Introduction-Day-1

22 December 2023

20:35

AWS-DAY-1

01 January 2024

18:53

* DEVOPS course alone is not sufficient to crack any devops interview now a days, as a devops engineers we should aware of any one of the cloud platforms like AWS/Azure/GCP.

* It's even much more better to have the knowledge on AWS & AZURE both, you will have chance to get more interview calls.

**Why because you will get more calls?**

Almost every company now following devops practices & also they are migrating their infra either to AWS or AZURE, If you have knowledge on both cloud platforms you will become one of the choice for them.

* In this course I will teach only the AWS & that too I will cover major resources in DEVOPS that are needed for 3-6 years' experience of devops engineers.

* Today's agenda is

* AWS Global Infrastructure
* Create AWS trail account
* Launch an EC2 & Connect to it

* In AWS global infrastructure we mainly discuss about **regions** and **availability zones**.

Now let's talk about regions.

* **What is region?** It's a geographical locations of aws across the world.

* So let me open the **AWS global infrastructure page.**

In this page you can see there are 30 regions across the world the AWS doing its operations.

Blue dots refers to the already existed and actively running regions and their Red dots refers to the this are the new upcoming regions.

* Each region consists of a cluster of two or many data centres and these data centres we can consider availability zones.
  + In India as of now we have to regions one is in Mumbai and another one is Hyderabad.
  + In Hyderabad region we have three availability zones and in Mumbai region we have to availability zones so you can simply assume availability zone is nothing but a data centre.
* Now let me explain this regions and availability zones in diagrammatic way for better clarity.

Machine generated alternative text:
"-south-I 

* Imagine this is an Mumbai region and in Mumbai region we have to AZ-A like AZ-B.

* AZ-A and AZ-B availability zones will have a data centre and inside the data centre there will be a physical server and inside the physical server there is a virtualization software and this virtualization software will help us to create many easy to instance on top of the physical missions.

* Many of the people will think in a way if we create a server in AZ-A and by chance if available to zone a got crashed we will think that easy to whatever we created in AZ-A will automatically up in the AZ-B but this is not correct. Since many of us come from db/virtualization background will think in this way only, so remove such kind of thought in your mind if you have it.

* **Create AWS trail account**

* **Launch Amazon Linux EC2 instance and connect to it.**

Linux-Day-1

13 October 2023

00:26

**LINUX**

Today session is on Linux operating system. Before we jump into the class,

* **You guys already having knowledge on Linux operating system?**
  1. If you don't have knowledge on LINUX operating system no worries I will explain it from the scratch.
  2. If you have knowledge on the Linux operating system it's well and good please listen one more time and don't get bored.

* **Can someone tell what is an operating system?**
  + It's an interface between the user and system.

**Image-1**

* So user will communicate with operating system and operating system will communicate with the machine.

* **What are the popular operating systems in the market?**

Windows

Unix

==================================================================================================================

**WINDOWS**

==================================================================================================================

* **Who is the owner/vendor for the windows operating systems?**
  + Microsoft is the owner of the windows operating system.
  + If anyone wants to install windows operating system on any laptops they have to purchase the licence from Microsoft.
  + I mean windows operating systems are not open source operating system.

* **What is open source? - Tuning required**
  + The **source code which is developed** for windows operating system is **not shared with public users for free**.
  + Only the vendor have authority to modify the source code of windows operating system so windows is not open source operating system.

* **What are the different flavours of the windows operating system?**

* Windows Vista
* Windows XP
* Windows 2008
* Windows 2009
* Windows 2011

* In this different flavours of windows operating system 90% of the core functionalities are mostly same and only 10% futures are different in the each operating systems.
* I mean there will be almost 10% delta portion between windows Vista and windows XP operating systems, rest of the core functionalities will be same.

* **Have you observe so far in the windows I discussed only user desktop level operating systems.**

I mean the operating systems windows Vista, windows XP, windows 2008 2009 2011 are installed in the users laptops.

* **Microsoft developed server level operating systems also at windows side, the server side flavours of windows operating systems are**
  + Windows 2012 server
  + Windows 2016 server
  + Windows 2019 server
  + Windows 2022 server

This flavours of windows operating systems are installed in server machines.

* **What is server and desktop?**

**Desktop:** it's a machine normally that you are using with the normal configurations.

**Server:** It's a machine with higher configurations in terms of number of CPU,Ram and disk size.

Machines that are provided from AWS/Azure/gcp those machines also will be considered as servers.

==================================================================================================================

**UNIX**

==================================================================================================================

Now let's jump into the Unix operating system

* **Many people assume UNIX and LINUX are same operating system is this really true?**

No the UNIX & Linux are both are supporting operating system I will discuss on this now,

* **Unix** operating system invented at **Bell Lab's** and this **UNIX** operating system is not an open source operating system if anyone wants to use in its operating system they need to purchase licence from Bell Labs.

* When this **UNIX** operating system invented this is become very popular at university level and research side.

* So now major companies like
  + IBM
  + HP
  + Sun Solaris
  + Fujistu

made agreement with **Bell Lab's for** developing their own operating system based on Unix operating system source code.

* **IBM developed ==> AIX based out of Unix operating system source code**
* **Sun Solaris ==> Solaris OS**
* **HP ==> HP-UX OS**
* **Fujistu ==> UXP/DS OS**

* So now all these are became the flavours of Unix operating system.

I mean it's

AIX

Solaris

HP-UX

UXP/DS all are **flavours of the Unix operating system**.

* Again if you want to use any **flavour of Unix operating system you have to purchase the licence from its vendor** so this flavours of Unix operating system or not free of cost.

* Now there is a guy call **Linus** a college student developed a free **open source operating system** based on the core principles of the Unix operating system and this operating system named as a **Linux**.

* So again companies coming to picture the developed they own operating system based on LINUX operating system which was developed by the Linus.

* OS that are developed based on Linux OS are
  + **RHEL ==> REDHAT(IBM)**
  + **Centos** it is a community OS.
  + **Amazon Linux ==> Amazon**
  + **Fedora**
  + **Ubuntu**

Like this many flavours created from the Linux operating system those operating systems are around **300 Plus**.

* We can see these all different operating systems in the **AWS cloud site**(Amazon) while creating the instances or servers.

* Here for your easy understanding **UNIX is a grandfather** and **LINUX is a son for UNIX** and **RHEL/Ubuntu/Fedora all these grandsons of UNIX**.

* The flavours of Linux almost 90% is same in terms of core functionalities only 10% might be different, so it will be fine if your fine in any one of the operating system you can manage rest of the flavours.

==================================================================================================================

**Which OS is easy to learn? Windows/Linux OS**

==================================================================================================================

* My opinion is windows operating system is easy to learn compared to the Linux operating system this is my opinion do you agree on this thing?

* Yes, windows is easy to learn compared to the Linux operating system because Windows OS is user friendly & GUI.
* So that anyone I mean any non-technical person can easily interact with windows operating system.

* But LINUX operating system if you want to operate you have to know the commands.
  + If you want to create file you should know command for that.
  + If you want to create folder you should know the command.
  + In Windows you can just right-click & create new file we can do easily since its GUI

* Because of this reason 80% of the desktop users I mean in the personal systems having windows operating system & only 20% of machines using the Linux operating system

==================================================================================================================

**How Linux OS became very popular than windows OS**

==================================================================================================================

Now let's discuss about the next features that made very popular then windows operating system.

* **First one is open source operating system**
  + We know that Linux is an open source operating system.
  + Since its open source operating system
    - You no need to purchase any licence.
    - The source code that is used to develop the Linux operating system is available for free so any public user whoever having access to the internet they can read modify the source code and they can develop their own flavour of Linux operating system.

* **Security**

In terms of security we mainly concerned about **hackers** and **viruses**.

* **Who is hacker?**
  + Hacker is a very good technical person on the systems with bad in tensions.
  + I mean he always try to connect to others machines and try to steal the information like credit/debit/bank details and any personal information & do scams.

* **Why hackers mostly target Windows based machines?**

* I don't say hackers only target the windows operating system and not the Linux operating system?

But the chances for hacking the windows operating systems are very high compared to the Linux operating system because the number of desktop uses for the windows operating system around 80% are there what the Linux operating system disturb only 2% so the hackers will target large number of people only instead of this number of less Linux people.

* **When we come to the virus section, windows 7 having more chances to get virus attacks**
  1. **Why there are more chances of having virus on the windows machines?**
     + **How a normal install software on windows machines will just go to the internet and directly will download .exe/.msi file &b will run it as an administrator & will click on next-next button on the software will get installed.**
     + **Suppose if the MSI are having virus that will affect the all the files & folders of the machine. Like entire C: drive & D: drive**
     + **So in this case we have to reach like the machine because of the machine will become very slow after the virus attack.**
     + **To remove virus on systems re-install the OS & sometimes this might lead to data loss also this will be a huge cost to the company.**

1. In case of Linux OS when we are installing any software we can download the software packages only from the official site of the respective operating system. So there is a less chances to get the virus affected software's or packages into the Linux systems.

1. Also any virus affected software or package installed on the Linux machine it don't spread to the all the files are folders of the Linux operating system, because of the file/folder permissions. So we can simply delete folder that affected with virus will resolve issue.

* **The next feature is less resources**

So why Linux building less resources compared to the windows operating system?

* **If you take example of MP3 file and MP4 file what is the main difference between this two files?**

* MP3 song audio file.
* MP4 is in video + audio file.
* The size of the MP3 file will be normally around 5 MB.
* The size of MP4 file how much it will be there it should be around 100MB

* **Why MP3 file size is very less compared to the MP4 file?**

Because MP3 saving only audio content but MP4 file having audio Plus video content. So this is causingMP4 file must have largest size compared to the MP3 files.

* **This is a similar case for windows & Linux also.**

**Linux is only command line operating syste**m but the windows is **command-line + graphical user interface** so it consumes lot of resources like CPU/RAM/HD

* When you install and windows operating system it will take around 80GB but when you install Linux operating system it will take around 5 to 10 GB.

* Also the CPU utilisation will be very high in windows machines because of the graphical user interface functionalities but it won't take that much of CPU usage in the Linux machine.

* So companies will mainly look for less resources obviously they will go for Linux  operating system.

* **Multi user and multitasking**

Before we understand this feature let's take an example like

* One user perform one task.
  + At a given time single user only can connect to machine & can perform only one action either create a file/folder & after this activity only user can do other task.
  + EX: DOS

Machine generated alternative text:


* One user perform multiple tasks.
  + At a given time single user only can connect to machine & can perform multiple actions like create a file & same time listen music
  + EX: WIN-2008, WIN-2010

Machine generated alternative text:
09 n 

* Many user perform multiple tasks.
  + At a given time multiple users can connect to machine & can perform multiple actions like create a file & same time listen music on their own space.
  + EX: Win-server-2012/2019/2022

Machine generated alternative text:
//IJ)V 
L-gch -D 

* **Highly available**
  + I mean the reboot time for the Linux is very long I mean that server will run for longer run without reboot, companies we can observe some of the machines not rebooted from two three years.
  + But in windows it is not like that so reboot will required after the patch updates or else if you action on server & it is not replicating correctly reboot will required.

**So all this features will out satisfying the company requirements that want to be number one position in the market.**

Futures like

* Open source & less resources - comes under expenditure & speed(CLI)
* Security feature - higher quality
* Longer runs without reboot - always available

These features making the choice of Linux on the servers instead of windows machines.

**============================================================================================================**

**Architecture of Linux**

**============================================================================================================**

Before we discuss about the architecture of Linux let's see how **windows architecture** will be there

Machine generated alternative text:
shell 

Let's see this picture here

* hardware is nothing but our physical tab laptops,
* on this laptops normally install windows operating system
* On top of Windows OS shell layer will be there
* And finally on top of the shell layer user will be available.

So user normally communicate with shell(user --> shell)

* shell will communicate with operating system(shell --> OS)
* operating system is a communicate with hardware(OS --> Hardware)
* And similarly in reverse way we will get the response from Hard ware to OS and OS to shell & shell to user.

* **Now let's see what is shell?**
  + Let me open your command prompt so here you can see the black screen & this black screen we consider as shell.
  + As a user I can communicate with shell and this shell communicate with OS and OS with hardware.
  + Let me find out number of folders/files present in current path, so I provided a **dir** command in the shell so this **dir** command will reach OS & hardware will get response with the list of the files/folders present in current path.

* So the Linux architecture also almost same with the windows architecture so there is a technical word difference between windows architecture and LINUX architecture in windows we call it as **OS layer** and in Linux we call it as a **kernel layer**.

Machine generated alternative text:
shell 

* But there is one more difference between these two architecture that made Linux is faster than the windows. That is **GUI** layer which is present on top of the shell layer.

* So when you create a folder from the shell it will be very quick but when you create your folder from windows machines first we are communicating in the GUI layer and this GUI layer again communicate with SHELL layer.

Machine generated alternative text:
GUI 
shell 
kernal 

* So there is a third person between the user and shell this is delaying speed like folder creation

Linux-Day-2

10 November 2023

16:56

**================================================================================================**

**Let's understand few terminologies between Windows & Linux**

**================================================================================================**

* **Tell me what is directory?**
  + It's a folder, sometimes in windows we call it as a folder and in Linux we can call it as a directory.
  + Don't feel they are different if I interchange these two words between the windows and LINUX operating.

* Similarly in windows we call it as an **administrator** and in Linux we call it as a **root**

* In windows we call it as a **software** and in Linux we call it as a **package**.

**================================================================================================**

**Linux file system hierarchy**

**================================================================================================**

* **What is meant by hierarchy?**
  + Just take your **family** as an example for hierarchy
  + In and family **grandpa** will be there and his **sons** comes under him and his sons having some **children**.
  + This looks like tree structure right? this is what we call as a hierarchy.

* **Linux file system architecture also similar to the tree structure hierarchy.**

* Before we discuss the Linux file system hierarchy let's discussion windows file system hierarchy because we already aware of windows operating system.

* **In windows what directories are available normally?**
  + C: drive or D: drive
  + In this two drives let's talk about C: drive
  + **Under C: drive what are the folders will be present?**
    - Program files
    - Program files 32 bit
    - Windows
    - Users
    - System 32
  + Like this many folders will be present and inside that folder again many files will be present.

* In the similar way Linux also all path will start with **slash (/)**
  + **/** - this slash we can call it as a **top level root directory**.
  + Under this top level root directory there will be many other directories.

* **Now let's see the directories under the top level root directory.**
  1. **root**
     + **here the root directory is under the top level root directory. root directory is a subdirectory of top level root directory.**
     + **Like under C: drive program files folder will be there similarly under slash root directory is there.**
     + **No let's open the command prompt, the default home directory here what is there?**
       - **C:/Users/User1**
     + **When you login Linux machine with root user the home directly for the root user is /root.**
     + **Home directory is a default directory for the user.**
     + **Like when you close the shell are comment from and when you open the shell are come and from again you will directly go to the default home directory.**

1. **home**
   * **home is a sub-directory under the top level root directory.**
   * **The home directory is similar to the c:/users folder in the windows.**
   * **All other non-root/non-admin users of the machine having the home directories under the  /home directory**
   * **Just assume there is a user1 on the Linux machine so the home directory of the user one present and /home/user1**
   * **Once user1 login to the machine by default it will go to the default home directory. So the default home directory for the /home/user1**

1. **boot**
   * **What is meant by booting?**
   * **Click on the power on button some process will run in background and bring windows initial desktop screen.**
   * **During this process what happened in background?**
   * **The operating system related files will get activated on reloaded into the Ram and will bring something initial desktop screen mirror whether it is a Linux or windows or Mac.**
   * **So the boot directory contains OS is related files.**
   * **If we delete the files in boot directory the Operating System will get corrupted and you will not login into the machine.**

1. **etc**
   * **Suppose you are my manager and you ask me to create you one Linux server.**
   * **I have created one liner and inform to my manager.**
   * **Next my manager ask me like what is the configuration of the server?**
     + **What I will tell?**
       - **It's a RHEL machine I mean ready at enterprise Linux machine.**
       - **Having 8GB of memory.**
       - **And I have created five users in this Linux machine to login.**
       - **And I have created this machine with so & so host name.**

* Like this all the configuration details I have told to my manager.

* **Now read this configuration files will get stored?**

It will be stored under the etc directory.

* /etc directory some of the files will created automatically when you launching machine.

We can customise the files in the etc directory based on our requirement.

1. **usr**
   * **When we install software in windows machine where it will get installed? C:/PROGRAMFILES**
   * **SIMILARLY INSTALL ANY PACKAGES IN LINUX IT WILL GET INSTALLED UNDER SPLASH USER DIRECTORY.**

1. **bin**
   * **The bin folder contains the Linux commands.**
   * **Suppose when I executive dir command in windows I get response with the list of files in the current directory.**
   * **And if I run some different command like bir I will get like this command is not present in windows like that.**
   * **Why like that you are getting error?**
     + **Because that command not  defined somewhere in the machine, so we are getting error.**
     + **The commands in the bin directory can be executed by any user whether it is a root user or non-root user.**

* Don't get confused what is the non-root user it's a user without admin privileges are root privileges in Linux machine.

1. **sbin**
   * **It's a directory contains the commands similar to the bin folder but this commands can be executed only by the root user and non root cannot execute these commands.**

1. **tmp**
   * **Contains temp files**

Some machines will contains different directories under the /var

**================================================================================================**

**Setup AWS account & Launch RHEL server**

**================================================================================================**

**================================================================================================**

**Different ways of creating files**

**================================================================================================**

* In windows normally how to create file?

Right-Click + New File

* Different ways of creating files
  + cat
  + touch
  + echo
  + vi
  + Nano

* **Let's see how to create a file using cat**

cat > file1.out

This is Chaitanya

I live in India

Ctrl+c

* **How to read file using cat**

cat file1.out

* **How to append data to file**

cat >> file1.out

I am Hyderabad city

Ctrl+c

* **How to create file using touch**

touch file2.out

The file2.out is empty file & size of file is zero

* **How to create file using vi editor**

**vi file3.out**

* Editor will be opened & it's only read only format.
* To write the data into file --> shift + I
* To exit from insert mode of file Esc
* To save file with modification --> :wq!
* Don't save changes & exit from file --> :q!
* Go to bottom of the file --> G
* Go to 10th line of file --> 10G

* **Create file using nano**

nano file4.out

1. To save the changes, press Ctrl + O (Hold down the Control key and press the letter 'O'). This will prompt you to confirm the filename. Press Enter to confirm.
2. To exit nano, press Ctrl + X. If you've made changes, it will ask if you want to save them. Press Y to confirm and then press Enter.

Linux-Day-3

10 November 2023

18:16

* **How to list the files in current directory**

ls

* **How do we create a folder in Linux machine**

mkdir dir1

Please don't get confused when I say something like directory is nothing holder in Linux operating system.

* **Now let's see whether directory is created or not**

Just enter "ls" command it will show the directory created

* **How to create multiple folders in single command**

mkdir1 file1 file2 file3 file4

* **How do we differentiate between the directories and files?**
  + Now you can see **directory in blue colour** and **files are in white colour** so is this differentiation enough to find which is directly and which is file?
  + For now it is okay directory is in blue colour and file is in white colour so that we can easily identify what is directly and what is file.
  + But in Linux there may be other situations some files will come in blue colour also so like the differentiation between directory and files based on the colours it's not a correct way

* **To clearly differentiate what is file or what is directory we can run LS command with a flag -l**

ls -l

* Here -l is nothing but longest format of the files/directories, In this output the first character defines what is directory and what is file.
* If the first character is "d" then it is a directory, if there is no values the first character then it is a file.

* **Hidden files/directories**

Files/directories are hidden from visibility called hidden files or directories.

* Suppose when you open C: drive we can find folders like
  + program files
  + users
  + windows …..etc
  + Apart from this there are some hidden files/folders are there but you don't see it by default
  + If you want to see those files/folders you have to go to the view section and view hidden items then only you can see the hidden folders/files.

* We will see now how to create hidden folder in windows
  + Create a folder
  + Right-Click + Properties + Hidden
  + Similarly we can do it if we want to hidden file

* In Linux we can create hidden files are directories by just prefix dot symbol before the file name or directory name.

touch .dir1

touch .file1

* Now type ls command you won't see the hidden files.

* If you want to see the hidden files that are present in current directory you have to add **-a flag** to the **ls** command.

**ls -a**

Now we can see hidden files and also the normal files that are present in the current directory.

* If you want to see the **hidden files as well as the longest format** of the files are directly and single is command we can run

**ls -al**

* We can use multiple flags in the single command and get the many details based on our requirements.

Suppose you want to print the **longest format** of the files & also want to see the **hidden files** and also you want to **sort the files based on the date of creation time**

ls -alt

* **Create multiple directories side by side**

mkdir dir2 dir3 dir4

* We can see that these directories are created or not by using

ls -ltr

* **Now let's see how to create directly inside another directory**

mkdir -p dirx/diry/dirz

* You can see whether this dir-x is created or not? just typing "ls" command
* Here you can see only dirx created.
  + And dir-y created under dir-x & dir-z created under dir-y

* To see directory created or not inside the another directory, you have to change directory to the directory using **cd** command

**cd dirx**

* Now when you type the ls command you can find diry folder inside dirx

* In the same way to check whether dirz is created or not we have to change the diry folder

cd diry

ls

* Now you want to go one step back from dirz folder

cd ..

* Now if you want to go back two steps back from the current folder I can type

cd ../..

Present you are at the home directory of the current user.

* If you see the negation symbol that terminal then you are at the home directory of that particular use whatever you login

====================================================================================================

**Next command will see PWD**

====================================================================================================

* If you want to find out in which path your cursor present you can use PWD

pwd ==> Print working directory

So here we can see the working directory is /root

root directory is present under / top level directory

* Under the root directory again many directors are presented dir1 dir2 dir3 dir4 dirx

Inside dirx-> diry present & inside diry -> dirz

* If you want to bring the full path of the dirz directory

You can change folder and type PWD command

You will get the full path of dirz from top level root directory

====================================================================================================

**COPY**

====================================================================================================

* **How to copy files in Linux**

To copy files in Linux you can use a CP command and we have to provide the source location & destination location

cp <file-name> <dest-folder>

cp file1 dir1

* **How to copy and directories from one location to another location?**

To copy directory from one location to another location we can use the cp similar to files copy,

Additionally will use -r flag to copy directly from one location to another location

cp -r dir1 dir2

* **How do you more files from one location to another location?**

**cp file1 file2 file3 dir1**

* **How to move file from one location to another location**

In Windows OS we will use cut and paste option to copy file from one location to another location

So if I want to know file1 into the dir4

mv file1 dir4

* How to move directory from one location to another location can you use same command

mv dirx dir1

* To rename file/folder

mv dir1 dir2

* How to delete the files?

You can delete the file using rm command

rm file1

We will get you prompt like whether you want to delete your file? You can enter Y, if you want to delete and you can enter 'N' don't want to delete.

* To delete a file without prompt we can use

rm -f file1

* How to delete directory?

rmdir dir1

Only useful when there is no files or sub directories  inside the directory then only we can delete the directory.

* If you want to delete the directory which is containing files or sub directories you can use the command like

rm -rf dir1

* How to get the full path of the file or directory?

Suppose if I ask you what is the full path of the program files in windows operating system what you will tell?

C:/program files

If I ask what is the full path of the Microsoft folder in windows operating system again what you will tell?

C:/program files/Microsoft

* In the similar way if I ask you full path of a particular file or folder in Linux operating system how to get I will show you now?
  + We just go to the dir or file location & enter pwd
  + Suppose I want to find the full path of dir1 we just change the directory to the dir1 and enter pwd command.

cd dir1

pwd

* If I ask you to create a file under dirz, how can we create?

We just go to the dirz & and run touch command with filename

cd dirx/diry/dirz

touch file5

* This will be hardest way to switch to the directory and creative file so the simplest way to create your file in the required directly is we have to provide the full path before the file name that we want to create.

touch /root/dirx/diry/dirz

Linux-Day-4

10 November 2023

18:16

* **grep**

grep command is to search for a particular string or pattern in your file.

**unix is great os. unix is opensource. unix is free os.**

**Unix linux which one you choose.**

**uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.**

* Let search for a unix  string  present in a file or

cat file1.out | grep unix

Normally Linux is a case sensitive language, so if we search "unix" only the matching string lines only will get printed.

* If you want to ignore the case sensitive while searching unix, you can use the -i flag in the grep command.

cat file1.out | grep -i unix

* If you want to display the lines that containing in existing and also the print the line number before they each line, you can use -n flag.

cat file1.out | grep -n unix

* If you want to bring the lines that is not containing the unix word you can use the flag -v

cat file1.out | grep -v unix

* Print the lines that started only with the Unix string, for this we can use cap^ character before the string during the grep command

cat file1.out | grep ^unix

* Print the lines that having a Unix keyboard as part of any string.

cat file1.out | grep unix\*

* If you want to search for a multiple strings we can use a egrep command.

cat temp.txt | egrep 'great|operating'

* **more**
  + Normally how do we read files?,  we just simply use the cat command.

cat file2.out

Suppose file2.out  is a very big file so to read all the content you have to scroll up and scroll down am I correct or not?

* The more command also useful to read your file but we can read a file page by page instead of displaying all the content in the screen.

more file2.out

* After running the more file2.out the first page of the file is getting display,
* so if you want to read the new line you can press enter.
* if you want to go to the new page you can press to space.
* if you want to quit the prompt you can press on q.

* **head**
  + head command will display top lines of the file.

cat /etc/passwd | head

head command by default display the first to 10 lines of the file

* If you want to print them first five lines of the file you can use the command like

cat /etc/passwd | head -5

* **tail**
  + tail command normally display them bottom lines of the file.

cat /etc/passwd | tail

tail command by default will display in bottom 10 lines of the file.

* Suppose if you want to display the bottom files of the file you can use they tail command as

cat /etc/passwd | tail -5

* You can case the tail command in different way also

tail -n 10 /etc/passwd

* **sort**
  + sort command will print the data inside your file in ascending order based on the first letter of the line.

sort file3.out

* sort - Sort command used to sort the lines of a file & arranges records in particular order.

**cat > file.txt**

abhishek

chitransh

satish

rajan

naveen

divyam

harsh

abhishek

**sort file.txt**

abhishek

abhishek

chitransh

satish

rajan

naveen

divyam

harsh

* **uniq**
  + uniq is the tool that helps to detect the adjacent duplicate lines and deletes the duplicate lines.

**cat kt.txt**

I love music.

I love music.

I love music.

I love cricket.

I love music.

* If you want to print only unique lines that are present in a file, we have to use sort & uniq command in combination

sort file.txt | uniq

* **history**
  + If you want to see the commands that you have executed since you login you can simply click on history command you will get the list of the commands that we have executed so far.

* History of commands stored in a file called **.bash\_history** file which is presented under the home directory of user.

* **hostname**
  + Every machine will have some name right?
  + That name we can call it as a hostname, if you want to find the hostname

Type on terminal **hostname**

* **ifconfig**
  + If you want to see the private-ip address of machine we can type the ifconfig command.
  + Here we can see lot of data and if you want to print only the private IP address we can use hostname command with -i flag.

**hostname -i**

* **/etc/os-release**

To find out what operating system where using we can open your file /etc/os-release

cat /etc/os-release

* **wc**
  + cat file1.out | wc -w
  + cat file2.out |wc -c
  + cat file2.out | wc -l

* **awk**
  + Print the particular columns in a file

firstName lastName age city ID

Thomas Shelby 30 Rio 400

Omega Night 45 Ontario 600

Wood Tinker 54 Lisbon N/A

Giorgos Georgiou 35 London 300

Timmy Turner 32 Berlin N/A

cat samp.txt | awk '{print $1, $4}'

* Here the default seperator between the columns in **/tab**
* If we want us different separator between the columns

cat samp.txt | awk -F ':' '{print $1, $4}'

* **yum**
  + **How do you install software in windows?**

Just double click on .exe/.msi file & will click on next-next

Ex: Let's install git on windows

* In Linux to install software/package we use yum/dnf command
  + Install git on Linux

yum install git -y

* Remove git package on Linux

yum remove git -y

* How to list packages available to install & already installed

yum list

* How to list packages installed

yum list --installed

* How to upgrade packages to latest version

yum upgrade git -y

* Here I used -y flag to proceed without prompting for Yes

Linux-Day-5

12 November 2023

17:17

**================================================================**

**Service**

**================================================================**

* In any operating system whether it is Windows/Linux some of applications/services continuously will run in background to make sure the features are working correct.

* Suppose if you take a windows operating system how to check water all the services are running in the windows machine?
  + Open **services.msc** in the run prompt
  + Here in the status column you can see the mini services are running in background.
  + Some services will be helpful to work properly our audio systems/video systems/login access things..etc.
  + Here my **audio** system working fine because of the service "**WindowsAudio**" running continiously, If I stop it I can't listen anything.
* So far whatever the services we seen like
  + Audio
  + Video
  + Bluetooth
  + CredentialManager

are related to the system level resources.

* Apart from system level services we have some other application related services also there like,
  + Database as service
  + Webserver as service
  + Application server as service
* The process for stopping/starting for any service(system/application) is same.

* Similar as windows in Linux as well we have services.

* Now let's install apache webserver & start as a service.

* **How to install the Apache web server on the Amazon Linux machine**

yum install httpd -y

* Now **how can we check whether my Apache server is running or not?**

Actually there are two ways to check whether my service is upon running or not.

service httpd status

systemctl status httpd

* **The service command is used in the older version of the Linux operating systems and systemctl is used in the latest version of the Linux operating systems.**

* Even though if you run service command in the latest operating system it will work correctly only but in background the service command rotate to the systemctl command.

* Now you can see that the status of the httpd web service are Apache web service is showing as a stopped state.

Let's **start the Apache web service** the command is

service httpd start

* In future if you **upgrade the Apache web service** versions you have to restart the service in those situations how can you restart the service?

service httpd restart

* **How can you stop the service?**

service httpd stop

* In place of service command you can always well and good to use the systemctl command based on your choice the latest operating systems.

* **Now let me ask you a one thing if I reboot my machine will my service automatically in running state?**

No it will not buy default the service will not run automatically on system reboot.

* **How can you enable the service to run automatically when systems get rebooted?**

systemctl enable httpd

**================================================================**

**Users and groups management**

**================================================================**

* **Scenario:** Company got a new project and for that project company recruited different skills people.
  + So now the company recruited four employees
    - user1
    - user2
    - user3
    - user4

* Just assume **user1** and **user2** are **developers** and

**user3** and **user4** are **Linux admin**

* Now when the four people need access to the servers first we have to create the groups that belongs to the users

* How to create a groups and Linux

groupadd developers

groupadd linuxadmins

* We have to create the users under these groups

useradd -g developers user1

useradd -g developers user2

useradd -g linuxadmin user3

useradd -g linuxadmin user4

* Now let's set the password for this users whatever we have created.

How to be set the password to the users for that will use passwd command

passwd user1

passwd user2

passwd user3

passwd user4

* Now let's try to login to the server using the user one account.

**Login to the server for the user one will failed, why it is failed?**

* The machines that are provided from the AWS by default login with username and password is disabled.

* So to enable login to the linux server using the username and password you have to do some small configuration change

Open your **/etc sshd\_config** file in this file password authentication was set into no so we have to make it as a then after we have to restart the sshd.

* Now let's try to login to the server again with the user1, now you are able to login to the server successfully. why we are able to login successfully now?

* we have update at the **passwordauthentication** and we have **restarted the sshd** service so now any user that is present on the server can login with their credentials to the server without any issues.

* **Sudo privileges**
  + Now the users that are present in the **developers group and LINUX admin group** both are having the **root/admin privileges** is it that correct? - yes

* **So now if you have a requirement to install any packages does the users user1/user2/user3/user4 can able to do?** - **no** right, why no these users are normal users they don't have any sudo religious so they cannot able to do any admin kind of actions.

* **Now I want to provide admin access to the users that are present in Linux admin group for that one what we have to do?**
  + There will be a **/etc/sudoers** file this configuration file need to be updated in order to provide admin privileges to any group or any user.

* Add the Linux admin group in the End of file in order to get the admin privilege.

* Now try to login the server using the users that are presented in the Linux admin group, so I am login with user3 and checking whether I am having sudo access or not.

* After login just type sudo su - see now user3 have the admin privileges, previously we were getting error right? This is how we provide should access to the group level.

* **Now let's see how to grant root access to the user level.**
  + Now login to the server with the user one which is presented in the developer group and check whether we have should access or not when we type sudo su - access denied.

* So now update the /etc/sudoers file by adding user1 now check whether user one having the privileges are not.

**user1 ALL=(ALL) NOPASSWD: ALL**

When we type **sudo su -** you are able to switch root users.

* Also it is not mandatory to switch root account to run the admin level commands(user/group creation, enable service, install packages..etc), you can just **prefix sudo command before the command that you want to run from the non-root user** it will get executed successfully same as from root privileges.

* **Find**
  + **Find all the files that are present in the current directory.**

find -type f

This command will displays all the files in the current directory and also the files that are presented in the nested directories.

* Find all the directories that are presented in the current directory.

find -type d

* Find that files that are presented under the particular path.

Let's search the files that are presented under the top level root directory.

find / -type f

* Find the director is that are presented under particular directory.

In similarly we will search the director is that our present under the top level root directory.

find / -type d

* Find the specific file location under the top level root directory.

find / -type f -name passwd

* Find the specific directory location under the top level root directory

find / -type f -name init

* free -h

* du -sh \*

* df -h

Linux-Day-6

14 November 2023

22:10

**======================================================================**

**Permissions management on files and directories**

**======================================================================**

* Whether it is a Linux operating system or windows operating system **what kind of operations will do on files?**

Normally we do

* Read
* Write
* execute

actions we will perform on the files is that correct or not? - **Yes it is correct**

* **What do you mean by read operation on the file?**

Like running cat command are command on your file and just read the content that is present inside the file that is read operation.

* **What is write operation on  file?**

We open your file using vi command and we will write some content inside that file and saving that file that is an right operation and also if you open your file and delete some content on the file and saving the file that is also that correct or not? - yes correct

* **What is meant by execute action on the file?**

In windows directory .EXE file will be there and when we double click on the .EXE file and we will install software that is an execution action.

In the same way in Linux also there are some script files with extension .sh, to run this files we need execute privilege.

* Hope now you understand what are the actions we can do on the files.

* **Now will discuss the level of permissions on each files/folders.**
  + what do you mean by level of permissions?

Level of permissions nothing but **who is having permissions on the files** and **what kind of permissions do they have**?

* In Linux we can control the permissions on the files mainly on three levels
  + Owner
  + Group
  + Public

* Here owner is nothing but creator of the file suppose if you creative file you are the owner of the file.
* Group is the where the owner is belongs to.
* Public nothing but all the users that have access to the Linux machine.

* How do we see the permissions on the files or directories?

This command already be discussed we use ls -l command which provide the longest format of the files.

**ls -l <file\_name>**

**-rw-r--r-- 12 linuxize users 12.0K Apr  28 10:10 file\_name**

|[-][-][-]-   [------] [---]

| |  |  | |      |       |

| |  |  | |      |       +-----------> 7. Group

| |  |  | |      +-------------------> 6. Owner

| |  |  | +--------------------------> 5. Alternate Access Method

| |  |  +----------------------------> 4. Others Permissions

| |  +-------------------------------> 3. Group Permissions

| +----------------------------------> 2. Owner Permissions

+------------------------------------> 1. File Type

* **How to modify the permissions on the files?**

There are actually 2 ways to Modify the permissions

I am going to explain only one way here now,

* The syntax to change permissions on files

**chmod <OCATAL-NUM> filename**

**chmod 777 filename**

In this 3-digits

* 1 digit - represent permissions related owner
* 2 digit - group
* 3 digit - all users

* Each action that we perform on file will have specific value
  + r - 4
  + w - 2
  + x -1

* The permissions of the specific user class I mean the owner/group/public is sum of values of the permissions.

* **Let's take a file & modify the permissions**
  + Provide read, write and execute permissions level at owner level
  + Read and execute permissions for a group
  + Read permissions for all users

* To modify the permissions first identify 3-digit value

|  |  |  |
| --- | --- | --- |
| **owner** | **group** | **public** |
| rwx | rx | r |
| 4+2+1 | 4+1 | 4 |
| 7 | 5 | 4 |

chmod 754 file1.txt

* Now check the permissions

ls -l file.txt

Now the permissions changed as 754

* Try few use cases
  + Provide rwx permissions to owner, group & public
  + Provide rw-rw-w to user-group-public

**================================================================================**

**SOFTLINK Vs HARDLINK**

**================================================================================**

**SOFTLINK**

* **In windows how do we create shortcuts?**

Here shortcut is just pointer to original file/folder & actual size of the shortcut is not same as original file/folder.

* Similarly as windows shortcuts in Linux we can us softlink to create shortcuts

ln -s <org\_file> <softlink>

* The soft link file size is not same as the Org\_File & If we delete the original file soft link will not work.

* Softlink we can create on files & directories as well.

* Softlink & orginal file will have different inode.

* Inode is the address of the file where exactly its store in Harddisk

Machine generated alternative text:
ORIGINAL FILE 
inode 
10001 
SOFTLINK 
Inode 
10002 
Data of file 

To checkInode of file/directory we can use command

ls =i file\_name

**Hardlink**

* Hardlink is mirrored copy of a file.

* The hardlink file size is same as the original file.

* We can create hardlink using command

ln org\_file hard\_link

* If we delete the original file still we can access the hard link file.

* Hardlink & Original files will have same inode.

Machine generated alternative text:
ORIGINAL FILE 
HARDLINK 
Inode 
10005 
Data of file 

* We can create only hardlink for the file not the directories.

Git-Day-1

15 November 2023

09:48

* **What is a version control system?**

**Version control system** also referred as **source code management**

* In simple words

Version control system we can call as **VCS &**

Source code management tool has **SCM** tool

* **Different type of VCS tools** in market,

1. GIT

2. SVN

3. Clear case

4. Mercurial

* Before we understand what is an version control system? we will discuss **what are all the problems that will occur them version control system**.

* **Scenario:** Just assume you are developer and working in MNC IT company & your manager came to you and told that we have new project from some

client and he gave you some inputs to start work on that project.

* So as a next step what you will do based on the inputs that you received from the manager and client you started working on developing of application.

* Developer will create a folder in his laptop & will write the code in the files.

Machine generated alternative text:
railway-project 

* Once the developer completed first version(v1) of project, developer will set up a meeting with client & manager for reviewing application.

Machine generated alternative text:
client 
Hotel bookin 
demo-I 
feedback-I 
demo-2 
feedback-2 
demo-3 
Freelance developer 
hotel-booking 

* During the first version of application review the manager & client are not satisfied with the futures that are present in application. So they suggested few more corrections to your application & new features.

* So as a next step the developer what will do?

Again he will go back to his system and based on the inputs that received from the review meeting he will correct the source code and add new other features, correct or not?

* After the completion of the code corrections & futures development based on the review-1 meeting and again you are going to set one more review meeting with client and manager to validate V2 code.

* During the  review-2 meeting also there are some more changes suggested by Manager & Client.

* So again developer what he will do?

He will update the code in existing folder based on the review-2 meeting inputs.

* Again developer will setup review-3 meeting with manager & Client to review V3 application & its code.

In this review-3 meeting manager & client said we don't need these changes latest but i want features that presented during the review-1 meeting that are fine.

* **Now it's a big trouble to developer, he can't go back to the exact v1 files of code, correct or not?** correct

* **Why correct?**

There are so many lines of code changed & many files are added, so he can't go back to exactly v1 code.

V3 ---> V2 ----> V1

* **Problems without VCS**
  + Overwritten the files & not having mechanism to rollback to previous changes on files.
  + Assume when multiple developers are working on same project, it will be very hard to find who is working on what & collaboration will be very difficult.

* **VCS will resolve all these kind of issues like,**
  + Record the changes happened to the files

- What is changed

- When it is changed

- Who is changed

* Allows to rollback to previous versions of files
* Allows to review the latest source code files with previous versions of files.
* Acts as a best collaborative tool for developers, so it's easy to know who is working on what & maintain stable code.

* Types of VCS
  + CVCS - Centralized Version Control system
  + DVCS - Distributed Version Control system

* **Centralized version control system**
  + In centralized version control system we will have one central server & which contains repository.

* **What is an repository?**

Repository is nothing **folder which is contains source code files** & also the **metadata about the files**.

* **What is metadata about files?**

Who is modified the file, when is it modified & who modified like this information stored.

* Now let's see how CVCS will work

Machine generated alternative text:
Main server 
Repository 
-1 
(30 
Filel 
Dev-A 
Filel 
Dev-B 

Assume there are two are developers are present developer-1 & developer-2 & these two developers are connected to centralized repository

* Developer-1 connected to the central repository & did checkout of files that he need to update.
* Now the changes made by the developer-1 present in laptop only.
* In order to make the changes visible to others the developer-1 should commit those changes to central server.
* Now developer-2 able to see the changes made by developer-1.

* **Problems in CVCS:**
  + Single central repository always developers are connecting, so if there is any network issues developers will sit idle.
  + If central server goes down that's it we lost the code, if we don't have proper backup mechanism.

* **SVN & IBM-Clear case comes under CVCS category**

* **Distributed version control system(DVCS)**

Machine generated alternative text:
Main Server 
Repository 
Local 
Workstatio 
DEV 
DEV 
DEV 

* In DVCS the **main server contains repository** & this repository we can call it also **remote repository**.
* Developers will **clone the copy of remote repository into laptop** & this repository called as **local repository**.
* On this local repository developers user will perform update/commit operations to make changes reflect.
* In order to make the changes visible to others developers can push changes to remote repository.

* **Benefits of DVCS are CVCS**
  + If main server goes down & we can recover the repo from latest copy local repository.
  + Even if there is network issue developers can work on local repositories & can push the changes to main repo once network available.

* **Git comes under DVCS category.**

**=============================================================================**

* **How to install Git on Windows**

**=============================================================================**

* Download .MSI way
* Using choco
  + **Install choco software in windows**

Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))

* **Install git**

choco install git

* **Check git is installed or not**

git --version

**=============================================================================**

* **How to install Git on Linux**

**=============================================================================**

yum install git -y

(or)

dnf install git -y

====================================

**Git config**

**====================================**

* This command will help us to introduce ourself to Git like who is using person using it & his mail id.

* git config --global user.name chaitanya

git config --global user.email chaitanya.rv99@gmail.com

* To check this configuration we will run command

git config --list

**=============================================================================**

**Create a Local repository & Explain different stages**

**=============================================================================**

Machine generated alternative text:
DEVELOPER MACHINE 
hotel-booking 
A.java 
Workspace 
gi 
A.java 
status 
Staging area 
gi 
status 
hotel-booking 
git push 
git init 
Github 
A.java 
Local repository 

* Imagine this is a **developer-1** machine with **windows OS** & git already installed on it.

* On this developer machine a folder called hotel-booking created by developer-1 & this directory will have source code files related to the project will be developed.

* To **convert this hotel-booking directory as a repository** we have to run "**git init"** command inside the **hotel-booking** directory.

* After initialization of the **hotel-booking** as a repository this **hotel-booking** directory will be divided into three sections logically.
  + **Working directory**
  + **Staging Area**
  + **Local repository**

* Now the developer started working on the project and he created **file A.java in the working directory** and he written some code inside A.java file.

* To move the file from **working directory to staging area** we can do it by running the add command.

**git add A.java**

* In order to make sure whether the file is staging area are in the work space we can run the **git status** command.

From this command we will come to know whether the file is in which stage.

So now we can see the file is in staging area.

* To move the file from **staging area to local repository** you can use the command **commit**.

* When we run commit command in the repository the snapshot of the changes will be taken and committed .

Along with that information you will get like

* who committed those changes
* when those changes are committed and
* commit message.

* **Explain this scenario practically**

* To check what commits present on the repository we can run a command **git log**.

* Here we can see the first comment we made on the repository.
  + 40-bit SHA code
  + Mail-id of the user
  + User ID
  + Commit message.
* Now the changes are present in developer machine only & in order push the local repository into internet & make it visible to other developers we have to use repository hosting platforms.
  + GitHub
  + Azure devops
  + Bit bucket
  + GitLab

**===============================================================**

* **Create GitHub account & empty remote repository**

**===============================================================**

* Now I will show you how to create a remote repository.

* Where do we create our remote repositories? GitHub/Azure devops/Gitlab

* First of all what is **GitHub**?

GitHub is the repository hosting platform, it means it will store all the repositories and make those repositories available to the other developers for easily collaboration.

* **People might think git and GitHub both same. Is this correct?** 
  + No it's not correct statement.

* **Git** is a **tool** &

**GitHub** is a .**repository hosting platform**

* Like camera is a tool and Facebook or Google drive or Google photos is a hosting platform like GitHub.

* **Now let's see how to create GitHub account.**
  + To create GitHub account you need an email id so since I already have the GitHub account I am just giving the random email ID.
  + After creating the GitHub account the email id will receive an 8-digit passcode and we have to enter this password then only our account creation will get complete.

* **Now let's understand what is an project?**
  + Under project we can create the number of a repositories that belongs to specific project.

* Suppose hospitals is my project then the different departments like

oncology

radiology

cardiology

gastrology

and all those things we can consider as repositories.

* Now we are skipping the part of creating the organisation as of now and we are going to create repo under the default organization & default organization name userid name.

* **Create remote repository in GitHub with same name as local repository.**   
  + To create repository under the default organisation we need to pass unique  repository name and

* Give the description about the repository the description is an optional thing, but we can provide it.

* Next we need to choose  repository type as a public or private.

* **so what do you mean by public & private repository?**
  + **Public** repository can be **accessible over the internet by any person** &
  + If repository **private** s a only the **specific people that you granted access**.

* Next don't choose the checkbox like readme.md file & create **empty remote repository**.

* Now we have an empty remote repository in GitHub & local repository in developer-1 machine. In order to push changes from local repository to remote repository connectivity required between them.

* First let's **check is there any connectivity between local repository & remote repository**

git remote -v

There is no connectivity b/w local & remote repo's.

* To setup connectivity between local & remote repo we need to do

git remote add origin <github\_url >

git remote -v

* Now developer-1 is ready to push the code.

git push origin main

* GitHub repo is now updated with local repository changes.

Git-Day-2

17 November 2023

12:14

======================================================================================================

**Clone repo**

======================================================================================================

* To **copy entire remote repository as a local repository into developer machine** can be done using git **clone** command.

* In two ways we can repo using
  + https GitHub URL
  + SSH URL

* **Let's see how to clone repo using https URL**

git clone <https\_github\_url>

Since our repositories are public repositories we don't need to enter user-name & password while cloning repositories.

If it’s a private repository we have to enter credentials.

* **Now how clone repo using SSH**

Machine generated alternative text:
DEVELOPER MACHINE 
cc.f-n 
Github 

* To clone repo from GitHub to Developer machine **SSH Connection** must be established.

* **How to setup SSH connection?**
  + On the developer machine create ssh-key's for the user.

**ssh-keygen -t rsa**

* Now public key & private key is generated for the user under current user home directory.

* Next add the public-key in GitHub account & don't share private key with anyone, it's like password.

* Clone repo

git clone <ssh url of repo>

**Homework:** Try to clone repo in Linux machine using SSH

======================================================================================================

**PULL**

======================================================================================================

* Let's assume there are two developers DEVELOPER-1 & DEVELOPER-2 are working on same GitHub repository & master branch having 3-commits.

* Now DEVELOPER-1 & DEVELOPER-2 are cloned repositories made few changes in code & committed in Local repository.

* Now DEVELOPER-1 pushed the code into GITHUB repo & this got successful.

* **In the same way If DEVELOPER-2 want to push the code into GITHUB it will get failed. Why its failed?**

Because when DEVELOPER-2 cloned repository it was in V3 & this version updated to V4 by DEVELOPER-1.

Now DEVELOPER-2 can only update the V4 & not the V3.

* To fix the issue first DEVELOPER-2 has to pull the latest changes of remote master branch in GitHub repo & integrate into DEVELOPER-2 master branch in local repo & after that he need to push it.

* Let's see this practically
  + **GitHub repo:** With 3 commits

* **Clone repo**  
  DEVELOPER-1: git clone <repo>

DEVELOPER-2: git clone <repo>

* **Push code**

DEVELOPER-1:

git add file1.java

git commit -m "Adding file1 by dev-1"

git push origin master

DEVELOPER-2: git push will be failed

git add file2.java

git commit -m "Adding file2 by dev-2"

git push origin master

* **Pull the code**

DEVELOPER-2: git pull origin master

git push origin master

Git-Day-3

19 November 2023

11:52

===================================================================================================

**Branch**

===================================================================================================

* Normally when you create a repository by default master/main branch will be get created inside the repository.

* The master branch will contains stable & production ready code for deployments. We shouldn't tamper this code at all.

* Suppose if we allow developers for directly pushing the code on master branch there will be a chance of disturb the production ready/stable code on main branch.

* so if the code on main branch is not stable then the deployments also will not work correctly.

* To overcome such kind of issues,  Whenever developers want to work on new feature Git allows to create a new branch & on this new branch developers can parallel work without disturbing the main line branch code. Once the feature development completed feature branch will merged to master.

* Now let's work on these branching practically,
  + Clone pets clinic repo

* By default present we are in master branch & its having 3 files.

* Create new branch called **feature-1**

**git branch feature-1**

* Switch to new branch

**Git checkout feature-1**

* Check on which branch currently you are in

**git branch**

* List all the branches present in local repository & remote repository

**git branch --all**

* Let's add a new file & create commit, now feature branch having 4-commits & master having 3-commits

* Let's merge the feature branch into master

**git checkout master**

**git merge feature**

* Delete the feature branch

**git branch -D feature-1**

* **Did you notice one issue here? while merging the code from feature branch into master branch**
  + Also **developers are directly not allowed to push the code from local repository master branch to remote repository master branch**.
  + **Code is not reviewed by other developers/lead** since the changes are present in Local machine.

* Now we will see how to overcome above two issues.

* **How to protect master branch on GitHub from direct code push from developers.**
  + To protect main branch from direct push, we have to setup protection policies

Repo --> Settings --> Branches

* Require a pull request before merging --> Users can't do direct push & only pull request is the way to merge code to master.

Require approvals --> The code must be reviewed & approved by other developers.

Do not allow bypassing the above settings --> No one have possibility to skip rules like above

Lock --> Whenever code freeze is there, we can lock the branch. So users can't merge pull requests.

Machine generated alternative text:
Branch name pattern 
main 
Applies to 1 branch 
maln 
Protect matching branches 
Require a pull request before merging 
When enabled, all commits must be made to a non-protected branch and submitted via a pull request before they can be merged into 
a branch that matches this rule. 
Require approvals 
When enabled, pull requests targeting a matching branch require a number of approvals and no changes requested before they 
can be merged. 
Required number of approvals before merging: 1 

Machine generated alternative text:
O Dismiss stale pull request approvals when new commits are pushed 
New reviewable commits pushed to a matching branch will dismiss pull request review approvals. 
O Require review from Code Owners 
Require an approved review in pull requests including files with a designated code owner. 
O Require approval of the most recent reviewable push 
Whether the most recent reviewable push must be approved by someone other than the person who pushed it. 
O Require status checks to pass before merging 
Choose which status checks must pass before branches can be merged into a branch that matches this rule. When enabled, commits 
must first be pushed to another branch, then merged or pushed directly to a branch that matches this rule after status checks have 
passed. 
O Require conversation resolution before merging 
When enabled, all conversations on code must be resolved before a pull request can be merged into a branch that matches this rule. 
Learn more about requiring conversation completion before merging. 
CJ Require signed commits 
Commits pushed to matching branches must have verified signatures. 
O Require linear history 
Prevent merge commits from being pushed to matching branches. 
O Require deployments to succeed before merging 
Choose which environments must be successfully deployed to before branches can be merged into a branch that matches this rule. 
Lock branch 
Branch is read-only. Users cannot push to the branch. 
Do not allow bypassing the above settings 
The above settings will apply to administrators and custom roles with the "bypass branch protections" permission. 

* Now next fix the other issue by raising pull request

===================================================================================================

**GIT PULL REQUEST**

===================================================================================================

* Git pull request will help us show the difference between personal branch & mainline branch. This will helps the peer developers to review changes.

* Also provide feasibility to suggest comment on the code developed if any modifications needed.

* Now let's assume

Machine generated alternative text:


* This is a developer machine & this is a GitHub repo with master branch
* If developer wants to work on new feature, he will create a branch called feature-1 in local repository.
* Once feature development completed, developer will push the feature branch into GitHub repo.
* Now In order to merge the feature branch into master in GitHub, pull request need to be raised.
* Once the pull request raised it can be reviewed by other developers & can be merged into master.

* Let's do this practically now.

* **Clone repo**

git clone <https://github.com/chaitanyaredd/pull-request-demo.git>

* **Create a feature branch** in local repository based out of master branch

git checkout -b feature1

* **Add new file & commit in feature1 branch**

echo "This is file1" > file1.out

git add .

git commit -m "Adding first file"

* Push the feature1 branch to GitHub

git push origin feature1

* **Raise PR** in GitHub
  + Review the PR by another developer(student).
  + Make sure the new developer must have collaborator access(settings --> Collaborator).

* Merge the Pull request to master branch.

Machine generated alternative text:
Changes approved 
1 approving review by reviewers with write access. Learn more about pull request reviews. 
V 1 approval 
This branch has no conflicts with the base branch 
Merging can be performed automatically. 
Show all reviewers 
Merge pull request 
You can also open this in GitHub Desktop or view command line instructions. 

=================================================================================

**Git Fetch**

=================================================================================

* Before we discuss on Git Fetch command let's discuss what are all the different branches we have in GIT.

* In Git mainly we have three kind of branches
  + Local branch
  + Remote branch &
  + Remote tracking branch

* **What is local branch?**

The branch that we have created in the **local repository** & local repository presented in our laptop

* **What is remote branch?**

The branch that we created in the **GitHub** that is called as a remote branch.

* What is remote tracking branch?
  + **It's local copy of the remote branch.**
  + How can we see the remote tracking branches?

git branch -r

* Git **Fetch** command will **download the changes from the remote branch** and **updates its corresponding remote tracking branch**.

* In this case the changes are downloaded from remote repository to local repository but not merged it with the local repository branch.

* **Now let's see scenario practically**
  + Create a repository in a GitHub with three commits in master branch and clone that repository into the local machine.

* Now we have two repositories one is local repository and another one is remote repository in GitHub.

* Now list the branches, git branch -a

Here we can see the local branch as master &

remote tracking branch as origin/master &

remote branch as master that is present in the GitHub

* Now let's make some changes in the master branch GitHub.

* Next step if you run git fetch command the changes whatever present in master branch of GitHub will download it to them remote tracking branch in the local repository.

* Can we see the changes that we have downloaded?

No we cannot see those changes that we have downloaded into local repository from master local  branch.

**cat filename**

See no changes are coming.

* Normally we should not edit the remote tracking branches.

* To make visible the changes that are presented in the remote tracking branches to your local branch we have to run git merge command

**git merge origin/master.**

=================================================================================

**GIT PULL**

=================================================================================

* Git pull command is combination of **fetch** as well as the **merge** command.

* It means that whenever developer runs the fetch command changes are downloaded from remote branch(master) to remote tracking branch(origin/master) & then after remote tracking branch will get merged to local branch.

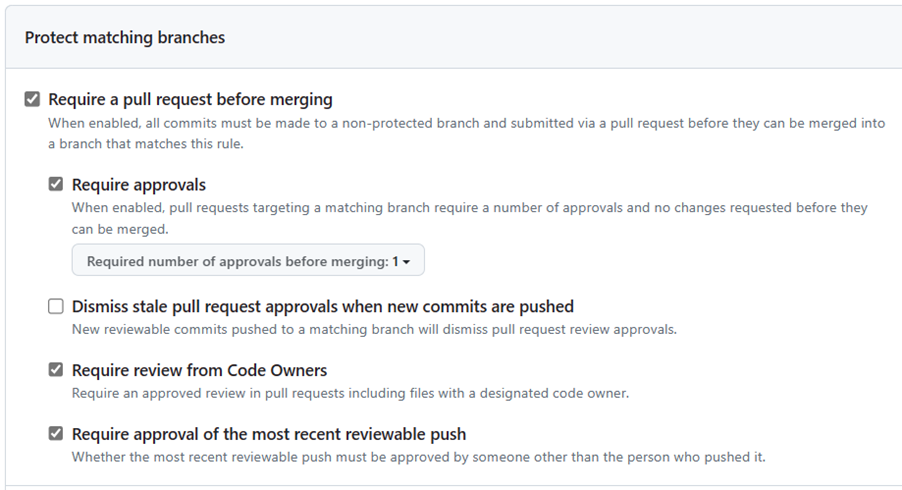
* Now let's see this scenario practical
  + Update the remote repository with the few commits.

* Run git pull command on master branch

   git pull origin master

cat filename

See we can see the changes.



Dismiss stale pull request approvals

Even last approved request get dismissed whenever new commit happens on this PR.

Require review from code owners:

Post approval on PR, if any new commit triggered on personal branch it will become as below screenshot & asks for approval latest changes keeping previous changes approved with affecting.

**CODEOWNERS** file added at root folder & these people are reviwing code before merging into master rbanch



Machine generated alternative text:
Imagine master branch 2 commits 
based of master devl:» devl-feature add commit 
based of master dev2 dev2-feature add commit 
merge the dev2-feature into master & complete it 
merge the devl-feature to master(you behind the master) 
merge master to devl-feature & git push dev -feature 
git pull origin main 
git checkout devl-feature 
git merge main 
git push origin devl-feature 

Machine generated alternative text:
Review required 
Show all reviewers 
Waiting on 1 reapproval from someone other than the last pusher. Review from chaitanyaredd is stale because it was submitted 
before the most recent code changes. Learn more about pull request reviews. 
V 1 approval 

Git-Day-4

21 November 2023

12:50

**====================================================================================**

**Merge conflict**

**====================================================================================**

When same file & same line getting modified & merged into main branches one after one.

1. Git on Windows ==> developer1 ==> update user config ==> feature1

2. Git on EC2 ==> developer2 ==> update user config ==> feature2

3. Merge feature1 into main branch by developer1

4. Merge feature2 branch into main branch by developer2

5. Merge conflict

Get the base branch changes into local machines

git pull origin main

git checkout feature2

git merge master

**====================================================================================**

**Forking repo**

**====================================================================================**

* Before we understand what is and Forking repository, let see this diagram

Machine generated alternative text:
Gue4VJo 

* **Assume there is a repository in the GitHub & you want copy of this repository in your local machine how can we do that?**

We can use git **clone** command to copy repository from the GitHub to your local machine.

* **Similar way your repository in GitHub account-1 & you want to create copy the repository in another GitHub account-2, how can you do that?**

**Fork** option will be used.

* When you do the fork repository we have two independent repos so the changes in one repo  will not affect the another repository.

* **Why do we need to fork repository?**
  + Suppose you have interest to **develop a hotel-booking** so you have one option like you have to write a from the scratch and complete the application development that is one way.

* Else we can check internet is there anyone having similar interest to develop a hotel-booking application & pushed code in GitHub.

* So now you can take that copy of code into your account and you can start developing your application according to your company requirements using the Fork option.

* When you do the fork repository we have two independent repos so the changes in one repo  will not affect the another repository.

* Now let's go to the GitHub repositories, here you can see there are many number of users forked this repository because other users considered my repositories are might useful for them and they want to maintain my repository in their account that's why they did the fork.

* Similarly I can create the others repositories is whatever present in the different type of account into my data account by clicking on the fork button.

* Now here we can see the owner of the repo is different user now when I click on the fork button it is coming under my own account and also by default we are copying the master branch code.

* The fork button you can see the number of users have made the forking of repository on this main repository.

**====================================================================================**

**REVERT Changes**

**====================================================================================**

* Now we will see how to revert the changes from the
  + working area
  + staging area
  + local repository in the developers machine.

* Now let's see how to **rollback changes in workspace area**

In order to revert changes from the working directory we can use a command

**git restore filename**

* How to **revert the changes from the staging area to working directory**

**git restore --staged file** #Discard changes from staging area

git restore file #Discard changes from working directory

* DISCARD THE CHANGES FROM LOCAL REPOSITORY

git reset HEAD~ # Revert the changes from local repo  to working directory

git restore file

* Revert the changes from the remote repository

We don't have any direct to it to reward the changes from remote repository.

We have to undo the changes by comparing the last to commits and push back those changes whatever you did again in order to under the changes in remote repository.

**.gitignore**

Maven-Day-1

21 November 2023

14:29

* Today our topic is on **BUILD TOOL**

* **Why do we need Build tool?**
  + Build tool used to convert the source code files into binary packages.

* Machine generated alternative text:
  push 
  Actor 
  java developer 
  maven 
  .java 
  •jar 
  .war 
  . ear 

Suppose you are a java developer, you write a code in .java files & push the code into GitHub. The source later converted into binaries packages like

* .jar
* .war
* .ear

* **Which build tool used to convert the .java source code files into .jar/.war/.ear binaries?**

maven

* These binary files only we deploy into our DEV/QA/PROD environments.

* Machine generated alternative text:
  push 
  Actor 
  .Net developer 
  maven 
  . my 
  . exe 

Same way if you are .Net developer, you write code in .cs files & push the code to GithHub. These sources code will be converted into binaries like

* .msi
* .exe
* .dll

* **Which build tool used to convert the .cs source code files into .msi/.exe binaries?**

msbuild

* Like this different build tools based on the different technologies
  + Android --> Gradle
  + Nodejs --> npm

* In this course we talk about maven as build tool since we have most of applications on Java

* **How to Install maven on Linux machine**

* **Install java-11**
  + List java-11 related packages are available

yum list | grep -i java-11\*

* Install the java-11

yum install java-11\* -y

* Check the java version

java -version

* **Download the maven binary**
  + Go to maven official site & get latest maven installer link

wget <https://dlcdn.apache.org/maven/maven-3/3.9.5/binaries/apache-maven-3.9.5-bin.tar.gz>

* Extract the maven binary

tar -xzvf apache-maven-3.9.5-bin.tar.gz

* Move the extracted maven folder as /opt/maven

mv apache-maven-3.9.5 /opt/maven

* Set the environment variables **maven\_home** & **path at user-level** 
  + Update the **maven\_home** in ~/.bashrc or ~/.bashprofile

**export maven\_home=/opt/maven**

* Update the **PATH** variable

**export PATH=$PATH:$maven\_home/bin**

**(OR)**

* Set the environment variables **maven\_home** & **path at system-level** 
  + Update the **maven\_home in /etc/profile**

**export maven\_home=/opt/maven**

* Update the **PATH** variable

**export PATH=$PATH:$maven\_home/bin**

* **Check the installed maven version**

mvn --version

* **How to Install maven on Windows machine - Home work**

* **Install jdk-20**

Download: <https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.exe>

Update: JAVA\_HOME & PATH System variables

* Download & extract maven binary

<https://dlcdn.apache.org/maven/maven-3/3.9.5/binaries/apache-maven-3.9.5-bin.zip>

Extract to C:\maven

* Update the MAVEN\_HOME & PATH variables at system level

* Maven is not only a **build tool** but it is also **project management tool**.
  + What is project management tool?
    - It's used to create project structure.

* Just check any java projects in GitHub, they proper directory structure.
  + Let me do search of java project in GitHub
  + If you see calculator repository, it's having proper directory structure src/main/com, src/main/test
  + If you see one more online book store repository, It's also having directory structure in proper way

* To create this directory structure manually it's
  + Time taking process
  + Chances for not creating project correctly

* This problems can be avoided with help maven.

* **Now who will use maven as project management tool?**

DEVELOPER

DEVELOPER - Will create project structure & will push into GitHub.

* So DEVELOPERS will use the maven as  **project management tool (To create projects) & Build Tools(To compile the files & generate binaries locally)**

* Now let's work on maven as a project management tool practically
  + Assume you are java developer & have requirement to create application for **sdbbank** in java.

* DEVELOPER will create hotel-booking project structure using maven
  + open cmd & run the command **mvn archetype:generate**

* It will display list of templates available & prompt for template number to enter
  + Each template will used for different purpose & creates different structure
  + Developers are responsible to choose correct number
  + As of now we go with default one & which is sample maven project.

* Now the template also will different versions, because in background maven team continiously enhancing this templates. We will choose latest one.

* **Groupid**: Normally sdbbank website url is like [www.sdbbank.com](http://www.sdbbank.com)

In this the groupid is **com.sdbbank**

Similarly for [www.facebook.com](http://www.facebook.com) the groupid is **com.facebook**

Machine generated alternative text:
www.sdbbank.com 
groupid: com.sdbbank 
artifactid: netbanking 

* **Artifactid**: What kind of operations banks will do?

Machine generated alternative text:
Net banking 
cards 
sdbbank 
Loans 
Stocks 

* Net banking
* Cards
* Loans
* Stocks

Out of these areas we can take anyone as an artifactId

* **Version**: we can choose as snapshot since our project is on development phase.

* **Package** we can as groupid.artifactid

* Now the project created with name netbanking.

* Let's check the directory structure
  + Install tree command & check directory structure

yum install tree -y

tree

* Here we can see

Machine generated alternative text:
L— netbanki ng 
pom. xml 
main 
java 
com 
L— sdbbank 
ne tbanking 
L App. java 
test 
1— java 
com 
sdbbank 
1— netbanki ng 
I AppTest. java 

* pom.xml
* src/main/java/com/sdbbank/netbanking/App.java
* src/test/java/com/sdbbank/netbanking/AppTest.java

* The files that are comes under the src/main --> Related to functional code
  + What is functional code?

The code written to develop the net banking application.

* The files that comes under the src/test --> Test code in order to validate the functional code. This code we can call as Unit Test code.

Machine generated alternative text:
Calibri 
18 
Core configuration file 
Functional code 
Unit test code 
netbanki ng 
pom. xml 
mai n 
I— java 
com 
I— 
test 
java 
com 
sdbbank 
netbanki ng 
I App. java 
sdbbank 
netbanki ng 
I AppTest. java 

* Why here developers are writing test code? Normally QA has to do testing correct?

Basic level of functional code can be validated instead of checking with QA engineers.

* **pom.xml** Core configuration file of the maven project & it's considered as heart of the java project.
  + groupId
  + artifactId
  + version

* **properties:**
  + Properties section used to **define parameters & reuse those parameters throughout the pom.xml file**
    - **<java.version>1.7</java.version>**

* <maven.compiler.source>${java.version}<\maven.compiler.source>

<maven.compiler.target>${java.version}<\maven.compiler.target>

* Normally we can install java7/java8/java11/java17 versions in same machine.

* Whenever developer write java files we have to tell maven, based on what java version source code files are developed.

* Here with these two parameters maven compiler will treat,
  + consider the source code is developed based on java7
  + Generated byte code byte code compatible with Java7

* **Dependencies:** 
  + **Dependencies are** external modules/libraries that you are project depends.
  + Here junit dependency used to run the unit test on functional code during build process.

**<dependencies>**

**<dependency>**

**<groupId>junit</groupId>**

**<artifactId>junit</artifactId>**

**<version>4.11</version>**

**<scope>test</scope>**

**</dependency>**

**</dependencies>**

* We can get above snippets by looking the maven official sites.

* **Plugins**
  + Plugins will provide additional functionalities to enable tasks compile the code, test, package & deploy as part of Maven build Lifecycle.

* I will explain Maven build Lifecycle in sometime & later you will come to know these plugin needed.

* Hope you are clear about maven project directory structure.

Maven-Day-2

23 November 2023

18:38

* + Now the we have code ready & pushed into GitHub
    - **Will you deploy the source code that is available in GitHub as it is in DEV/QA/PROD environments?**

No

* + We have to convert the source code into binary package. How you do it?

Go to the repo & run command

**mvn install**

* + This command will

Machine generated alternative text:
Downloading from central : 
Downloaded from central : 
Downloading from central : 
Downloading from central : 
Downloading from central : 
Downloading from central : 
Downloaded from central : 
Downloaded from central : 
t 3.5 MB/s) 
Downloaded from central : 
Downloaded from central : 
https : // repo. maven. apache.org/maven2/j uni t/ j uni t/ 3.8. I/ juni t-3. 8. I. jar 
https://repo. maven. apache.org/maven2/junit/junit/3.8. 1/junit-3.8.1. jar (121 kB at 5.5 MB/s) 
https : // repo. maven. apache.org/maven2/cIassworl ds/cl assworl ds/l. I-al pha-2/cI assworl ds-I. I-al pha-2. jar 
https : // repo. maven. apache. org/maven2/commons-codec/commons -codec/ 1.6/commons-codec-1.6. jar 
https : // repo. maven. apache . org/mave n2.org/apache/maven/sha red/maven-shared-uti 1 s/(). 4/maven-shared-uti s -O. 4 . jar 
https : // repo. maven. apache.org/maven2/org/codehaus/pI exus/pl exus-uti I s/ 3. O. 15/ pl exus-uti I s-3. O. 15. jar 
https://repo. maven.apache.org/maven2/c1asswor1ds/c1asswor1ds/1. 1-a1pha-2/c1asswor1ds-1.1-a1pha-2.jar (38 kg at 1.4 MB/s) 
https://repo. maven. apache.org/maven2/org/apache/maven/shared/maven-shared-utiIs/O.4/maven-shared-utiIs-O.4.jar (155 kB a 
https://repo. maven.apache.org/maven2/commons-codec/commons-codec/I.6/commons-codec-I.6. jar (233 kg at 4.8 MB/s) 
https://repo. maven. apache.org/maven2/org/codehaus/pIexus/pIexus-utiIs/3.O.15/ plexus-utils-3.O.15.jar (239 kB at 4.3 MB/ s 
L INFO) Installing / root/netbanki ng/target/netbanking-l. O-SANPSHOT. jar to / root/. m2/repository.com/sdbbank/netbanking/I. ng-I. O- 
SANPSHOT. jar 
CINFOJ Installing / root/netbanki ng/pom. xml to / root/. m2/repository.com/sdbbank/netbanking/I. O-SANPSHOT/netbanking-I. O-SANPSHOT. pom 
CINFOJ BUILD SUCCESS 
r INFOI 

* + **Download required dependencies from internet - Without maven it will be very difficult to download libaries**
  + **Generate binary file (.jar)**

* + Now one more directory got created called **'target'**

Machine generated alternative text:
target 
cl asses 
sdbbank 
I— netbanking 
L— App. class 
generated-sources 
I— annotati ons 
generated-test-sources 
1— test-annotati ons 
maven-archiver 
pom. properti es 
maven-status 
L— maven-compiler-plugin 
compi e 
L— default-compi le 
createdFi les. 1st 
i nputFi1es.1st 
testcompi 1 e 
1— default-testcompile 
createdFi les. 1st 
inputFi1es. 1st 
netbanking 1. 0-SANPSHOT.jar 
surefi re- reports 
TEST -com. sdbbank. netbanki ng. AppTest. xml 
com. sdbbank. netbanki ng . APPTest. txt 
test-cl asses 
sdbbank 
I— netbankinq 

* + **Here functional code related .class files we can see**
  + **Also we can see unit test code related .class files**
  + **jar file created based out of .class files**

* + The generated .jar we will deploy into dev/qa/prod Tomcat/Kubernetes/Cloud foundry environments.

* + Let's understand **MAVEN Build Lifecycle**
    - In Maven, the build process organized in series of well-defined phases & the sequence of these phases referred as Build Lifecycle.

* + Machine generated alternative text:
    Instan 
    Validate 
    Compile 
    Test 
    8 Phases of the Default Maven Lifecycle 

* + If you see above diagram there are multiple phases in build process in sequence to generate & deploy the binaries.

* + Let's discuss how each phase will help

* + Validate -

Its primary purpose is to validate the project is correct and all necessary information is available for a successful build like the references in pom.xml are valid values or not.

This phase is often used to check whether the project's configuration files are correct and all necessary resources like dependencies are available

Ex: If I change dependency from log4j to log4jx the mvn validate will failed, beacuase the refernce log4jx not present anywhere maven repos like central/local/remote

mvn validate

Check the directory structure created correctly or not.

* + Compile - Convert the .java files into .class files & we can see these files in target folder

mvn compile

Here in target folder we can .java files are converted into .class files

* + Package - Will generate the binary file .jar based out of .class files

mvn package

Here in target folder we can see .jar file

* + Verify - Verify that the project is valid and meets the quality standards, Code coverage should be > 80%

mvn verify

* + Suppose if you are a developer & wrote 100 java files for functionality but written only 2 java files for unit test.
  + Which means only 2% of code is tested at developer side.
  + We can restrict the developers by making build failed until they wrote test code more than 80% by using **Jacoco** plugin in pom.xml

* + Install - Copy the generated .jar file from target folder to local repository(~/.m2).

* + Deploy - It copies the packaged .jar file to the remote repository for sharing it with other developers

* + Along with these goals we can use clean command like

mvn clean install

Clean - Will delete the target folder before fresh build process started.

* + Next we will understand types of repositories in maven, so we will come to know install & deploy commands clearly.

* + In maven we mainly 3 - Kind of repositories,
    - **Local repository**
    - **Central repository**
    - **Remote repository**

Machine generated alternative text:
Local repository 
N/.m2 
Download required 
dependencies 
Remote 
repository 
Central 
repository 
https://repo.maven.apache.org 

* + Whenever developers run any maven command like(mvn install/deploy) maven connects to maven central repository which is present in Internet & downloads the required binaries into ~/.m2 folder on the machine to complete build process.

* + The binaries are copied remote repository to local repository only at first time & next time onwards maven will consumes the dependencies from local repositories from local repository.

* + Let's remove the files are downloaded in local maven repo & run **mvn install** command

Here we can see binaries are downloaded from maven central repo to the local repository.

* + Now again run **mvn install** command, this time see binaries are downloading & it's getting consuming from the local repository.

* + Machine generated alternative text:
    Local repository 
    N/.m2 
    Download required 
    dependencies 
    Download required 
    dependencies 
    Remote 
    repository 
    Central 
    repository 
    ttps://repo.maven.apache.org 
    0 

* + Some organizations will not allow to connect the maven central repository & download binaries.

In these situations maven will connect to remote repositories that are create in the organization level. Maven remote repository is in sync with the central repository.

* + Normally we tell maven where to download binaries either from central/remote repo based on configurations in settings.xml(%MAVEN\_HOME% /conf/)

**<mirrors>**

**<!-- mirror**

**| Specifies a repository mirror site to use instead of a given repository. The repository that this mirror serves.**

**-->**

**<mirror>**

**<id>central</id>**

**<url>https://repo.maven.apache.org/maven2</url>**

**<mirrorOf>central</mirrorOf>**

**</mirror>**

**<!-- ... other mirrors ... -->**

**</mirrors>**

If we don;'t have above configufration in settings.xml by default we are using central repo for downloading binaries.

* + If we want to download binaries from custom-central repo url

**<mirror>**

**<id>central-custom</id>**

**<url>custom-https-url</url>**

**<mirrorOf>central</mirrorOf>**

**</mirror>**

<mirrorOf>: Specifies which repository or group of repositories this mirror should replace. In this case, it's set to "central," indicating that it's a mirror for the Maven Central Repository.

* + **mvn deploy explain practically during Jfrog setup. - Try this during Jfrog session**

* + Setup maven repository in Jfrog

* + Configure Deployment in pom.xml: In your project's pom.xml file, you need to configure the distribution management section to specify the URL and authentication details for the remote repository. Here's an example:  
      
    <distributionManagement>  
     <repository>  
     <id>your-repo-id</id>  
     <url>https://your.repository.url/releases</url>  
     </repository>  
     <snapshotRepository>  
     <id>your-snapshot-repo-id</id>  
     <url>https://your.repository.url/snapshots</url>  
     </snapshotRepository>  
    </distributionManagement>  
      
    Replace your-repo-id, <https://your.repository.url/releases>, your-snapshot-repo-id, and <https://your.repository.url/snapshots> with your repository details. Make sure to use the correct URL for release and snapshot repositories.

* 1. Configure Authentication: Maven needs authentication details to deploy artifacts. You can configure these details in the settings.xml file, which is typically located in the <Maven\_Home>/conf directory. Alternatively, you can configure authentication directly in the pom.xml file (although this is less secure):  
       
       
     <servers>  
      <server>  
      <id>your-repo-id</id>  
      <username>your-username</username>  
      <password>your-password</password>  
      </server>  
      <server>  
      <id>your-snapshot-repo-id</id>  
      <username>your-username</username>  
      <password>your-password</password>  
      </server>  
     </servers>  
       
     Replace your-repo-id, your-username, and your-password with your repository and authentication details.

* 1. Run mvn deploy: Now, you can run the following command in your project's root directory to deploy the artifacts:  
       
     mvn deploy  
       
     Maven will compile your project, run tests, and if successful, deploy the artifacts to the specified remote repository.

**Home-work**

* 1. Download the binaries from custom central maven repository
  2. Build any java application & generate binaries & automatically copy it to local repository on fresh ec2 machine
  3. Explain the Maven Build Lifecycle.
  4. Explain the difference between a Maven Snapshot and a Release.
     + A Maven Snapshot is a version of a project under development, typically not intended for release. A Release is a stable version of a project that is considered ready for production use.
  5. Deploy artifacts to remote repositories

* 1. Explain Maven life-cycle

* 1. Explain with scenario how plugins downloaded from the Jfrog remote repositories

* 1. Install Jenkins

* 1. Install Git & Maven on server

* 1. Integrate Git with Jenkinsfile

* 1. Integrate Maven with jenkins

* 1. Create maven project

* 1. Create pipeline project

* 1. Execute the pipeline

Sonarqube-Day-1

28 November 2023

17:36

Sonarqube-1

* In this you will get understand

* What is sonarqube?

* Why do we need Sonarqube?

* How to setup sonarqube on Linux server?

* Sonarqube is **Static code analysis tool.**

* **Why do we need Static code analysis tool?**

Before we understand what is Sonarqube let's jump into the development phase of the project

* During project development the developer will write a code, once the code development of completed this code has to be reviewed by the another peer developer like Team lead/Architect of the project.

* So now what Team Lead/Architect will check the code & try to identify,

* Is it containing any bugs?

* Is it secure issues?
  + It means that any passwords or personal email ids are mentioned in the source code?

* Is there any duplication code?
  + There might be situation developers has to write **same piece of code in many locations of the files related to project**.

* Using same lines of code at many files can be called as duplication of code.

* **Architect** will check possibility of **creating function for same piece of code at one location & will call that code rest of the places**.

* Is that code tested properly or not?
  + As we know **developer job is not only writing code for the functionality of the application but also they have to write test code to check the functionality**. This test code called as **UNIT TEST** Code.

* So Architect will check will the code sufficient UNIT TEST code or not?

* Is there any complex code written?
  + As an architect/team-lead if there is any complex code, you have to find-out is there any better way of rewriting the coding easiest way of understanding without affecting functionality.

* Easy to integrate with another developers code?
  + Architect will review the code and check if that code is easy to integrate or not with the another developers code when he is working in a group of team.

* All this actions are doing manually by the people like Architect/Team Lead, so every time whenever developer pushed/check-in latest code this review process will come into picture & it will kill the time of developers.

* We can automate all these actions with static code analysis tool came into picture.

* We have so many **static code analysis tools in the market**
  + Sonarqube
  + Coverity
  + Codescene
  + Veracode

* Apart from these static analysis tools why most of the companies are choosing sonarqube?
  + Sonarqube is not only **static code analysis but it is also a code quality management tool**.

* Just assume you are a java developer & you have a requirement to **develop a calculator** application for mobile phones.

* So developer what you will do you write a code for functionality development like,
  + addition
  + subtraction
  + multiplication and
  + division,  correct or not? - Correct

As a developer you are not only taking care of the functionality development but you also have to write a code to test your functionalities, this code called as unit testing.

* **Sonarqube** as a code quality management tool it will **provide the unit test code reports** of your project.

* Even Sonarqube will provide the details about **code coverage**.

**What is an code coverage?**

* Suppose as a developer you wrote a hundred functions to complete your application development so out of those hundred functions how many functions are successfully tested based on that this code coverage will be calculated the projects.

* The projects which are having the more coverage those projects will be considered as stable projects.

* So all these information whatever we have discussed so far like

Machine generated alternative text:
Quality Gate ? 
Passed 
Reliability 
O 
Bugs 7 
6 Security 
o 
Vulnerabilities 
Coverage 
76% coverage ? 
On 21k New Lines to cover 
New Code Overall Code 
New Code: since about 1 month ago 
Maintainability 
O 
Code Smells ? 
Security Review 
o 
Security Hotspots ? 
Duplications 
0.0% Duplications ? 
On 46k Newv Lines 

* Number of bugs your project.
* Number of vulnerabilities
* Code coverage
* Duplications code
* Code smells

like this information we can easily see in the sonarqube dashboards.

* Now let's understand the **sonarqube components**
  + In sonarqube we have mainly two sections
    - Sonarqube server
    - Sonar-scanner

* In sonarqube-server we have mainly 4 components,

Machine generated alternative text:
Web 
component 
Rules 
Elastic search 
Database 
EC2-SERVER With SonarQube server 

* **Rules**
  + Rules are nothing but the guidelines/best practices that developers has to follow during the code development.

* Whenever we install sonarqube server by default we get rules for each programming language that developer has to follow while writing code.

* If there is **any deviation in the code for the rules defined** it will be considered as a **bug/vulnerability/code smell** & displayed in the sonarqube dashboard.

* **Database**
  + Once the rules are executed successfully & it will generate the analysis reports.

* The analysis report which is created based on the sonarqube rules triggered on source code of the developer.

* **Web component**
  + It will display the analysis report that is stored in the database in nice graphical interface.

Machine generated alternative text:
Quality Gate ? 
Passed 
Reliability 
O 
Bugs 7 
6 Security 
o 
Vulnerabilities 
Coverage 
76% coverage ? 
On 21k New Lines to cover 
New Code Overall Code 
New Code: since about 1 month ago 
Maintainability 
O 
Code Smells ? 
Security Review 
o 
Security Hotspots ? 
Duplications 
0.0% Duplications ? 
On 46k Newv Lines 

* **Elastic search**

This component will help the web component to fetch the information from database in quickest way.

* **Sonar-scanner**
  + Sonar-scanner agent which is normally installed on the machine very were the source code is existed.

* Sonar-scanner will run on the source code and generates the analysis report later that this analysis report is push to the Sonarqube server.

Machine generated alternative text:
Source-code 
Developer-Machine 

* Sonarqube  supports almost **27 plus programming languages** and generates analysis reports.

Machine generated alternative text:
Java- 
VB 
JS 
TS 

* **How communication established between sonar-scanner & sonarqube-server,**

Machine generated alternative text:
4. Display report in GUI format in URL 
2. Rules: Sonar-scanner 
gather applicable rules 
from SonarQube based on 
programming language 
identified & these rules will 
execu e on source 
generate analysis reports. 
re orts &store in database 
Source-code 
so AG.9--'l 
Web 
component 
Rules 
Elastic search 
Database 
EC2-SERVER With SonarQube server 
I. Sonar-scanner will collect information about 
source code based sonar-scanner.properties file 
a. In what language source written? 
b. Sonar project name used to create? 
Develooer-Machine 

* **How to install sonarqube server with inbuilt database**

* Let's discuss the pre-requisites to install Sonarqube server

**Perquisites:**

* **JAVA**

* **Why we need Java as it request it to install sonarqube?**

Because Sonarqube is developed based on Java programming language.

* To install Java on Linux machine we have to run a command like

**yum install java-17\* -y (sonarqube LTS 9.9.3)**

* Sonarqube provides LTS(Long Term stable) sonarqube-server products & latest release sonarqube-server products.
  + LTS means its stable version of product & if its containing any bugs/vulnerabilities in feature you can get fixes for those without being impacted with other functionalities.
  + Latest products continuous improvements for the product & can't guarantee on few of features.

* Now to download the sonarqube installer from LTS model & go to the official site of sonarqube.

There we can find the different level of products will be available like

* Community
* Enterprise level
* Cloud level

Like this different products available.

* **Installation steps:**
  + So in this session we are going to install the community edition.

<https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-9.9.3.79811.zip>

* **Extract the zip file**
  + unzip sonarqube-9.9.3.79811.zip
  + mv sonarqube-9.9.3.79811 /opt/sonarqube

* Create sonar user

useradd sonar

* Change ownership of /opt/sonarqube to sonar

chown -R sonar:sonar /opt/sonarqube

* Start sonarqube as non-root user

cd /opt/sonarqube/bin/

./sonar.sh start

* Now access the sonarqube in browser http://<ip-address>:9000

Username: admin

Password: admin

* **How to install sonar-scanner**
  + Download sonar-scanner

<https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-5.0.1.3006-linux.zip>

* Extract it

unzip sonar-scanner-cli-5.0.1.3006-linux.zip

mv sonar-scanner-cli-5.0.1.3006-linux /opt/sonar-scanner

* Check sonar-scanner version

Sonar-scanner --version

Sonarqube-Day-2

28 November 2023

23:20

* **Console overview of Sonarqube**

* **Projects:**
  + Once you logged into the console, when you click on the projects it will be empty for the first time and here we can create projects based on our requirement.

* This project console will started getting filled up when you try to push the analysis reports of your source code projects.

* In sonarqube we can create projects in two ways,
  + Manually
  + Another one is by integrating sonarqube with other source code management tools like
    - GithHub
    - Gitlab
    - Azure devops
    - Bit bucket

* Now let's start create a project manually,

To create a project manually we have to enter owner project key & now I gave

project key as **calculator-dev**

sonar.login as calculator-dev

* sonar.login is used to authenticate either from Jenkins/cmd prompt to the sonarqube.

* Now we have to find out the command that need to be executed to publish analysis reports of your projects into sonarqube.

Click on continue, it will give the command to execute.

* As of now I am just saving this command into the notepad after sometime I will show you how to execute this command from the command prompt.

* The next tab **issues** it will show you the list of the issues that are present in your source code related project as of now it will be empty since we don't have any projects.

* **Next rules**
  + As I said earlier rules are best practices that each developer keep in mind while writing a code for his project.

* Sonarqube supports almost  27  programming language.

* Each programming language having some set of rules
  + Java is having 670 rules &
  + .NET having 400

Like that each programming language having some number of rules if there is any deviation from this rules that will be report as an bug or duplication or vulnerability.

* Next **quality profiles**
  + Quality Profile is nothing but **collection of the rules & these collection of rules(quality profile) will be applied on code during analysis of the project.**

* By default each programming language associated with **sonar-way** quality profile.

If you see the Java related sonar-way quality profile it's having 483 rules are active & 143 rules are inactive by default applicable to any Java related projects.

* We can create a quality profiles on our own instead of default one and we can enable/disable some more extra rules in that quality profiles depend on the need.

* Show this practically.

Create java-custom-profile with 626 active rules.

* **UC: Now we will see practically how to generate static code analysis report & publish it to sonarqube.**

* **Pre-requisites**
  + EC2 server
  + Git
  + Maven
  + Sonar-scanner
  + Sonarqube server
  + GitHub(java-repo):

* **Clone repo:**

git clone <https://github.com/chaitanyaredd/onlinebookstore.git>

* **Build:**

mvn clean install

* **Run static code analysis:**

mvn sonar:sonar \

-Dsonar.projectKey=calculator

-Dsonar.projectName=calculator

-Dsonar.login=XXXXXXXXXXX

-Dsonar.host.url=http://<ip-address>:9000

* Now in during code analysis sonar-way quality profile applied on code which was containing 400+ rules for java code & generated static code analysis report. This report is available in sonarqube with project name calculator.

* Here we can see all details like
  + Bugs
  + Vulnerabilities
  + Code coverage
  + Code duplications
  + Code smells

* **UC: Change quality profile for the sonar-project & new custom profile must have 600+ rules.**

* Modify the quality profile of project to custom-java-profile.

* Re-run static code analysis on code,

mvn sonar:sonar \

-Dsonar.projectKey=calculator

-Dsonar.projectName=calculator

-Dsonar.login=XXXXXXXXXXX

-Dsonar.host.url=http://<ip-address>:9000

* Now quality profile of the project modified from **sonar-way** to **custom-java-profile**

* **Next one is quality gates**
  + Quality gates are nothing but setting threshold levels for the each measurements of project & if any measurement is below that threshold level that Sonar project status will be should failed.

* Like if project is containing
  + bugs more than one that project status should be filled.
  + If project would coverage is less than 80% that project status will be failed.

* Let's setup quality gate(custom-java-gate\_ instead default quality gate.

* **UC: Modify the quality gate of calculator application to custom-java-gate & make project status as failed.**
  + **Clone repo:**

git clone <https://github.com/chaitanyaredd/onlinebookstore.git>

* **Build:**

mvn clean install

* **Run static code analysis:**

mvn sonar:sonar \

-Dsonar.projectKey=calculator

-Dsonar.projectName=calculator

-Dsonar.login=XXXXXXXXXXX

-Dsonar.host.url=http://<ip-address>:9000

* Now the calculator project status will be in failed state.

* **Homework:**
  + **Repo:** <https://github.com/chaitanyaredd/petsclinic.git>
  + Do the static code analysis on java project & publish report to sonarqube
  + Create custom quality profile & update it sonar project
  + Create custom quality gate & make the project as failed
  + Generate custom sonar.login token & use it this token during latest scan.

Jfrog-Day-1

29 November 2023

14:55

* In this session we are going to talk about

* what is an Artifact?

* what is an Artifact repository?

* Types of Artifact repositories?

* Jfrog Artifactory?

* How to install the JFROG Artifactory on ec2

* **First of all can you tell me what is an Artifact?**

Machine generated alternative text:
source code 
GitHub 
Binary file/package 
Build process 
Artifact 

* It's a binary file or package which is generated after build process completed on the source code.

Machine generated alternative text:
Java --0 Maven(BuiId) .jar/.war/.ear 
.Net MSBuiId .exe/.msi 

* Suppose just assume your Java source-code is available on the GitHub after the build process the binary files will get created & the format of the files like JAR/WAR/EAR files.

* Similarly the .Net source code will generate the binary files in the format like .exe/.msi packages after the build process gets completed.

* In general the binary files that are created after the build process created we can refer them as artifacts.

* So now the files like JAR/WAR/EAR/MSI/EXE we can call as a artifacts.

* Normally whenever there is a change in the suits that is presented in the GitHub an Artifact will get generated.

* **What we will do with this Artifact?**

we will deploy this Artifact in DEV/QA/PROD environments.

* Mostly the developers will check-in latest code into GitHub many times in a day, for every latest check-in made into GitHub repository an Artifact will get generated. The Artifact whatever is created will do deployment in to the DEV environment.
* Just Imagine the application was working without any issues before the deployment. But after the latest Artifact deployment application is not working correctly.

* **How do you fix it?**
  + We have to revert the changes in code & get the Artifact for reverted changes & deploy it. This is one way and its time taking process.

* Maintain storage for artifacts & deploy previous artifacts whatever needed.

* **Which way we will normally after in this to options obviously will go with the second option only correct or not? -** Correct

* Now Artifact repository tool will help us to take a backup of artifacts that are created as part of the build process for ever latest check-in in GitHub. So if there is a requirement to rollback your environment to the previous Artifact version, you can simply pull that Artifact from the Artifact repository and you can make it your application working successfully.

* Artifact repository is not only used for storing the binaries but it is also  provide dependence libraries that are required to complete build process successfully.
  + Normally when we run mvn commands the binaries are downloaded from internet instead of that we can download it from Jfrog.

* Now let's see this practically
  + When you run **mvn install** command the Maven command connects to the central repo which is on the Internet & download the libraries that are needed to build your application.

* In some organisations it is not allowed to connect directly to the internet and download the required packages.

* In those situations in order to get the binaries the Maven command will connect to the Jfrog artifactory and Jfrog will get will get in sync with the central repository which is stored on the internet.

* In the market there are number of Artifact repository tools are there

* Helix
* Pulp
* Nuget
* Jfrog
* Nexus
* Docker registry

* In this set of tools we are going to discuss about the JFROG artifactory tool.

* Jfrog artifactory tool mainly available in two editions
  + Pro-edition
  + Open source solution

* Now we'll see **how to install the open source solution model Jfrog artifactory on Amazon Linux ec2 machine.**
  + **Prerequisites**
    - Create EC2 with instance type of t2.small
    - Allow the ports
      * 8081
      * 8082
    - Java
    - Maven
    - Git

* **Installation steps**
  + There two types of installable available for Jfrog
    - pro installer --> It's Licensed version
    - oss installer --> It's a free open source solution

* Login to EC2 with root user & install java
  + yum install java-1.8\* -y

* Download the artifactory installers into /opt

There two types of installable available for Jfrog

* pro installer --> It's Licensed version
* OSS installer --> It's a free open source solution

Now let's install OSS installer

cd /opt

wget -O <https://releases.jfrog.io/artifactory/bintray-artifactory/org/artifactory/oss/jfrog-artifactory-oss/6.23.42/jfrog-artifactory-oss-6.23.42.zip>

* Extract the archive

unzip jfrog-artifactory-oss-6.23.42.zip

mv jfrog-artifactory-oss-6.23.42 jfrog

* set **JFROG\_HOME** variable

echo "export JFROG\_HOME=/opt/jfrog" > /etc/profile.d/jfrog.sh

source /etc/profile/jfrog.sh

* Run a start-up script

cd /opt/jfrog/app/bin/

./artifactory.sh start

* Access the artifactory from browser

http://<dnshostname>:8081

* Default credentials to login jfrog artifactory

user: admin

password: password

* Change the default password

* Here we can create different type Artifact repositories based on the source code,
  + If we have **java** source code we can create **maven** repository
  + If we have **php** source code we can create **php** repository
  + If we have **python** source code we can create **py** repository
  + If we are **not sure about the source code** we can create **Generic** repository

* In Jfrog main we will have 3 types of repositories
  + **Local** --> Local repositories are the place to store artifacts generated as part of build

* **Remote** --> This repository get in sync with central maven repository

* **Virtual** --> It's a combination of local + remote repositories

In Local repositories we can see

libs-snapshot-local

libs-release-local

In Remote repositories we can see

jcenter

In Virtual repositories we can see

libs-snapshot

libs-release

* **UC: Create a maven local repository with name "online-bookstore-local" for online-bookstore java application**.
  + This will help us to store different versions of artifacts like
    - jar
    - war
    - ear

* **UC: Create maven remote repository with name "jcenter"**
  + This repository will help to download the dependencies from jcenter repo from Jfrog instead of the central repository in Internet

* **UC: Create maven virtual repository with name "online-bookstore"**
  + online-bookstore-local + jcenter

* Check Artifact repository browser & we don't see any artifacts on those repositories, those are empty.

Jfrog-Day-2

01 December 2023

17:09

* **UC1: How to integrate Jfrog Artifactory with maven**
  + **Pre-requisites**
    - Source code: <https://github.com/yankils/hello-world.git>
    - Tools:
      * Git
      * Maven
      * Jfrog

* **Integration maven with Jfrog steps**

In order to publish the **Artifact into the jfrog online-bookstore-local repository**

* Go to Jfrog **online-bookstore-local** repository & setup ==> Deploy

* Copy the <distribution Management> XML code paste into pom.xml

* Run mvn deploy {Will tries to deploy Artifact into **online-bookstore-local** repository}

* Deploy failed due to authorization issue

* Change the user-name & password for in ~/.m2/settings.xml(The user must have admin/write privileges)

* mvn deploy

The jar file along with pom.xml into Jfrog **online-bookstore-local** repository

* Why snapshot pushed to libs-snapshot-local?

due to version tag contains 1.0-SNAPSHOT

b. Publish the Artifact into jfrog release repository

A. Remove -SNAPSHOT in pom.xml & run mvn deploy

B. Copy the <distribution Management> XML code of libs-release-local into pom.xml

C. Artifact will be pushed into libs-release repo

* **UC2: Pull the dependencies from Jfrog artifactory rather than maven central repo**
  + Rename the ~/.m2/setting.xml

mv ~/.m2/setting.xml ~/.m2/setting\_bkp.xml

* Go setup of "jcenter" repository in Jfrog & generate settings.xml file, next create ~/.m2/settings.xml file with generated content
* Machine generated alternative text:
  «epos i tori e» 
  «reposi to rp 
  «enabl ecbfal se«/enabl ecb 
  "snapshot» 
  cbcent ral cb 
  «nameA i bs- re 1 ease«/nam» 
  Qurbhttp : // 54.172.2.32 : 8081/artifactory/I i bs-rel ease4urI 
  K/ reposi torp 
  q-eposi to rp 
  «snapshots 
  Ki CBs naps hots«/i 
  i bs-s 
  «urbhttp : // 54.172.2.32 : 8081/artifactory/1 i bs-snapshot«/url 
  g' reposi torp 
  "reposi to ri e» 

* mvn clean install

Artifacts will be downloaded from jfrog remote repo

* Rename the ~/.m2/setting\_bkp.xml to settings.xml

mv ~/.m2/setting\_bkp.xml ~/.m2/setting.xml

* mvn clean install

Now artifacts will be downloaded from central maven repository present in internet.

* **UC3: How to integrate Jfrog Artifactory with Jenkins Maven project - Try this scenario once**

* Install "Artifactory" plug-in
  + Manage Jenkins -> Jenkins Plugins -> available -> artifactory
  + Configure Artifactory server credentials
* + Manage Jenkins -> Configure System -> Artifactory

**Artifactory Servers**

Server ID : Artifactory-Server

URL : Artifactory Server URL

Username : jenkins

Password : jenkins@123

* + Create a Maven Project
    - **Job Name** : artifactory-project
    - Source code management
      * Git URL : <https://github.com/chaitanyaredd/hello-world.git>
    - Build Environment

Resolve artifacts from Artifactory : <provide Artifactory server and repository details>

Build - Goals: clean install

Post-build Actions

Deploy Artifacts to Artifactory : <provide Artifactory server and repository details>

Execute job

* **UC4: How to download Jfrog Artifact with Jenkins Freestyle project-Try this scenario once**

* **Webhook**

Jenkins-Day-1

01 December 2023

18:39

* Today onwards we are going to start discussion on the Jenkins tool.

* Before we are going to understand
  + **what is Jenkins tool?**
  + **Why do we need Jenkins tool?**
  + **How to install Jenkins on AWS Linux EC2 instance**

* Jenkins is a **CI/CD** tool**.**
  + **CI** - Continuous Integration
  + **CD** - Continuous Delivery (or) Continuous Deployment

* Now let's understand what CI/CD tool will do,

Machine generated alternative text:
Actor 
Actor 
••i iiii iiii iiii 
mvn dean insta 
mvn sonar:sonar 
Jenkins 
DEV 
http://dev.8012 
Send mail to 
Maven 
Artifactory 
Get approval from PO 
httpWQA8012 
pROD 
http://prod:8012 
iiii iiii 
CONTINIOUS DELIVERY 

* Assume there are 3 developers are working in a project.

These three developers normally will develop the code for the project in their laptops, correct or not? - Yes

After code development completed these developers will keep the code in common place, correct?

**What is that common place?** GitHub

* Now assume this is a Jenkins.

* What are all other tools we have covered so far?

I have discussed

* build tool as a Maven
* Static code analysis tool as a sonarqube

* Once application development code completed & you generated packages like .jar/.war/.ear file, will you deploy it to client environment directly? - No

Why? There will be a chances for issues/bugs in that code & so whenever we deploy that .jar/.war/.ear there will be a chances of application will not work correctly.

* So before we deploy application into client environment first we do test in DEV & QA environment, after the regression test completed successfully we do deployments in PROD environments.

Let's assume this is

* DEV environment & It's automated testing setup
* QA environment & It's automated testing setup
* PROD environment & prod-checkouts

* Now all these tools are integrated with Jenkins
  + GitHub
  + Maven
  + Sonarqube
  + DEV/QA/PROD servers

* So whenever there is a
  + Code change in a GitHub

* Jenkins will download that code and inform maven to run build command mvn clean install & will generate packages .jar/.war/.ear

* Once the packages are generated next Jenkins will inform sonarqube to run static analysis & publish generated report into sonarqube dashboard.

* Now packages are deployed into DEV servers automatically with help of Jenkins.
  + For each environment there will be a endpoint URL like **https:// <dev-server-ip>:8012**
  + Next QA engineers will do end to end testing in DEV environment.
  + Once the testing completed successfully in DEV environment QA will green signal to proceed to deploy on QA environment.

* Now Jenkins will do deployment to QA servers,
  + Now QA environment will have one more endpoint URL like **https:// <qa-server-ip>:8012**
  + Again QA engineer will do end to end test manually in QA environment.
  + Now we see there is no issues & ready to take prod & we have to take approval from product-owner to deploy application in to PROD.

* After product-owner approval application will get deployed into PROD environment with help of Jenkins.

* **Continuous Integration** ==> Once developer check-in code into GitHub & on that code
  + Automated build
  + Automated unit-test
  + Automate code analysis
  + Artifact will stored

* **Continuous Delivery** ==> It's a process to automate deployment in DEV & QA environments & automate testing as well for those two environments(non-prod), but we need approval of application owner to do production deployment.

* **Continuous Deployment** ==> It's a process to automate deployment in DEV & QA environments & automate testing as well, & no need approval from application owner to do production deployment.

**===========================================================================================**

* **How to install Jenkins in EC2**

**===========================================================================================**

**Go to the official site:** [**https://jenkins.io**](https://jenkins.io)

* **Pre-requisites**
  + EC2 instance: Amazon-Linux
  + Allow port 8080
  + Install Java-11  
    yum install java-11-openjdk-devel -y

* **Installation steps**
  + **Add the Jenkins repo to download the packages:**

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

* **Import key file to authenticate the Jenkins repo in-order to install the Jenkins package**

sudo rpm --import <https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key>

* Install Jenkins

yum install jenkins -y

* Start jenkins server

systemctl start jenkins

* To auto restart jenkins on server reboot  
  systemctl enable jenkins
* **Configure Jenkins**
  + By default jenkins runs on port 8080

* We can access jenkins using URL http://<ip\_address>:8080

* Default user name for jenkins is **admin**

* Password for admin stored by default at **/var/lib/jenkins/secrets/initialAdminPassword**
* Do the installation of recommended plugins

* Change password of admin user
* Next I will show you how to create jenkins job, configure it & execute it.
  + Click on new-item to create job
  + We have enter the name of the job in the text box.
  + There are different kind of job types are present to create.
    - Free-style
    - Pipeline
    - Folder
    - Maven
      * **Freestyle:**
        + This is the most common & basic job type in Jenkins.
        + Using this job type we can process for build, test & deploy processes with various configurations & settings options in GUI way.
      * **Pipeline:**
        + Using this option we define the process for build, test & deploy with code either in scripted type/declarative type.
      * **Folder:**
        + This option used to organize the jobs, I mean that if you have 50 jenkins-jobs, out of these 25-related to Team-A & other 25-related to Team-B. So we can create two folders like Team-A & Team-b, we can move the Team-A related jobs to Team-A folder & Team-B related jobs to Team-B folder.
      * **Maven:**
        + This option is mainly designed for build, test & deploy process for Java based applications.
  + Let's choose the freestyle job option & continue.
    - Here what are the different sections to define process for build, test deploy.
      * General
      * Source code management
      * Build Trigger
      * Build
      * Post-build
    - First we go with basic configurations,
      * In General section under **description** we can write **the purpose of the job.(Like: This is my first jenkins freestyle-job)**
      * Under the build section choose execute sh script & enter command to print message **"This is my first jenkins job"**
        + **echo "This is my first jenkins job"**
        + Why did I entered echo command? because jenkins present running on Linux node.
        + **Wherever we created the jobs in jenkins, that node we can refer as master node.**
      * **Save:** If you click on save button you remain in the same page & configuration are applied.
      * **Apply:** When we click on apply if any configurations those will get applied & will go back jenkins main page.
    - Now Execute the job --> See the output of job execution --> **"This is my first jenkins job" got printed**
    - This job triggered for only one time so far based on job history.
    - When you click on execute one more time & we can find the job execution number.

* Hope now you got idea how to create, configure & execute the job.

**================================================================**

* **Home-work**

**================================================================**

* **Create freestyle Jenkins job to print the date**
  + **Job name:** date-job
  + **Description:** To print the date of Jenkins EC2 server
  + **Execute shell:** echo “The date & time is: ” `date`

Jenkins-Day-2

04 December 2023

17:25

* + **How to integrate Jenkins with GitHub**
    - Create a freestyle jenkins job by clicking on New-Item ==> **online-bookstore**
    - Under source code management chose Git
      * Provide the repository URL that we are looking to clone & build the Artifact
      * Also mention branch name that we are looking to clone repository
      * Mostly the branch name will (\*\master)
    - Do we need to enter credentials?
      * The online-bookstore is a public repository, so there is no need of providing credentials.
      * If repository is private repository we have to enter credential's.
    - Apply ==> Execute
    - The job is got failed & If we see job execution log we can find the error.
    - Here what is the error? Git command not found
    - Go & install the Git jenkins server
    - Now re-execute the job, next if we check the Jenkins job log we see repository is cloned successfully.

* + **How to integrate jenkins with Maven**
    - For the same online-bookstore job only after cloning completed we will run the **mvn clean install** command.
    - Build ==> Execute shell ==> mvn clean install
    - The mvn command will run at the location wherever POM.XML present, if we trigger it from any other location it will be failed.

The pom.xml present at top root directory of the repo.

* + Apply ==> Execute
  + Build will failed, why? No maven installed on server
  + Install maven on jenkins server.
  + Global Tools Configuration ==> Provide any random name ==> Enter the maven path

* + **So far the job online-bookstore runs only on the master branch by default, now I want to run the jenkins job one time on master branch & other time on some different branch, how can you do that?**
    - Go to online-bookstore ==> General ==> This project is parameterized ==> Git-Parameter ==> Branch(parameter-name) ==> Apply
    - Now while trying to execute the job now there is no option for **Build & only Build with parameters** option present since this job is enabled for parameterized.
    - After click on the **Build with Parameters** we can see option to choose the branch names & these names of the branches that present in repository.

* + **Create new job similar to (online-bookstore) another one without creating a job from scratch.**
    - Present we have job with name **online-bookstore** & this will clone online-bookstore repository & will build it.

* + Now we will create another job with name **pets-clinic** with the same job configurations of online-bookstore

* + To create new job parameter\_ jobs

New-Item  pets-clinic **copy-from-template**  online-bookstore

* + We can place the repository URL of pets-clinic, so job configuration completed for the pets-clinic completed with less efforts.

* + **Delete job in Jenkins**
    - **Option1:** On job-name  Click on Down arrow  Delete project
    - **Option2:** Click on Job-name  Delete-Project

* + **Jenkins-Dashboard:**

Machine generated alternative text:
25 • 

* + S  Indicates whether job is success/failure

* + W  Sunny, It means all the builds of the jobs are successfully completed.

Cloudy, it means few of the builds of the jobs failed.

Rainy, most of the builds completed with failed status.

* + Last successful build

* + Last failed build

* + Last Duration  For how long last trigger build run to get completed.

* + Scheduled build job  Trigger it to run build for the job “First Jenkins Job”

* + **Build-Triggers:**
    - **So far how are we running the jobs?** We are running manually correct, right? - Yes- just click on build & job is executing
    - There are different ways to execute the jobs rather than click on build icon,
      * **Build the job remotely:** 
        + Provide the authentication token & copy that URL in the separate cognito mode to run the job.
        + Previous the job executed x times now we can it got executed with y times.

* + **Build periodically**
    - We can run the job schedule basis like every day 10AM or every one hour like that
    - To schedule to job fill this five \* with values
      * \*minutes
      * \*hours
      * \*date
      * \*month
      * \* year
  + **Poll-scm**
    - This option make the jenkins to check the GitHub repository based on scheduled windows & if there is any changes build will get triggered.
    - \*\*\*\*\* every minute jenkins check GitHub repository & if there is any changes in repository build will get triggered.

* + **GithHub hook trigger for SCM polling:**
    - Web hooks help us to notify the external services like jenkins whenever certain event occurs in GithHub repositories.
      * External services are nothing but Jenkins
      * Event in GitHub refers branch-creation/push to repo/merge request
    - Suppose whenever developers push/check-in the latest commit into the repository automatically jenkins will run.
    - Let see this scenario practically
      * Create a free-style job
        + Source code <https://github.com/chaitanyaredd/onlinebookstore.git>
        + Invoke top level maven targets  mvn clean install
        + Choose the Build trigger option  GithHub hook trigger for Git SCM polling

* + Create web hook in GithHub repository
    - Go to <https://github.com/chaitanyaredd/onlinebookstore.git> Settings  Add web hook
    - Payload URL https://<jenkins\_url>:8080/github-webhook/
    - Content-Type  Application/json
    - Which event you would like to trigger this web hook?  Push
  + **Trigger other jobs**
    - Assume there are two jobs - job1 & job2
    - You want job-2 run automatically after job1 finish, then we can use this option.
    - Let's try this practically
      * Job1 -->Trigger other job(job-2) --> Shell(echo "This is job1")
      * Job2 --> shell(echo "This is job2")

* + **Homework**
    - Clone repo in jenkins from private GitHub repo
    - Create Maven job & build it
    - **Create string parameterized job & print the value of that string parameter**
      * To create parameterized jobs we have to choose "**This project is parameterized**" option we can create jenkins job.
      * So During job runs we can pass the values to parameters & can us those values of parameters.
      * During parameterized jobs creation we have different options,
        + String
        + Boolean
        + File
        + Password
        + Choose
      * Let's continue job configuration
        + Let's choose the string parameter
        + **Description:** to print the name of Ex-president of US
        + **Execute shell:**

Name=”Trump”  
echo “The Ex-President of US is $Name”

* + Apply
  + Execute the job with option **"Build with parameters"**
    - It will prompt to enter string value
    - We can provide any values as we need.

* + **Create parameter job2: Create choose parameterized job & print the value of that choose parameter**.
    - Pass the Name as string parameter & Execute it
    - Pass the Name as choice parameter & Execute it

Jenkins-Day-3

* + Today session we will see basic configurations that we need to do in Jenkins.
    - For that let's go to

Manage Jenkins  Configure system

* + **home directory** 
    - It represent the home directory of jenkins & contains the configuration details of Jenkins server like
      * Jobs
      * Plugins
      * Logs
      * Secrets ..etc

* + By default on Linux servers the jenkins home directory is **/var/lib/jenkins**

* + Sometime if there is any space restriction issues we can change this home directory to different location(/home/jenkins/jenkins\_home).

* + **System Message** 
    - What is System Message?
      * The message whatever we give in this text box it will be displayed on Jenkins console page.

* + Let me put message here “This is used of DEVOPS Training”.

* + Now we can see some message on Jenkins console.

* + We write the system message in html format, So we can get the message in correct font on jenkins console page.

<h1> This is used of DEVOPS Training </h1>

* + Preview
  + Apply
  + Check in Console

* + If you are not able to use the html
    - Go to Configure Global Security  Markup Language  choose safe html

* + **No. of Executors**
    - It indicates **how many number of parallel jobs can run at same time**.

* + Assume there are 10 jobs & if we trigger these 10 jobs at same time only 2 two jobs will executed & after that other 2 jobs will get executed.

* + **Quiet Period**
    - No. of seconds, that Jenkins will wait to trigger the job.

* + Why Jenkins will wait this many seconds to trigger job?
    - Normally Jenkins job will get run on scheduled basis/whenever there is push/check-in happened in GitHub, there might be chances sometimes check-in may not immediately reflect in GitHub but after few seconds will updated. So by waiting some sleep time will fix this kind of issues.

* + **SCM Checkout retry count**
    - There might be scenarios Jenkins may failed to run job because of network glitches while cloning the code from GitHub.

* + Instead of failing job if its failed to clone repository for first time, we mention the number of times that jenkins has to try to clone the repository in jenkins. Since we are re-trying multiple times to clone repository there will be chances getting completed are high.

* + Create Jenkins Job to download source code
    - New Item  SCM checkout  Clone repo from master  the error message we are getting jenkins not sure where git installed.
    - New Item  SCM checkout  Clone repo from master  Job will passed
    - SCM Checkout  Update branch abc  Job Failed
    - Update the SCM Checkout retry count in Jenkins, so Jenkins makes the job to run this many times to clone repo.

* + **Restrict project Naming**
    - As of now we creating jobs with any name, correct/not? - Correct
    - If you want to create job name with specific name, give pattern name as test\*
    - Now try to create job with name Check-test, it will not allow us to create.
    - Try to create a job with name Test, it will allow us to create job.
    - As of we are going with Default

* + **Global Properties**
    - Suppose you want to declare the environment variables for all the jobs, we can declare those variables here.
    - Global Properties  Environment Variables  Key1  Value1
    - Now the environment variable Key1 & its value available for all the jobs as $Key
    - Tool locations  We can mention tools that should be used for all the projects.

* + **Jenkins Location**
    - Jenkins URL  Mention correct IP address of machine where Jenkins installed.
    - Email  Email that is used to send notification once builds are passed/failed.

* + Shell Executable 
    - Mentions location of the shell that Jenkins to be used.
    - Jenkins by default use bash shell
    - If you want to use windows shell you can C:\cmd.exe

* + **Manage Plugins**
    - **Why do we need plugins?**

Plugins will provide features that helps to integrate the Jenkins with other tools like,

* + Git
  + Maven
  + Sonarqube
  + Docker
  + Ansible
  + Kubernetes …etc.

* + **Why do we need to integrate Jenkins with other tools?**

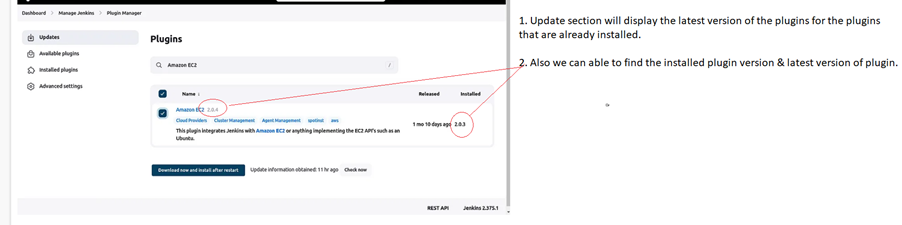
In order to meet the pipeline setup requirements.

* + The Plugins are in format like **.hpi**

* + **Where can we find the plugins?** 
    - Manage jenkins --> Manage Plugins

* + **Installed plugins:** 
    - This section will display list of installed plugins.
    - These are the basic plugins that are installed during jenkins installation.
    - On Jenkins server we can find plugins that are installed in **<Jenkins\_homedir>/plugins** folder

* + **Available plugins:**
    - From this section we can choose the plugin that are needed to integrate with Jenkins.
    - Suppose If I want integrate jenkins with Sonarqube I have to install the **sonar-scanner** plugin.
      * Manage jenkins --> Manage plugins --> Available --> sonar-scanner
      * Now jenkins will install the sonar-scanner plugin along with dependencies.
      * Next open jenkins job it will show the options to configure sonarqube related commands.
      * Without sonar-scanner plugin you can't able to see this options.
  + **Updates:** 
    - Click on **Updates.**
    - Search the plugin you want to upgrade in the search box
    - Here, we can see the latest version and the currently installed version of the plugin.
    - Click the checkbox to the required plugin.
    - After that, Click the **Download now and install after restart**button.



* + **How to uninstall the plugin in Jenkins?**
    - Click on **Installed plugins**.
    - Search the plugin that you want to uninstall in the search box
    - Click on the **Uninstall** icon.
  + **Install plugin by uploading hpi**



* + How to integrate Jenkins with sonarqube
    - **Pre-requisites**
      * Install sonar-scanner plugin
      * Repo: <https://github.com/chaitanyaredd/onlinebookstore.git>
      * Git
      * Maven
      * Sonar-scanner
      * Jenkins
      * Sonarqube(In separate EC2)

* + Integration of Jenkins with sonarqube
    - **First step Generate User Token on sonarqube server**
      * Log in to SonarQube Server and go-to the “**My Account**” section on your profile. And move to the “**Security**” tab. Then, Generate a “**User Access Token**.”

* + **Login > Profile > My Account > Security > Generate Token**

* + **Add SonarQube Authentication Token Into Jenkins**

Head-over to Jenkins server and go-to **Jenkins > Credentials > System > Global Credentials > Add Credentials**

**Kind**: Secret test

**Secret**: SonarQube Authentication Token

**Description**: Provide a descriptive name

Click **OK** to add new credentials.

* + **Add SonarQube Server on Jenkins**

Now, We need to add SonarQube server settings into Jenkins.

**Manage Jenkins > Configure System > SonarQube servers**

* + **Add Sonar-Scanner For Jenkins in Global tools configuration**

Go to Jenkins > Manage Jenkins > Global Tool Configuration > SonarQube Scanner [Scroll Down] > Add SonarQube-Scanner

* + Create free-style job & execute sonar-scanner analysis(work-out practically)

Jenkins-Day-4

07 December 2023

18:10

* **What is Tomcat server?**
  + Tomcat is web application server.

* What is web application?
  + Sites like [www.amazon.com](http://www.amazon.com) / [www.facebook.com](http://www.facebook.com) / [www.netflix.com](http://www.netflix.com) are considered as web applications.

* The web applications artifacts like .war files we can deploy on tomcat server & can access those deployed programs via browsers

* **How to install & configure Tomcat?**

* To install tomcat we need **pre-requisites** like
  + **Install java-11**
    1. List java-11 related packages are available

yum list | grep -i java-11\*

1. Install the java-11

yum install java-11\* -y

1. Check the java version

java -version

* **Download the tomcat binary from internet & extract it, later copy it to the /opt folder**

wget <https://archive.apache.org/dist/tomcat/tomcat-10/v10.1.13/bin/apache-tomcat-10.1.13.tar.gz>

tar -xzvf apache-tomcat-10.1.13.tar.gz

mv apache-tomcat-10.1.13 tomcat

mv tomcat /opt/

* **Start tomcat**

./opt/tomcat/bin/startup.sh

* **Check tomcat is running or not**

ps -ef | grep -i tomcat

* Now tomcat is accessible on browser with URL

**http://<ip-address>:8080**

* Update content.xml file to in-order to have manger-app/server-status check access,

vi /opt/tomcat/webapps/manager/META-INF

Machine generated alternative text:
--4'aIve apache. catal i na. val ves. RemoteAddrvaIve 
al Nd+X. Nd+X. : 

* **To login tomcat setup via GUI credentials**

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

Machine generated alternative text:
krole 
(user username:" tomcat" "0 

* **To deploy applications into tomcat we have to user role "manager-script"**

* **Let's manually deploy sample web application & check manually we are able to access or not.**

cd /opt/tomcat/webapps

wget <https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/sample.war>

Now we are able to sample application through tomcat.

* In this case for your understanding purpose only I download the .war file directly from internet, but in real-time we have to create it out of source code & later we deploy to tomcat server.

* **Deploy web applications on tomcat via jenkins pipeline.**

* Now we will discuss how to download latest code from GitHub, build that code & deploy tomcat server with help of jenkins we will see.
* Machine generated alternative text:
  Clone 
  GitHub 
  Build 
  Maven 
  deploy 
  Tomcat 

* **Pre-requisites:**
  + EC2-SERVER-1(t2.medium)
    - Git
    - Maven
    - Java
    - Jenkins
    - Repo: online-bookstore

* EC2-server-2(t2.micro)
  + Tomcat

* Deploy to container plugin

* **Free-style job configuration**
  + src: <https://github.com/chaitanyaredd/onlinebookstore.git>
  + Build: mvn clean install
  + Post build steps:
    - Provide location of war file: \*\*/.war {recursively look under the workspace folder for war file}
    - Provide context as you want, since we are configuring deployment for online-bookstore, I am giving as online-bookstore
    - Choose tomcat container as tomcat-9.x
      * Enter tomcat url
      * Enter tomcat credentials - (The credentials must have the manager-script role, then we can deploy application to tomcat from jenkins)

Machine generated alternative text:
«role rolename-"manager-scri pt"b 
«user 

* Save --> Apply
* Now execute the job
* Now we can able the access the online-bookstore application

http://<tomcat-server>:8080/online-bookstore

* **Discard old builds**

* **Choose custom workspace**

* **Archive the jar/war files**

* **Publish junit reports**

Jenkins-Day-5

08 December 2023

18:51

* **How to integrate Jfrog with Jenkins & upload artifacts**

* **Pre-requisites**
  + EC2-SERVER-1
    - Git
    - Maven
    - Jenkins
    - Repo: online-bookstore

* EC2-SERVER-2
  + JFROG

* EC2-SERVER-3
  + Tomcat

* Artifactory plugin

* Create a new Freestyle project in Jenkins.

* Configure your source code repository if your job involves source code.

* Configure Build Steps:
  + mvn clean install
  + To upload artifacts to Artifactory, you can use the "Deploy artifacts to Artifactory" build step. Provide the necessary details like the target repository, path, and artifacts to deploy.  
    Example for uploading artifacts:  
      
    sh 'echo "Build completed"'  
    rtUpload (  
     serverId: 'ArtifactoryServer',  
     spec: '''{  
     "files": [  
     {  
     "pattern": "target/\*.jar",  
     "target": "libs-release-local/"  
     }  
     ]  
     }'''  
    )

* **How to integrate Jfrog with Jenkins & download artifacts**

To download artifacts from Artifactory, you can use the "Download artifacts from Artifactory" build step. Provide the necessary details like the source repository, path, and artifacts to download.  
Example for downloading artifacts:  
  
  
rtDownload (  
 serverId: 'ArtifactoryServer',  
 spec: '''{  
 "files": [  
 {  
 "pattern": "libs-release-local/\*.jar",  
 "target": "downloaded-artifacts/"  
 }  
 ]  
 }'''  
)

Jenkins-Day-6

08 December 2023

19:12

* **Jenkins file**

* **Normally in Jenkins how do we create & configure the jobs?**

GUI way.

* **What is the problem when your jenkins server crashed?**
  + You have to recreate jenkins server and **recreate the jobs and it's configurations**.

* This will be tedious task because you will have many number of jobs and lot of configurations so there is no chance of remembering to all those job details and configurations.

* Also when multiple devops engineers are working on same jerkins instance it is very difficult what another person made the configuration changes on the jobs.

* To avoid this problem we can use jenkinsfile script in order to maintain the configurations of the jobs & and we will maintain this script file in GitHub.

* Since we are maintaining the jenkinsfile in the GitHub, it is easy to track the pipeline changes and get review those changes.

* We can write the script for the Jenkins file two ways
  + one is **scripted pipeline**
  + another one is **declarative pipeline**

Declarative pipeline is much more easy compared to the scripted pipeline this is developed by the Jenkins team to make the people more convenient to write the Jenkins file.

But if you want to write scripted by plane you need to have knowledge on the groovy script.

Scripted pipeline syntax looks like

node(

  //groovy script

)

Declarative pipeline syntax looks like

Now let's go to the official site of the chin games and we will take a look how the scripted pipeline and declarative pipelines syntax.

Here you can see scripted by plane there is a stage test stage and placed similarly in the declarative pipeline also we can see build stage and depli station so here the stage nothing but job in the Jenkins.

Now let's copy this declarative pipeline code and do some modifications in order to run the pipeline.

As of now in each stage I am just printing the statement what is stages doing but in our real time we have to enter the commands that we need to run.

Pic-3

Usecase1: Run Jenkins file from the Jenkins job only instead from GithHub.

**To use the declarative pipeline we have to install pipeline plugin.**

Pic-4

* **Usecase2: how to run multi step shell commands in jenkinsfile**

pipeline {

agent any

stages {

stage('Test-Multi-shell-commands'){

steps {

sh 'echo "Executing first command"'

sh '''

echo "Executing second command"

pwd

ls -ltr

'''

}

}

}

}

* **Usecase3: use retry option whenever them are any command is failing frequently.**

pipeline {

agent any

stages {

stage('Timeout'){

steps {

retry(3){

sh 'echo "retriggering the command"'

}

}

}

}

}

* **Usecase4: Fail the job if it is taking more than 60 seconds using time out option.**

pipeline {

agent any

stages {

stage('Timeout'){

steps {

retry(3){

sh 'echo "retriggering the command"'

}

timeout(time: 30, unit: 'SECONDS') {

sh 'sleep 60'

}

}

}

}

}

* **Usecase5: how to set environment variables in the pipeline**

pipeline {

agent any

environment {

NAME = "Chaitanya"

COUNTRY = "India"

}

stages {

stage('Env variables') {

steps {

sh 'echo $NAME $COUNTRY '

}

}

}

}

* **Usecase6: post steps after the job completed**

pipeline {

agent any

environment {

NAME = "Chaitanya"

COUNTRY = "India"

}

stages {

stage('Env variables') {

steps {

sh 'echoxea $NAME $COUNTRY '

}

}

}

post {

success {

echo "This will executed when job is success"

}

failure {

echo "This will executed when job is failed"

}

unstable {

echo "This will executed when job is unstable"

}

always {

echo "This will executed irrespective of job status"

}

}

}

* **Usecase7: How to execute maven command**

Jenkins-Day-7

05 October 2023

12:39

* 1. **Requirement:**

Machine generated alternative text:
clone repo 
Pipeline that will run during the latest check-in in the master branch. 
Download binary from 
artifactory 
Deploy binary 
to tomcat DEV 
environment. 
Build 
Sonarqube scanning 
Binary push to 
Quality gate check 
artifactory 

* 1. **Pre-requisites**
     1. EC2 instance with t2.large
     2. Git
     3. Maven
     4. Sonarqube
     5. Jenkins
        1. Sonarqube plugin
        2. Artifactory plugin
     6. Artifactory --> t2.large
     7. Tomcat --> t2.large
     8. Plugins
        1. Artifactory
        2. Sonarqube scanner
        3. Pipeline utility steps(Find files for binary deploy to tomcat deployments)
        4. Deploy to container

* 1. **Clone repo using jenkinsfile**

pipeline {

agent any

stages{

stage('clone') {

steps {

checkout scm

}

}

}

}

* 1. **Build the code**

pipeline {

agent any

stages{

stage('clone') {

steps {

checkout scm

}

}

stage('Build') {

steps {

sh 'mvn clean install'

}

}

}

}

* 1. **Build will failed due to maven command not found & define the maven**

pipeline {

agent any

tools {

maven 'maven-3.6.3'

}

stages{

stage('clone') {

steps {

checkout scm

}

}

stage('Build') {

steps {

sh 'mvn clean install'

}

}

}

}

* 1. **Do sonar scanning, it will fail due sonar scanner plugin not present & sonar. Login not defined**

pipeline {

agent any

tools {

maven 'maven-3.6.3'

}

stages{

stage('clone') {

steps {

checkout scm

}

}

stage('Build') {

steps {

sh 'mvn clean install'

}

}

stage('sonarqube-scanning'){

steps {

withSonarQubeEnv('My SonarQube Server') {

sh 'mvn sonar:sonar'

}

}

}

}

}

* 1. **Do sonar scanning with the parameters**

pipeline {

agent any

tools {

maven 'maven-3.6.3'

}

stages{

stage('clone') {

steps {

checkout scm

}

}

stage('Build') {

steps {

sh 'mvn clean install'

}

}

stage('sonarqube-scanning'){

steps {

withSonarQubeEnv('My SonarQube Server') {

sh 'mvn sonar:sonar \

-Dsonar.projectKey=online-book-store \

-Dsonar.projectName="online-book-store" \

-Dsonar.login=sqa\_f78d0d4e2a2be94550489789968eff5b801e53f9'

}

}

}

}

}

* 1. **Add quality gate to the jenkinsfile**

pipeline {

agent any

tools {

maven 'maven-3.6.3'

}

stages{

stage('clone') {

steps {

checkout scm

}

}

stage('Build') {

steps {

sh 'mvn clean install'

}

}

stage('sonarqube-scanning'){

steps {

withSonarQubeEnv('My SonarQube Server') {

sh 'mvn sonar:sonar \

-Dsonar.projectKey=online-book-store \

-Dsonar.projectName="online-book-store" \

-Dsonar.token=squ\_c7fbb3bf3647607967ac2ed5a162ed0c5603bc67'

}

}

}

stage("Quality Gate") {

steps {

retry(3) {

sh 'sleep 30s'

waitForQualityGate abortPipeline: true

}

}

}

}

}

* 1. **Fail the pipeline by enabling quality gate in sonarqube**

pipeline {

agent any

tools {

maven 'maven-3.6.3'

}

stages{

stage('clone') {

steps {

checkout scm

}

}

stage('Build') {

steps {

sh 'mvn clean install'

}

post {

always {

archiveArtifacts artifacts: '\*\*/\*.war', onlyIfSuccessful: true

}

}

}

stage('sonarqube-scanning'){

steps {

withSonarQubeEnv('My SonarQube Server') {

sh 'mvn sonar:sonar \

-Dsonar.projectKey=online-book-store \

-Dsonar.projectName="online-book-store" \

-Dsonar.token=squ\_c7fbb3bf3647607967ac2ed5a162ed0c5603bc67'

}

}

}

stage("Quality Gate") {

steps {

retry(3) {

sh 'sleep 30s'

waitForQualityGate abortPipeline: true

}

}

}

stage('Upload to Artifactory') {

steps {

script {

// Define a new Artifactory server instance

def server = Artifactory.server 'jfrog-instance' // Replace 'ServerID' with your Artifactory server ID from Jenkins configuration.

// Define the artifact

def uploadSpec = """{

"files": [

{

"pattern": "\*\*/\*.war",

"target": "online-bookstore/${BUILD\_ID}/onlinebookstore-1.0.war"

}

]

}"""

// Upload the artifact

server.upload(uploadSpec)

}

}

}

}

}

* 1. **Download the artifact from jfrog to server**
     1. Install JFROG
     2. Create repo in jfrog maven based "online-bookstore" repo
     3. Install artifactory plugin
     4. Integrate Jenkins & Artifactory

pipeline {

agent any

tools {

maven 'maven-3.6.3'

}

stages{

stage('clone') {

steps {

checkout scm

}

}

stage('Build') {

steps {

sh 'mvn clean install'

}

post {

always {

archiveArtifacts artifacts: '\*\*/\*.war', onlyIfSuccessful: true

}

}

}

stage('sonarqube-scanning'){

steps {

withSonarQubeEnv('My SonarQube Server') {

sh 'mvn sonar:sonar \

-Dsonar.projectKey=online-book-store \

-Dsonar.projectName="online-book-store" \

-Dsonar.token=squ\_c7fbb3bf3647607967ac2ed5a162ed0c5603bc67'

}

}

}

stage("Quality Gate") {

steps {

retry(3) {

sh 'sleep 30s'

waitForQualityGate abortPipeline: true

}

}

}

stage('Upload to Artifactory') {

steps {

script {

// Define a new Artifactory server instance

def server = Artifactory.server 'jfrog-instance' // Replace 'ServerID' with your Artifactory server ID from Jenkins configuration.

// Define the artifact

def uploadSpec = """{

"files": [

{

"pattern": "\*\*/\*.war",

"target": "online-bookstore/${BUILD\_ID}/onlinebookstore-1.0.war"

}

]

}"""

// Upload the artifact

server.upload(uploadSpec)

}

}

}

stage('Download from Artifactory') {

steps {

script {

sh 'mkdir -p ${WORKSPACE}/ready\_deploy\_binary/'

// Define the Artifactory server instance

def server = Artifactory.server 'jfrog-instance' // Replace 'ServerID' with your Artifactory server ID from Jenkins configuration.

// Define download spec

def downloadSpec = """{

"files": [

{

"pattern": "online-bookstore/${BUILD\_ID}/onlinebookstore-1.0.war",

"target": "${WORKSPACE}/ready\_deploy\_binary/"

}

]

}"""

// Download the artifact

server.download downloadSpec

}

}

}

}

}

* 1. **Deploy Artifact to tomcat server**
     1. Install tomcat
     2. Create users with Manage-gui & Manage-script role
     3. Install deploy to container plugin
     4. Define tomcat username & password in jenkins
     5. Install credentials binding plugin
     6. Install BUILD\_TIMESTAMP
     7. Correct the TIMESTAMP as "yyyy-MM-dd-HH-mm-ss-z"

pipeline {

agent any

tools {

maven 'maven'

}

stages {

stage('clone-repo'){

steps {

checkout scm

}

}

stage('Build') {

steps {

sh 'mvn clean install'

}

}

stage('sonarqube-scanging'){

steps {

withSonarQubeEnv('my-sonarqube-server') {

sh 'mvn sonar:sonar'

}

}

}

stage('quality-gate-check'){

steps {

sh 'sleep 30s'

waitForQualityGate abortPipeline: true

}

}

stage('artifactory-upload'){

steps{

script {

// Define a new Artifactory server instance

def server = Artifactory.server 'jfrog-instance' // Replace 'ServerID' with your Artifactory server ID from Jenkins configuration.

// Define the artifact

def uploadSpec = """{

"files": [

{

"pattern": "\*\*/\*.war",

"target": "online-bookstore/${BUILD\_TIMESTAMP}/onlinebookstore-1.0.war"

}

]

}"""

// Upload the artifact

server.upload(uploadSpec)

}

}

}

stage('download from artifactory'){

steps{

script{

sh 'mkdir -p ${WORKSPACE}\readytodeploy'

def server = Artifactory.server 'jfrog-instance'

def downloadSpec = """{

"files": [

{

"pattern": "online-bookstore/${BUILD\_TIMESTAMP}/onlinebookstore-1.0.war",

"target": "${WORKSPACE}/ready\_deploy\_binary/"

}

]

}"""

// Download the artifact

server.download downloadSpec

}

}

}

stage('Deploy to Tomcat') {

steps {

// Use Jenkins credentials binding

withCredentials([usernamePassword(credentialsId: 'TomcatCredentials', usernameVariable: 'TOMCAT\_USER', passwordVariable: 'TOMCAT\_PASS')]) {

script {

def tomcatUrl = '<http://18.234.161.208:8080/manager/text/deploy>'

def warFile = '${WORKSPACE}/ready\_deploy\_binary/${BUILD\_TIMESTAMP}/onlinebookstore-1.0.war' // Adjust the path to your WAR file.

echo "Executing: curl -T ${warFile} -u $TOMCAT\_USER:$TOMCAT\_PASS '${tomcatUrl}?path=/your-context-path&update=true'"

sh """

curl -T ${warFile} -u $TOMCAT\_USER:$TOMCAT\_PASS "${tomcatUrl}?path=/online-bookstore&update=true"

"""

}

}

}

}

}

}

* 1. **Make sure jfrog & tomcat deploy runs only on master branch**

when {

// Only run this stage when the BRANCH\_NAME is 'master'

expression {

env.BRANCH\_NAME == 'master'

}

}

Ansible-Day-1

11 December 2023

17:06

===================================================================================================================

* **Introduction**

===================================================================================================================

* Before we discuss about Ansible first will discuss about configuration management tool.

* **What is configuration Management?**
  + Normally how we will install some software in any machine,
    1. We just login to server manually
    2. Install the required software (tomcat)
    3. Do the changes in configuration of software files to make our application run smoothly.

* Now assume when you have situations to install same software on large number of servers,
  + Consume lot of time to perform installation/configuration
  + Chances for human errors
  + Lot of human efforts required

* To overcome this issues configuration management tool came into picture.

CM tool taken responsibility for making the target systems & software's in desired and consistent state.

* In market we have different types of CM tool's
  + Ansible
  + Chef
  + Puppet
  + Salt

* **Ansible** not only acts as **configuration management tool** but it also support **orchestration tool.**

* **Orchestration tool nothing provisioning/creating resources like ec2 on public clouds like ec2,alb..etc. similar to terraform.**

* But best for choosing Ansible for CM & terraform as orchestration tool since some of advanced feature in terraform.

* Out of these CM tools Ansible is the widely used CM tool.

Machine generated alternative text:
Ansible 
Chef 
Puppet 
Salt 

===================================================================================================================

* **Ansible architecture**

====================================================================================================================

* Create & Explain diagram ansible\_day\_1.png

Machine generated alternative text:
Node-I 
RHEL 
Ansible controller 
Linux 
Node-2 
Ubuntu 
Node-3 
Windows 

* **Ansible-controller:** 
  + It's a server where Ansible is installed .

* From this server we can control the install/update/delete configurations for the software on particular machines.

* Mostly the controller servers is on RHEL OS.

* **Node:** 
  + It's a server **where we have to install/Update/Configure software** through the Ansible controller.

* Node machine can run on any OS Linux/Ubuntu/Windows

* The connectivity established from Ansible controller to target machines using SSH key based authentication.

* On target servers we don't required to install any agent software to establish connectivity to controller.

* **hosts** 
  + Contains the list of target machines IP/hostnames.

* **Playbook**
  + The script files that we have written to automate the tasks.

* **What is task?**

Downloading file from internet/Unzipping the file ….etc are consider as tasks.

* The playbooks will be written in yml format.

* YAML nothing but Yain't mark-up language.

* It's a declarative language. Easy to understandable & writable.

=========================================================================================================

* **Why Ansible is popular than chef?**

=========================================================================================================

* Ansible works on push based mechanism but chef works on pull based mechanism.

Machine generated alternative text:
Node-I 
RHEL 
Ansible controller 
Linux 
Node-2 
Ubuntu 
Node-3 
Windows 
NJ 

* Suppose you want to install tomcat software on node machines, the controller machine pushes the commands/configurations that needs to be executed node machines.

* Chef client on node machines pulls the configuration/commands that needs to be executed from chef server. So Chef is working on pull based mechanism.

* Ansible is a agentless configuration management tool but chef is an agent based configuration management tool.

Machine generated alternative text:
Node-I 
RHEL 
Ansible controller 
9 
Node-3 
chef-cli nt 
Windows 
chef-server 
Linux 
Node-2 
Ubuntu 
NJ 
chef-cli nt 
o 
chef-clie 

* On Node machines chef-client software need to be installed & this client will connect to the chef-server and pulls the configurations on regular intervals.

* For Ansible we don't need to install any software on machines.

* Ansible is supported by Red-hat systems, so this is add-on all Unix flavours.

=====================================================================================

* **Ansible setup**

=====================================================================================

* Create one Ansible controller & 3 target machines (RHEL, Amazon Linux, Ubuntu)

* Prepare Ansible controller server/EC2
  1. Create EC2 instance
  2. Create ansible user
  3. Add ansible user to sudoers file
  4. Generate SSH keys for ansible user
  5. Enable password based authentication
  6. Check python presence
  7. Install ansible

* Prepare target node
  1. Create EC2 instance
  2. Create ansible user
  3. Add ansible user to sudoers file
  4. Enable password based authentication
  5. Check python presence

* Copy the key of ansible user from controller to target machine ansible user

**ssh-copy-id ansible@<target-ec2-ip>**

* Adding target machine details into inventory file

* Test the connectivity from controller to target machines

**ansible -m ping all**

* Provide some wrong IP address & check connectivity

Ansible-Day-2

11 December 2023

19:13

==========================================================================================

* **Ansible components**

==========================================================================================

* Before jump into ansible further we have to get familiar with few of terminologies in ansible

* **/etc/ansible/ansible.cfg** 
  + Is the default configuration file of ansible, which is useful to manage ansible playbooks or ansible commands.

* I will discuss more details with example in sometime.

* **/etc/ansible/hosts** 
  + Is the default inventory file.

* It contains list of target machine IP addresses/hostnames.

* **tasks** 
  + Activities that we are going to managed on target servers.
    - Installing git activity on target server will be considered as one activity.

yum install git -y

* Starting the **httpd** service activity on target server will be considered as one activity.

systemctl start httpd

* **playbooks** 
  + Playbooks are collection of tasks

* **modules** 
  + Modules are predefined commands/units which are used inside the playbooks or commands in order to execute on target machines.

==========================================================================================

**Ansible Ad-hoc commands**

==========================================================================================

* When you have a need of run repeated task on setup of servers then we can use ansible ad-hoc commands.

* Assume in your company there are 1000 servers are there & out these servers how many servers are running & how many servers are not running, how can you find it using ansible?

ansible -m ping all

syntax:

ansible -m <module\_name> -a "argument" <group\_name/all>

* **Command module**:- Used execute any shell command target machine, then we use shell command

Find out uptime of the list of servers.

ansible -m command -a "uptime" all

Find out the home directory of ansible user in target machines

ansible -m command -a "pwd" all

* **stat:** Displays the statics of the file,
  + **what is statistics of a file?**
    - Who is the owner of file?
    - What permissions are there on file?
    - Is it directory or file?

ansible -m stat -a "/etc/hostname" all

* **group**: Used to create/delete groups on target machines {Cache issue}
  + Create a Unix user group with ansible ad hoc command

* Create a user name group named **dbadmin** using the ansible group module.

ansible -i ~/hosts -m group -a "name=dbadmin state=present" all

* Delete a user name group named **dbadmin** using the ansible group module.

ansible -i ~/hosts -m group -a "name=dbadmin state=absent" all

* **COPY**: Used to copy the files to target machine from controller machine

ansible -i ~/hosts -m copy -a "src=/etc/hosts dest=/tmp/hosts mode=600"

* **file:** To create a file/directory on target nodes
  + Create a new directory with 755 permission

ansible -m file -a "path=/opt/oracle state=directory mode=0755" all

* Create a new file with 755 permission

ansible -m file -a "path=/opt/oracle state=file mode=0755" all

* **yum:** Install a package or software in Linux using yum module of ansible

ansible -m yum -a "name=httpd state=installed" all

* Remove sudo access on target & repeat yum module, if we don't have sudo access on remote installation will be failed.

* **service:** start/stop httpd service on target machine

* Start httpd service & enable it to auto restart on server reboot

ansible -m service -a "name=httpd state=started enabled=yes" all

* To Stop

ansible -m service -a "name=httpd state=stop enabled=yes" all

* **get\_url:** Download a file from URL in ansible

ansible -m get\_url -a "url=https://nodejs.org/dist/v14.17.4/node-v14.17.4-linux-x64.tar.xz dest=/tmp mode=0755" all

* **setup:** Displays facts about the target machines.
  + What do you meant by facts of target machines?
    - Ip-address
    - Hostname
    - OS family
    - Architecture
    - Network configurations
    - RAM
    - File system details

ansible -m setup all

==========================================================================================

**Inventory file**

==========================================================================================

* The inventory file have a set of nodes on which we are going to install/update/delete/configure software.

* The default inventory file is **/etc/ansible/hosts**

* We can't change every time the default inventory file we can change it using -i option

**-i hosts**

* Use inventory file other than default inventory file

ansible -i ~/hosts -m command -a "uptime" all

* We can change the inventory file location to different file permanently by modifying ansible.cfg
  + inventory=/root/hosts

* library ==> Contains modules that are present ansible

* remote\_tmp ==> When we execute ansible commands it requires some temporary space required on remote side.

* local\_tmp ==> When we execute ansible commands it requires some temporary space required on local side.

* forks ==> The number of parallel executions on target machine
  + suppose if you have 100 machines & fork count is 5 only first 5 servers completed & next go for another 5 like that till 100

* poll\_intervel ==>

* module = command {default module that can picked automatically when we haven't provided in adhoc commands}

ansible all -a "uptime"

* Change the module from command to yum

ansible all -a "name: git state: present"

* Change module to yum & execute

ansible all -a "name: git state: present"

Explain Idempotency

Explain change/without &

Ansible-Day-3

13 December 2023

12:05

=========================================================================================================

**Ansible modules**

=========================================================================================================

* Modules are predefined commands/units which are used inside the playbooks or commands in order to execute specific tasks on target machines.

* Ansible have large extent number of modules, we can view it on ansible community side.

* Explain user module & required parameters

=========================================================================================================

**Ansible playbook syntax**

=========================================================================================================

* Ansible playbook is a text file written in YAML(Ain't markup language) & stored with format .yml

* The sample format of that ansible playbook appears as

---

- hosts: all

become: yes

gather\_facts: yes

tasks:

- task1

- task2

- task3

Where

* --- Indicates the starting of the playbook

* - hosts: Points the list of hostnames that we are targeting to configure.

* gather\_facts: Capture the facts about the target machines

* become: The ansible user becomes the root.

* tasks: List of the activities that we are going to do.

* task1: activity that we are going to do

**Ex:**

**- name: "Installing git"**

**yum:**

**name: git**

**state: present**

* **Create a playbook for userid creation**

---

- hosts: all

become: yes

gather\_facts: yes

tasks:

user:

name: dhoni

state: present

* ansible-playbook -i ~/hosts -v user\_creation.yml

The output appears in the following format

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

client-0x0001 : ok=5 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

* changed ==> The state of the target machine modified after executing particular task.

* ok ==> means there was no change on target machine state after executing particular task.

* unreachable ==> Not able to connect target machine, try with wrong ip address

* failed ==> Specific task failed due to some errors , try with error wrong package name

* skipped ==> Particular task skipped to run on specific host machine, run a task only on Ubuntu {Discuss during conditional statements}

tasks:

- name: "Install git on rhel"

yum:

name: git

state: present

when: ansible\_os\_family == rhel8

- name: "Install tree on ubuntu"

apt:

name: tree

state: present

when: ansible\_os\_family == ubuntu

rescued: Need to work on this. {Discuss during conditional statements}

Explain every line ansible-day-3.png image

* The playbooks uses the indentation with spaces in order to indicate structure of the data.

---

- hosts: all

become: yes

tasks:

- user:

name: john

state: present

Since we didn't followed indentation in right format we received syntax errors.

* gather\_facts ==>

* Used to fetch target machine meta data details/facts

Example:

"ansible\_nodename": "ip-172-31-21-77.ec2.internal",

"ansible\_os\_family": "RedHat",

"ansible\_pkg\_mgr": "yum"

* By default gather\_facts are enabled

* When we don't have any need of facts we can disable it by setting gather\_facts to no

---

- hosts: all

become: yes

gather\_facts: no

tasks:

- user:

name: john

state: present

* **become** 
  + This used to get privilege escalation option & converts ansible user to get sudo privilege's,

* similarly that we used "-b" option in ad-hoc commands during group creation.

* **UC:- Create a user with become no option.**

---

- hosts: all

gather\_facts: no

become: no

tasks:

- name: "create user"

user:

name: john

state: present

* ansible-playbook -i ~/hosts -v user\_creation.yml

* The execution will get failed due to the permissions issue & we can solve it by setting become to yes.

* By default become is set to no, if we don't use become option in our playbooks.

* **--check**:
  + When ansible-playbook is executed with --check it will not make any changes on remote systems.

It will display report what changes they would have made rather than making them.

* **UC:- Create a playbook for kohli user creation & before execute the playbook run it on check mode**

**On target machine we don't see the kohli user after mode, since mode id dry run,**

It just report user creation changes is going to happen

---

- hosts: all

become: yes

gather\_facts: no

tasks:

- name: "create a user"

user:

name: kohli

state: present

* ansible-playbook -i ~/hosts -v user\_creation.yml --check

Even we have executed our playbook, it hasn't created kohli user, It's just reported the changes what are all going happen.

* Remove the --check option & execute playbook, now kohli user added on target system.

ansible-playbook -i ~/hosts -v user\_creation.yml

* **UC1:- Create a playbook for installing "git" package on ubuntu & rhel machine.**

---

- hosts: all

gather\_facts: no

become: yes

tasks:

- name: "Install git"

yum:

name: git

state: present

* ansible-playbook -i ~/hosts -v git\_install.yml

* The execution will be failed due to By default

"yum" will acts package repo for RHEL machines &

"apt" will acts package repo for Ubuntu machines.

* To overcome this issue we use "package" module to support installation of packages on any Linux flavours when package name same for all flavours.

---

- hosts: all

gather\_facts: no

become: yes

tasks:

- name "Installing git"

package:

name: git

state: present

ansible-playbook -i ~/hosts -v git\_install.yml

* **UC2:- Playbook for Install apache(httpd) webserver, start it & later configure it**

yum install httpd

service httpd start

echo "Welcome to ansible" > /var/www/html/index.html

---

- hosts: all

become: yes

gather\_facts: no

tasks:

- name: "Installing httpd"

yum:

name: httpd

state: present

- name: "starting httpd service"

service:

name: httpd

state: started

- name: "Configure the httpd service"

command: echo "Welcome to ansible" > /var/www/html/index.html

**ansible-playbook -i ~/hosts -v httpd\_setup.yml**

* Grouping & subgrouping of servers

* LAMP stack setup with installation commands & customize it with loop

Ansible-Day-4

13 December 2023

13:07

=========================================================================================================

**LOOP in ansible**

=========================================================================================================

* loop keyword used to execute the task multiple times.

Example:

- Creating multiple users with user module

- Changing the ownership for multiple files

* Just few versions before instead of loop, with\_items was used to support loop functionality & now its depricated.

* **usecase1: create testuser1,testuser2 & testuser3 using loop in the single task**

---

- hosts: all

become: yes

gather\_facts: no

tasks:

- name: "Creating user"

user:

name: "{{ item }}"

state: present

loop:

- testuser1

- testuser2

- testuser3

* **usecase2: create 3 users(testuser1,testuser2 & testuser3) in three different groups(dev,test,prod) using loops**

---

- hosts: all

become: yes

gather\_facts: no

tasks:

- name: "Creating user"

user:

name: {{ item.user }}

group: {{ item.group }}

state: present

loop:

- { name: 'testuser1', groups: 'dev' }

- { name: 'testuser2', groups: 'test' }

- { name: 'testuser3', groups: 'prod' }

* **usecase3: Install multiple packages using loop {git,tree,wget}**

=========================================================================================================

**When condition**

=========================================================================================================

* "When" statement is a conditional statement that runs the particular task if the condition is met.

* **usecase1:- Create a playbook install apache webserver setup in Linux{httpd} & Ubuntu{apache2} machines**

---

- hosts: all

become: yes

gather\_facts: no

tasks:

- name: "Installing httpd"

yum:

- name: httpd

state: present

- name: "Start httpd service"

service:

name: httpd

state: started

- name: "Configure httpd"

copy:

src: index.html

dest: /var/www/html/index.html

- name: "Installing apache2"

apt:

name: apache2

state: present

- name: "start apache2 service"

service:

name: apache2

state: started

- name: "configure apache2"

copy:

src: index.html

dest: /var/www/html/index.html

* Once I execute above playbook all the tasks will run in sequential mode & execution failed on all ubuntu & rhel servers.

To overcome this is issue I am going to use when condition particular task, so when condition met then only respective task will executed.

* Capture OS family details,

ansible -m setup -a "filter=\*os\*" all

gather\_facts set to be true.

---

- hosts: all

become: yes

gather\_facts: yes

tasks:

- name: "Installing httpd"

yum:

- name: httpd

state: present

when: ansible\_os\_family == "RedHat"

- name: "Start httpd service"

service:

name: httpd

state: started

when: ansible\_os\_family == "RedHat"

- name: "Configure httpd"

copy:

src: index.html

dest: /var/www/html/index.html

when: ansible\_os\_family == "RedHat"

- name: "Installing apache2"

apt:

name: apache2

state: present

when: ansible\_os\_family == "Ubuntu"

- name: "start apache2 service"

service:

name: apache2

state: started

when: ansible\_os\_family == "Ubuntu"

- name: "configure apache2"

copy:

src: index.html

dest: /var/www/html/index.html

when: ansible\_os\_family == "Ubuntu"

=========================================================================================================

**Grouping of Inventory**

=========================================================================================================

=========================================================================================================

**Handlers**

=========================================================================================================

* It's a feature of ansible, appears similar to the task.

* Handlers will works only when it's called from another task using notify keyword.

* When there is a need of running repetitive task after running particular task then we will use handlers.

* Assume we have Apache installed & configured, after the configuration normally we required to restart apache

* In this case we will use handlers.

* **Usecase4:- Create playbook Apache setup using handlers**

---

- hosts: all

become: yes

gather\_facts: no

tasks:

- name: "Installing httpd"

yum:

name: httpd

state: present

- name: "Configuring httpd"

copy:

src: index.html

dest: /var/www/html/index.html

notify: Restart apache

handlers:

- name: "Restart apache"

service:

name: httpd

state: restarted

Ansible-Day-5

13 December 2023

14:25

* **How to declare variable in ansible playbooks**

---

- hosts: all

become: yes

var:

pkg\_name: git

tasks:

- name: Install git

yum:

name: {{ pkg\_name }}

state: present

Execute the playbook the and show installation done successfully.

* **Override the git variable explicitly during ansible playbook execution**

ansible-playbook -i hosts -v -e "pkg\_name=tree" samp.yml

* **Define variables in /etc/hosts as host variables & group variables**

[webserver]

10.1.2.100 pkg\_name=wget

[dbserver]

10.2.1.200

10.2.2.201

[dbserver:vars]

pkg\_name=apache2

* **Write a playbook to install the package using host vars**

---

- hosts: webserver

become: yes

tasks:

- name: print the variable

debug:

msg: The variable pkg\_name is: "{{ pkg\_name }}"

o/p:-

The variable pkg\_name is: wget

* **Write a playbook to install the package using group vars**

---

- hosts: dbserver

become: yes

tasks:

- name: print the variable

debug:

msg: The variable pkg\_name is: "{{ pkg\_name }}"

o/p:-

The variable pkg\_name is: apache2

* **Priority levels of variable**

0 --> group\_vars

1 --> host\_vars

2 --> variable in playbook

3 --> variable passed through the command line

* **Example: Maintain variables in** 
  + **group\_vars(apache2)**
  + **host\_vars(wget)**
  + **playbook(git)**
  + **command line(tree) & test the priority levels**

* **Write a playbook & execute**

---

- hosts: all

become: yes

vars:

pkg\_name: git

tasks:

- name: print the variable

debug:

msg: The variable pkg\_name is: "{{ pkg\_name }}"

**exec1-**

ansible-playbook -i hosts -e "pkg\_name=tree" -v smp.yml

o/p:

The variable pkg\_name is: tree

**exec2:-**

ansible-playbook -i hosts -v smp.yml

o/p:

The variable pkg\_name is: git

**exec3:- Disable variable & define host vars for dbserver group also**

ansible-playbook -i hosts -v smp.yml

o/p:

The variable pkg\_name is: wget

**exec4:-**

ansible-playbook -i hosts -v smp.yml

o/p:

The variable pkg\_name is: apache2

* **New way of declaring the group\_vars & host\_vars**

cd /home/ansible

ls

hosts

group\_vars/webserver.yml

host\_vars/10.1.2.100.yml

* **Create host file with this contents**

The host file contains details like

[webserver]

10.1.1.100

10.1.1.101

[dbserver]

10.1.2.100

* **Define variables at group level**

webserver.yml

---

pkg\_name: git

* **Define variables at host level**

10.1.2.100.yml

---

pkg\_name: tree

* **Provide hosts details as web-server & check which package is getting executed**

---

- hosts: webserver

become: yes

tasks:

- name: print the pkg\_name

debug:

msg: value of pkg\_name is "{{ pkg\_name }}"

**Exec:** On webserver related nodes git package will get installed

* **Provide hosts details as ip-details & check which package is getting executed**

---

- hosts: 10.1.2.100

become: yes

tasks:

- name: print the pkg\_name

debug:

msg: value of pkg\_name is "{{ pkg\_name }}"

**Exec: On 10.1.2.100 node tree package will get installed**

* **Provide all hosts & check which package is getting executed**

---

- hosts: all

become: yes

tasks:

- name: print the pkg\_name

debug:

msg: value of pkg\_name is "{{ pkg\_name }}"

**Exec:** On webserver related nodes git package will get installed

On 10.1.2.100 node tree package will get installed

Ansible-Day-6

13 December 2023

14:58

===================================================================================================

**Ansible vault**

===================================================================================================

* This feature allows us to encrypt the sensitive information like passwords & secret key's rather than using it as a plain text in ansible playbooks or roles.

* **Create a file with secret key in plain text**

echo "API\_KEY: ThisIsSuperSecretDontDisplay" > vars/api\_key.yml

* **Create a sample playbook to display value of the variable API\_KEY**

---

- hosts: all

become: yes

vars\_file:

- vars/api\_key.yml

tasks:

- name: "Display the API\_KEY value"

debug:

msg: "{{ API\_KEY }}"

ansible-playbook -v test.yml

**o/p:** ThisIsSuperSecretDontDisplay

* **Let's encrypt the vars/api\_key.yml file**

* **Encrypt the file**

ansible-vault encrypt vars/api\_key.yml

cat vars/api\_key.yml

The file whatever present vars/api\_key.yml is encrypted now.

cat vars/api\_key.yml

* So we can push this file to GitHub along with our playbook/role.

* Execute ansible playbook

ansible-playbook -v test.yml --ask-vault-pass

* Execute ansible playbook with password stored in a file

ansible-playbook -v test.yml --vault-password-file

* Having a separate script file for getting the passwords is also possible.

you need to make sure the script file is executable and the password is printed to standard output for it to work without annoying errors.

ansible-playbook launch.yml --vault-password-file ~/ .vault\_pass.py

* **Decrypt the file**

ansible-vault decrypt vars/api\_key.yml

* View an Encrypted File in Ansible

ansible-vault view vars/api\_key.yml

* Edit an Encrypted File in Ansible

ansible-vault edit vars/api\_key.yml

* Change the password for the encrypted file

ansible-vault rekey vars/api\_key.yml

===================================================================================================

**What is ansible role?**

===================================================================================================

1. Normally when we write playbook for particular requirement we mention everything in a single file

like

- tasks

- variables

- handlers

- metadata

2. Ansible roles allow you to develop reusable automation components by grouping related automation things,

- configuration files

- templates

- tasks

- handlers

3. Roles will have structured layout.

4. Ansible roles consists of multiple folders & each folder by default contains main.yml file.

===================================================================================================

**How to create a role**

===================================================================================================

1. Create ansible role for apache-setup

ansible-galaxy init apache-setup

2. The role structure of the apache-setup will be like

apache-setup

├── defaults

│ └── main.yml

├── files

├── handlers

│ └── main.yml

├── meta

│ └── main.yml

├── README.md

├── tasks

│ └── main.yml

├── templates

├── tests

│ ├── inventory

│ └── test.yml

└── vars

└── main.yml

3. Now let's understand every folder purpose,

a. defaults/main.yml ==> In this configuration file we declare default values for the variables.

Example: httpd is the default package name for installing apache on Linux machine,

so we will declare it as variable here.

b. files ==> We copy the static files like .tar.gz/.zip files into this directory.

During ansible role execution files that are present over here will be transfer to target machine.

c. handlers/main.yml ==> In this section we will declare handlers like restart apache.

d. meta/main.yml ==> It's metadata file, which contains the information like

- author

- license

- dependency management

From this we can another dependency role for current apache-setup role.

Example: Let's assume we have two roles for maven-setup & java-setup

When we create maven-setup role, we can make java-setup role as dependecy in meta/main.yml

e. README.md ==> In this file we write Description about the role & its usage.

f. tasks ==> In this folder we will create the list of the .yml files related to our activities

Example:

Installing apache {apache.yml}

Configure apache {configure.yml}

g. templates ==> This folder contains the template files(jinja2) used by the role to create the actual configuration files.

Example: After installation there might be need to update the configuration files as well,

like including usernames/os-details in-order to do this we will use templates.

h. vars ==> Used to declare the variables that are needed for ansible playbooks

Here we can see variables we can declare at defaults/main.yml & vars/main.yml.

The variables which are declared in vars/main.yml have high priority.

4. Create a ansible for apache-setup & push it GitHub

Docker-Day-1

14 December 2023

10:12

* **What are VM's?**

The EC2 instances that we created on AWS management console are considered as VM's

* **How these VM's are created?**

* We just go to the AWS management console & click launch instance, correct right? - Yes

* But in background we will discuss what will happen,

Machine generated alternative text:
Appl 
•ava based a 
Libs/bin 
ava 
Guest OS 
(RHEL) 
App2 
java based web ap 
Libs/bin 
tomcat 
GUEST OS 
Ubuntu 
Hypervisor 
OPERATING SYSTEM 
App3 
.Net based app 
I S In 
Guest OS 
Windows 
Physical server 

* Assume there is **physical server** with Linux OS on datacentre with good amount of
  + CPU
  + RAM
  + HD
  + Network utilities

* Whatever machine that physically present it can be considered as physical server.
  + I mean laptop/any computers

* On top physical server Hypervisor software is installed.

* **What Hypervisor software will do?** This helps to create virtual machines.

* On these virtual we can install OS like
  + RHEL
  + Ubuntu
  + Windows

* On top of these OS we will install s/w's (or) binaries (or) libraries, assume on
  + VM1 we installed Tomcat
  + VM2 we installed Java
  + VM3 we installed IIS

* Post that we deploy applications on these VM's
  + VM1 ==> Tomcat ==> War
  + VM2 ==> Java ==> .jar
  + VM3 ==> IIS ==> .Net application

* **What is the problem that may happened with VM's?**

* **In real time, how will bring any application to the live environment?**

First, we will deploy application into DEV environment & next we will deploy the application into QA environment & finally we will deploy application into production environment, Correct or not? - Correct

Machine generated alternative text:
online-bookstore.war 
Java app 
Tomcat 
RHEL OS 
DEV 
java app 
Tomcat 
RHEL OS 
java app 
Tomcat 
RHEL OS 
PROD 

* **Assume you have a Java web application called Online Bookstore and you want to take it live.**

* Once the developers completed the coding part, as a DevOps engineer will generate a binary file called **online-bookstore.war** file.

This file we will deploy into DEV environment VM first.

* **To deploy java application on DEV environment what is required?**

We will create a VM with Linux OS on top of that tomcat software & finally we will deploy the Java application.

So, post deployment the developers will do the testing & confirmed the application is working fine in DEV environment & asked to promote into QA environment.

* As a DevOps engineer again, we will create a VM to the QA environment with Linux OS, tomcat software & finally deploy application into QA environment.

* Here again, the QA engineer will do the testing.

During QA testing identified a lot many issues, so simply QA engineer will create a bug in the JIRA & will assign to developers.

* Now the developers will replies, there is no issues in the code at their side & the application also working fine in the dev environment. This might be configuration issues at QA VM itself.

* Now who did this configuration of VM like software installation, and deployment? DEVOPS ENGINEER

* So devops engineer will check all software's are installed or not in QA VM & since all software's installed simply revered the mail/ticket to DEVELOPERS there are no configuration issues at QA VM side.

* Like this conversation happened between developers & ops person for couple of days.

* One fine day the was issue identified, the tomcat version is different in both servers.

Due to this environment mismatching some of the future not working in the QA environment which are working in the development.

* Finally time got wasted this kind of environmental issues.

* To avoid this kind of situations containers came into picture.

* **What containers will do?**
  + Containers will have the software's that are needed to run your application & Artifact of the application code.

* Now the probability for getting errors related to the environment changes between different environments very less, since we packed the .war file & tomcat software. Simply we can container on any environment & can access application.

* Now let's see how to create containers in background.

Machine generated alternative text:
Conl 
Con2 
DOCKER 
RHEL OS 
PHYSICAL SERVER / VM 
Con3 

* Assume this is a physical server or VM with good amount of RAM, CPU and hard disk on top of that
  + RHEL was installed & again on top of that docker is installed. So, this **docker software what it will do, it will help you create the containers.**

* Let's create containers
  + container-1
  + container-2
  + container-3

* Assume on these containers we install libs/bin's & finally we deploy our application.

* On container-1 we can install tomcat binary & deploy java web application .war file.

* On container-2 we can install java binary & deploy java application .jar file

* On container-3 we can install IIS & deploy .Net application .jar file

* These containers are independent each other & we can create n-no of containers on single VM.

* **You may have doubt how containers will work with-out OS?**

Container will communicate directly with the OS of physically as read only.

* For your easy understanding following are containers in real-time,
  + Shipping container

Machine generated alternative text:


* Lunch box

Machine generated alternative text:


* Makeup box

Machine generated alternative text:


* Here containers like the boxes for us & the **items that present in boxes**/containers are considered as **applications & dependencies needed** for project.

Machine generated alternative text:
Application + Dependencies 
war 
(Java+Tomcat) 

* **Now let's see how to containers practically.**

Machine generated alternative text:
docker image 
dockerhub 
docker image 
Container 

* To create any docker container we require **docker image**.

* **What is docker image?**
  + It's a read only template which contains the list of instructions to create the container.
    - Suppose you want to create Ubuntu OS container you can find Ubuntu image &
    - Same way if you want to create tomcat container we need tomcat docker image.

* **Where do this docker images stored?**
  + That docker images will stored under the registry called Docker Hub.
  + In the Docker hub we can find docker images for any software.
    - Let's search for **Ubuntu** docker image here we can find n-no.of docker images created by different people.
    - Same way if you search **jenkins** you can find jenkins docker image details
    - Out of these docker images we will only the **Official** Image which is maintained docker community.
    - You may have doubt what are the rest of the docker images?
      * Those are docker image created by the **people like us & pushed into Docker hub**. We see this part on next sessions.

* How to list the active containers that running in VM

**docker ps**

Since we don’t have any active docker container we don't see any output.

* Now the run the **docker ps -a**

This command will help to list the inactive & active containers, present we don't have any active/inactive container. So the output is nothing.

* Let's **Create Ubuntu container on docker**, for that first we need to
  + **Download Ubuntu docker image from docker hub** to the server, for that we will go Docker hub & look for docker image Ubuntu

Next we will run

**docker pull Ubuntu**

If I don't mention anything after Ubuntu by default latest docker image will be picked & if we mention specific tag particular docker image will be downloaded.

* Based on this output we can confirm docker image is downloaded but to confirm what docker images available or not on the server run command

**docker images**

* Ubuntu image is ready on server & we can create Ubuntu container based on this by running command

**docker run ubuntu**

* Now we can check container is running or not with the command

docker ps

* **To connect actively running container**

docker exec -it <cid> /bin/bash

* Only we can login to the OS related containers.

* Now we are inside the container & here we can see the
  + login user on VM was ec-user but now we are under root
  + file system structures are different.

* To come-out this container we can enter a command exit, now we are on the server.

* **If you run docker ps command we don't see any docker containers, what happened?**

**After click on exit in docker container, it will be automatically stopped**. You can find stopped container using

docker ps -a

* **If you want to connect container & come outside of it without stopping container we will see now.**

* Create new docker container from Ubuntu image

docker run Ubuntu

* Connect to docker container

docker exec -it <cid> /bin/bash

* To exit from container type

Ctrl + p +q

* We can stop actively running container in docker using

docker stop <cid>

* We can also start container by running

docker start <cid>

* To remove the container from VM

docker rm <cid>

* If you want to delete actively run command

docker rm -f <cid>

* Same way if you want to remove the docker images present on server

docker rmi <image\_name>

* If you want to remove all the docker images

docker rmi -f (docker images -q)

Docker-Day-2

22 December 2023

19:34

==========================================================================================

* Docker volumes

==========================================================================================

* Normally the data inside the docker container will be deleted automatically if container stopped/killed.

* In order to make the data that was present inside the container even after the container stopped/killed we use docker volumes.

* Now let's practically see whether data loss will happen or not after container destroy

Create a container & login

docker run -it Ubuntu /bin/bash

echo "Testing the data storage availability" > /storage. out

cat /storage.out

* Suppose if we kill/exited container the /storage.out file also deleted, but in real-time there will be a situations we need to have file/data present inside the container.

exit the container & we will not able to see the /storage.out file.

* So docker volumes will help here

* There are different ways of using docker volumes
  + **Anonymous volumes**
  + **Named volumes**
  + **host based volumes**

==========================================================================================

**Anonymous volumes**

==========================================================================================

* **Create a container a with mount point /data\_01 & the data/files present under /data\_01 will be accessible on docker host if container destroyed.**

docker run -it --name myubutnu -v /data\_01 ubuntu:latest

* To know **more details about docker containers** we use inspect command,

**docker inspect mybuntu**

* The output of the docker inspect command is in json format.

* Here in **mount section we find the source tag which present on docker host machine** & pointing to the **/data\_01** of docker container.

* So Inside the docker container under /data\_01 if we create any file that will show in docker host on path /var/lib/dcpler/volumes/8491375021/\_data

* And same if we create any files/directory inside docker host machine will be showed in container as well.

* Now kill the container & check whether the data of container is present on docker host or not.

* With this method we will get confused to which volume the container was used if we have many containers.

==========================================================================================

**Named volumes**

==========================================================================================

* **During named volumes we provide name for the docker volume on host machine**

docker run -it --name myubutnu -v volume\_test:/data\_01 ubuntu:latest

* To know more details about docker containers we use inspect command,

docker inspect mybuntu

* The output of the docker inspect command is in json format.

Here in mount section we find the source tag which present on docker host machine & pointing to the /data\_01 of docker host.

We can observe volume name generated as volume\_test

* So Inside the docker container under /data\_01 if we create any file that will show in docker host /var/lib/dcpler/volumes/8491375021/\_data

* And same if we create any files/directory inside docker host machine will be showed in container as well.

* Now kill the container & check whether the data of container is present on docker host or not.

* With this method we will get confused to which volume the container was used if we there many containers

==========================================================================================

**host based volumes**

==========================================================================================

* In this method we will mount specific directory on docker host machine to docker container.

* Create new directory on host machine

mkdir /root/volume-test

* Mount the new directory(/root/volume-test) as mountpoint(/data\_01) inside the container

docker run -it --name volume\_test3 -v /root/volume-test:/data\_01 ubuntu:latest

* Create a file inside the docker container on /data\_01

mkdir IamInsideContainer

* On new terminal check the directory "IamInsideContainer" present or not under "/root/volume-test"

cd /root/volume-test

ls

* Create new directory on host under /root/volume-test

cd /root/volume-test

mkdir IamOutSideContainer

* Verify inside docker container IamOutSideContainer folder present or not under /data\_01

cd /data\_01

ls

* Stop & kill the container

docker stop <container\_id>

docker rm <container\_id>

* Verify the files which are present under /data\_01 directory of container are not lost.

cd /root/volume-test

ls

========================================================================================================

Port Mapping

========================================================================================================

* Normally every application runs on a specific port.

Like

* tomcat will run on port 8080
* sonarqube will run on port 9000

* Similarly when we spin container for any application it will run particular port,

We can't directly connect to the docker container via port, so we will bind docker container exposed port with host machine port.

**docker run -it -p 8888:8080 tomcat**

Docker-Day-3

22 December 2023

19:51

=====================================================================================

**dockerfile**

=====================================================================================

* It is a normal script file contains the set of instructions which are useful to build a Docker image.

* dockerfile ==> docker image ==> container

* Machine generated alternative text:
  docker image 
  dockerhub 
  docker image 
  Container 

* Now let's instructions that we use inside the dockerfile

=====

**FROM**

=====

* In every dockerfile the first line is **FROM** instruction.
* The **FROM** used to pull the base image & that we are looking to customize it.
* Search of dockerfile in internet we can use every file started with from instruction.

Ex:-

FROM ubuntu:latest

======

**LABEL**

======

* LABEL instruction used to stores meta data information about docker Image, like who is the author of the Docker Image.

* As best practice purpose only we use LABEL instruction.

Ex:-

LABEL owner chaitanya

LABEL team XYZ

=====

**RUN**

=====

* RUN instruction used execute shell commands, I mean any Linux commands

Ex:-

RUN apt-get install wget -y

RUN apt-get install tree -y

RUN apt-get install vim -y

**usecase-1:**

* Create a custom docker image with git presence

FROM ubuntu:latest

LABEL owner chaitanya

RUN apt-get install git -y

* To Create docker image based out of the docker file we use command

docker build -t custom\_git:latest -f dockerfile

docker images

docker run -it custom\_git:latest /bin/bash

git --version

**usecase-2:**

* Use single RUN instruction for instead of multiple run instructions

**Way-1**

FROM ubuntu:latest

LABEL owner chaitanya

RUN apt-get install git -y

RUN apt-get install wget -y

RUN apt-get install tree -y

RUN apt-get install vim -y

**Way-2**

FROM ubuntu:latest

LABEL owner chaitanya

RUN apt-get install git -y && \

apt-get install wget -y && \

apt-get install tree -y && \

apt-get install vim -y

=====

**CMD**

=====

* CMD instruction is used to provide default command & parameters to docker container.

* We can easily override the default commands that are mentioned in CMD while spinning up container.

* If we have multiple CMD instructions in dockerfile the last CMD instruction only will be applied.

**Usecase3: Create docker image with default echo command for docker container using CMD**

* **Create docker file**

FROM ubuntu

LABEL Chaitanya

RUN apt-get update

CMD ["echo", "Hello World"]

* The each string in shell command will be double quoted, then only CMD will consider it as command.

* **Create docker image**

docker build -t basic\_cmd:1.0 -f dockerfile

docker images

docker run -d basic\_cmd:1.0

* The container will gets spin-up & execute the default echo command inside container & container will get stopped automatically since there is no long running command inside it.

**Usecase4: Create docker image such way that it should be up & running as long you stop it explicitly using CMD**

* **Create docker file**

FROM ubuntu

LABEL owner chaitanya

RUN apt-get update -y

RUN apt-get install iputils-ping -y

CMD ["ping", "google.com"]

* **Create docker image**

docker build -t basic\_cmd:2.0 -f dockerfile

docker images

docker run basic\_cmd:2.0

**Usecase5:- Override the shell commands in CMD instruction**

* docker run basic\_cmd:2.0 hostname

* The hostname command overrides the default ping command & prints the hostname of the container.

============

**ENTRYPOINT**

============

* Similar to CMD instruction, ENTRYPOINT instruction also is used to provide default command & parameters to docker container.

* **What is the difference between CMD and ENTRYPOINT?** 
  + You cannot override the ENTRYPOINT instruction by adding command-line parameters to the docker run command.

* By opting for this instruction, you imply that the container is specifically built for such use.

**Usecase6:- Create docker image with default echo command for docker container using ENTRYPOINT**

* **Create docker file**

FROM ubuntu

MAINTAINER sofija

RUN apt-get update

ENTRYPOINT ["echo", "Hello World"]

* **Create docker image**

docker build -t basic\_entrypoint:1.0 -f dockerfile /root

docker images

docker run basic\_entrypoint:1.0

* Prints the Hello World & container will be stopped automatically

**Usecase7:- Create docker image such way that it should be up & running as long you stop it explicitly using CMD**

* **Create docker file**

FROM ubuntu

LABEL owner chaitanya

RUN apt-get update -y

RUN apt-get install iputils-ping -y

ENTRYPOINT ["ping", "google.com"]

* **Create docker image**

docker build -t basic\_entrypoint:2.0 -f dockerfile /root

docker images

docker run basic\_entrypoint:2.0

**Usecase8:- Try to override the shell commands in ENTRYPOINT instruction**

* docker run basic\_entrypoint:2.0 hostname

* The hostname command just append as string to the default ping command & prints the hostname

**Usecase9:- Using CMD + ENTRYPOINT**

* **Create docker file**

FROM ubuntu

LABEL Chaitanya

RUN apt-get update

ENTRYPOINT ["echo", "Hello"]

CMD ["World"]

==========

**WORKDIR**

==========

* WORKDIR instruction used to define the working directory of a Docker container at any given time.

**Ex:- Create dockerfile & image based based out of it without workdir instruction**

FROM ubuntu:latest

RUN apt-get update -y

RUN apt-get install git -y

RUN touch 1.out 2.out 3.out

* Create docker image

docker build -t custom\_img:latest .

docker run -it custom\_img:latest /bin/bash

* Here the we can observe that the default working directory "/" & files are created under "/"

**Usecase10:- Create a custom WORKDIR**

* **Create docker file**

FROM ubuntu:latest

WORKDIR /project

RUN apt-get update -y

RUN apt-get install git -y

RUN touch 1.out 2.out 3.out

* Here the we can observe that the default working directory "/project" & files are created under "/project"

=====

**ADD**

=====

* The ADD command is used to copy files/directories into a Docker image. It can copy data in three ways:

* Copy files from the local storage to a destination in the Docker image.
* Copy a **tar ball from the local storage and extract** it automatically inside a destination in the Docker image.
* Copy files from a URL to a destination inside the Docker image.

* Ex:- **Create a docker image to copy files from VM to docker container**

FROM ubuntu:latest

WORKDIR /project

RUN mkdir test

ADD codes /project/test

ADD abc.out /project/test

ADD samp.tar.gz /project/test

=====

**COPY**

=====

* The COPY command is used to copy files/directories into a Docker image. It can copy data in three ways:

* Ex:-

FROM ubuntu:latest

WORKDIR /project

RUN mkdir test

COPY codes /project/test

COPY abc.out /project/test

==========

**EXPOSE**

==========

* EXPOSE instructions informs the docker that the container listens on specific port at runtime.

* Jenkins container will listens on port 8080

* Ex:-

EXPOSE 8080

=========================================================

**Usercase10: Build tomcat docker image with deployable package**

=========================================================

* Take Ubuntu as base image
* Install tomcat inside docker Image
* Copy the sample war Artifact
* Expose port 8080, since tomcat listens on port 8080
* Run tomcat as service

**Way-1**

FROM Ubuntu

LABEL Chaitanya

RUN mkdir /opt/tomcat/

WORKDIR /opt/tomcat

RUN curl -O <https://www-eu.apache.org/dist/tomcat/tomcat-8/v8.5.40/bin/apache-tomcat-8.5.40.tar.gz>

RUN tar xvfz apache\*.tar.gz

RUN mv apache-tomcat-8.5.40/\* /opt/tomcat/.

RUN yum -y install java

RUN java -version

WORKDIR /opt/tomcat/webapps

RUN curl -O -L <https://github.com/AKSarav/SampleWebApp/raw/master/dist/SampleWebApp.war>

EXPOSE 8080

CMD ["/opt/tomcat/bin/catalina.sh", "run"]

**Way-2: Actually you don't need to install tomcat on Ubuntu we have docker image for tomcat itself, for your understanding on instructions I craeted dockerfile based on Ubuntu image**

FROM tomcat:8.0-alpine

LABEL maintainer=”abc@gmail.com”

ADD sample.war /usr/local/tomcat/webapps/

EXPOSE 8080

CMD [“catalina.sh”, “run”]

======

**ARGS**

======

* ARG instruction defines the variables during docker image creation.

* **Usecase: Define variable & call it inside the dockfile using ARG instruction**

* **Create docker file**

FROM ubuntu:latset

LABEL chaitanya

WORKDIR /

ARG user1 chai

RUN echo $user1 > user\_name.out

* **Create docker image**

docker build -t myarg:1.0 .

docker run -it myarg:1.0 /bin/bash

cat /user\_name.out

* Override the user1 value

docker build -t myarg:2.0 --build-arg user1=tom .

======

**ENV**

======

* ENV instruction is used to set the Environment variables(system-level/user-level in windows) for the future containers which are created from custom docker image.

FROM ubuntu:latest

WORKDIR /project

ENV http\_proxy <http://10.0.0.0:8000>

RUN echo $http\_proxy

======

**VOL(use case step required)**

======

* Create a volume inside a container & persist data even container killed/terminated

FROM ubuntu:latest

RUN mkdir /data

WORKDIR /data

RUN echo "Hello from Volume" > test

VOLUME /data

**Home work:**

1. Dockerize jar & spin up container & access application
2. Automate the ci/cd process
   1. Maven build
   2. Sonar-scanning
   3. Dockerize war file
   4. Push docker image to docker registry
   5. Deploy image &Spin as container
   6. Access the application

DEVOPS Syllabus

04 March 2024

21:38

============================================================================================================================================

**DEVOPS**

============================================================================================================================================

Introduction about DEVOPS

Linux:

• Linux Introduction & architecture

• Setup AWS Account & Linux server provision

• Basic Linux commands - file/dir creation, cp, mv..etc

• User managment & softlink/Hardlink

• Permissions management & Package management

Git:

• Git and github repository creation

• git commands - all 21 commands with practical

• working with linux and windows environment

• working with java code on git and github.

Maven:

• Build the code using maven tool

• maven dependencies

• maven repository

• maven lifecycle/phases

• java artifacts.

• pom.xml configuration

Sonarqube:

• Sonarqube server setup & Purpose of sonarqube setup

• Publish sonarqube reports to sonarqube server

• Setup quality gates

• Quality profile setup

Jenkins

• Continous integration

• Jnekins overview

• Jenkins installation

• dependencies of jenkins

• Job configuration

o Free style job

o Pipeline job

• scripted pipeline

• declarative pipeline

• Plugin managment

• Global tool configuration and integration

• Security in jenkins

o global securty

o user management

o credential management

o role based autherisation

• master slave configuration

• github webhook with jenkins.

• maven and other tool configuration

• Jfrog integration

JFROG:

• Artifactory management

• Jfrog overview

• Jfrog installation

• types of repository in Jfrog

• creating maven repository

• integrating with jenkins and push the artifact to Jfrog.

Ansible

• Continous delivery/deployment

• Configuration management

• push based/pull based, agent based/agentlesss

• Ansible overview

• IT orchestration

• Ansible installation

• Playbook

o YAML basics

o Modules

o Roles

o Galaxy

o tasks

o variables

o Adhoc cmd execution

• Tower

• Inventories

o Static inventory

o Dynamic inventory

• Playbook to install any package in target

• playbook to CD operation

Docker:

• Containerisation

• difference between container and vm

• Docker components

• Docker deamon/engine/clinet

• Docker container

• docker images

• dockerfile

• dockerhub

• docker port

• docker volume

• docker instructions

• write dockerfile to create an apache image.

• dockerising an applicaiton by writing the dockerfile

• managing the images in dockerhub-

• Deploying the applicaiton in web - apache server

• deploymeing spring boot applciation in tomcat server.

Kubernetes:

• overview and highlevel session.

• What is pod

deploy

service

ingress

ConfigMap

Project1: CI & CD pipeline setup using jenkinsfile into Kubernetes by handling the different environments

Project2: Handling the CI/CD pipeline with different branching strategy into dev/qa/prod environments

Linux-Day-7

18 March 2024

22:35

1. Shell script is used to automate the repeated tasks that we do daily basis on servers.
   1. Checking the file system size on server
   2. Install of software & configuration.

1. Shell script files are usually with extension .sh

Example: sample.sh / file\_system\_size\_check.sh

1. The shell script file contains the Linux commands in sequence format in order to achieve particular requirement like checking file system size.

1. **Usecase1:** Now I will write sample shell script to print date & time using shell script

vi get\_today\_date.sh

#!/bin/bash

echo "Today date & time is:" $date

Ignore the first line whatever I mentioned over here & will discuss about it after sometime .

Now let see how to execute this script file

./get\_today\_date.sh

1. **Useacase2:** Next we will discuss about how to declare variable & call it

vi variables\_usage.sh

#!/bin/bash

age = 10

name = "chaitu"

Echo "The value of a is: " $a

Echo "The value of name is: " $name

1. Usecase3: In above script file we are hardcoded the values of a & name variables, normally we shouldn't hardcode the values & instead of that we pass the values to variables during script execution.

#!/bin/bash

name=$1

age=$2

Echo $name age is: $age

./print\_person\_age.sh chaitu 28

The values that I given during

1. Command line arguments

Take example from tutorials

1. Example: Chck

Maven-Day-2

29 March 2024

12:44

1. Explain Maven life-cycle

1. Explain with scenario how plugins downloaded from the Jfrog remote repositories

1. Install Jenkins

1. Install Git & Maven on server

1. Integrate Git with Jenkinsfile

1. Integrate Maven with jenkins

1. Create maven project

1. Create pipeline project

1. Execute the pipeline