

"This content is shared with the learners for use in the training session and does not infringe on copyright held by anyone."

By Pavan Wankhade - Date : 9th May 2022
For sharing common history:

PROJECT Help

Project1 : [Link1](#)

Project2 : [Link2](#)

=====

```
6 hostname
7 hostname -f
8 ifconfig
9 ip a
10 hostname -i
11 free
12 free -m
13 free -g
14 free -h
15 lscpu
16 cat /etc/os-release
17 clear
18 date
19 cal
20 clear
21 history
```

=====

```
22 touch sample.txt
23 ls
24 ls -l
25 mkdir pavandata
26 ls -l
27 pwd
28 cd pavandata
29 pwd
```

```
30 ls -l
31 cd ..
32 echo "Hello all"
33 echo "Hello all" > sample.txt
34 cat sample.txt
35 tree
36 sudo apt-get install tree
37 tree
38 clear
```

=====

```
40 whoami
```

```
41 sudo -i
```

```
    pwd
    whoami
    exit
```

```
42 whoami
```

```
43 sudo su
```

```
    pwd
    whoami
    exit
```

```
44 pwd
```

```
45 whoami
```

```
46 sudo -i
```

```
    pwd
    whoami
    useradd -m userx
    exit
```

```
47 useradd -m usery
```

```
48 sudo useradd -m usery
```

```
49 mkdir /projectX
```

```
50 sudo mkdir /projectX
```

```
51 ls -l /
```

```
52 sudo chown userx /projectX
```

```
53 ls -l /
```

```
54 id userx
```

```
55 id usery
```

```
56 id pavanwankhade4u
57 id sam
58 cat /etc/passwd
```

```
=====
+++++
## Day2 : 10th May2022
=====
```

```
67 mkdir projectX
68 ls
```

```
69 cd projectX/
70 ls
71 ls -la
72 git init
73 ls -la
74 ls .git/
75 vi hello.py          ## add some lines to file and save it
76 vi bye.py            ## add some lines to file and save it
77 vi credentials.txt   ## add some lines to file and save it
78 ls
79 touch .gitignore
80 echo "credentials.txt" >> .gitignore
81 cat .gitignore
83 ls -la
```

```
90 git config --global user.name pavansw
91 git config --global user.email pavan@simplilearn.com
```

```
84 git add hello.py
85 git add .gitignore
86 git status
87 git log
88 git commit -m "Initial commit by developer1"
93 git log
94 git log --oneline
95 git status
```

```
=====
## Lets do it one more time with same project directory
```

```
96 ls
97 vi hello.py          ## add some lines to file and save it
98 vi bye.py            ## add some lines to file and save it
99 vi pavan.py          ## add some lines to file and save it
100 git status
101 git add .
102 git status
103 git commit -m "Second commit as discussed"
104 git log
```

=====

Try git Diff

```
114 git status
115 vi hello.py          ##### delete few lines and add new lines in file contents
116 git status
117 ls
118 git log --oneline
119 git diff hello.py
120 git add .
121 git commit -m "Updated hello.py"
122 git log --oneline
```

=====

Create GitHub personal account and add developers ssh key as authorized development vm

=====

```
127 ssh-keygen
128 cat /home/pavanwankhade4u/.ssh/id_rsa.pub      ### copy the key from the file
```

On Git Hub :

Right upper corner Avatar/Profile picture : ⇒ Setting ⇒ Access (on Left Panel) ⇒ Select “SSH and GPG Keys” ⇒ Add ssh key and paste here.

=====

create repository with Public visibility on Git Hub and clone it

```
131 git clone git@github.com:pavansw/devopsVodafoneMay2022.git    ## use your repository
132 ls
```

```

133 cd devopsVodafoneMay2022
134 ls -l
135 ls -la
136 touch .gitignore
137 touch Readme.md
138 touch planning.txt
139 touch secret.txt
140 echo "secret.txt" >> .gitignore
141 vi hello.py          ### add some lines
142 ls -la
143 git add .
144 git commit -m "Initial Files are ready"
145 git log
146 git push

```

+++++

Git fetch and git pull

make some changes in any file using Git hub web console and commit it ⇒ we can identify changes between Local Repo and Origin Repo with fetch

If fetch replies some output then apply git pull to adopt those changes locally

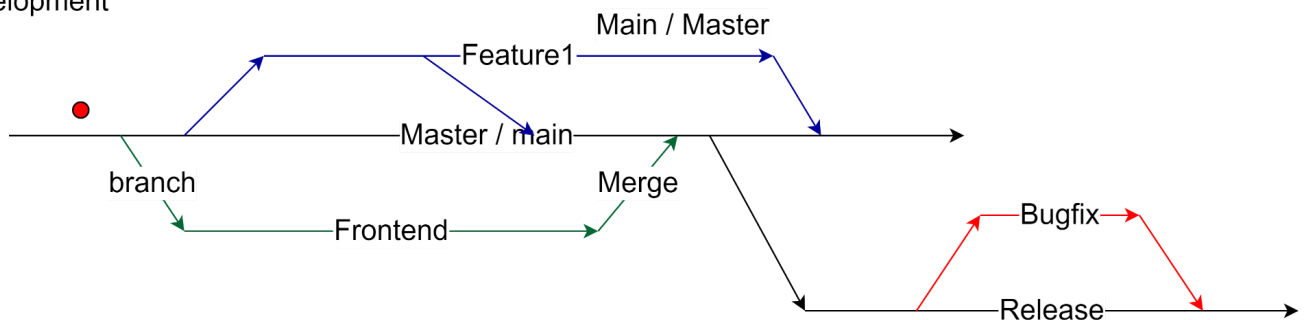
```

150 git log
151 git fetch
152 git pull
153 git log
154 git fetch
155 cat hello.py

```

===== ### Branching and merging

Parallel
Development



```

158 git log --oneline
159 git branch
160 ls
161 git branch frontend

```

```
162 git branch
163 git checkout frontend
164 git branch
165 ls
166 touch frontend1
167 touch frontend2
168 touch frontend3
169 ls
170 git add .
171 git commit -m "First Frontend"
172 git log --oneline
173 touch frontend4
174 touch frontend5
175 git add .
176 git commit -m "Second Frontend"
177 git log --oneline
178 ls
179 git branch
180 git checkout master
181 git branch

182 ls
183 git log --oneline
184 touch master1
185 touch master2
186 git add .
187 git commit -m "Master Development 1"
188 git log --oneline
189 ls
190 git log --oneline
191 git branch
192 git merge frontend
193 ls
194 git log --oneline
```

=====

#####

Day 3

=====

GitHub Branching - Merging - Pull request

```
203 pwd
204 git status
205 git log
206 git log --oneline
207 git push
208 git branch
209 git branch bugfix1
210 git branch
211 git checkout bugfix1
212 git branch
213 ls
214 rm frontend5
215 rm master2
216 touch bugfix1
217 touch bugfix2
218 git add .
219 git commit -m "2 files deleted and 2 new added"
220 git log --oneline
221 git push origin bugfix1
#### After this : From Git hub web console create Pull request and then by accepting the pull request
merge into master
=====
```

Extras : Java - Maven sample project

```
264 git clone https://github.com/pavansw/simpleMavenJUnit.git
265 cd simpleMavenJUnit/
266 ls
267 ls src/main/java/hello/Greeter.java
268 cat src/main/java/hello/Greeter.java
269 cat src/test/java/hello/GreeterTest.java
270 mvn --version
271 ls
272 vi pom.xml
273 mvn compile
274 ls
275 ls target/
276 mvn test
277 ls target/surefire-reports/
278 ls tar
279 ls target/
280 mvn package
281 ls target/
```

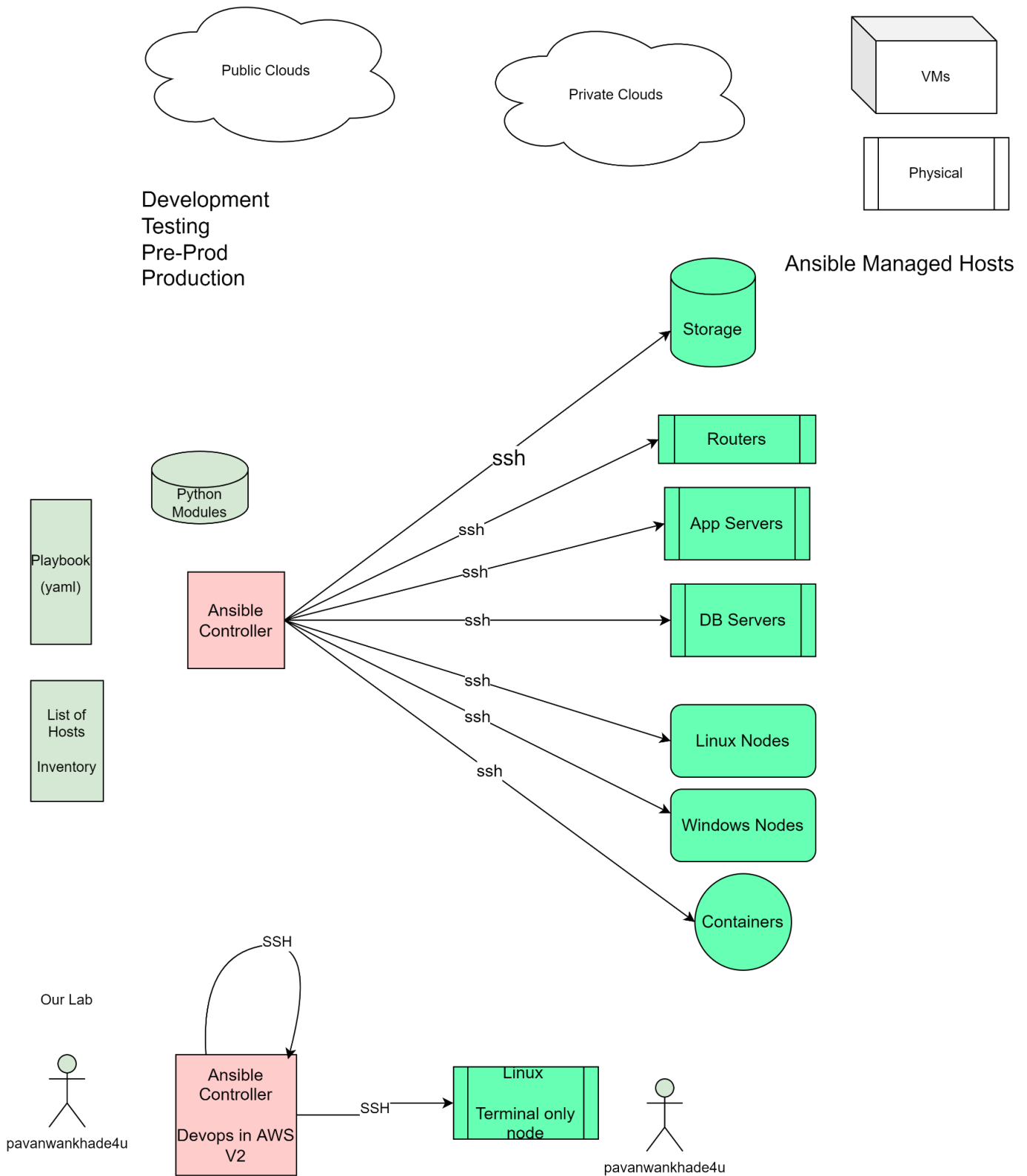
282 java -jar target/gs-maven-0.1.0.jar

=====

Day 4:

=====

CMT : Ansible



On Master/ V2 lab node Perform following

```
-----
289 ansible --version
290 cat .ssh/id_rsa.pub      ### if no output then generate key "ssh-keygen"
291 cat .ssh/id_rsa.pub >> .ssh/authorized_keys
292 ssh localhost -p 42006
    exit                    ## to logout
```

To add the second Terminal only node as Managed host need to share ssh key

```
294 ssh-copy-id pavanwankhade4u@3.89.164.44 -p 42006      ### as per your lab Terminal only
node details
```

Lets create inventory

```
296 sudo vi /etc/ansible/hosts
```

[simplilearn]

localhost:42006

3.89.164.44:42006

Ip of your terminal only node lab

```
297 ansible simplilearn --list
298 ansible simplilearn -m ping
```


Ad-Hoc Commands

```
301 ansible simplilearn -m file -a 'name=/tmp/projectA state=directory'
302 ansible simplilearn -m command -a 'ls /tmp'
303 ls /tmp
304 ansible simplilearn -m file -a 'name=/tmp/projectA state=directory'
305 ansible simplilearn -m file -a 'name=/tmp/projectA state=absent'
```

Some more examples with User module and apt module

```
309 ansible simplilearn -m command -a 'free -h'
310 ansible simplilearn --become -m user -a 'name=samual state=present'
311 id samual
```

```
312 ansible simplilearn --become -m user -a 'name=samual state=present'
313 ansible simplilearn --become -m user -a 'name=samual state=absent'
314 id samual
315 ansible simplilearn --become -m apt -a 'name=tree state=present'
316 ansible simplilearn --become -m apt -a 'name=tree state=absent'
317 tree
318 ansible simplilearn --become -m apt -a 'name=tree state=present'
```

=====

First playbook to deploy apache webserver

```
320 vi webdeploy.yaml
```

```
- name: Apache Webserver Implementation on Ubuntu nodes
  hosts: simplilearn
  become: yes
  tasks:
    - name: Install apache2 package
      apt: name=apache2 state=present

    - name: Service Apache2 getting started
      service: name=apache2 state=started enabled=yes

    - name: Copy Webpage to Document root
      copy: src=webpage.html dest=/var/www/html/index.html
```

```
321 vi webpage.html
```

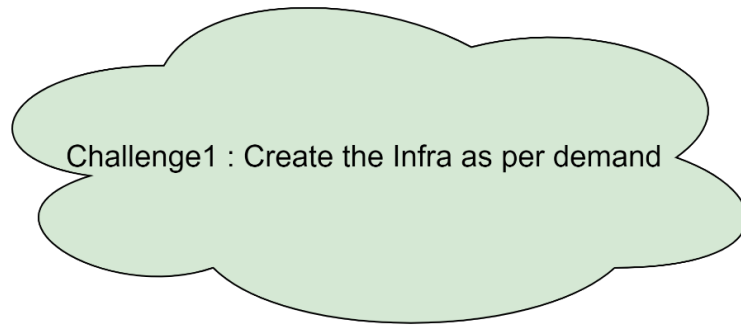
Hello all , Pavan wants to welcome you all

```
322 ansible-playbook webdeploy.yaml --syntax-check
323 ansible-playbook webdeploy.yaml
324 curl http://localhost
325 ansible-playbook webdeploy.yaml
```

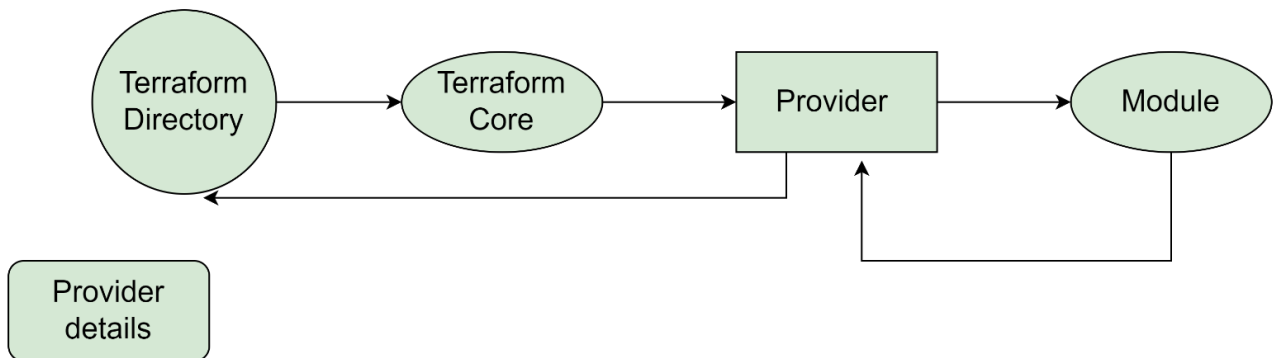
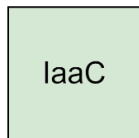
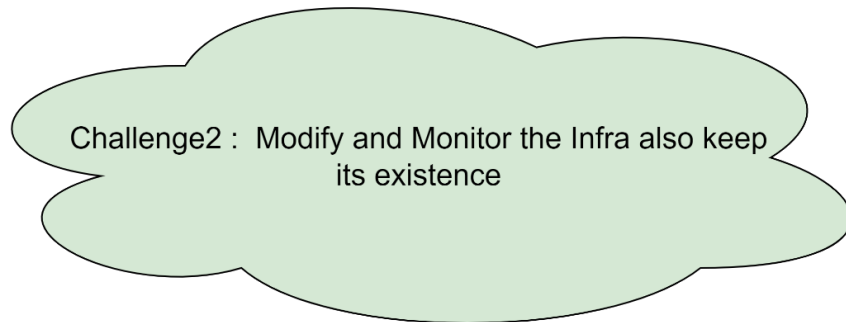
=====

Terraform

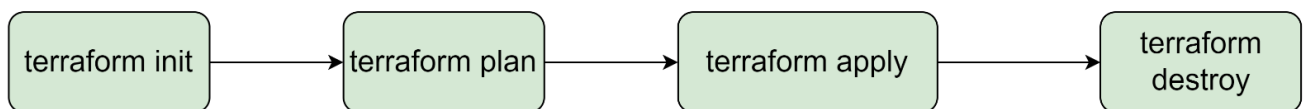
=====



Terraform



Workflow



Ref : Providers : <https://registry.terraform.io/browse/providers>
Installations : <https://www.terraform.io/downloads>

```
330 terraform --version
331 mkdir terra-myproject
332 cd terra-myproject/
```

Use your aws login keys and tokens

```
337 vi credentials.tf
provider "aws" {
  region = "us-east-1"
  access_key = "*****"
  secret_key = "*****"
  token = "*****"
}
```

```
338 vi aws.tf
resource "aws_s3_bucket" "bucket1" {
  bucket = "mys3bucket-pavan-training-vodafone" ##### Name has to be unique on the region
  acl = "private"

  tags = {
    Name = "My Bucket1"
    Environment = "Development"
  }
}
```

```
340 terraform init
341 ls
342 ls -la
343 terraform plan
344 terraform apply
345 ls
346 cat terraform.tfstate
347 terraform destroy
```

=====

+++++

Jenkins CI/CD:

Start : 13th Jun 2022

=====

Demo 1 : Git gub and Git

Create a Private repository on Git Hub and clone it on the terminal

```
73 git clone git@github.com:pavansw/vodafoneJenkins.git
```

```
74 ls
```

```
75 git config --global user.name pavansw
```

```
76 git config --global user.email pavansw@example.demo.com
```

```
77 cd VodafoneJenkins
```

```
78 pwd
```

```
80 ls -l
```

```
81 vi hello.sh
```

```
82 ls
```

```
83 git add hello.sh
```

```
84 git commit -m "Initial hello.sh ready"
```

```
85 git log
```

```
86 git push origin master      OR      git push origin main
```

=====

```
88 vi myprogram.py
print ("Hello all")
```

```
89 vi README.md
Type some Project Intro Lines text
```

```
90 touch personaldata.txt
```

```
91 echo "personaldata.txt" >> .gitignore
```

```
92 git add .
```

```
93 git status
```

```
94 git commit -m "Second commit for demo"
```

```
95 git log
```

```
96 git push origin master
```

```
97 git log
```

=====

