

**DV1C04 AY2022 TERM 1**

**Individual Coursework Submission**

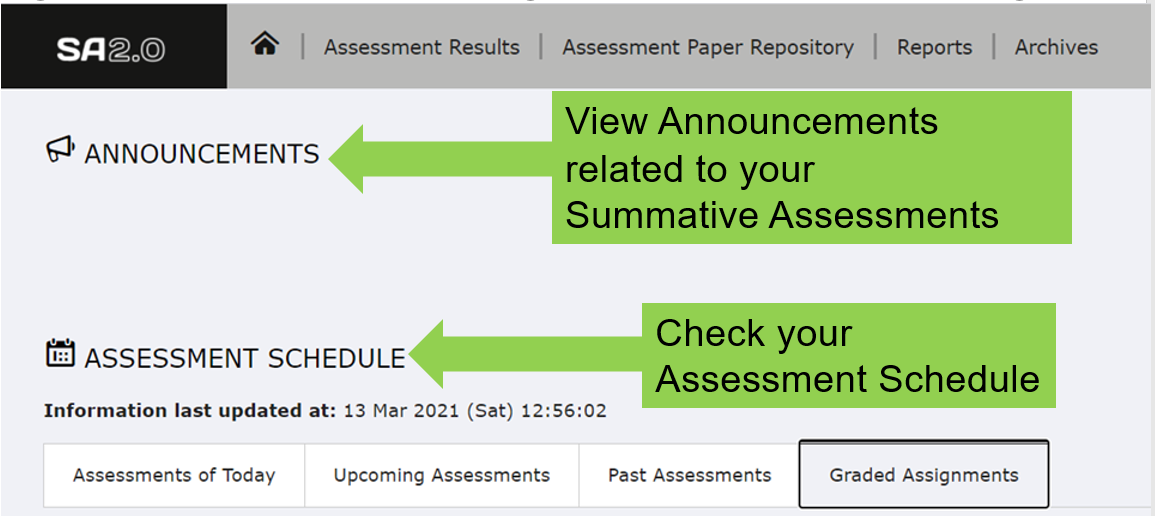
|  |  |
| --- | --- |
| **Personal Details** | |
| **Name** | **Siti Nur Alyshyia Binte Hashim** |
| **Student No.** | **21051265** |

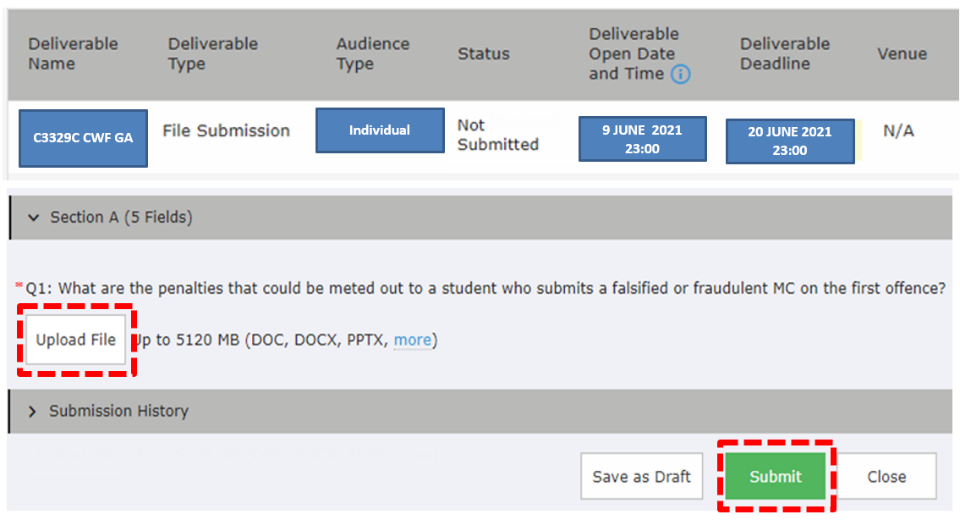
|  |  |  |
| --- | --- | --- |
| **Compliance Statement** | | |
| **Plagiarism**  I declare that this report is my original work. I understand that if I am suspected of plagiarism, my enrolment in the programme may be terminated. | | **** |
| **Retention of Backup Copy**  I declare that I have a back-up electronic copy of this report for immediate submission. | | **** |
| **Signature** |  | |
| **IMPORTANT:** Non-compliance to these clauses will result in unconditional rejection of your submission | | |

Instruction to the Candidate

This coursework assignment is an individual assignment. The assignment must to be completed and submitted by each candidate.

1. **Upload your completed assignment report to SA2.0 as shown in the screen shot below. Access** [***https://mysa.rp.edu.sg***](https://mysa.rp.edu.sg/)***,* enter your RP Student ID and Password, and click LOGIN.**





Full student guide for SA2.0 can be found at:[**BITS@RP**](https://myrp.sharepoint.com/:b:/r/sites/SA/User Guide/Republic Polytechnic SA 2.0 Student User Guide v2.0_Final.pdf?csf=1&web=1&e=xNH3Kp)

1. **Please note the deadline your assignment report submission is 24th April 2022, 23:00 on SA2.0.**
2. **The following table shows your assigned VM for your design implementation:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Name** | **VM** | **IP Address** | User Name | Password |
| 1 | LEOW TANG QING | DC1C04-CWF-01 | 172.20.113.210 | dockeradm | <Your Student ID> |
| 2 | LIM SHEN HUI | DC1C04-CWF-02 | 172.20.113.211 | dockeradm | <Your Student ID> |
| 3 | LIN JINGZHOU | DC1C04-CWF-03 | 172.20.113.212 | dockeradm | <Your Student ID> |
| 4 | MUHAMMAD FAISAL BIN SHAIK HASSAN | DC1C04-CWF-04 | 172.20.113.213 | dockeradm | <Your Student ID> |
| 5 | LYNN LEE QING XIA | DC1C04-CWF-05 | 172.20.113.214 | dockeradm | <Your Student ID> |
| 6 | NG CHEE KIONG | DC1C04-CWF-06 | 172.20.113.215 | dockeradm | <Your Student ID> |
| 7 | PARAMASIVAM S/O VANNU GOPAL | DC1C04-CWF-07 | 172.20.113.216 | dockeradm | <Your Student ID> |
| 8 | SAHLATUL-FARIHAH BINTE M ASARI | DC1C04-CWF-08 | 172.20.113.217 | dockeradm | <Your Student ID> |
| 9 | SHAIKH FAID BIN OMAR | DC1C04-CWF-09 | 172.20.113.218 | dockeradm | <Your Student ID> |
| 10 | CHIU JING XIONG | DC1C04-CWF-10 | 172.20.113.219 | dockeradm | <Your Student ID> |
| 11 | KELLY WONG XUE YEE | DC1C04-CWF-11 | 172.20.113.220 | dockeradm | <Your Student ID> |
| 12 | LIM SI YING | DC1C04-CWF-12 | 172.20.113.221 | dockeradm | <Your Student ID> |
| 13 | LIN LI YI | DC1C04-CWF-13 | 172.20.113.222 | dockeradm | <Your Student ID> |
| 14 | MUHAMMAD MUQTADIR BIN SADIQ BASHA | DC1C04-CWF-14 | 172.20.113.223 | dockeradm | <Your Student ID> |
| 15 | NUR HIDAYAH BTE RAMLEE | DC1C04-CWF-15 | 172.20.113.224 | dockeradm | <Your Student ID> |
| 16 | NUR NADIA ASHBOLLAH BINTE | DC1C04-CWF-16 | 172.20.113.225 | dockeradm | <Your Student ID> |
| 17 | NUR THAQIFAH AQILAH BINTE JURAIMI | DC1C04-CWF-17 | 172.20.113.226 | dockeradm | <Your Student ID> |
| 18 | RAUDHATUNNISHA BTE RAMLI | DC1C04-CWF-18 | 172.20.113.227 | dockeradm | <Your Student ID> |
| 19 | SITI NUR ALYSHYIA BINTE HASHIM | DC1C04-CWF-19 | 172.20.113.228 | dockeradm | <Your Student ID> |
| 20 | TAN TEE BING | DC1C04-CWF-20 | 172.20.113.229 | dockeradm | <Your Student ID> |

Grading Criteria:

This assignment has a maximum of **200** marks.

* Criteria 1, 2, 3 & 4 make up **140** marks.
* Criteria 5 makes up **60** marks.

|  |  |
| --- | --- |
| **S/N** | **Criteria** |
| **1** | **Documentation** |
|  | *Consistent and clear formatting with content page, page numbering, consistent font type/size and relevant headings.*  *Free of typographical, grammar, and other such errors*  *Appropriate use of tables, graphs, charts, equations, etc. as a visual aid to explain* |
| 2 | **Design Completeness** |
|  | *Completeness in design (e.g. Jenkins file etc) to meet assignment requirements.*  *Relevancy and correctness of solutions to the assignment requirement* |
| **3** | **Design Implementation** |
|  | *Demonstrate ability to implement solution on resources provided.*  *Provide screenshots for implementation.* |
| **4** | **Comprehension of Design (Video Recording)** |
|  | *Demonstrates understanding of design and solutions.*  *Coherence and logical development of the ideas; sequencing and flow of the design and solution.* |
| **5** | **Monitor** |
|  | *Provide relevant answers to the questions asked.* |

**Submission Procedures**

1. Candidate must produce a report using this **DV1C04** Coursework Final Assignment Report Template
2. All sections in the template MUST be completed
3. All font, header, footer, margin, page layout settings must be followed
4. Candidate is to upload his/her assignment to SA2.0. in **two** files  
   - **One file is candidate’s assignment report**  
   - **One file is candidate’s recorded video explaining his/her design in the submitted report**. (Video recording should last only 5-10minutes.)
5. For the submission to SA2.0, the report file must be saved in PDF format with the following file naming convention:

<Student ID>-<Name>-DV1C04-AY20XXCWF.pdf

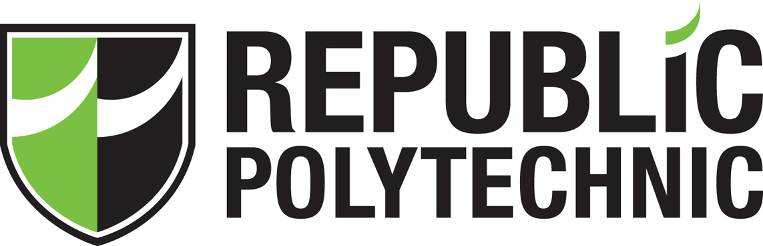
e.g. 2001111-John Khoo-DVC04-AY20XXCWF.pdf

1. Candidate must submit his/her final report to the Programme Co-ordinator no later than **2300h** on **24th Apr 2022 to SA2.0**. Late submissions of assignment-based coursework component without leave of absence (LOA) for the module will be subjected to the following late penalty:



1. Candidate is expected to produce his/her own report. Any copying of other’s report will be reported and dealt with.

**Next Page is the start of Assignment Report**



graded Assignment

DV1C04 Deploy & Monitoring in DevOps

Date of Submission: DD-APR-2022

Submitted By:

|  |  |
| --- | --- |
| 21051265 | Siti Nur Alyshyia Binte Hashim |
|  |  |
|  |  |
|  |  |

TABLE OF CONTENTS

[1. Assignment Overview 8](#_Toc99260444)

[1.1 Assignment Requirements 8](#_Toc99260445)

[2. Pipeline Design Overview 9](#_Toc99260446)

[2.1 Pipeline Design Explanation 9](#_Toc99260447)

[3. Pipeline Implementation 10](#_Toc99260448)

[4. Monitor 11](#_Toc99260449)

[References 14](#_Toc99260450)

[Appendices 15](#_Toc99260451)

# Assignment Overview

***Section Explanation:*** *This section should contain a brief description of*

* *Assignment Scope (What are the major tasks and deliverables of the assignment?)*
* *Assignment Assumptions (Are there assumptions which are made regarding how the end deliverables will work, etc.?)*

The major tasks are to set up a Proof-of-Concept (POC) to automate ItnGreen Organization’s operation which to set up a DevOps Operate pipeline with 5 stages. The deliverables are to provide a successful screenshot of a successful build.

## Assignment Requirements

***Section Explanation:*** *This subsection should clearly state the major requirements of your assignment.*

The requirements are to create one docker container with ubuntu version 18.04 and one Jenkins operate pipeline with 5 stages to replace the original apache2’s index.html file with an updated index.html file. The third requirement is to create a GitHub repository that has 3 files named 21051265\_jenkinsfile, index.html and 21051265\_script. The fourth requirement is a screenshot of a Console Output of a successful build in Jenkins.

# Pipeline Design Overview

***Section Explanation:*** *This section describes the overview of pipeline design. It describes your pipeline design approach.*

Diagram

Description automatically generated

*<Pipeline Diagram>*

## 2.1 Pipeline Design Explanation

***Subsection Explanation:*** *This subsection focuses mainly on your pipeline design for this assignment. Care must be taken to ensure that the pipeline design diagram(s) is/are clear and carefully organized.* ***Explains in details*** *the pipeline design in addressing the coursework requirements.*

As shown in the pipeline design diagram, when the developer of ITnGreen Organisation pushes and commit his codes to GitHub, Jenkins Pipeline named “operate\_21051265” is triggered as it is configured to use Poll SCM as the Trigger Method of every minute interval.

In the “operate\_21051265” Jenkins’s pipeline, it will go through 21051265 Stage One. It will print out “21051265 Start of Pipeline” in the Console Output. It will then proceed on to 21051265 Stage Two where it performs a delay to wait for a user input from the Console Output to choose an option between Proceed and Abort. Depends on the user input, if the user clicks on Abort, the pipeline is aborted at Stage Two. However, if the user clicks on Proceed, it will proceed to 21051265 Stage Three where it will run 21051265\_script using bolt command from the puppetclient docker container. After running the script, it will proceed to 21051265 Stage Four for testing and print out “Production website tested working”. Finally, it will proceed to 21051265 Stage Five where it will print out “Production website is update successfully” and publish index.html to production.

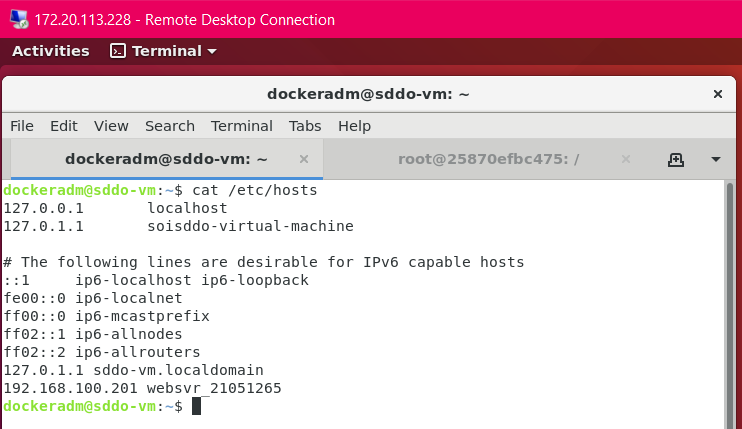
# Pipeline Implementation

***Section Explanation:*** *This section focuses the pipeline implementation. It includes the implementation approach. Care must be taken to ensure that the* ***screenshots*** *of your implementations are clearly and carefully organized.* ***Screenshots serve as evidences for your solution implementation in meeting this coursework assignment.***

To implement the pipeline, I will have to create the following requirements:

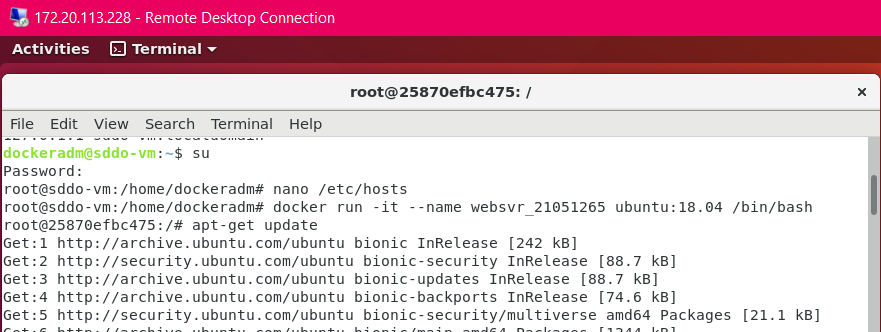
1. Create Docker Container
2. Create a GitHub repository
3. Create a Jenkinsfile
4. Create an index.html file
5. Create a script file
6. Create a Jenkin Pipeline
7. **Create Docker Container**

Update the hosts file to add websvr\_21051265 hostname as puppet client.

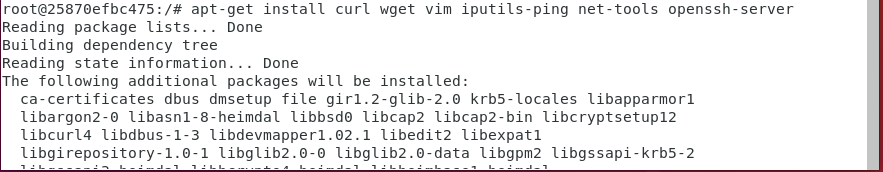


Docker run -it –name websvr\_21051265 ubuntu:18.04 /bin/bash

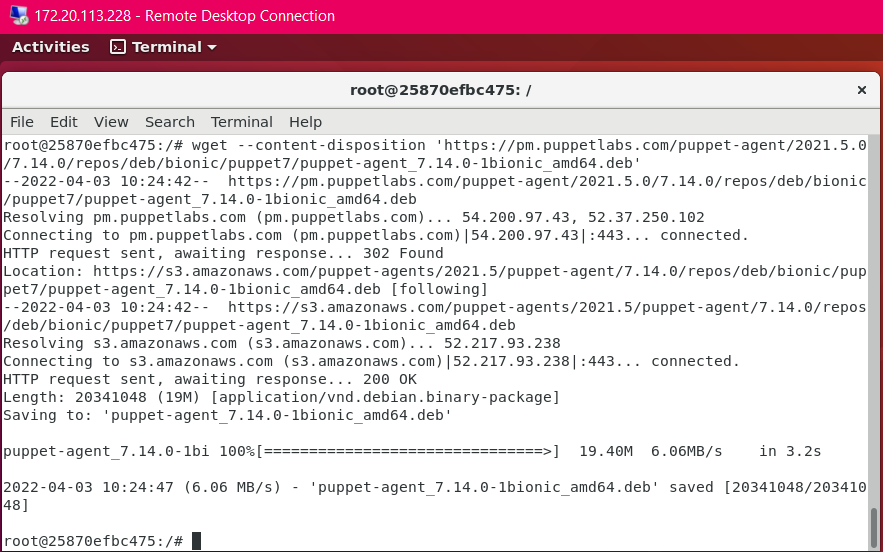
Apt-get update



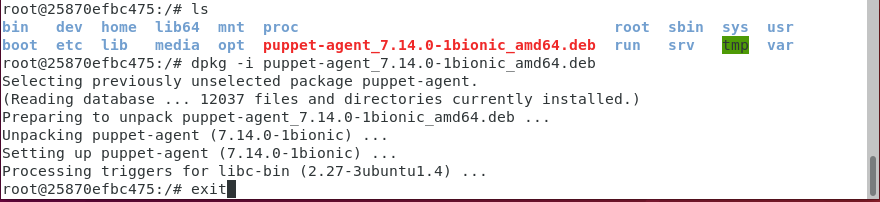
**apt-get install curl wget vim iputils-ping net-tools openssh-server**



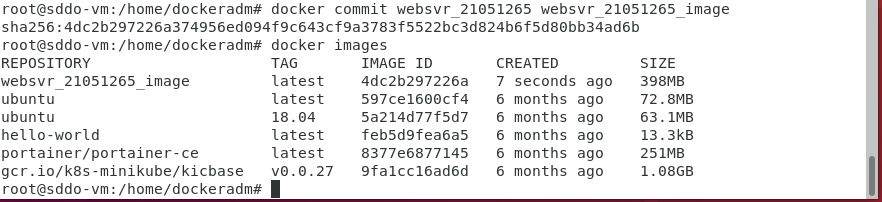
root@puppetclient1: **wget --content-disposition 'https://pm.puppetlabs.com/puppet-agent/2021.5.0/7.14.0/repos/deb/bionic/puppet7/puppet-agent\_7.14.0-1bionic\_amd64.deb'**



root@puppetclient1: **dpkg -i puppet-agent\_7.14.0-1bionic\_amd64.deb**



root@sddo-vm: **docker commit websvr\_21051265 websvr\_21051265\_image**

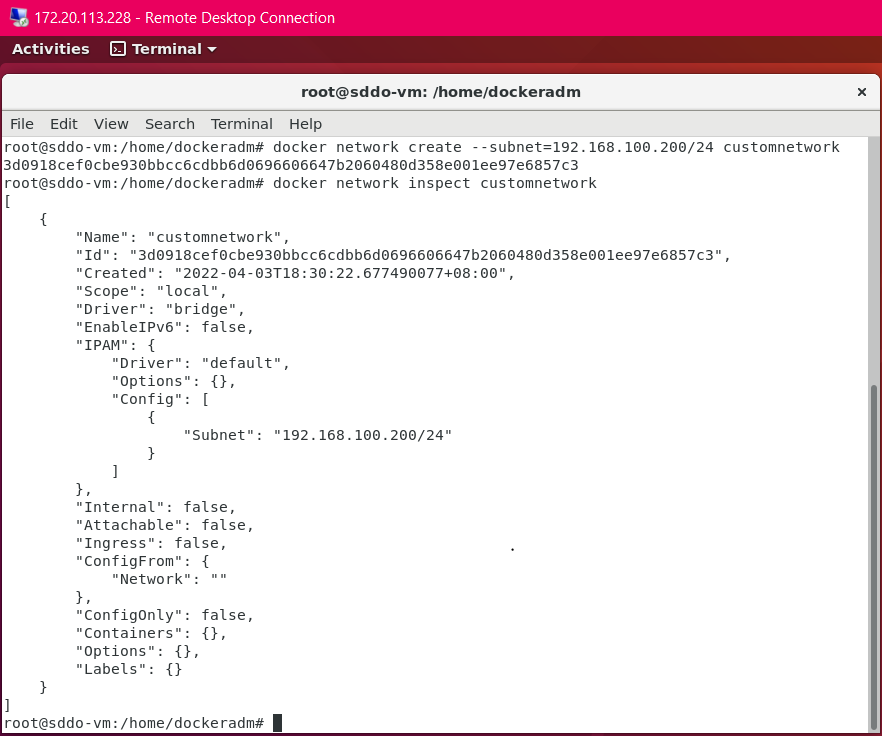


**docker rm websvr\_21051265 to remove existing container.**

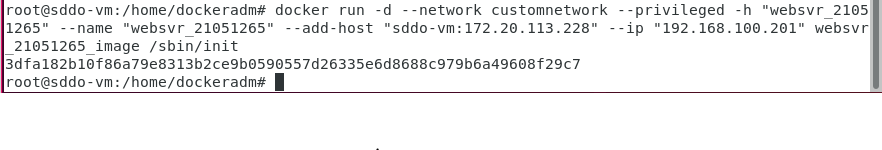


**Create custom network**

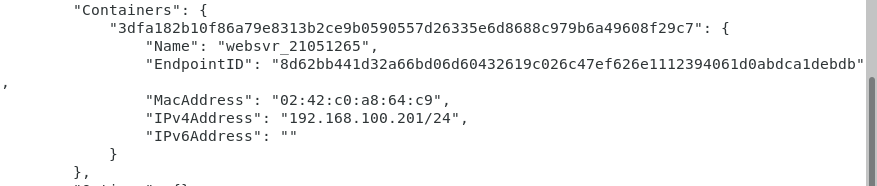
**Docker network create –subnet=192.168.100.200/24 customnetwork**



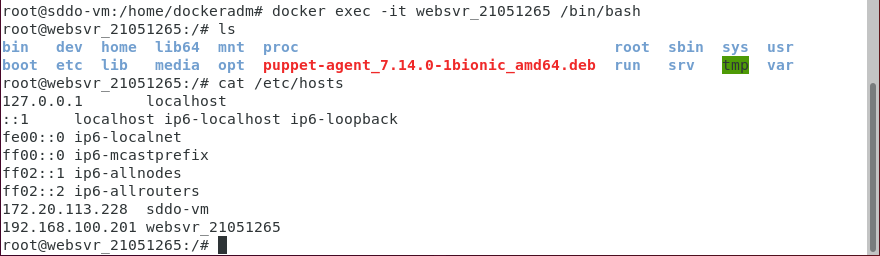
**Docker run -d –network customnetwork –privileged -h “websvr\_21051265” –name websvr\_21051265” –add-host “sddo-vm:172.20.113.” –ip 192.168.100.201” websvr\_21051265\_image /sbin/init**



**Docker network inspect customnetwork**

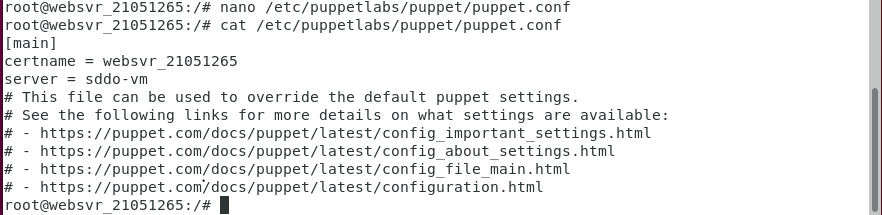


**Docker exec -it websvr\_21051265 /bin/bash**

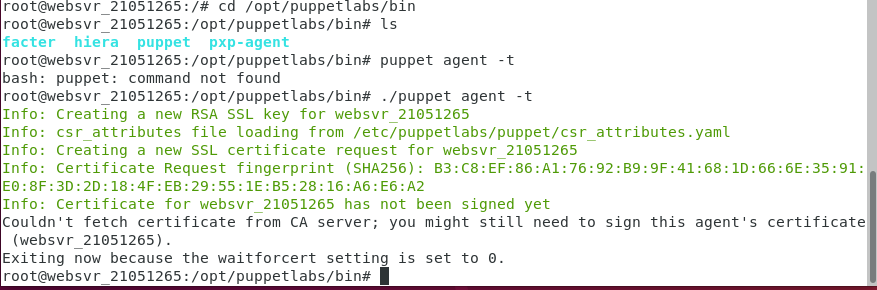


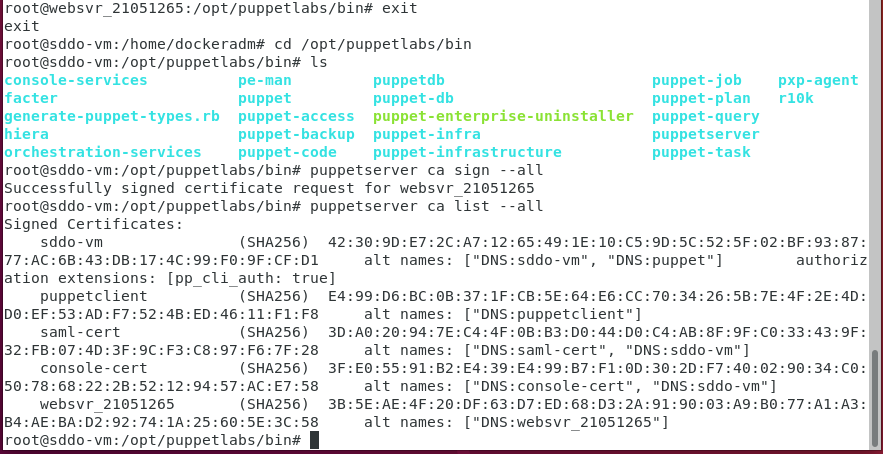
**Vim /etc/puppetlabs/puppet/puppet.conf**

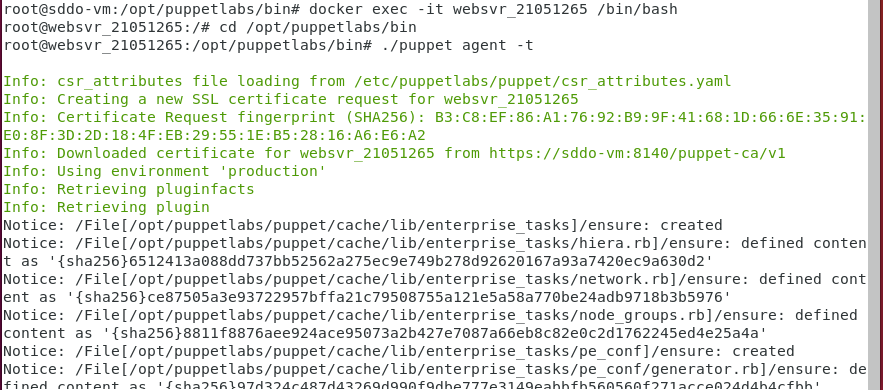
**Configure as Puppet Client:**



Follow lab 6

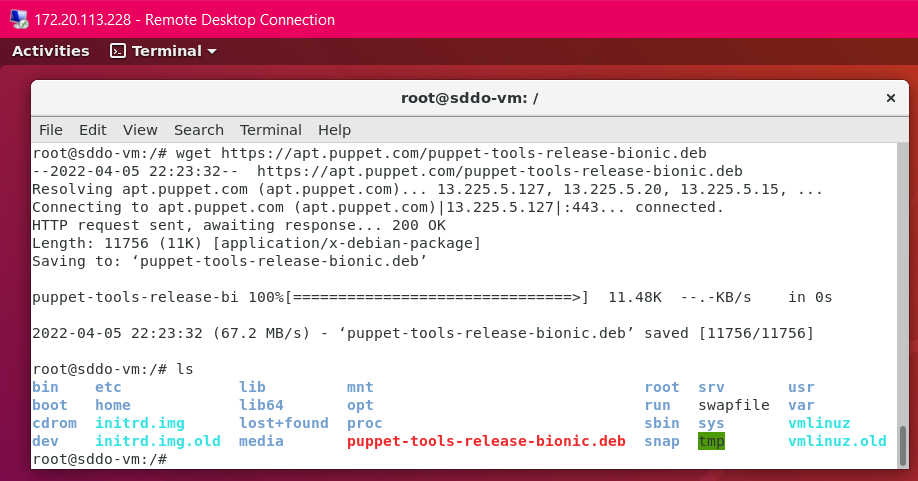


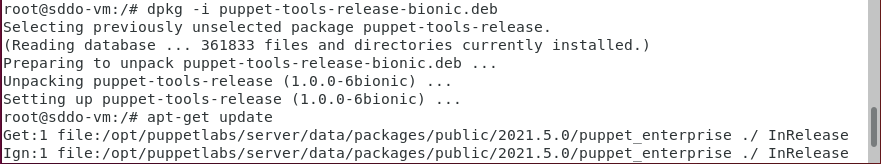


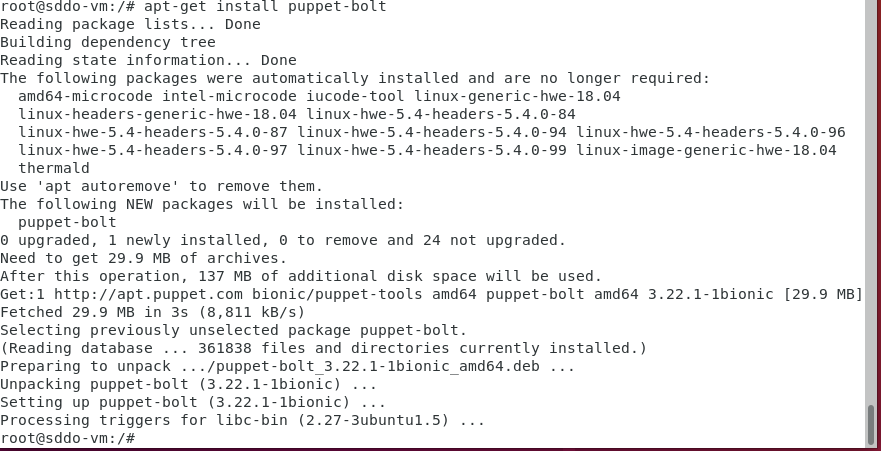


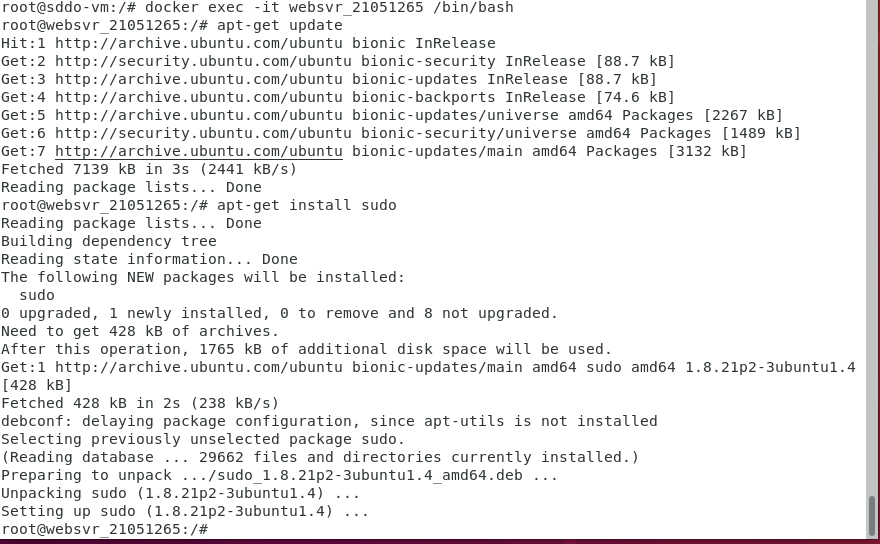
Bolt installation lesson 7

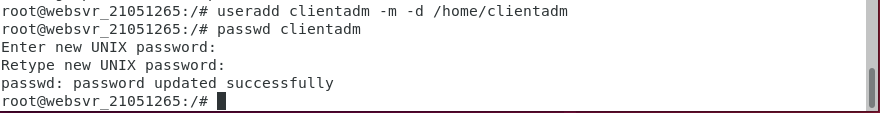
Install puppet bolt.

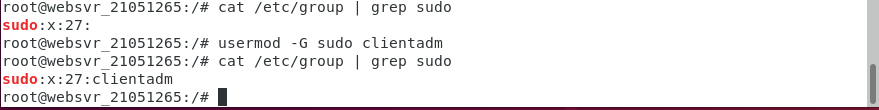










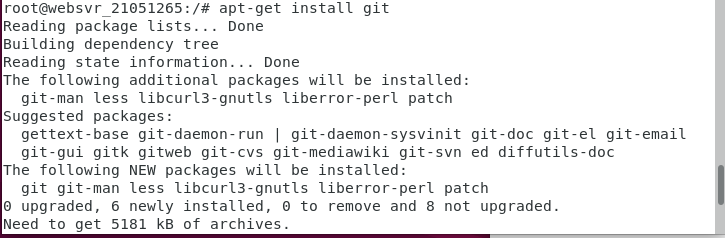


Puppet resource package apache2 ensure=present

Git clone [https://<token>@github.com/username/reponame.git](https://%3ctoken%3e@github.com/username/reponame.git)

Cp -p /var/www/html

/21051265/script\_dir



Sh ‘’’#!/bin/bash

Targets= websvr\_21051265;

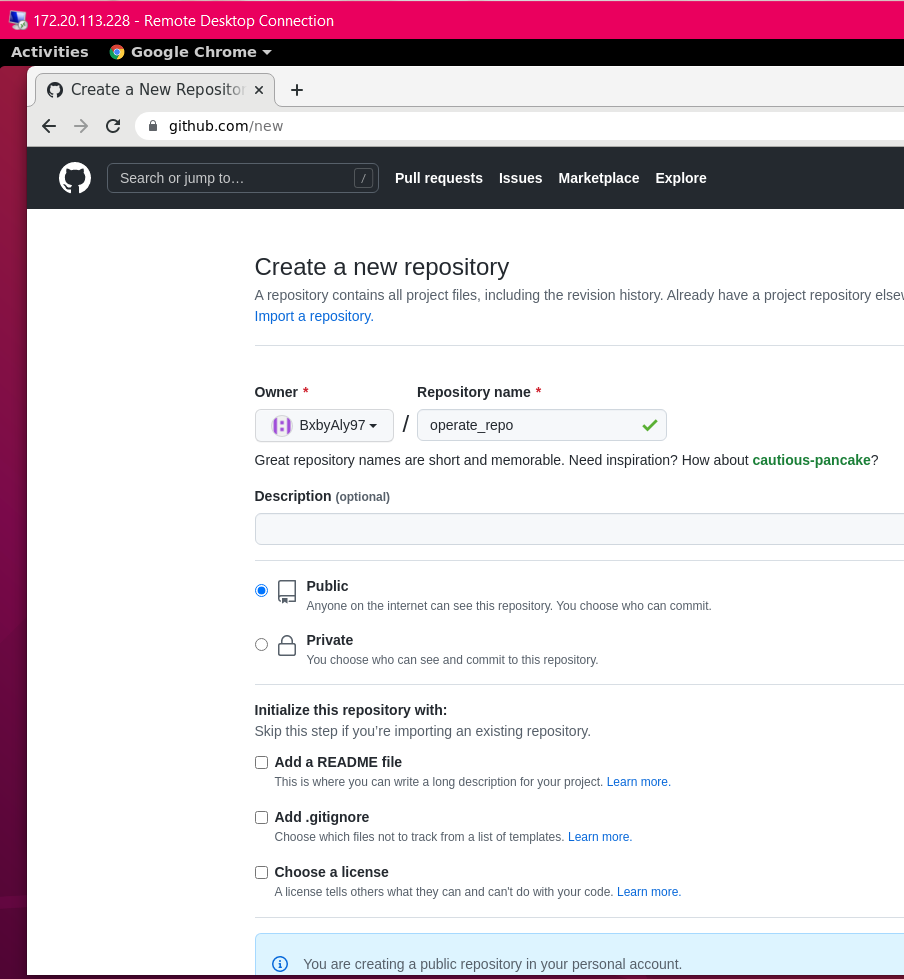
Locate\_script=’/21051265/script\_dir’;

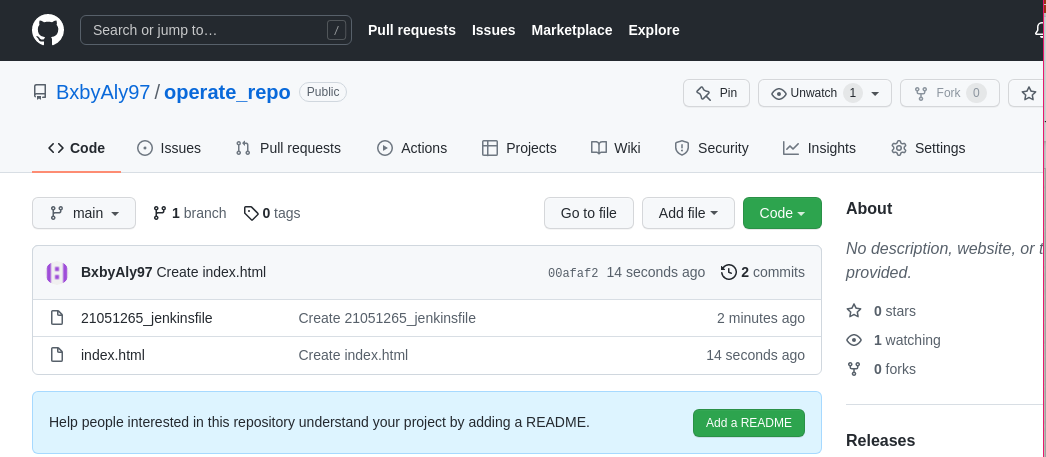
Docker cp $locate\_script $targets://$locate\_script

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

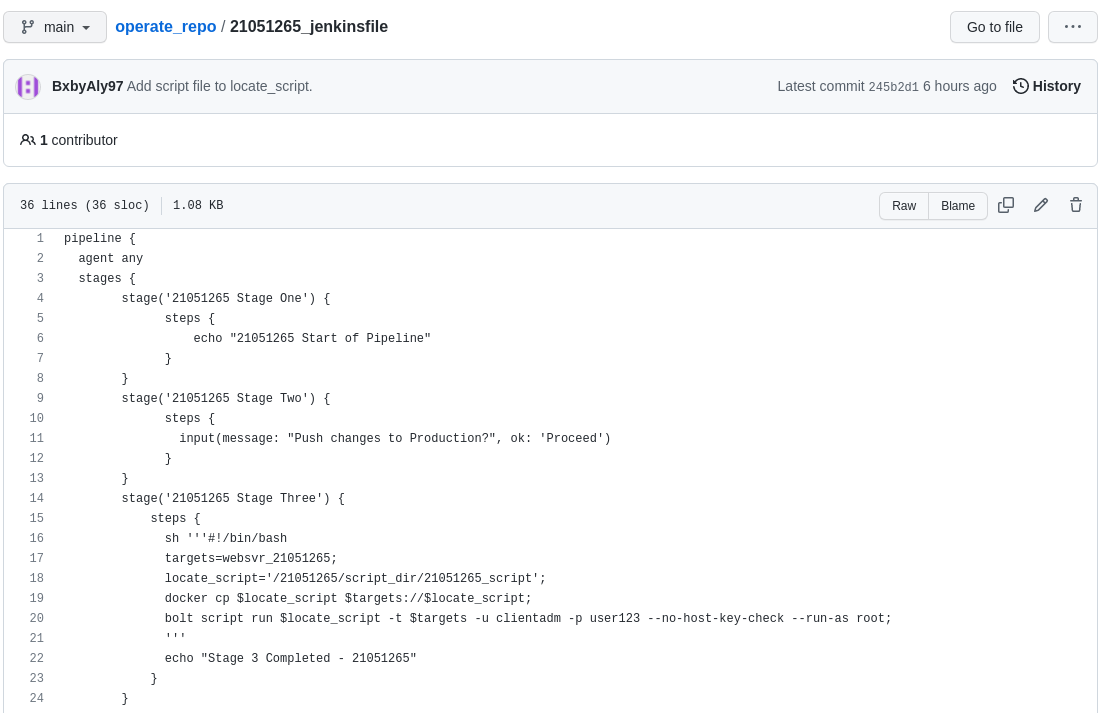
‘’’

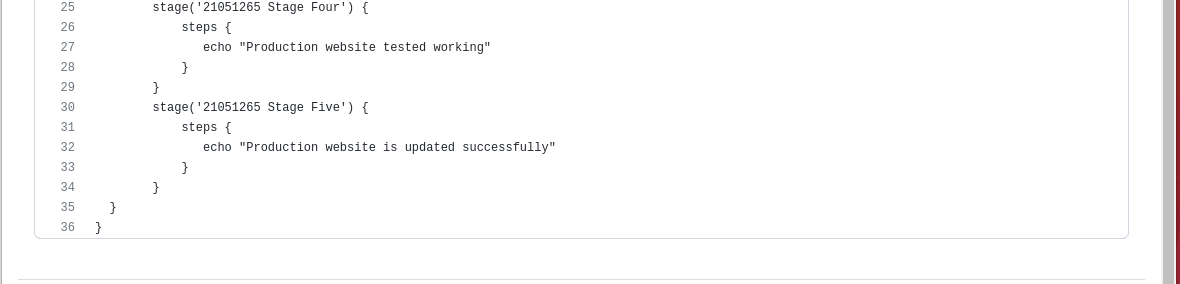
1. **Create GitHub Repository**



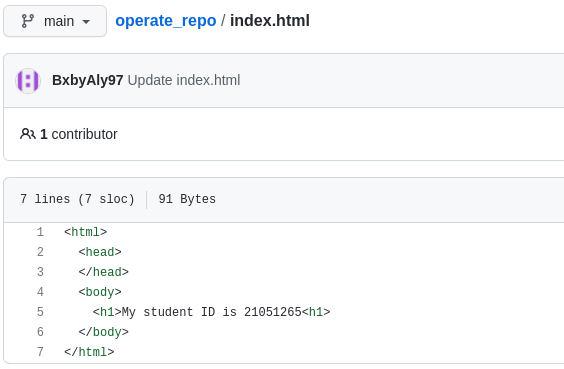


1. **Create a Jenkinsfile**

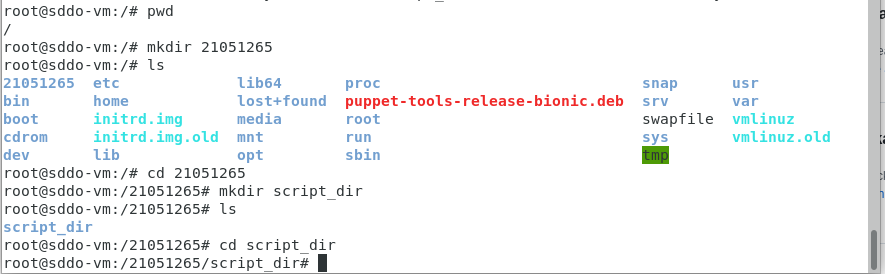


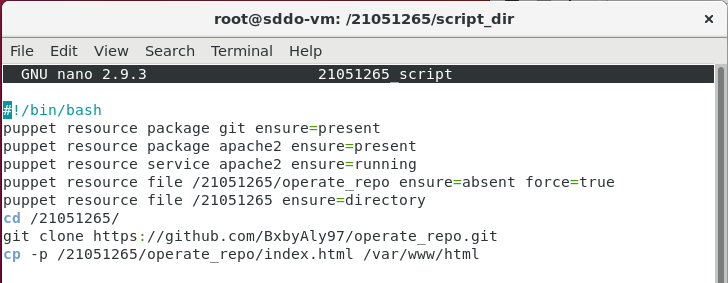


1. **Create an index.html file**

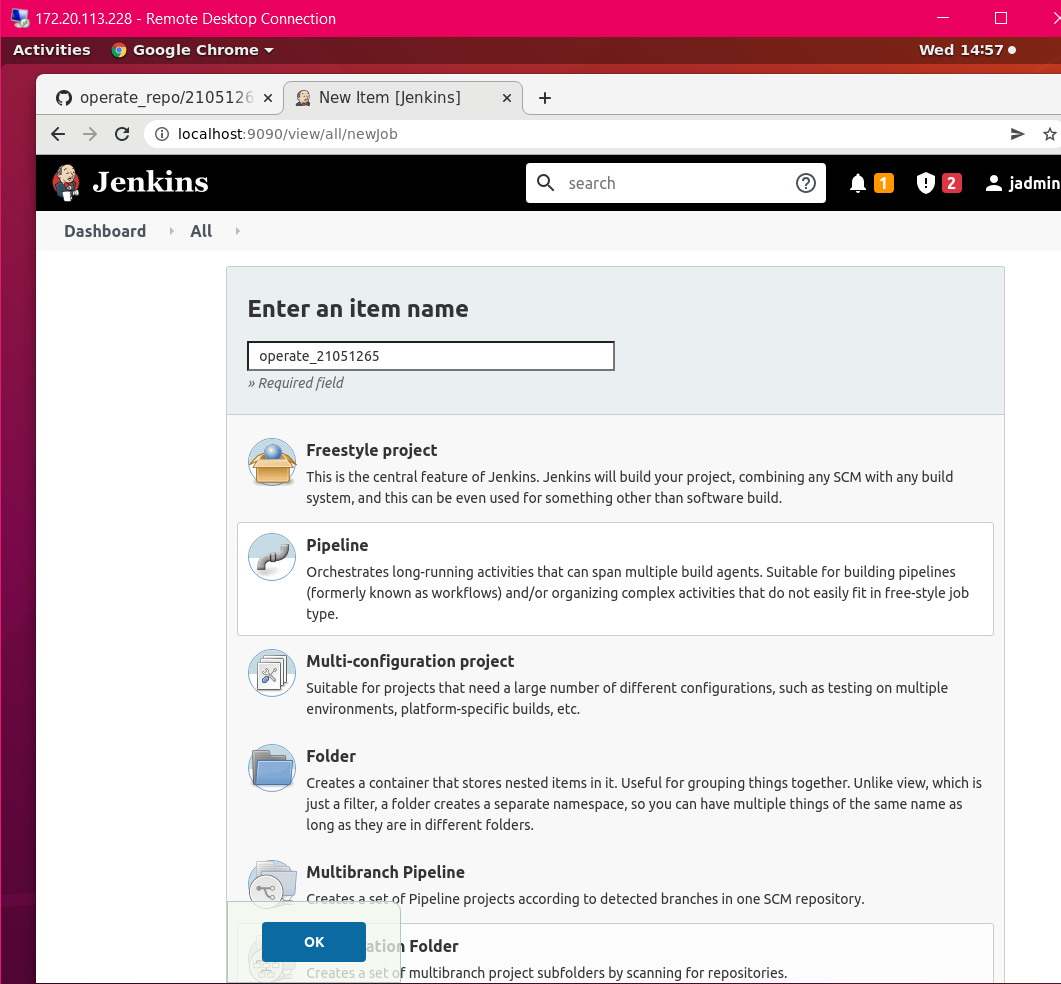


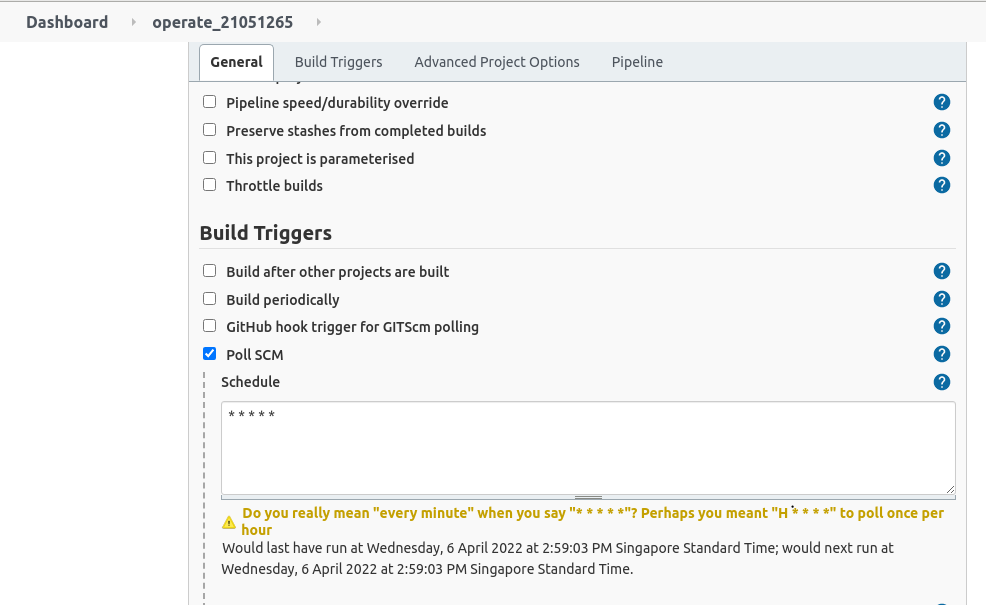
1. **Create a script file**

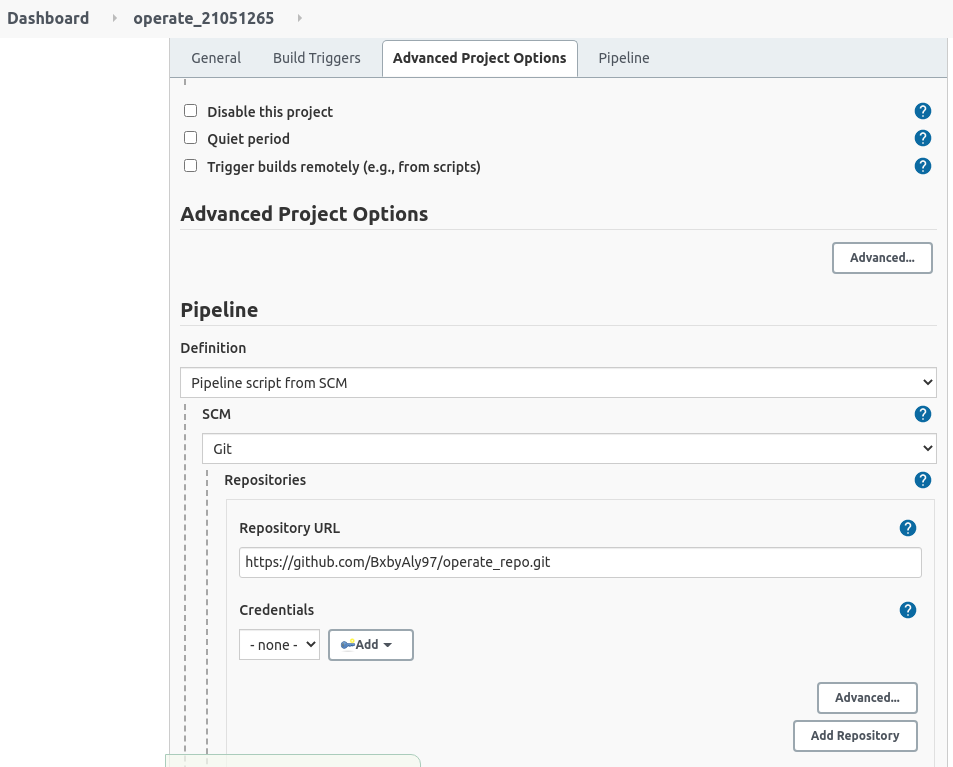


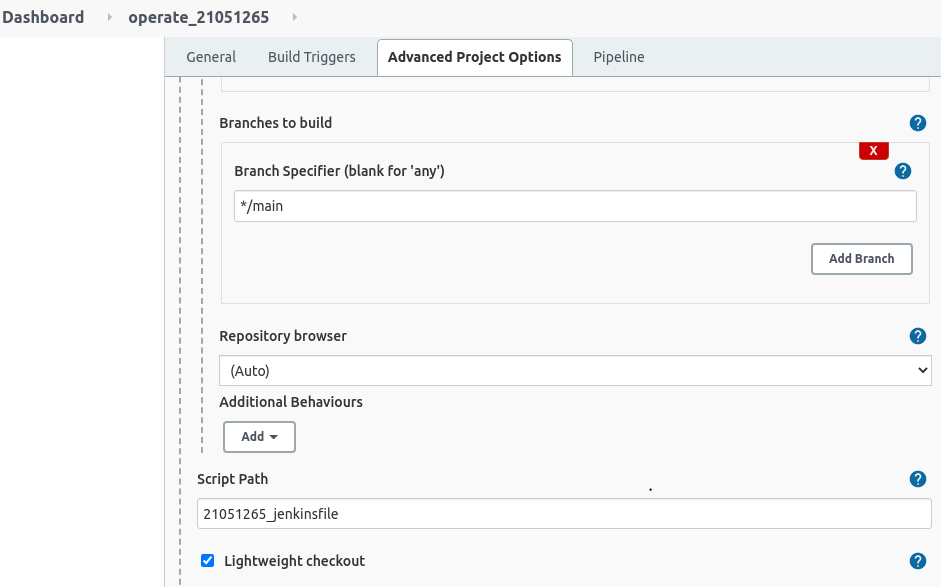


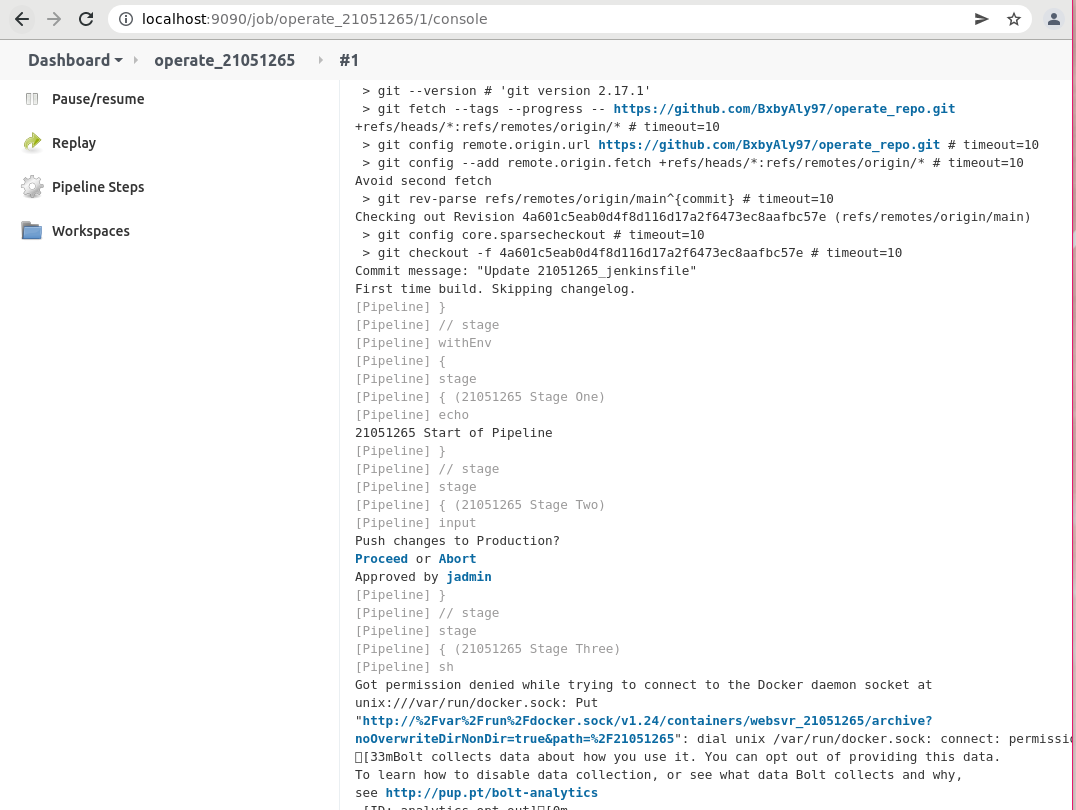
1. **Create a Jenkin Pipeline**

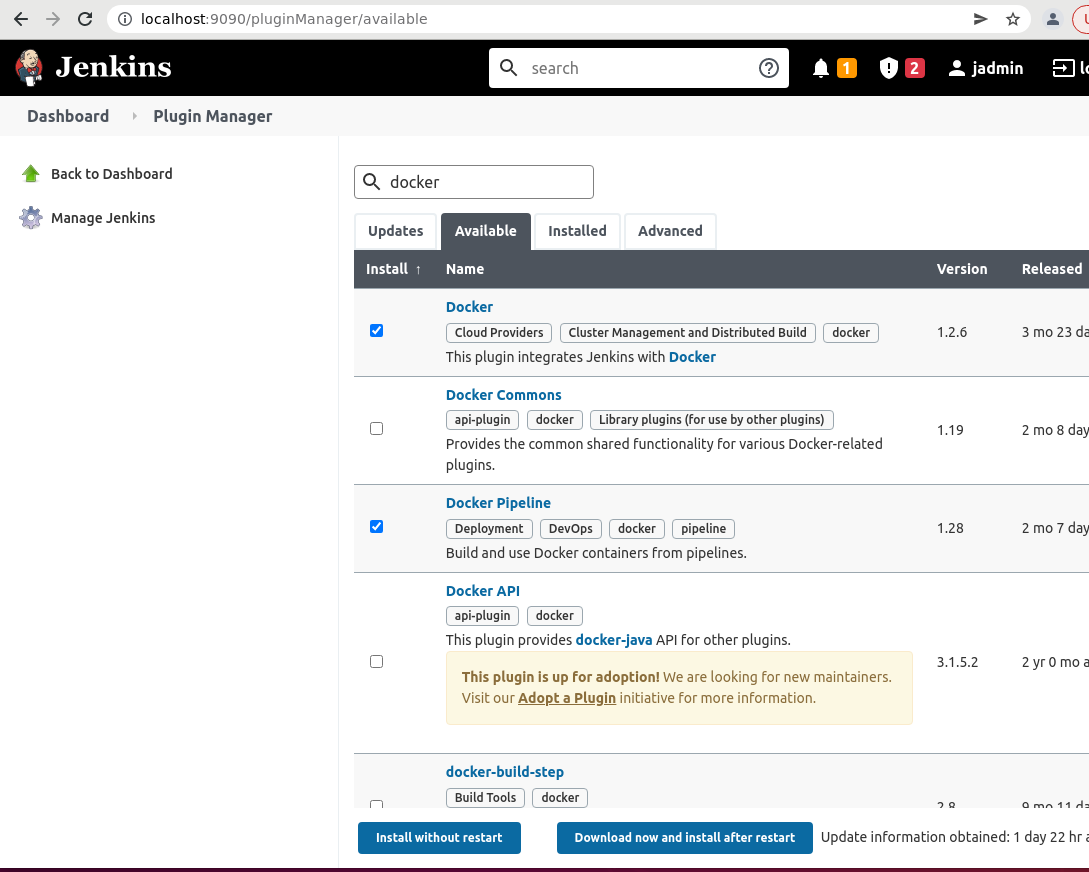




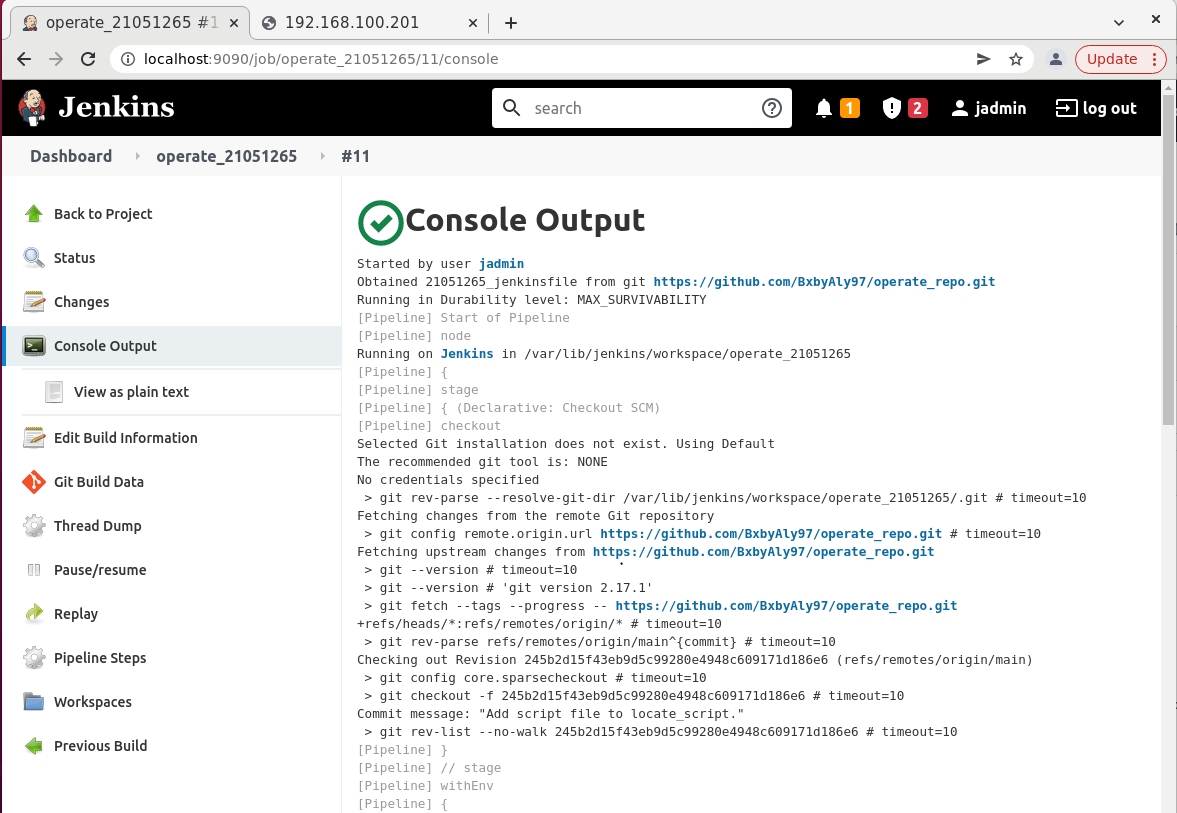


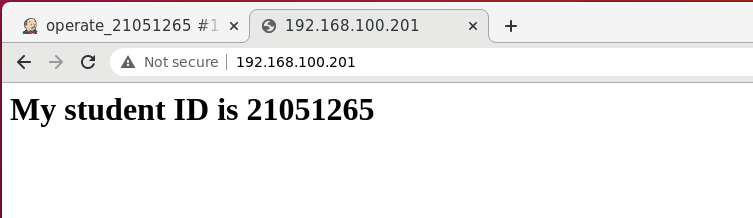












# Monitor

**Answer ALL questions in this section. (60 Marks)**

**Q 4.1: Name the SIX (6) components of DevOps Monitoring, and provide a one-sentence description for each component. (12m)**



|  |
| --- |
| The Six (6) components of DevOps Monitoring are Application Performance Monitoring, Network Performance Monitoring, Infrastructure Monitoring, Database Performance Monitoring, API Monitoring and Quality Assurance Monitoring.  Application Performance Monitoring (APM) is the process of monitoring the backend architecture of an application to resolve performance issues and bottlenecks on time.  Network (N/W) Performance Monitoring is the practice of consistently checking a network for deficiencies or failures to ensure continued network performance that includes monitoring network components such as servers, routers, firewalls, etc.  Infrastructure monitoring is to verifies the availability of IT infrastructure components in a data center or cloud infrastructure (IaaS) which involves monitoring the resources, their availability, checking under-utilized and over-utilized resources to optimize the IT infrastructure and operational cost that associated with it.  Database Performance Monitoring is to track performance, security, backup, file growth of the database which mainly to examine how a database server performs.  API Monitoring is the practice of examining applications’ APIs in a production environment that gives visibility of performance, availability, and functional correctness of APIs.  Quality Assurance (QA) Monitoring ensures that all processes, procedures, and standards of the application development are in compliance by writing test cases. |

**Q 4.2: Nagios can perform Active or Passive Monitoring of enterprise IT infrastructure using SNMP (Simple Network Management Protocol).**

1. Explain how we can use SNMP Traps for Passive Monitoring. **(4m)**

|  |
| --- |
|  |

1. Explain how does Nagios perform Active Monitoring using SNMP. **(4m)**

|  |
| --- |
|  |

1. What are the security concerns with the use of SNMP for Nagios monitoring? **(8m)**

|  |
| --- |
| SNMP v1 and v2 are not secured as it uses one authentication to access devices and the authentication information is being send in plaintext across the network which is susceptible to sensitive data leakage via Man-In-The-Middle (MITM) attacks. |

1. When using SNMP for Nagios monitoring, what are the recommended best practices to address the security concerns mentioned above in c)? **(4m)**

|  |
| --- |
| Use SNMPv3 which includes encryption, configure as read-only to accept connections from specific hosts that uses non-default community string names and physically or virtually segregate network as an isolated subnet for SNMP monitoring traffic only. |

**Q 4.3: Traditional Monitoring Systems face two main security issues that can be exploited by malicious users.**

1. What is the main security issue of Monitoring Systems that is addressed by the SSH protocol? **(2m)**

|  |
| --- |
| The main security issue is unencrypted channels and privileged root access at the clients. |

1. What features of the SSH protocol can we use to address the security issue mentioned above in a)? **(2m)**

|  |
| --- |
| Use off-by-default policy. |

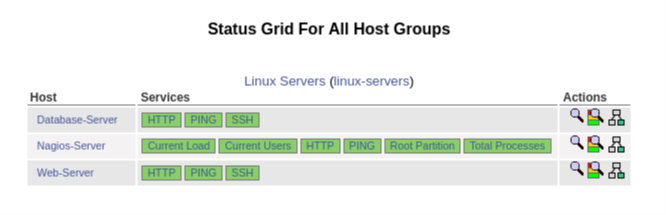
1. Does a successful execution of the "*ssh root@apacheweb.localdomain*" command address the rootkit security issue? If yes, how does it address? If no, what improvement can you suggest? **(4m)**

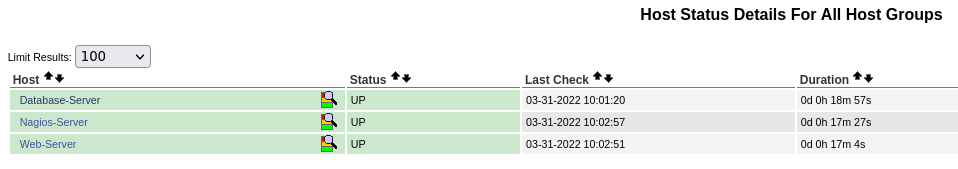
|  |
| --- |
| Yes, it addresses by prompting a password. If the password entered correctly, an ECDSA key fingerprint is added and successfully connected to the remote server. |

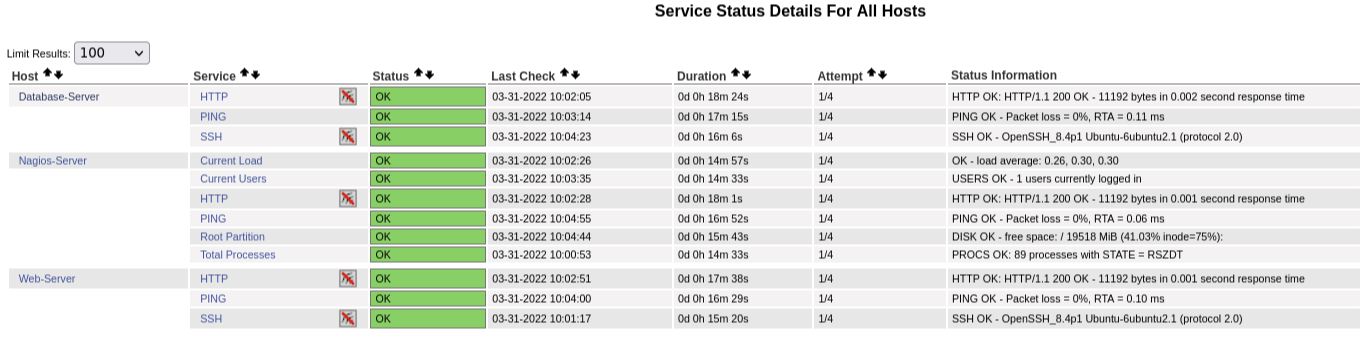
1. Explain with a diagram the complete SSH configuration required for secured Nagios remote monitoring that addresses the two main security issues. You have to indicate which system is each command executed at. **(8m)**

|  |
| --- |
|  |

**Q 4.4: Refer to these screenshots of the Nagios web interface.**







1. Paste the complete "*localhost.cfg*" file contents here, that will produce the above **EXACT** screenshots. Remove all redundant configurations in the file. **(6m)**

|  |
| --- |
| define host {  use linux-server  host\_name Nagios-Server  alias Nagios-Server  address 127.0.0.1  }  define host {  use linux-server  host\_name Database-Server  alias Database-Server  address 192.168.100.202  }  define host {  use linux-server  host\_name Web-Server  alias Web-Server  address 192.168.100.201  }    define hostgroup {  hostgroup\_name linux-servers  alias Linux Servers  members Nagios-Server,Database-Server,Web-Server  }  define service {  use local-service  host\_name Nagios-Server  service\_description PING  check\_command check\_ping!100.0,20%!500.0,60%  }  define service {  use local-service  host\_name Nagios-Server  service\_description PING  check\_command check\_ping!100.0,20%!500.0,60%  }  define service {  use generic-service  host\_name Web-Server  service\_description PING  check\_command check\_ping!100.0,20%!500.0,60%  }  define service {  use generic-service  host\_name Database-Server  service\_description PING  check\_command check\_ping!100.0,20%!500.0,60%  }  define service {  use local-service  host\_name Nagios-Server  service\_description Root Partition  check\_command check\_local\_disk!20%!10%!/  }  define service {  use local-service  host\_name Nagios-Server  service\_description Current Users  check\_command check\_local\_users!20!50  }  define service {  use local-service  host\_name Nagios-Server  service\_description Total Processes  check\_command check\_local\_procs!250!400!RSZDT  }  define service {  use local-service  host\_name Nagios-Server  service\_description Current Load  check\_command check\_local\_load!5.0,4.0,3.0!10.0,6.0,4.0  }  define service {  use generic-service  host\_name Web-Server  service\_description SSH  check\_command check\_ssh  notifications\_enabled 0  }  define service {  use generic-service  host\_name Database-Server  service\_description SSH  check\_command check\_ssh  notifications\_enabled 0  }  define service {  use local-service  host\_name Nagios-Server  service\_description HTTP  check\_command check\_http  notifications\_enabled 0  }  define service {  use generic-service  host\_name Web-Server  service\_description HTTP  check\_command check\_http  notifications\_enabled 0  }  define service {  use generic-service  host\_name Database-Server  service\_description HTTP  check\_command check\_http  notifications\_enabled 0  } |

1. Change the Web-Server and Database-Server names to your name following this format:

Web-Server name (e.g. *John-Web*): Siti Nur Alyshyia Binte Hashim-Web

Database-Server name (e.g. *John-Database*): Siti Nur Alyshyia Binte Hashim-Database

Make the corresponding configuration changes in "*localhost.cfg*", and provide your own new screenshots here. **(6m)**

|  |
| --- |
|  |

# References

***Section Explanation:*** *This section is for you to acknowledge materials that you refer or copy.*

* **Acknowledge** any work, statement or definitions copied
* https://insights.daffodilsw.com/blog/introducing-devops-monitoring-and-its-6-significant-components

# Appendices

***Section Explanation:*** *This section is for you to include any materials which you may not put in the main body of your report.*

*Appendices should be organised for ease of readability etc.*