Statement":[{  
    "Effect":"*effect*",  
    "Action":"*action*",  
    "Resource":"*arn*",  
    "Condition":{  
      "*condition*":{  
        "*key*":"*value*"  
        }  
      }  
    }  
  ]  
}

There are various elements that make up a statement:

* Effect: The effect can be Allow or Deny. By default, IAM users don't have permission to use resources and API actions, so all requests are denied. An explicit allow overrides the default. An explicit deny overrides any allows.
* Action: The action is the specific API action for which you are granting or denying permission. To learn about specifying action, see [Actions for Amazon EC2](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-policy-structure.html#UsingWithEC2_Actions).
* Resource: The resource that's affected by the action. Some Amazon EC2 API actions allow you to include specific resources in your policy that can be created or modified by the action. To specify a resource in the statement, you need to use its Amazon Resource Name (ARN). For more information about specifying the ARN value, see [Amazon Resource Names for Amazon EC2](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-policy-structure.html#EC2_ARN_Format). For more information about which API actions support which ARNs, see [Supported Resource-Level Permissions for Amazon EC2 API Actions](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-supported-iam-actions-resources.html). If the API action does not support ARNs, use the \* wildcard to specify that all resources can be affected by the action.
* Condition: Conditions are optional. They can be used to control when your policy is in effect. For more information about specifying conditions for Amazon EC2, see [Condition Keys for Amazon EC2](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-policy-structure.html#amazon-ec2-keys).

To learn policies we need to know JSON

JSON stands for *Java Script Object Notation,*

JSON is used for couple of reasons

1. To exchange information between applications (Facebook server and facebook mobile app)

All mobile apps exchange data with their server using JSON

1. To represent/store data

JSON syntaxes

*{*

***"name"****: "John",*

***"age"****: 31,*

***"city"****: "New York"*

***“isManager”****: true*

*}*

The above is an JSON document, which represents three properties name, age and city.

JSON Data types

* Text
* Number (Integer and floating point)
* Boolean ( true/false)
* List
* Objects and Nested Objects

{

 "name": "John",

 "age": 31,

 "salary": 10000.5,

 "isManager": false,

 "address": {

   "houseNo": "12-45",

   "landmark": "Dentalcollegeroad",

   "state": "KA",

   "pin": 560037

 },

 "languages": [

   "English",

   "Hindi",

   "French"

 ]"

}

ARN (Amazon Resource Names)

arn:aws:ec2:*ap-southeast-2*:aws-account:*resource-path*

Few examples, the following resource represents ec2 with instance id

arn:aws:ec2:ap-southeast-2::instance/i-05c217fc44889449e

Requirement:

Create a policy document which allows, start and stop permissions for a specific instance

{

   "Version": "2012-10-17",

   "Statement": [

       {

           "Sid": "Stmt1506566739000",

           "Effect": "Allow",

           "Action": [

               "ec2:StartInstances",

               "ec2:StopInstances"

           ],

           "Resource": [

"arn:aws:ec2:us-west-2:425139712349:instance/i-0499de4a913a8cf75"

           ]

       },

       {

           "Sid": "describeinstances",

           "Effect": "Allow",

           "Action": [

               "ec2:Describe\*"

           ],

           "Resource": [

               "\*"

           ]

       }

   ]

}

Requirement:

Create a policy document which allows, start and stop permissions for a specific instance type, i.e. t2.micro

{

   "Version": "2012-10-17",

   "Statement": [

       {

           "Sid": "Stmt1506912387000",

           "Effect": "Allow",

           "Action": [

               "ec2:DescribeInstances"

           ],

           "Resource": [

               "\*"

           ]

       },

       {

           "Sid": "Stmt1506912495000",

           "Effect": "Allow",

           "Action": [

               "ec2:StartInstances",

               "ec2:StopInstances"

           ],

           "Condition": {

               "StringEquals": {

                   "ec2:InstanceType": "t2.micro"

               }

           },

           "Resource": [

               "\*"

           ]

       }

   ]

}

## AWS CLI (Command Line Interface)

We also can manage AWS resources from command prompt, CLI is import because we use it for automation.

### Installing and configuring AWS CLI

1. Install AWS CLI
2. Create a user with programmatic access in IAM
   1. access key ID (Username)
   2. secret access key(Password)
3. Configure AWS CLI with above credentials, by running following commands
   1. aws configure

This prompts for access key ID and secret access key and region

Note: From this machine we can, we can access services in AWS via CLI, API, SDK (programs)

Which internally uses the credentials configured via CLI.

### AWS CLI Example

1. Launch ec2 from AWS CLI

*aws ec2 run-instances --image-id ami-aa5ebdd2 --instance-type t2.micro*

1. Describe all the instances

*aws ec2 describe-instances*

1. Stop EC2 instance

aws ec2 stop-instances --instance-ids i-0fda0100e2a447421

1. Start EC2 instance

aws ec2 start-instances --instance-ids i-0fda0100e2a447421  i-0bde64e68ef435562

1. Create EBS volume

create-volume --availability-zone us-west-2a --size 10 --volume-type gp2

1. Attaching a volume to instance

aws ec2 attach-volume --instance-id i-01b71ab575ae3f5e5 --volume-id vol-09f368f82e0ced451 --device xvdh

## IAM Roles

Role works similar to users, but we assign role to any entity.

### Create a Role

Roles → Createrole → Select EC2 → EC2 → Next → Select policy → Review → create policy

## Use cases for roles

1. If one AWS service wants to access other aws service it needs permissions, that is assigned by creating a role.

Example: EC2 instance wants to access S3, EC2 needs permissions, it is grated by creating and attaching a role to EC2.

1. IAM roles is also used for cross account access

### IAM Identity federation

We can federate with the users who are not present in AWS, i.e. for example a company is maintaining users and their credential in AD (Active Directory), we can grant access to users in  AD to AWS, this is done with identity federation in AWS.

### IAM integration with Identity providers

We can integrate AWS with identity providers like (Facebook, google, linkedin, Amazon, etc)

# AWS CloudWatch

Cloud watch is monitoring tool, we can monitor resources and applications, we also can schedule tasks using cloud watch, we also can gather logs generated by applications, we also can setup alerts.

## Monitoring:

We can monitor aws resources like (EC2, EBS, ELB, S3, RDS, etc).

Cloudwatch automatically collects lots of metrics, for EC2 (CPU Utilization, Disk Reads, Disk Writes, Network packets in, network packets out, etc..)

We also can publish custom metrics and publish to the cloud watch

Note: *Cloud watch does not collect Memory and Volume metrics*

We also can create custom metrics and publish.

AWS supports two types of monitoring

* Basic monitoring, free ( metric values are published to cloudwatch every 5 minutes)
* Detailed monitoring, is charged (metric values are published to cloudwatch every minute)

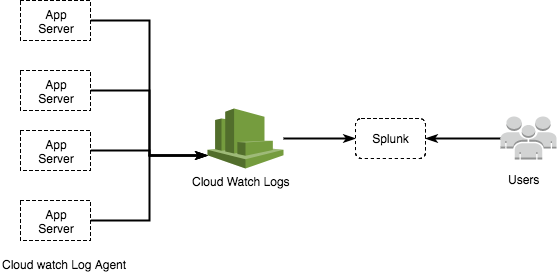
## Alarms:

Based on metric values we can create alarms and send notifications.

Alarms can be integrated with

* Notifications(Email/SMS)
* Auto Scaling Groups
* EC2 Actions

## Cloudwatch Logs:



## Cloud watch, Scheduling Tasks

# AWS Lambda (AWS Development)

AWS lambda Is ***serverless architecture***, i.e. we have to write the code and submit it to AWS, AWS takes care of provisioning servers.

As of today it supports four languages

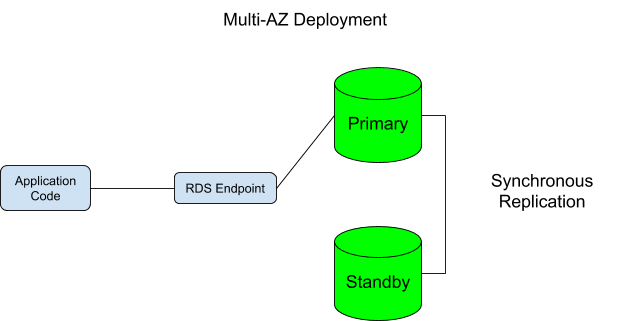
* Node JS
* Python (Most prefered language)
* Java
* C#

# Databases

## RDS (Relational Database Service)

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups. It frees you to focus on your applications so you can give them the fast performance, high availability, security and compatibility they need.

## RDS Multi-AZ Deployments

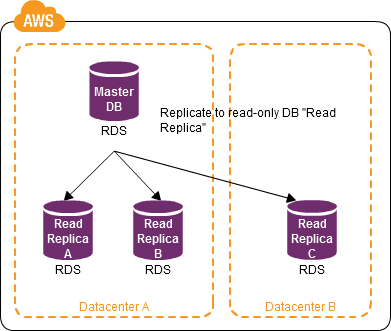


Amazon RDS Multi-AZ deployments provide enhanced availability and durability for Database (DB) Instances, making them a natural fit for production database workloads. When you provision a Multi-AZ DB Instance, Amazon RDS automatically creates a primary DB Instance and synchronously replicates the data to a standby instance in a different Availability Zone (AZ)

Each AZ runs on its own physically distinct, independent infrastructure, and is engineered to be highly reliable. In case of an infrastructure failure, Amazon RDS performs an automatic failover to the standby (or to a read replica in the case of Amazon Aurora), so that you can resume database operations as soon as the failover is complete. Since the endpoint for your DB Instance remains the same after a failover, your application can resume database operation without the need for manual administrative intervention.

**Note: Multi-AZ deployments provide enhanced availability and durability, it’s not meant for improving application performance.**

## Amazon Read Replicas



Amazon RDS Read Replicas provide enhanced performance and durability for database (DB) instances. This replication feature makes it easy to elastically scale out beyond the capacity constraints of a single DB Instance for **read-heavy database workloads**. You can create one or more replicas of a given source DB Instance and serve high-volume application read traffic from multiple copies of your data, thereby increasing aggregate read throughput. Read replicas can also be promoted when needed to become standalone DB instances.

## Working With Backups

Amazon RDS creates and saves automated backups of your DB instance to S3(Simple Storage Service). backing up the entire DB instance and not just individual databases.

Default retention period is 7 days & it can be max upto 35 days, if you do not need automated backup then retention period should be 0.

Amazon RDS creates automated backups of your DB instance during the backup window of your DB instance. If necessary, you can recover your database to any point in time during the backup retention period.

You can also backup your DB instance manually, by manually creating a DB snapshot

## The Backup Window

Automated backups occur daily during the preferred backup window. If the backup requires more time than allotted to the backup window, the backup continues after the window ends, until it finishes.

During the automatic backup window, storage I/O might be suspended briefly while the backup process initializes (typically under a few seconds). You may experience elevated latencies for a few minutes during backups for Multi-AZ deployments.

For MariaDB, MySQL, Oracle, and PostgreSQL, I/O activity is not suspended on your primary during backup for Multi-AZ deployments, because the backup is taken from the standby.

For SQL Server, I/O activity is suspended briefly during backup for Multi-AZ deployments.

If you don't specify a preferred backup window when you create the DB instance, Amazon RDS assigns a default 30-minute backup window which is selected at random from an 8-hour block of time per region.

## Restore From a DB Backups

If any failure occurs to the database instance, we can restore it from the snapshots.

You can create a DB instance by restoring from this DB snapshot. When you restore the DB instance, you provide the name of the DB snapshot to restore from, and then provide a name for the new DB instance that is created from the restore. You cannot restore from a DB snapshot to an existing DB instance; a new DB instance is created when you restore.

## Redshift

Amazon Redshift is a fast, fully managed [data warehouse](https://aws.amazon.com/data-warehouse/) that makes it simple and cost-effective to analyze all your data using standard SQL and your existing Business Intelligence (BI) tools. It allows you to run complex analytic queries against petabytes of structured data, using sophisticated query optimization, columnar storage on high-performance local disks, and massively parallel query execution. Most results come back in seconds. With Amazon Redshift, you can start small for just $0.25 per hour with no commitments and scale out to petabytes of data for $1,000 per terabyte per year, less than a tenth the cost of traditional solutions.

## Elasticache

Amazon ElastiCache is a web service that makes it easy to deploy, operate, and scale an in-memory data store or cache in [the cloud](https://aws.amazon.com/what-is-cloud-computing/). The service improves the performance of web applications by allowing you to retrieve information from fast, managed, in-memory data stores, instead of relying entirely on slower disk-based databases. Amazon ElastiCache supports two open-source in-memory engines:

* [Redis](https://aws.amazon.com/elasticache/what-is-redis/) - a fast, open source, in-memory data store and cache. [Amazon ElastiCache for Redis](https://aws.amazon.com/elasticache/redis/) is a Redis-compatible in-memory service that delivers the ease-of-use and power of Redis along with the availability, reliability and performance suitable for the most demanding applications. Both single-node and up to 15-shard clusters are available, enabling scalability to up to 3.55 TiB of in-memory data. ElastiCache for Redis is fully managed, scalable, and secure - making it an ideal candidate to power high-performance use cases such as Web, Mobile Apps, Gaming, Ad-Tech, and IoT.
* Memcached - a widely adopted memory object caching system. ElastiCache is protocol compliant with Memcached, so popular tools that you use today with existing Memcached environments will work seamlessly with the service.

# S3 - Versioning

1. To avoid accidental deletion of objects
2. It maintains multiple versions of same objects
3. We have control to retrieve/delete any version.
4. When we delete an object it adds delete marker, instead of actually deleting an object.
5. Downside is, every version occupies its complete storage capacity
6. Ones versioning is enabled there is no option to disable, we can only suspend new versions.

# S3 - Lifecycle management

You can manage an object's lifecycle by using a lifecycle rule, which defines how Amazon S3 manages objects during their lifetime.

Using lifecycle rules we can autom transition of objects to cheaper storage like (Standard IA, Glacier) to save money.

Using lifecycle rules we can autom expiration of objects.

Lifecycle use cases:

Requirement -1

I want to expire previous versions, if it is 30 days old

Requirement -2

We are working for a bank and they are storing log files in S3, to same cost, they want the following lifecycle

Standard --after 6 months → IA  --- after 3 months → Glacier

# S3 - logging

This allows us to track all API activities happening on this bucket

S3 - Hosting Static Websites

We can host static websites without need of servers.

It does not support dynamic websites like servlets, jsp, python django, .Net etc...

S3 - Events

Receive notifications when specific events occur in your bucket

Usecases:

1. Resize image when image is uploaded to the S3 bucket.
2. Create thumbnails for the images
3. When a CSV file is uploaded to S3, read the file content and store in dynamodb.
4. Whenever a mp4 videos are uploaded send email notification

Cross Region Replication

IF we upload an object into source bucket, it can be automatically replicated into destination bucket which is in different region.

Usecases:

1. Disaster recovery
2. Improve application performance
3. If there is a government rule which mandates the data should be available in a s specific geographical regions

Note:

1. Cross Region Replication works only if versioning is enabled on both the buckets.
2. It replicated only future uploads, it will not replicate the objects which exists before enabling CRR

…………………………………

1. Can we span VPC across multiple regions?
   1. No, VPC is specific to the region
2. Can we attach more than one Internet Gateway to the VPC?
   1. No, at a given point of time we can have only one Internet Gateway attached to VPC
3. How do you check an internet access is present for a subnet?
   1. By checking the route table of the subnet, if there is a route to internet gateway, subnet has internet access otherwise not.
4. We have connectivity issue with EC2, where do we troubleshoot this issue?
   1. Check security group rules, and make sure it is not blocking the connectivity
5. Difference between Public Ip and Elastic IP?
   1. Public IP is dynamic, Elastic IP is static.
6. Difference between Public & Private Subnets
   1. If a subnet is accessible from internet then it is Public subnet, if a subnet is not accessible from internet it is private subnet
7. We need internet access to the private subnet for some server maintenance how do you grant the internet access?
   1. NAT Instance or NAT Gateway
8. Where we should put NAT instance/Gateway?
   1. It must be in public subnet with Public/EIP
9. For a machine in public subnet we do not wanna grant internet, how?
   1. Do not assign public/ elastic ip
10. What are differences between NACL and Security groups?
    1. Find it from the document
11. What is VPC peering
    1. VPC peering is to establish network connectivity between VPCs
12. What is user data in EC2?
    1. User data is the option to execute an script at the time of launching EC2 instances.
13. Can we resize EC2 instance?
    1. Yes, Select Instance → Actions → Instance Settings → Change Instance Type
    2. Note: *We must stop the instance before resizing*.
14. As a root volume which volume types we can use?
    1. As of today we can use
       1. General Purpose SSD (Solid State Disk)
       2. Provisioned IOPS SSD
       3. Magnetic
15. How to increase EBS IOPS beyond 65,000?
    1. Configure RAID0
16. What is Instance store?
    1. Is an temporary store, data is lost when we poweroff ec2 instance.
17. Can we load balance ec2 instance present across multiple regions using ELB?
    1. No, ELB can load balance instances in same region
18. How do we load balance instances across multiple regions in AWS?
    1. Using Route53.
19. What is connection draining in ELB?
    1. If an instance is not healthy and before taking it out of service, ELB gives certain duration to the instance to complete its pending requests.
20. What is Stickiness ELB?
21. Auto Scaling enables horizontal scaling, can we implement automation for vertical scaling?
22. In auto scaling group, how to protect an instance from termination when scale-in activity is performed by ASG?
    1. We can enable scale in protection on instances

……………………………………….