**Lab 10**

**Laboratory Exercise**

**LAB EXERCISE: Puppet operation in Jenkins Pipeline**

**What You Learn**

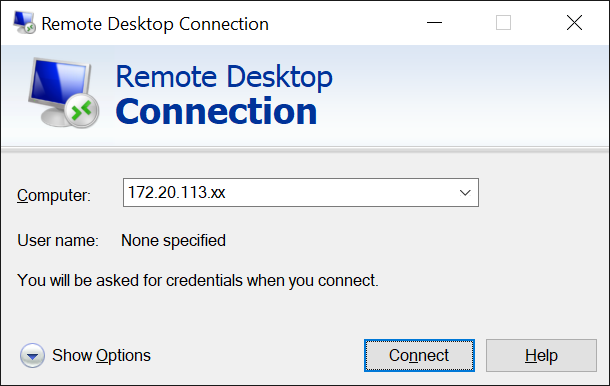
* Puppet in Jenkins Pipeline

**Time to Complete**

Approximately 90 Minutes

From your machine logged-in to RP VPN, run Remote Desktop Connection to connect to the ubuntu Linux Virtual Machine (VM). Please login based on your assigned VM as shown below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Name** | **VM** | **IP Address** | **User Name** | **Password** |
| 1 | LEOW TANG QING | PDC2-Ubuntu-03 | 172.20.113.184 | dockeradm | docker!2 |
| 2 | LIM SHEN HUI | PDC2-Ubuntu-02 | 172.20.113.183 | dockeradm | docker!2 |
| 3 | LIN JINGZHOU | PDC2-Ubuntu-04 | 172.20.113.185 | dockeradm | docker!2 |
| 4 | MUHAMMAD FAISAL BIN SHAIK HASSAN | PDC2-Ubuntu-05 | 172.20.113.186 | dockeradm | docker!2 |
| 5 | LYNN LEE QING XIA | PDC2-Ubuntu-06 | 172.20.113.187 | dockeradm | docker!2 |
| 6 | NG CHEE KIONG | PDC2-Ubuntu-07 | 172.20.113.188 | dockeradm | docker!2 |
| 7 | PARAMASIVAM S/O VANNU GOPAL | PDC2-Ubuntu-08 | 172.20.113.189 | dockeradm | docker!2 |
| 8 | SAHLATUL-FARIHAH BINTE M ASARI | PDC2-Ubuntu-09 | 172.20.113.190 | dockeradm | docker!2 |
| 9 | SHAIKH FAID BIN OMAR | PDC2-Ubuntu-10 | 172.20.113.191 | dockeradm | docker!2 |
| 10 | CHIU JING XIONG | PDC2-Ubuntu-11 | 172.20.113.192 | dockeradm | docker!2 |
| 11 | KELLY WONG XUE YEE | PDC2-Ubuntu-12 | 172.20.113.193 | dockeradm | docker!2 |
| 12 | LIM SI YING | PDC2-Ubuntu-13 | 172.20.113.194 | dockeradm | docker!2 |
| 13 | LIN LI YI | PDC2-Ubuntu-14 | 172.20.113.195 | dockeradm | docker!2 |
| 14 | MUHAMMAD MUQTADIR BIN SADIQ BASHA | PDC2-Ubuntu-15 | 172.20.113.196 | dockeradm | docker!2 |
| 15 | NUR HIDAYAH BTE RAMLEE | PDC2-Ubuntu-16 | 172.20.113.197 | dockeradm | docker!2 |
| 16 | NUR NADIA ASHBOLLAH BINTE | PDC2-Ubuntu-17 | 172.20.113.198 | dockeradm | docker!2 |
| 17 | NUR THAQIFAH AQILAH BINTE JURAIMI | PDC2-Ubuntu-18 | 172.20.113.199 | dockeradm | docker!2 |
| 18 | RAUDHATUNNISHA BTE RAMLI | PDC2-Ubuntu-19 | 172.20.113.200 | dockeradm | docker!2 |
| 19 | SITI NUR ALYSHYIA BINTE HASHIM | PDC2-Ubuntu-20 | 172.20.113.201 | dockeradm | docker!2 |
| 20 | TAN TEE BING | PDC2-Ubuntu-21 | 172.20.113.202 | dockeradm | docker!2 |



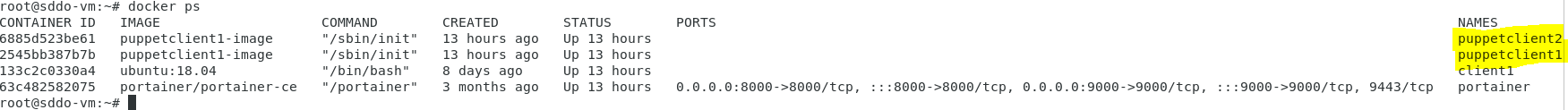
Replace xx with the IP address of the VM that you have been assigned.

**Part 1: Prepare simulated environments and containers**

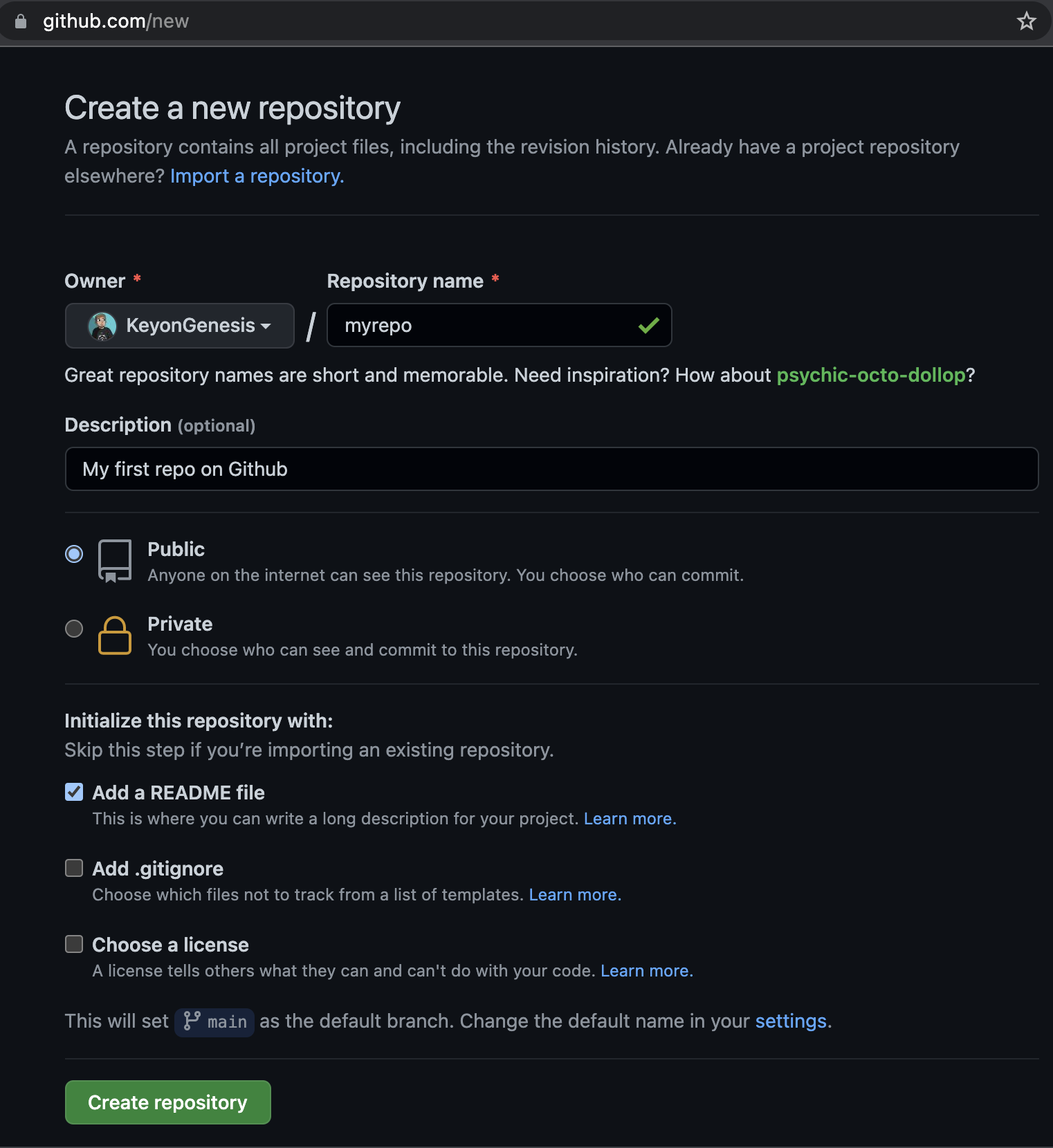
On sddo-vm, create the following directory, subdirectories and files:

**  
1.1** Verif**y** that there are 2 client containers, namely puppetclient1 and puppetclient2. If there is none of only one, please follow lab 6 to create the missing container.

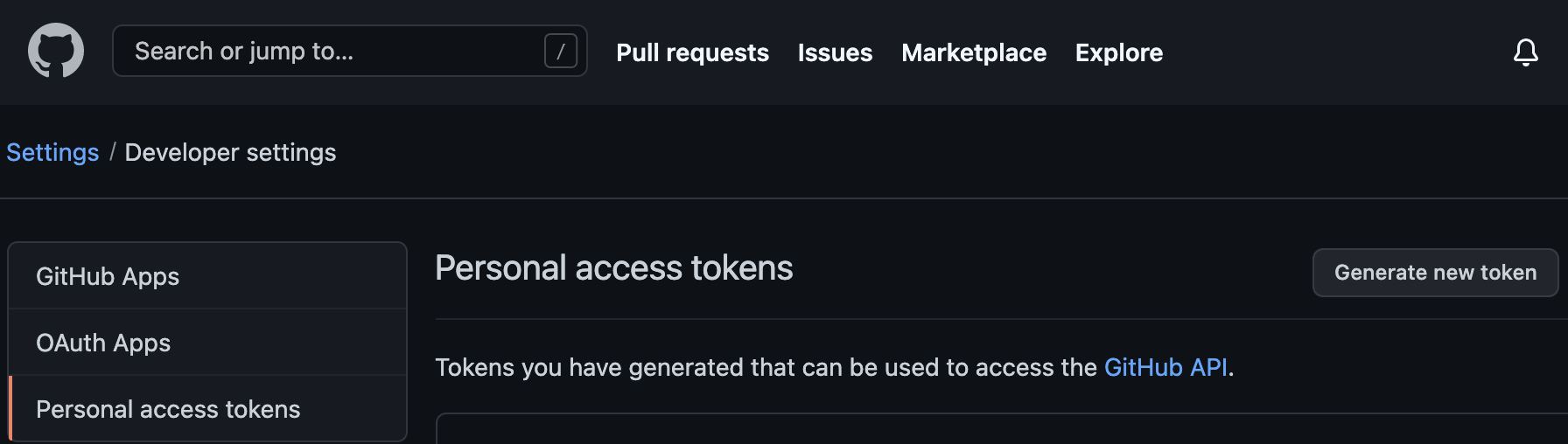
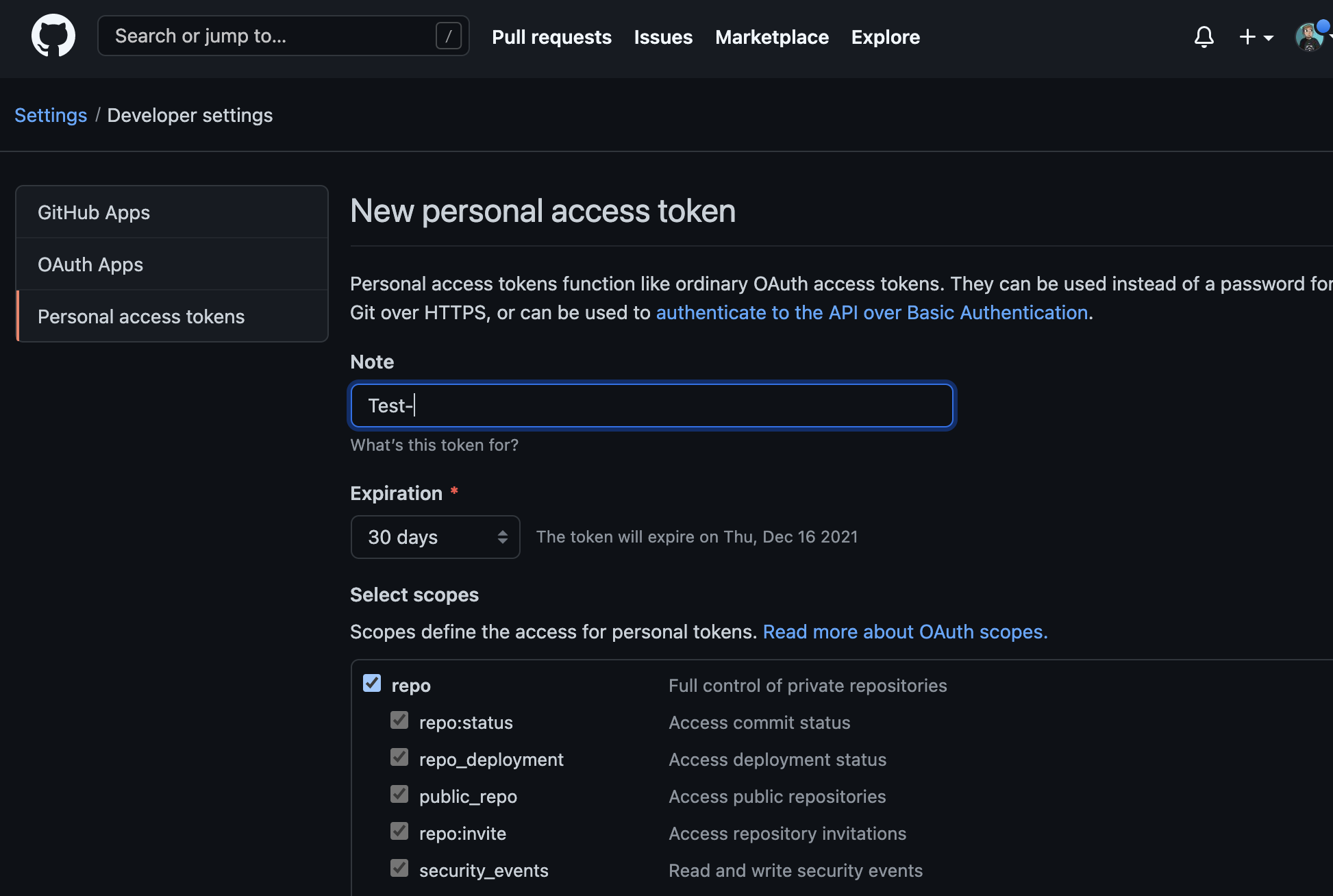
root@sddo-vm: **docker ps**



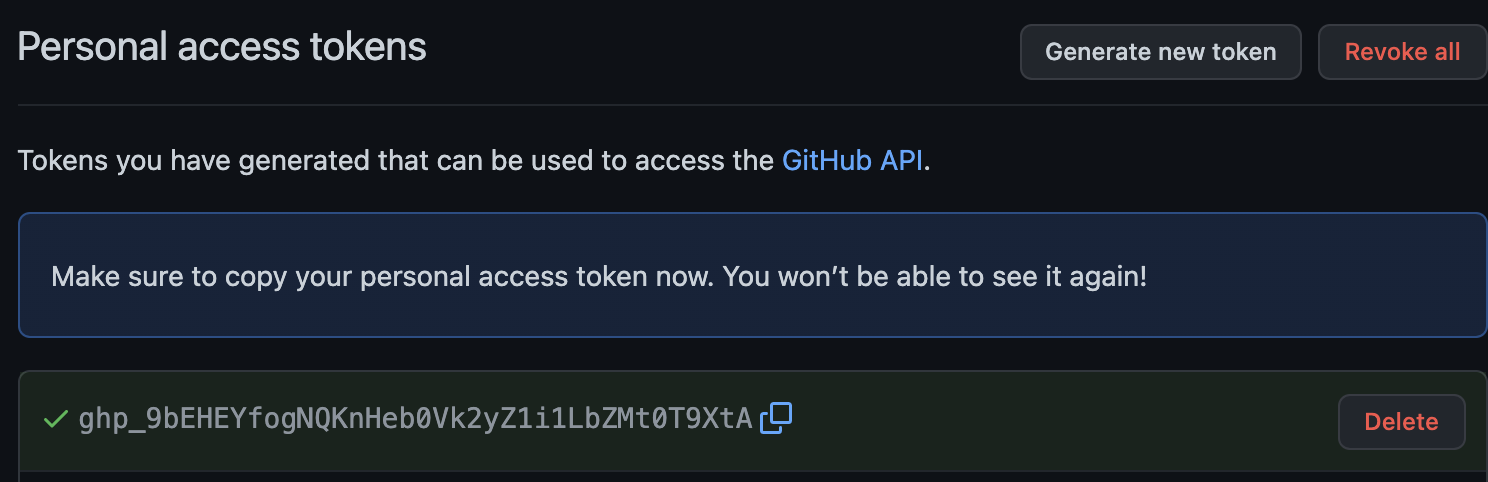
**1.2**  Login to GitHub using the your credential

1.2.1 Start a project on GitHub  
1.2.2 Create a new repository, name as **devops\_repo**  
 

1.2.3 Github now uses personal token. If you have not generated a token, follow the following steps.

• Click the profile icon and find the settings menu   
 • In the profile settings page, scroll down to “developer settings”  
 • Click “Personal access tokens”   
 • And then “Generate new token”  
   
  
1.2.4. Give a name for the note to the access token  
1.2.5. Check the “repo” box   
 

1.2.6. “Generate token”

• Copy your token to a safe place, e.g. notepad   
 • You won’t see the token again   
 • This token will be the password when you push/pull your repository  
 

1.2.7 Create 3 files in repository **devops\_repo** under **Prod** branch:

* Jenkinsfile
* script\_to\_run
* Index.html

1.2.8 Create the following content for the file **Jenkinsfile** on Github repository **devops\_repo**

pipeline {

agent any

stages {

stage('One') {

steps {

echo 'Begin of Pipeline: Stage one completes'

}

}

stage('Two') {

steps {

input('Do you want to update to Development container?')

}

}

stage('Three') {

when {

not {

branch "Development NOT updated"

}

}

steps {

sh '''#!/bin/bash

targets=puppetclient1;

locate\_script='/testdir/work/devops\_repo/script\_to\_run';

docker cp $locate\_script $targets://$locate\_script;

bolt script run $locate\_script -t $targets -u clientadm -p user123 --no-host-key-check --run-as root;

'''

echo "Development container updated"

}

}

stage('Four') {

steps {

input('Do you want to update to Production container: Proceed to Production')

}

}

stage('Five') {

when {

not {

branch "Production NOT updated"

}

}

steps {

sh '''#!/bin/bash

targets=puppetclient2;

locate\_script='/testdir/work/devops\_repo/script\_to\_run';

docker cp $locate\_script $targets://$locate\_script;

bolt script run $locate\_script -t $targets -u clientadm -p user123 --no-host-key-check --run-as root;

'''

echo "Production container updated"

}

}

stage('Completed updating Operation') {

steps {

echo 'Completed updating to Production Container'

}

}

}

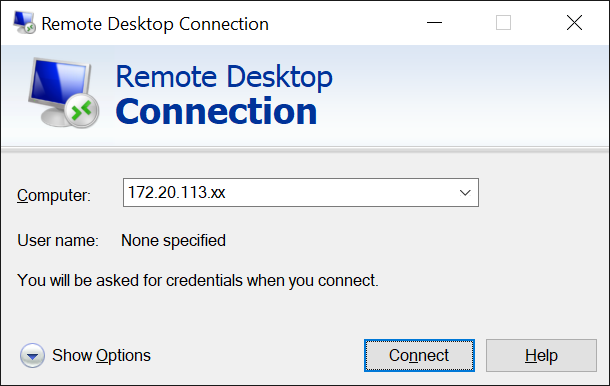
}

1.2.9 Create the following content for the file **script\_to\_run** on Github repository **devops\_repo**

|  |  |
| --- | --- |
| #!/bin/bash  puppet resource package git ensure=present  puppet resource package apache2 ensure=present  puppet resource service apache2 ensure=running  puppet resource file /testdir/clone ensure=absent force=true  puppet resource file /testdir ensure=directory  puppet resource file /testdir/clone ensure=directory  cd /testdir/clone  git clone https://ghp\_iAekp3iByYsyPonXwsJW39XYzWWeSy3gUgkv@github.com/bcsim-git/devops\_repo.git  cp -p /testdir/clone/devops\_repo/index.html /var/www/html | |
|  | |  | |
|  | |  | |
| 1.2.9 Create the following content for the file **index.html** on Github repository **devops\_repo** | |  | |
| <html>  <head>  </head>  <body>  <h1>Hello World<h1>  </body>  </html> | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |

1.3 Preparing **sddo-vm**:

1.3.1 Remotely login to **sddo-vm**:



Replace xx with the IP address of the VM that you have been assigned.

Create the follow directory and subdirectory:

root@sddo-vm: **mkdir -p /testdir/work**

**1.3.2 Clone github repository devops\_repo to sddo-vm’s directory /testdir/work:**

root@sddo-vm: **cd /testdir/work**

root@sddo-vm: **git clone https://<your access key token>@github.com/<your account name>/devops\_repo.git**

After the above command, the following file structure should be created.

****

**1.3 Preparing puppet client nodes:**

1.3.1 Access **puppetclient1**:

root@sddo-vm: **docker exec -it puppetclient1 /bin/bash**

root@puppetclient1: **mkdir -p /testdir/work**

1.3.2 Access **puppetclient1**:

root@sddo-vm: **docker exec -it puppetclient2 /bin/bash**

root@puppetclient1: **mkdir -p /testdir/work**

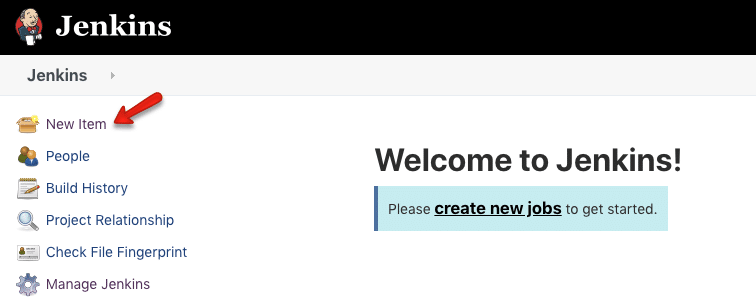
1.4 Creating & Building a Jenkins Pipeline Job

Access <http://localhost:9090>

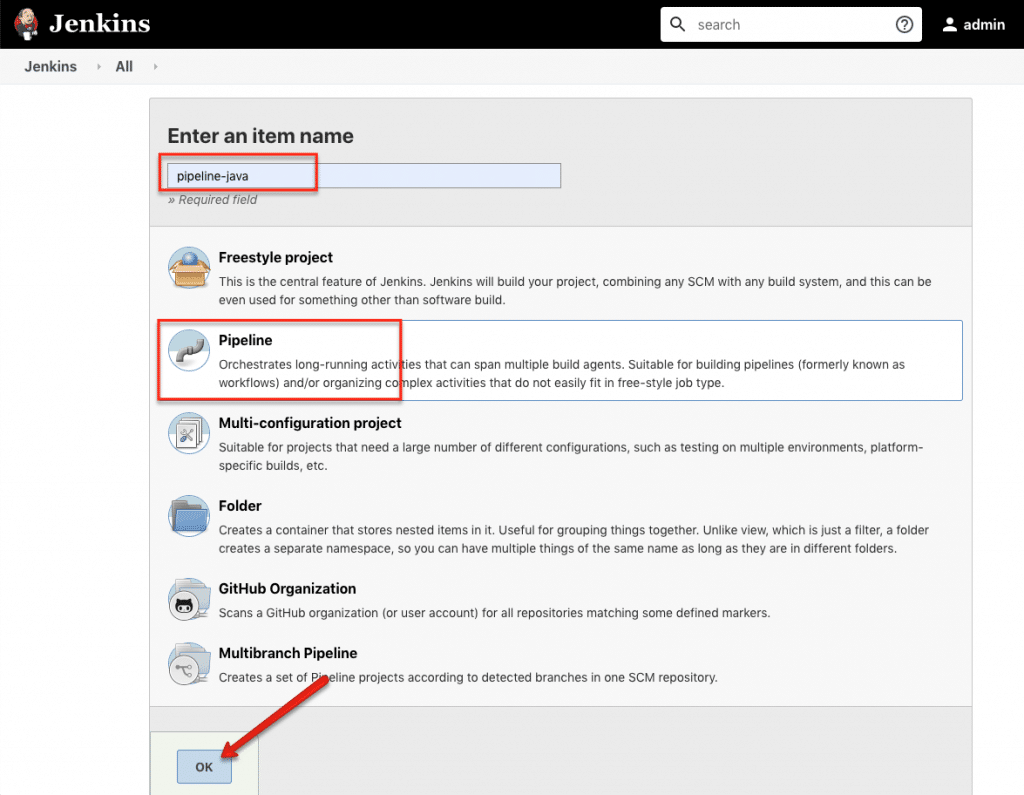
Username: **jadmin**

Password: **Jadmin!2**

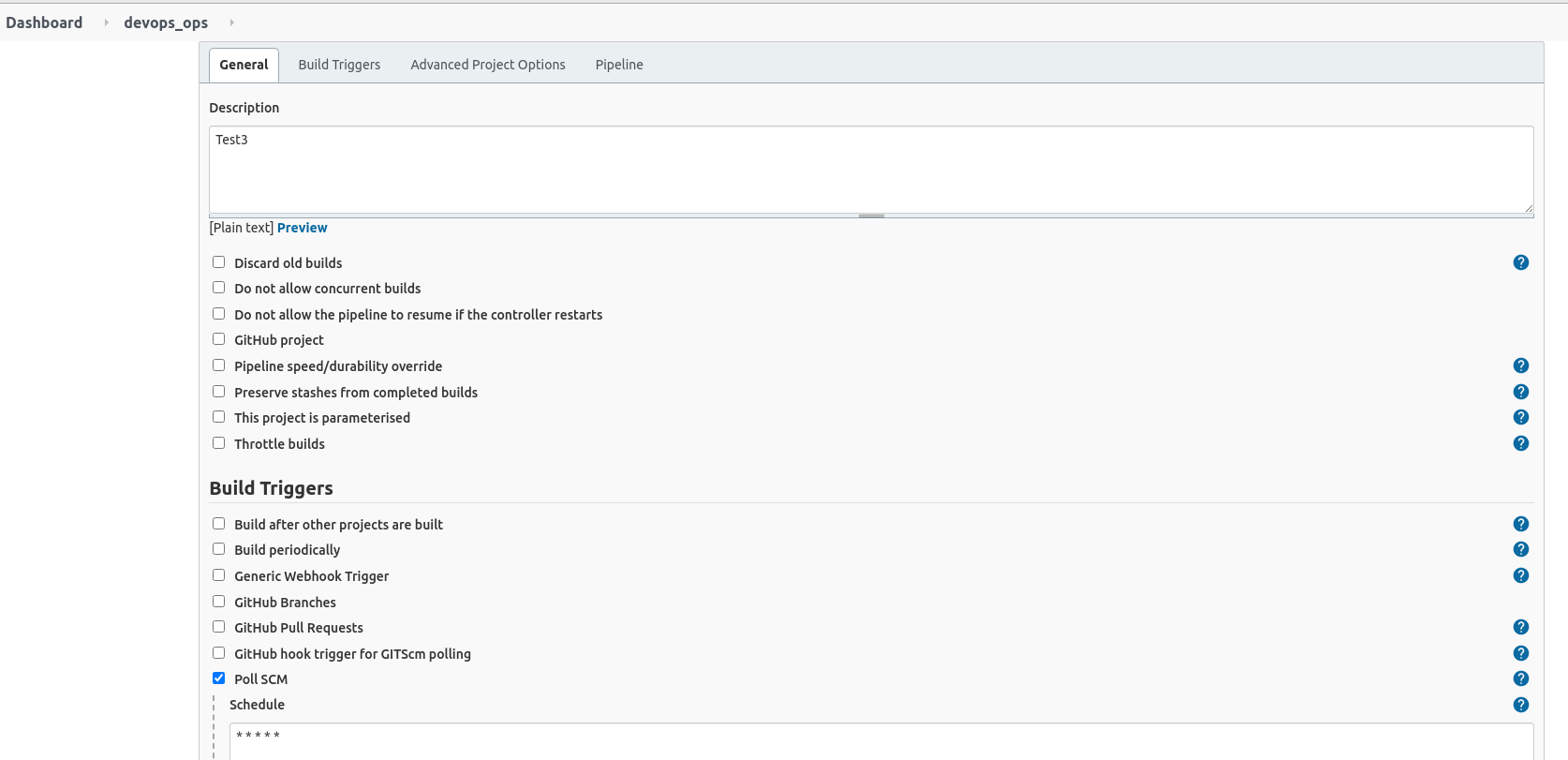
3.1. Go to Jenkins home and select “New Item”

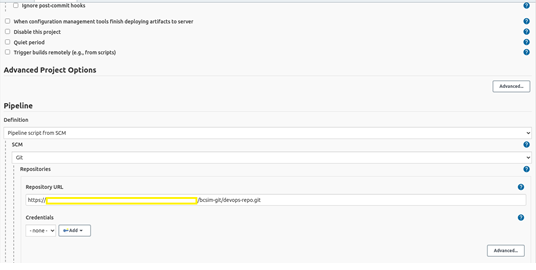


3.2. Give a name, select “devops\_ops” and click ok.



Configure the pipeline as below:



****

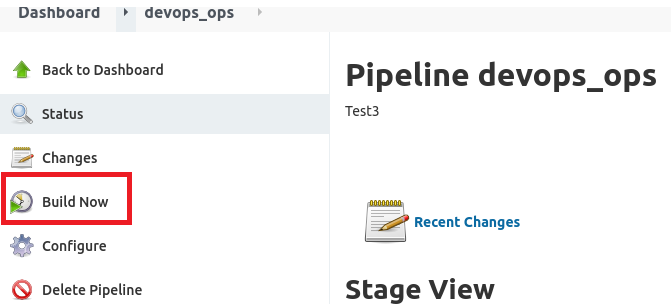


**Part 2: Testing the operation**In this part of the lab, we are going to test the pipeline and the update to index.html.

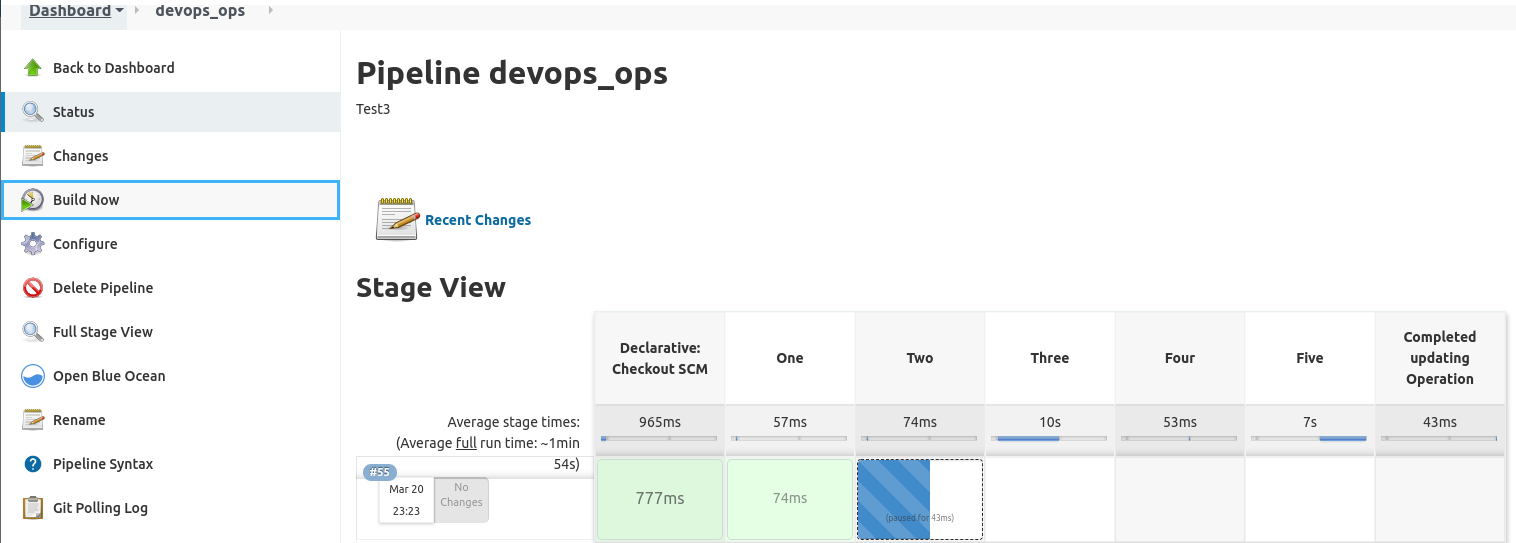
Let’s look at the content of index.html file again:

|  |
| --- |
| <html>  <head>  </head>  <body>  <h1>Hello World<h1>  </body>  </html> |
|  |

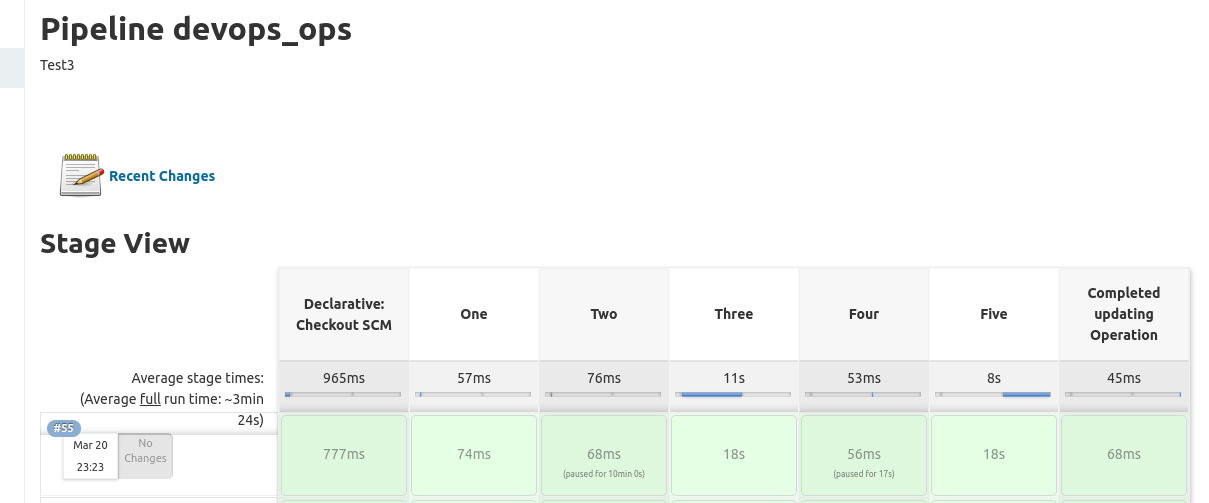
2.1 Trigger “Build Now” on devops\_repo pipeline.



While the job starts you can view each stage executing in stage view.

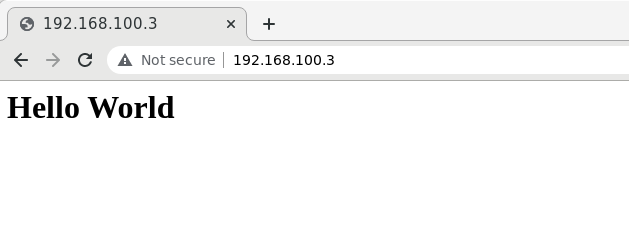


2.2 The pipeline should complete without error as shown below.

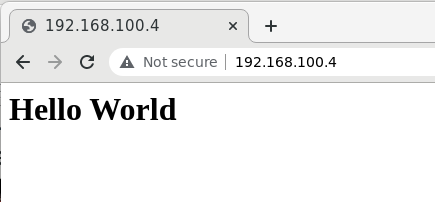


2.3 From **sddo-vm**, access the website which is running on puppetclient1 and puppetclient2. You should see the following:

For **puppetclient1**:



For **puppetclient2**:



2.4 Now, make the following change to the index.html file on sddo-vm.

root@sddo-vm: **cd /test/work/devops\_repo**

root@sddo-vm: **nano index.html**

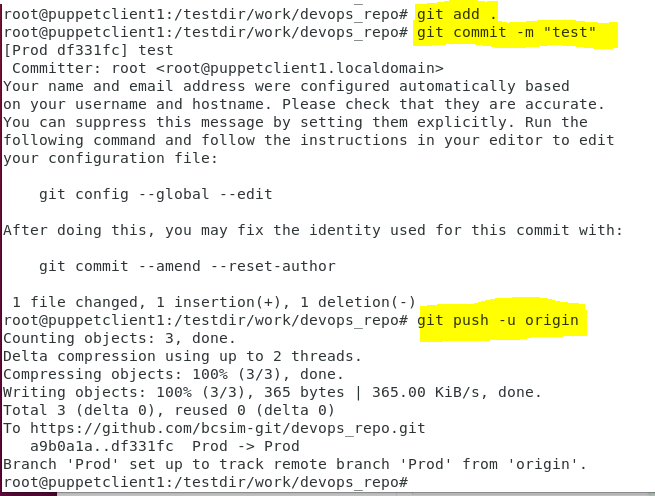
|  |
| --- |
| <html>  <head>  </head>  <body>  <h1>Great Planet<h1>  </body>  </html> |
|  |

2.5 Update the change to github:

root@sddo-vm: **git add .**

root@sddo-vm: **git commit -m “test”**

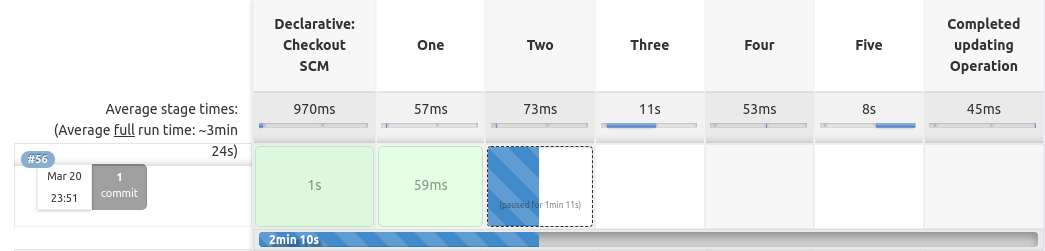
root@sddo-vm: **git push -u origin**

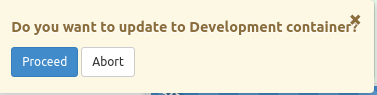


2.5 Wait for 1 minutes, Jenkins Pipeline **devops\_ops** should be auto trigger.

Do you know why?

The pipeline stops at stage 3, awaiting for a response:



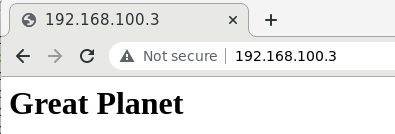


Click on proceed. It will again stop at stage 4.

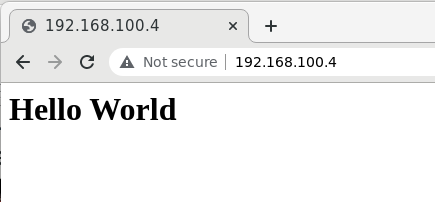


2.6 From **sddo-vm**, access the websites of **puppetclient1** and **puppetclient2** again.

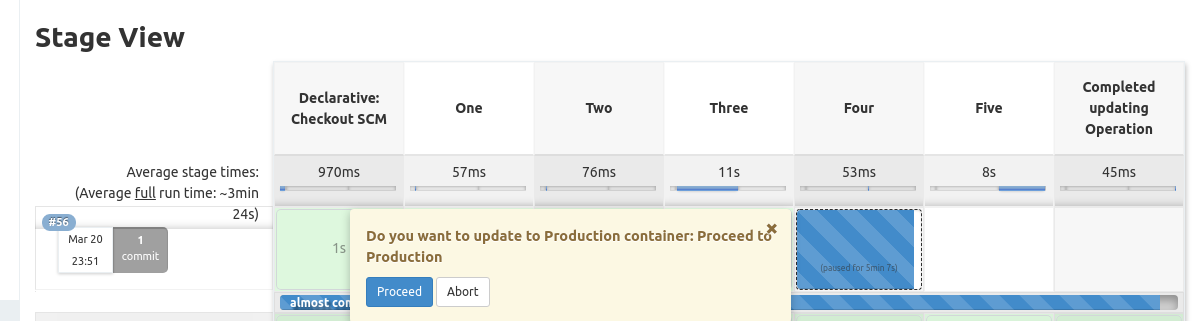
For **puppetclient1**:



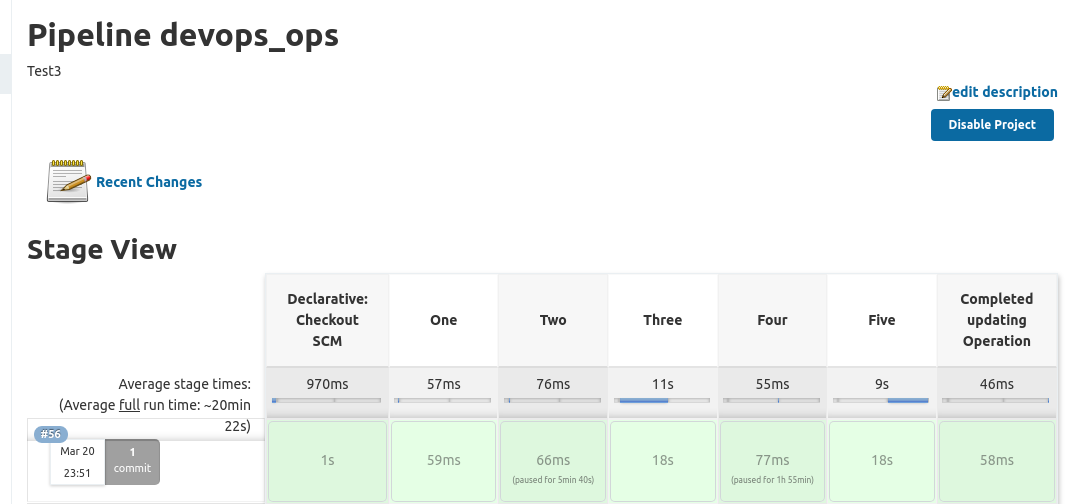
For puppetclient2:



2.7 Click on proceed.

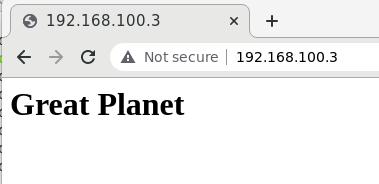


2.8 The pipeline should complete without error.

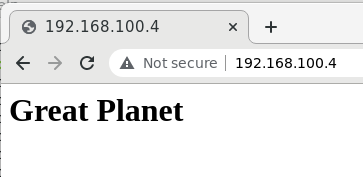


2.9 Access the websites of puppetclient1 and puppetclient2 again. You should see the following

For puppetclient1’s website:



For puppetclient2’s website:



**--End of Lab Exercise --**