**Root User:** srinivasreddyjuly@gmail.com

**Root User PWD**: 3Vc15cv@108

**DevOps**

\* Devops is process it increases the ability of delivering your applications at higher velocity (or)

\* Devops is a process of improving the application/package delivery by ensuring that there is automation in place; code quality, continuous monitoring & testing

\* SDLC is process that is followed by the industry to design, develop and test high quality product.

\* It consists of cycle of 6phases which includes planning, defining, designing, Building, testing and deploying an application

\* In planning we plan and take feedback/Requirements from customers or clients

\* In defining they collect the data and store it in category or priority wise

\* In Designing part they design in two levels like HLD (high level design) and LLD (low level design)

\* In Building they first develop the requirements in code form and push it to Git where it acts as center line service provider or source code repository

\* In Testing they build an server where Quality Assurance engineer will Test it in this server and assures quality or gives feedback if any issue

\* In Deployment they deploy the code into a certain tool for the application to be market ready or for clients

\* Hypervisor, also known as a virtual machine monitor, is a process that creates and runs virtual machines (VMs) or it is software that can install VMs on our physical server.

it generally breaks & separates physical servers into logically VMs which has its own RAM,CPU & HDD

Hypervisor like VMware, xen... etc.

\* AWS instance VMs are called “EC2"

**The 7 Phases Of SDLC (Software Development Life Cycle)**

Stage 1: Project Planning

The first stage of SDLC is all about “What do we want?” Project planning is a vital role in the software delivery lifecycle since this is the part where the team estimates the cost and defines the requirements of the new software.

Stage 2: Gathering Requirements & Analysis

The second step of SDLC is gathering maximum information from the client requirements for the product. Discuss each detail and specification of the product with the customer. The development team will then analyze the requirements keeping the design and code of the software in mind. Further, investigating the validity and possibility of incorporating these requirements into the software system. The main goal of this stage is that everyone understands even the minute detail of the requirement. Hardware, operating systems, programming, and security are to name the few requirements.

Stage 3: Design

The design phase (3rd step of SDLC), the program developer scrutinizes whether the prepared software suffices all the requirements of the end-user. Additionally, if the project is feasible for the customer technologically, practically, and financially. Once the developer decides on the best design approach, he then selects the program languages like Oracle, Java, etc., that will suit the software.

Once the design specification is prepared, all the stakeholders will review this plan and provide their feedback and suggestions. It is absolutely mandatory to collect and incorporate stakeholder’s input in the document, as a small mistake can lead to cost overrun.

Stage 4: Coding or Implementation

Time to code! It means translating the design to a computer-legible language. In this fourth stage of SDLC, the tasks are divided into modules or units and assigned to various developers. The developers will then start building the entire system by writing code using the programming languages they chose. This stage is considered to be one of the longest in SDLC. The developers need certain predefined coding guidelines, and programming tools like interpreters, compilers, debugger to implement the code.

The developers can show the work done to the business analysts in case if any modifications or enhancements required.

Stage 5: Testing

Once the developers build the software, then it is deployed in the testing environment. Then the testing team tests the functionality of the entire system. In this fifth phase of SDLC, the testing is done to ensure that the entire application works according to the customer requirements.

After testing, the QA and testing team might find some bugs or defects and communicate the same with the developers. The development team then fixes the bugs and send it to QA for a re-test. This process goes on until the software is stable, bug-free and working according to the business requirements of that system.

Stage 6: Deployment

The sixth phase of SDLC: Once the testing is done, and the product is ready for deployment, it is released for customers to use. The size of the project determines the complexity of the deployment. The users are then provided with the training or documentation that will help them to operate the software. Again, a small round of testing is performed on production to ensure environmental issues or any impact of the new release.

Stage 7: Maintenance

The actual problem starts when the customer actually starts using the developed system and those needs to be solved from time to time. Maintenance is the seventh phase of SDLC where the developed product is taken care of. According to the changing user end environment or technology, the software is updated timely.

\* A physical server is a machine with a hard drive, CPU, processor, memory, and network connection to run programs and applications.

\* A virtual server is a software-based server that allows multiple virtual machines (VMs) to run simultaneously as computing resources are virtualized.

\* If AWS EC2 instance receives a request or ticket to API server in expected format & in response it sends EC2 instance (VMs details) and API expects requests of valid, authenticated and authorized

\* Which Type of scripts available?

@ Automate the creation of ur VM or EC2 instance through custom scripts like

AWS CLI (command line interface)

AWS Cloud formation template (CFT)

AWS API (Application Programming Interface) (boot 3 service)

AWS Cloud deployment Kit (CDK)

Terraform

**Shell Scripting Points :**

**https://linuxsimply.com/100-shell-script-examples/**

\* In general touch command is used in all Linux machines to create a file but vim is also used for file creation but it opens new slide to create & write files, when it comes to touch(only for file creation) it write in existing terminal which is very imp in automation process for time saving

\* When u open vim(file creation & write) command it asks whether u want to write r read r copy by showing INSERT at below of file opened then u need to select ESC+I for writing and to know ur in I mode it displays INSERT at bottom of that file , then start writing with shebang & next matter u want to write

and to save that written file press ESC+:wq! And press Enter

\* To use vim command and is used basically to write inside a file

\* We start Shell script with Shebang(#!/)/bin/bash or /bin/dash or /bin/sh or /bin/ksh where these 4 are executable of shell scripting in Linux like ex #!/bin/bash/ or #!/bin/dash or #!/bin/sh or #!/bin/ksh

\* What is diff btw /bin/sh/ and /bin/bash??

@ In some projects/scripts it will change from sh to bash because of linking concept but in recent many OSs are using they are using /bin/dash & linking from sh to dash and if ur writing start with bash but linking should not be from sh to bash

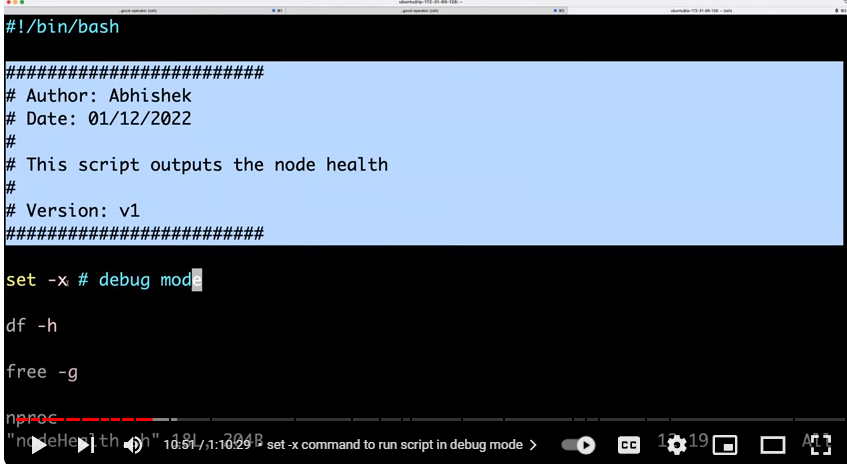
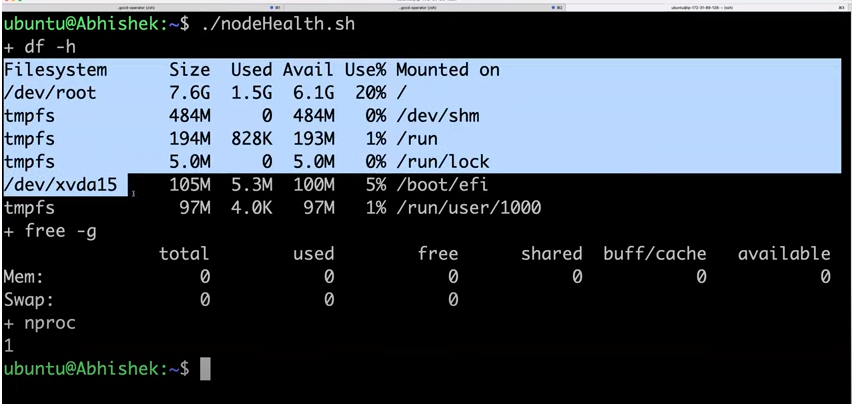
\* To print a name or texts in general we use print for java, python but in shell script we use ECHO like echo "some text"

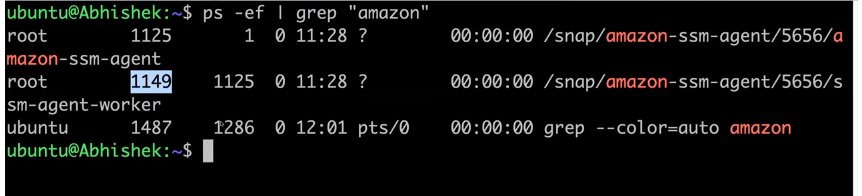
\***./ filename** for executing a file (or) sh filename for executing

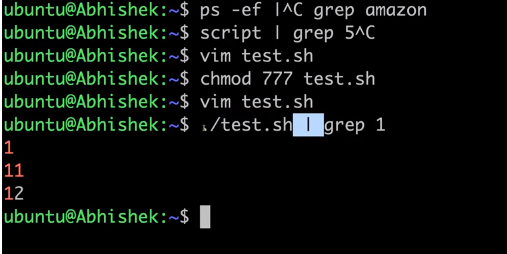
\* If u get permission denied for **./**filename while executing bec of permission like who needs to access that file for that use CHMOD command

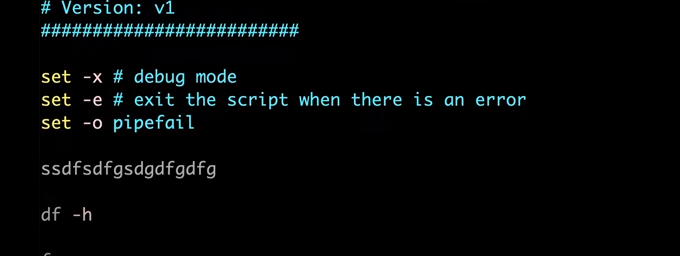
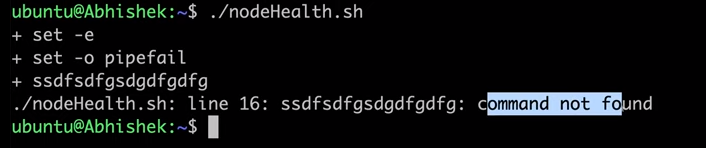
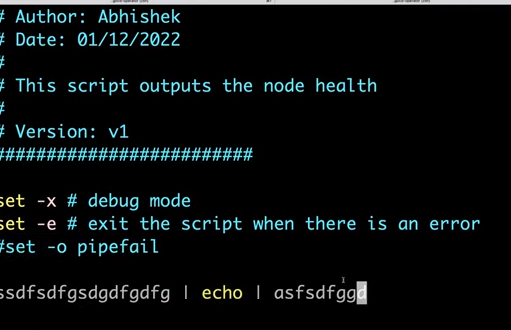
\* In shell script we give metadata to know exactly for what it is meant for and Metadata makes finding and working with data easier – allowing the user to sort or locate specific documents.

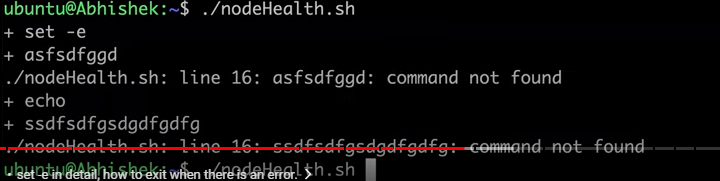
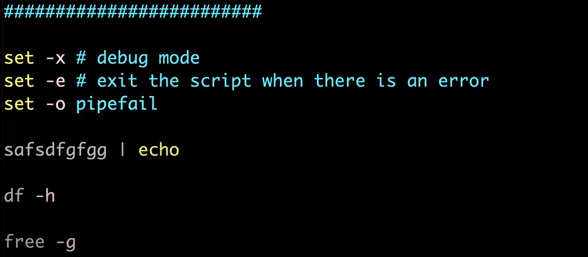
Some examples of basic metadata are author, date created, date modified, and file size can be seen in below pic which is selected that is called Metadata.

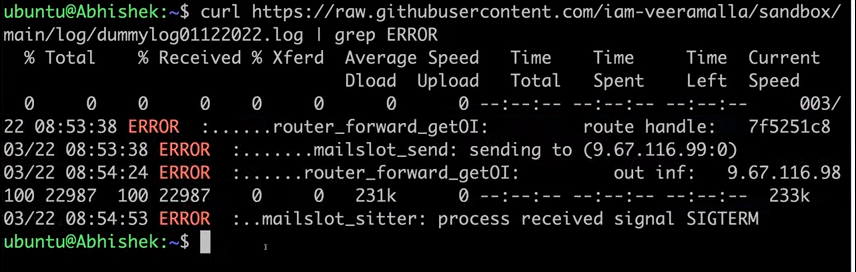
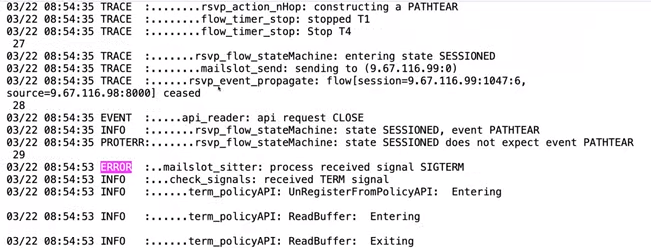
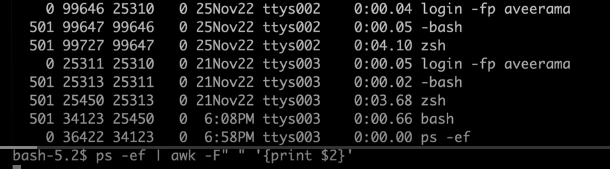
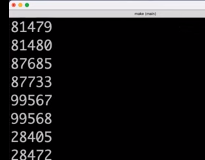
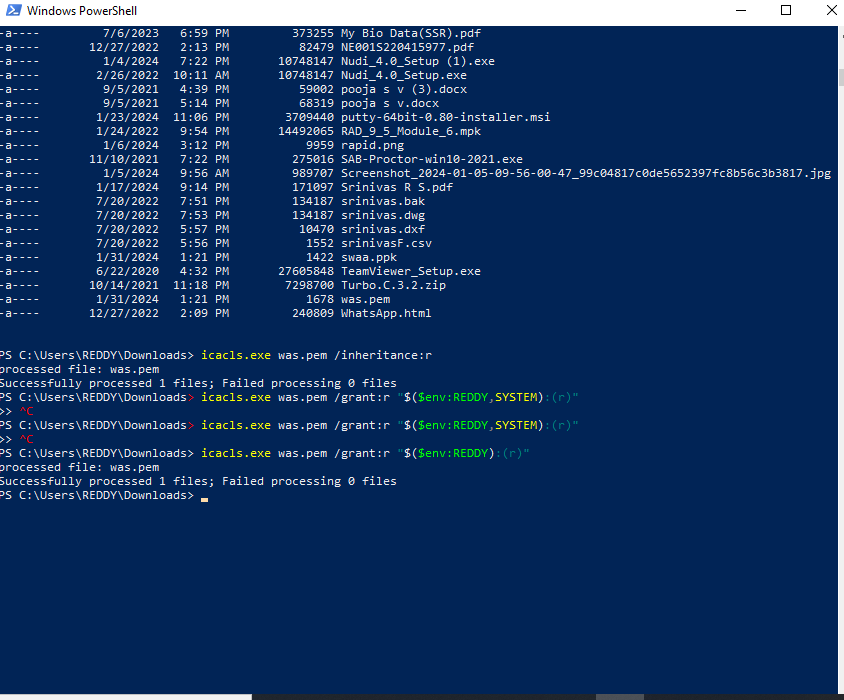
\*Shell script with metadata which prints disk space, total memory and no of CPU running \* **set –x** (sets/prints script in debug mod) before starting with shell script gives command which ur executing and prints the output (see below pic of output) ex: df –h ,free -g in pic below  


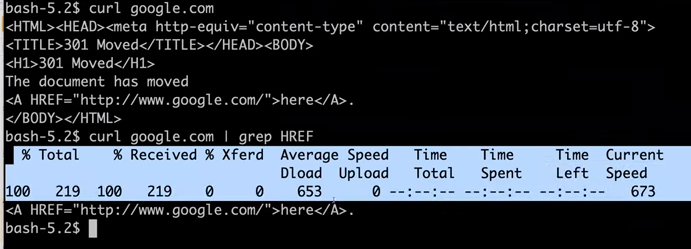
\* If u or any don’t want to see commands along with output then use #before **set-x** to show only output. And whenever pipe command is used use **set-e** & **set-o**  
\* **ps –ef** is command where **ps** is processes & **-ef** provides entire details of the processes in a full format  
\* To know only a particular process running in the machine use **ps –ef | grep** “processes name” which u want to know, here **grep** is used to find/pick the processes in total running at a time on machine in total on screen like amazon(see below pic) **|** ispipe command.  


\* Pipe command(**|**) which sends output of first command to send command like **,/test** is first command for execute **.sh** file and in btw pipe command passes second command **grep** 1 to print which all r similar to 1 in **.sh** file ( | it sends output of first command to second command)  


\* What is output of date | echo “today is “command?  
@ it gives today is because here date command is default and pipe command doesn’t send first command to second command as here pipe command sends output to STDIN but pipe will not be receiving response/information from STDIN to display along with second command and pipe can only print both commands when it receives info from STDIN  
\* Every VMs has different channels like STDIN,STDOUT & STDERR  
\* If your using pipe command(**|**) in script should first write **set –e** & **set –o   
\* set –e** is used to exit the script when there is an error and finds error at entry level and stops that script to execute further so use this at most importantly (one drawback is it will not show pipefail failure like it catches error of only last command in pic3 )  
  
  


 \* **set –o** is used to stop pipefail and by default should be used when set-e is used to stop command reading error   
  

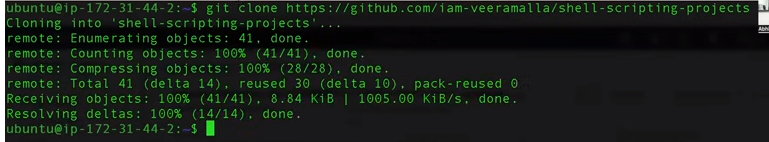
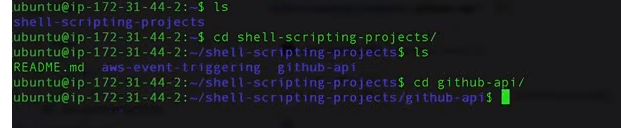

650.121944 hours remaining 598.121944 hours remaining 26.84 GB remaining24.54 GB remaining  
\* To find & get the errors in logfiles we use CURL command as **curl** **file location link | grep error** to get only errors ( a log file from github with error highlighting in 2nd pic)  
  
  
\* What is difference btw **wget** & **curl** command?  
@wget generally download the given file or link & stores the data in logfile/localdisk, whereas curl gives the output to u to get the data and store in our database   
if u don’t want to store the output and save the file in local disk then use CURL, if u want to get or download the file with saving that into ur disk then use WGET  
\* awk –F “ “ ‘{print $1}’, here in this command the $1 stands for the first field, in this case of first column and $2 for second column and it filters output(pic2) of specific line/file which u want from specified one from first command   
   


**Shell script Q & A with some examples:**\* Write a simple script to list all processes ?  
@ command : ps –ef | awk –F “ “ ‘{print $1}’  
here $ specifes which cloumn no u want to print from this process file u mentioed, -F used for separtion of one command to other or giving space btw and ps –ef lists all the processess in a list  
  
\* Write a script to print only errors from a remote log/logfile?  
@ here in below pic we use curl command for getting some erorrs from a link/website,from that if u want any specific word/text related info means use curl google.com | grep HREF (as example) it gives logfiles details and u can find errors from it   
\*Write a shell script to print no which are divisible by 3,5 & not divisble by 15  
# print numbers btw 1 to 100 of which is div by 3,5 but not by 15

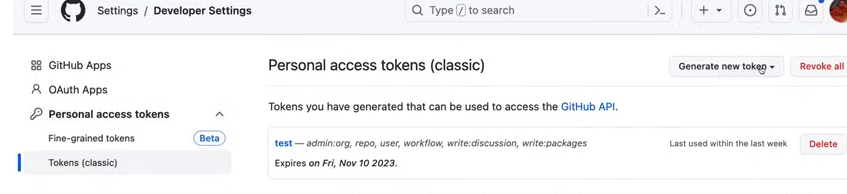
for i in {1..100}; do

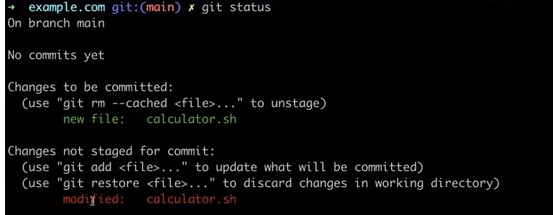
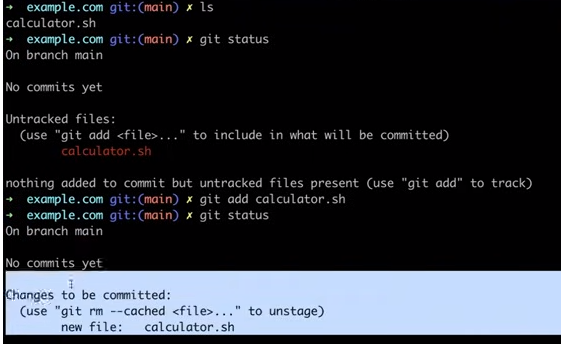
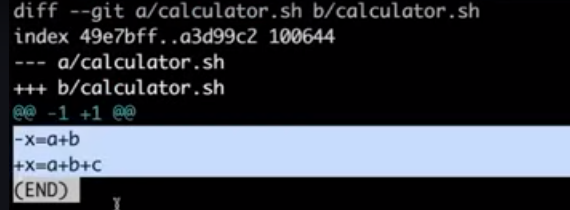
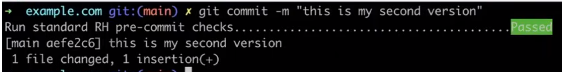
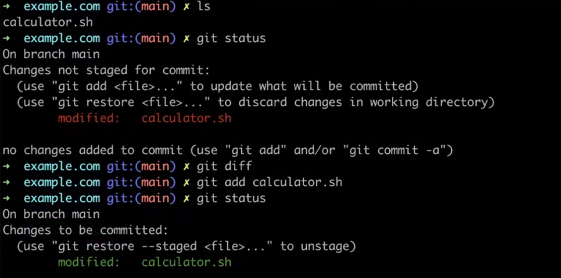
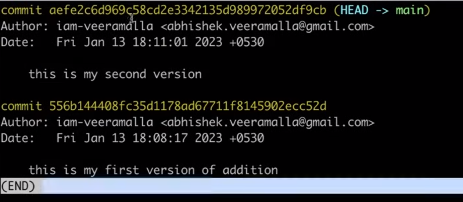
if ([ 'expr $i % 3' == 0 ] || [ 'expr $i % 5' == 0 ]) && [ 'expr $i % 15' != 0 ];  
then  
echo $i  
fi;  
done  
  
\*What is crontab?can u give an example of usage?  
@ it is used to set alert for daily updates/reporting of essential work related like if we want to get report from IAM,EC2,LAMBDA & S3 we can set it through crontab to report at EOD to certain time to anyone or to save daily reports in some s3 storage etc…  
  
\*What is difference btw soft(ln-s) & hard link(ln)?  
@ A symbolic or soft link is an actual link to the original file, whereas a hard link is a mirror copy of the original file. If you delete the original file, the soft link has no value, because it points to a non-existent file.  
\* read –p : reads the file   
ex : read –p “enter the number:” num   
 echo “The result is:$num”  
\* read –sp : it hides the password u type   
\* read –t 5 –p : it is timed reading type for 5 seconds  
ex : #!/bin/bash  
read -t 5 -p "Enter your name within 5 seconds: " name  
Enter your name within 5 seconds: Anonnya  
\* len=${#str} : to find length of a string ex ; str=”ssr” len=${#str} echo “length of string is : $len”  
\*-w : To check if a file is writable   
ex : read -p "Enter a File Name:" fname

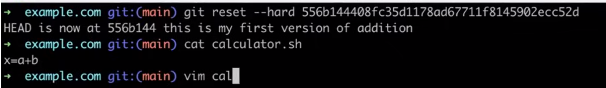
if [ -w $fname ]  
then  
echo "The File $fname is writable."  
else  
echo "The File $fname is not writable."  
fi  
\* How do I give a repo access to DevOps?

Open project settings-> Repositories->click one repo-> select the repositories which you want to give access to another team->add the permission group and set the permission Read to Allow. Then the group users can access these repositories.  
\* We need to first connect to ec2 insance and then **git clone linkofgitrepos** where ur .sh file is stored in github  
  
\*After cloning to github we can see list and enter directory as below and start using the files inside repository   


\*here after cloning we need to export username and token for API access   
for token genrt : settings>dev settings>per access tokens>tokens(clsic)>Gen new token at right  

**Git & Github :**  
\* Git is a tool used for source code management. It is a free and open-source version control system used to handle small to very large projects efficiently. Git is used to tracking changes in the source code, enabling multiple developers to work together on non-linear development,  
as it keeps track of every version of your codebase  
\* GitHub is a web-based hosting service for git repositories. i.e solution built on Git  
\* We can use Git without Github but we can’t use Github without Git  
\* GitHub is an online software development platform. It's used for storing, tracking, and collaborating on software projects. It makes it easy for developers to share code files and collaborate with fellow developers on open-source projects  
\* They have built Github to overcome asability, raising issues, commenting, talking to peers, Reviweing the code and project management (like project tracking etc)  
\* What is Fork?  
@A fork is a copy of a repository that allows you to make your own changes without impacting the original project  
\* In Github comman commands are **git push**,**git add** & **git commit** , Everthing or every file we create in Git is tracked as Objects  
\***Git hooks** are scripts that run automatically every time a particular event occurs in a Git repository. They let you customize Git's internal behavior and trigger customizable actions at key points in the development life cycle or they can help you automate your development lifecycle  
\*The **Git config** command is a convenience function that is used to set Git configuration values on a global or local project level. These configuration levels correspond to .gitconfig text files. Executing git config will modify a configuration text file.  
\***Git status** is a command which shows any files not getting tracked in repos and says it add that file to track it by using git status filename and from now on it looks after its changes and shows its changes of a file if we use command **git add** filename   
  
modified file will be shown in red color   
  
\***Git diff** is a command used to see changes in a file or check the changes u have modified to a file here first it was x=a+b but changed to x=a+b+c  
\* when we do some changes to a file we need to add changes by git add filename and see status by using git status then if it shows as changes to be committed then use git commit –m “this is first/second version” here we r commiting this changes as versioning , in 2nd pic we can see that   
\* If we want to know the changes or exact commits we made we can use **git log** to know the no of versions it has and this way got can track & commit 100s of versions   
\*If we want to go back to old version we need use commit id(highlighted one) and command for it is **git reset –hard commitId** & this one way of versioning   


\* If we want to share or send data from local repo to github use **git push origin master/branchname** and before that use **git remote add origin githubrepolink** from github to connect ur local repository to online github repo  
\* In git there are 4 process – **untracked>modified>staged>commit**  
@untracked means in this the new files which we added will be present but not saved   
@ staged means in this files will be saved but not committed   
@commit means in this after committing this files new versioning will start with first commit on each file(**git commit –m “messagefor version like first/second version**”  
@modified means in this we can get committed file to re-edit and use **git add** to get staged  
& then commit that file after staged   
\* **git add .** is used to convert files from modified untracked files to staged then commit them to modified to commit it  
\* after editing a file use **git add** . as command to convert modified to staged and use **git commit** command to add oneline text in editor and save it and use command **git log –oneline** u will get text message in oneline u edited saved last version   
\*command **git branch** is used to know the branches present in repository and to create branches use **git branch newbranchname** and to switch from one branch to another use **git checkout branchname**   
\*to move from untracked to staged use command **git status** and if we need to remove files from staged then use **git rm --cached filename**(it will be removed ur file from staged & it will move to untracked)  
\*git log is used to check no of committed files and before commiting at first time we should use configure it use git config –global user.email “ssrssr@gmail.com” & git config –global user.name “ssrssr”  
\*to check oue connected mail username use **git config –global user.email/user.name**   
\* to connect from ec2 to github we need to do first see the location in which ur staying like ur in pem file location or not when u connect to ec2 and then use **git clone linkofgitrepo**>**export username=”username of github**”>**export token=”token no”**  
\*to connect from from local repo of git to github use **git clone linkof online/githubrepo** > **git config –global user.email=” “> git config –global user.name=” “> git init(only if u want ur local repos to show in github then use this orelse just move from last step to add file & next add>commit>log>remote(only first time)>>push>check in github.  
\*** 