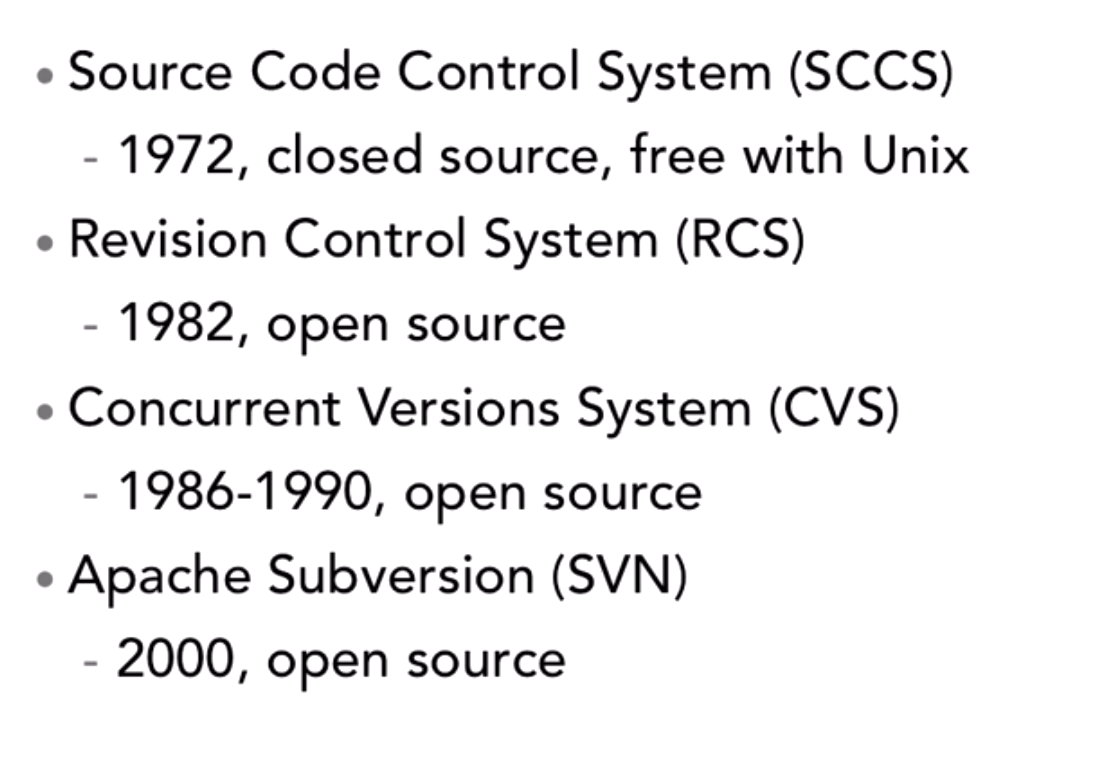
jGit is run locally

* It is a version control system (VCS) or source code management (SCM).
* It is used to track the changes in files.
* It will maintain multiple versions of the same file.
* It is platform independent.(redhat,mac,linux)
* It is free and open source.
* They can handle larger projects efficiently.
* They save time and developers can fetch and create pull requests without switching.

**VCS(version control system) Histroy**



SCCM-(1972)To track only one file –access to single person

RCS-Track multiple files but not directories-access to single person

CVS-Track multiple files and directories.-access to single person

SVN-Track multiple files and directories access to multiple people

GIT-(2005) Distributed version control system

**RVS(Revision Control System)**

* Is an early version control system. IT is a set of Unix commands that allow user to develop and maintain program code or documents. With RCS, users can make their own revisions of documents, commit changes, merge them.
* It will track only multiple files but not directories.
* Allowed for single user only.

**Concurrent Versions System:**

* An important component of source configuration management(SCM).using it you can record the history of source files and documents.It fills a similar role to the free software RCS,PRCS,Aegis packages.CVS is a production quality system in wide use around the world,including many free software projects.
* Tracks multiple files and directories.
* Allowed for single user only.

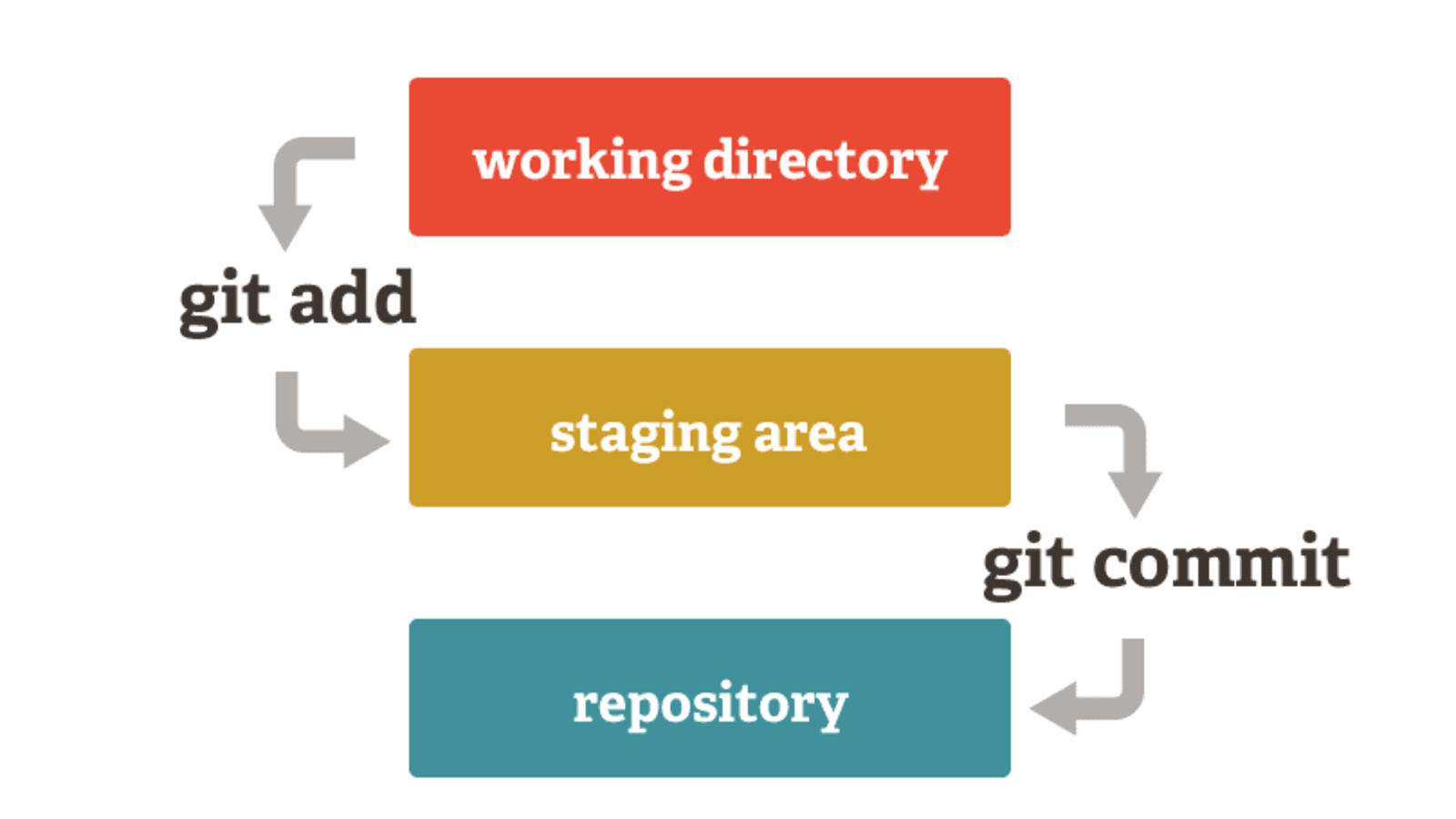
**Subversion**

* Subversion is an open-source, centralized version control system that is available for everyone at zerocost.
* It is designed to handle minor to major projects with speed and efficiency.It is developed to coordinate the work among programmers. The version control allows you to track and work together with your team members at same workspace.
* Allowed multiple users.

FREE : GIT,SVN

PAID : BITBUCKET,P4,STASH

GIT STAGES



**working directory**

The working area is where files that are not handled by git.

These files are also referred to as "untracked files."

In this stage git is only aware of having files in the project.

It will not track these files until we commit those files.

**Staging Area**

Staging area is **files that are going to be a part of the next commit**, which lets git know what changes in the file are going to occur for the next commit.

In other words ,in the next version of your project.

**Repository**

Repository in GIT is considered as your project folder.

A Git repository tracks and saves the history of all changes made to the files in a Git project.

It saves this data in a directory called . git , also known as the repository folder.

Git uses a version control system to track all changes made to the project and save them in the repository.

**Types of repos**

* **Local Repository:** Local repositories is everything in your .GIT directory.

Mainly what you will see in your local repository are all of your checkpoints or commits.

It is the area that saves everthing .(don’t delete it)

* **Central Repository:**It will be present in the Github where you can share all your filles.

You need to add and commit your files before you push into Github.

* **Remote Repository:**It will be present on remote hosts where you can share all youe files to remote machines.

**How to install Git in EC2 instance**

**Note:**YUM is **the primary package management tool for installing, updating, removing, and managing software packages in Red Hat Enterprise Linux**.-y is permission

yum install git –y

git init

To check if git is installed or not type

git –version(2.32.0)=>(2-Major Version for 1-2 years,32-Miniversion every 6months,0-Patches everyweek2,4)

**To install git repository i.e local repository**

git init . (. Is current directory)

**STEPS TO COMMIT A FILE**

1. Create a file : touch filename
2. Now add that file : git add.(Dot represents current directory)
3. Commit the file with message : git commit –m”commit message you want” filename
4. To see details of that file : git log

Now all those things will be done under root user

If you want to be done by another user or as under your name we need to configure it.

**CONFIGURATION OF USER**

1. Open the command line.
2. Set your username: git config --global user.name "FIRST\_NAME LAST\_NAME"
3. Set your email address: git config --global user.email [MY\_NAME@example.com](mailto:MY_NAME@example.com)

**GIT-HUB**

**-**Github us a web-based platform used for version controland collaboration.

-It lets you and others work together on projects from anywhere.

-Team members can work on files and easily merge their changes in with the master branch of the project

Now if you want to push your code Github

**git remote add origin url**

**git push –u origin branch-name**

**token:** **ghp\_4nUrsKBfgCCULJ4DQBQUUc6RCy7tBo0K9Mtg(june152022token)**

go to Github and check the files that you have pushed

**\*\*Repositories in GIThub are Public,Private**

**GIT MERGE**

-If you want to merge branch-1 with branch -2 switch to branch-1 first and give command

**git merge branch-2**

-Now that command had merged the content of branch-1 and branch-2.

-Whatever the content in branch-1 will be seen in branch-2 now.

**GIT FORK**

**-**A fork **is a rough copy of a repository.Forking a repository allows you to freely test and debug with changes without affecting the original project.**

**GIT BRANCHES**

**A branch represents an independent line of development or also**

Git, branches are a part of your everyday development process. Git branches are effectively **a pointer to a snapshot of your changes.**

The git branch command lets to you create,list,rename,and delete branches.

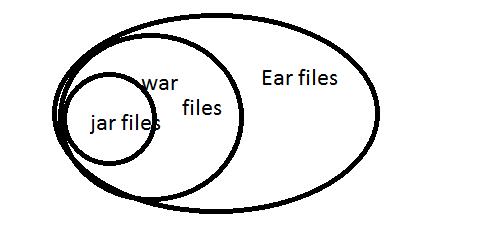
The default branch name in GIt is master.

* To see current branch : git branch
* To add new branch : git branch branch-name
* To switch branches : git checkout branch-name
* creates a new branch and checks out the new branch: git checkout –b branch-name
* To rename a branch : git branch –m old new
* To clone a specific branch : git clone –b branch-name repo-URL
* To delete a branch : git branch –d <branch>

The –d option will delete the branch only if it has already been pushed and merged with the

**MAVEN**

* Maven is a build automation tool developed by Apache s/w foundation.
* It is based on POM(Project Object Model).(POM.xml)xml:eXtensible Markup language.
* It is build tool and manages dependencies.
* It can build any no. of projects into desired output such as .jar(JavaArchive-.DotClassfiles(executable files)),.war(WebArchive ex -htmlfiles),and metadata(Data about data i.e folder having folder info).

****

* It is mostly used for java based projects.
* It was initially released on 13 july 2004.
* Maven is written in JaVa.
* Meaning of Maven is Accumulator of Knowledge in Yiddish language..
* Maven helps in getting the right jar file for each projects as there may be different versions of separate packages.
* For downloading dependencies visit mvnrepository.com.

The term **build** may refer to **the process by which source code is converted into a stand-alone form that can be run on a computer or to the form itself**.

A artifact is **a by-product produced during the software development process**. It may consist of the project source code, dependencies, binaries or resources, and could be represented in different layout depending on the technology.

**Dependencies:** It refers to the java libraries that are needed for the project.

**Repositories**: Refers to the directories of Packaged jar.files.

**Build tools:**

* C,C++ :make file
* .Net :Visual Studio
* Java :Ant,Maven,Gradle

**Java Project Structure**

**#Source code**

**#Test code**

**#Project Structure(Assets,directories,resources)**

**#Dependencies/Library**

**#Configuration**

**#TaskRunner(build,test,run)**

**#Reporting**

**Problems Without Maven**

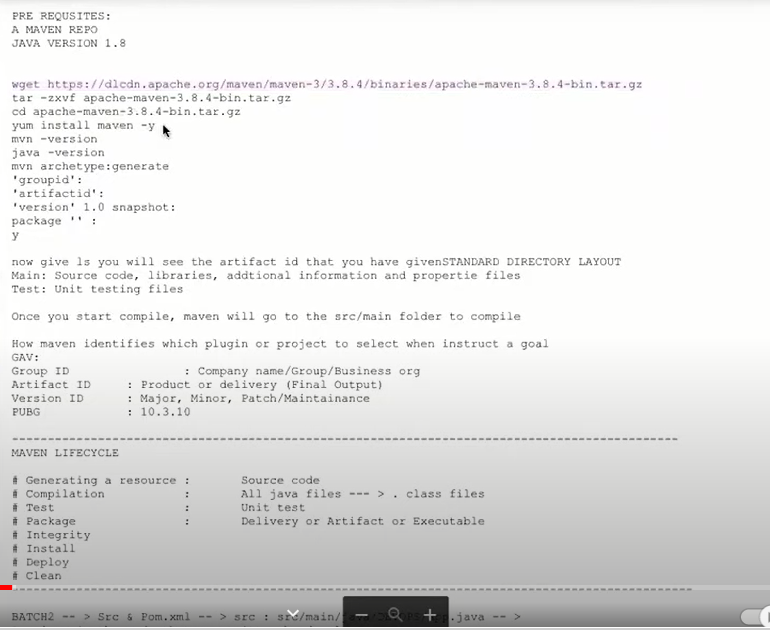
* **Adding set of jars in each project:** In case of Struts, springs we need to add jar files in each project. It includes all dependencies of jars also.
* **Creating the right project structure:** We must create the right project structure in Servlet,Struts etc. Otherwise it will not be executed.
* **Building and Deploying the project:** We must build and deploy the project,so it may work.

**What Maven does?**

* It makes project easy to build.
* Provides Project Information(log document, cross reference,sources,mailing list, unit test)
* Easy to add New Dependencies.
* Apache Maven helps to manage:Build,Dependencies,Releases,Distribution,Reports.

**WHAT IS BUILD TOOL?**

* A build tool takes care of everything for building process. It follows
* Generate source code
* Generate Documentation for source code.
* Compile source code.
* Install the package code in the local repo. Server repo or central repo .
* POM refers the XML files that have all information regarding project and configuration details.
* Main configuration file is in pom.xml.
* It has description of the project details regarding version and configuration management.
* The XML file is in the project home directory.

****

**POM.xml contains**

**ANSIBLE**

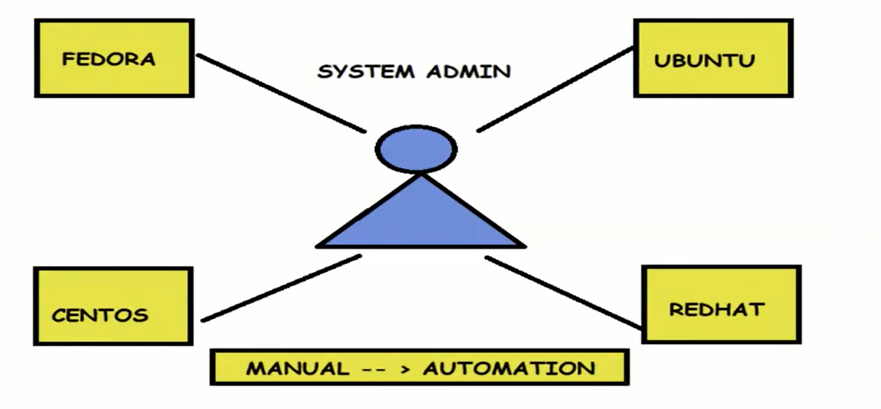
* It is configuration management tool.
* Configuration :Ram, Storage, OS, Software and IP address of device.
* Management: Update, Delete, Add.
* Ansible is simple open source IT engine which automates applications deployment.
* Orchestration ( orchestration **allows you to define your infrastructure once and use it wherever and however you need**.),Security and compliance
* Uses YAML scripting language which works on KEY-VALUE PAIR
* Ansible GUI is called Ansible Tower. It was just Drag and Drop.
* Used PYTHON for Backend.

**HISTORY**

* Michael DEhhan developed Ansible and the Ansible project began in Feb

2012.

* Ansible was taken over by Red-hat.
* Ansible is available for RHEL, Debian , CentOS , Oracle Linux
* We can use this tool whether you servers are in On-perm or in the cloud
* It turn your code into infrastructure i.e. your computing environment has some of the same attributes as your application.



If a system admin has to install those Linux flavors across all the systems on his company ,then he has to do manually. In manual work there might be some errors so we use here automated tools like Ansible, Chef, Puppet etc.

ANSIBLE :PUSH CHEF :PULL

**PUSH**: If we have many servers then it will push the notification for updates in all devices.

**PULL**: It will go to client server and ask for the notifications update.

**Advantages**

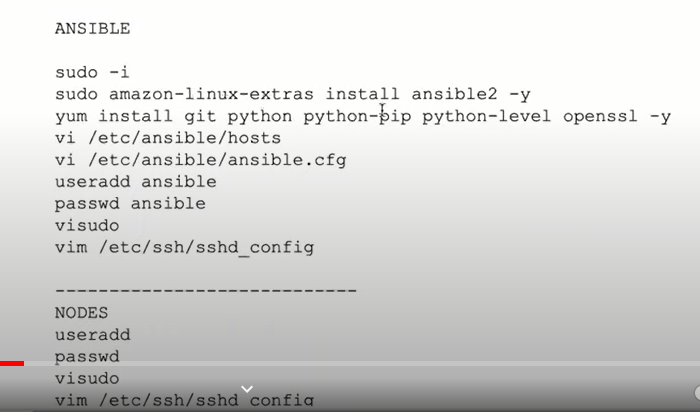
* Very consistent and light weight and no constraints regarding the OS or underlying HardWare.
* Secure less due to Agent less capability and Open SSH Security features.
* Doesn’t require any special system admin skills to install and use it (YAML).
* Push mechanism.

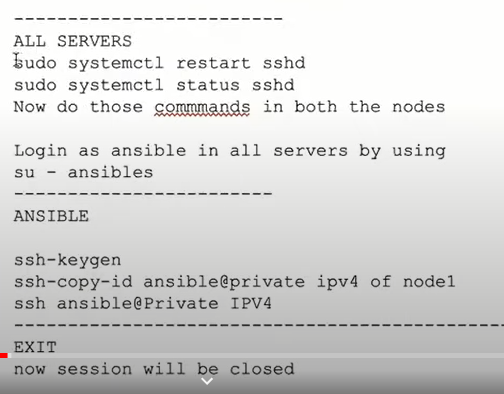
**Disadvantages**

* Ansible does not have any notion of state like other automation tools such as Puppet.
* Ansible does not track dependencies and simply executes sequential tasks and stops when tasks finish, fail or any error comes.
* Ansible has external dependencies to Python modules.
* Windows interaction requires some something.

**ANSIBLE WORKFLOW**

**Ansible installation**

****

****

If we want to push the code from Ansible server to nodes it can be done in 3 ways:

1.Ad-hoc Commands(Simple linux) Ad-hoc means temporary & it will over-ride commands.

2.Modules-A single command.

3.Playbooks-More than one module is called playbook.

Both module and playbook is in YAML.

**Ad-Hoc Commands**

* These commands can be run individually to perform Quick functions.
* Not used for configuration management and deployment ,because the commands are one time usage.
* The ansible ad-hoc commands uses /usr/bin/ansible/ command line tool to automate single task.

Go to ansible server and switch to ansible server

ansible remo –a “ls: [remo:Group name, -a:argument, ls:command]

ansible remo[0] –a “touch file1”

ansible all –a “touch file2”

ansible remo –a “sudo yum install httpd –y”

ansible remo –ba “yum install httpd –y” (b:become you will become sudo user)

ansible remo –ba “yum remove httpd –y”