**Certificate Authority Module**

**Lab 2 The Hierarchy of Certificate Authority**

[Introduction 2](#_Toc476560058)

[What‘s the hierarchy of certificate authority? 2](#_Toc476560059)

[The species of CA 2](#_Toc476560060)

[Implementation 3](#_Toc476560061)

[Network topology on GENI 3](#_Toc476560062)

[Enable web server function 4](#_Toc476560063)

[Install LAMP 4](#_Toc476560064)

[Enable SSL connection 5](#_Toc476560065)

[Build our hierarchy of certificate authority 6](#_Toc476560066)

[Run and open a browser on GENI 16](#_Toc476560067)

[For Windows operation system: 16](#_Toc476560068)

[For MacOS operation system 19](#_Toc476560069)

[For other Linux operation system 21](#_Toc476560070)

[View results 22](#_Toc476560071)

# Introduction

## What‘s the hierarchy of certificate authority?

Certificate Authority (CA) is verified using a chain of trust. The trust anchor for the digital certificate is the Root Certificate Authority (CA), and any Certificate Authority (CA) which comes under Root Certificate Authority (Root CA) is known as a subordinate Certificate Authority (CA). Their structure is like the tree’s root; the trust is like the oxygen passed from the top of the root to the tip, where is the internet entity needed to be verified.

## The species of CA

There are three kinds of the certificate authority. The root certificate authority (CA), the intermediate certificate authority (CA) and issuing certificate authority (CA).

Root certificate authority (CA) is the topmost Certificate Authority (CA) in a Certificate Authority (CA) hierarchy. Each Certificate Authority (CA) hierarchy begins with the Root CA

An intermediate Certificate Authority (CA) is a CA that is subordinate to another CA (Root CA or another intermediate CA) and issues certificates to other CAs in the CA hierarchy.

Issuing CAs are used to provide certificates to users.

The following picture can explain the relationships of these three kinds of the certificate authority (CA).

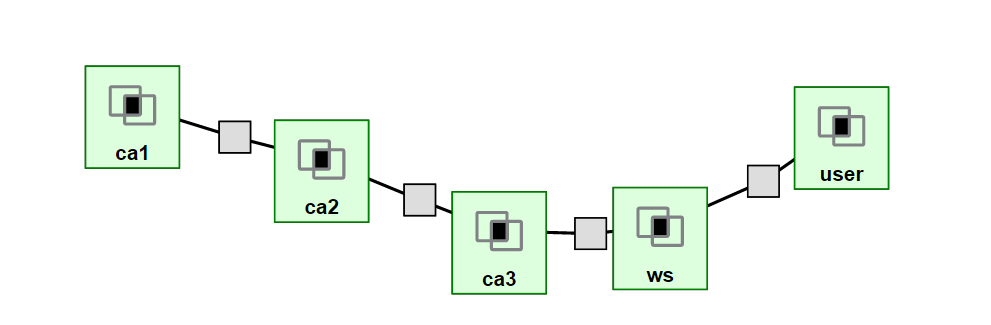
# Implementation

## Network topology on GENI

The network topology of our experiment will be the following one:

1. The node named CA1 will be the root CA1 in this experiment.
2. The node named CA2 is the child certificate authority of the CA1; it’s the intermediate Certificate Authority (CA).
3. The node named CA3 is the child certificate authority of the CA2; it’s the Issuing Certificate Authority (CA).
4. The node named WS will be the web server in this experiment, and we will install LAMP on this node to enable it to be a web server.

Notice: we must wait for all the GENI node turn green, which means the remote machines are ready for us to use, then we can continue the following steps. The above process may take a while.



## Enable web server function

In this part, we need to configure the WS node to let it able to play the role as a web server.

### Install LAMP

Follow the following commands to install the Apache, MySQL, and PHP:

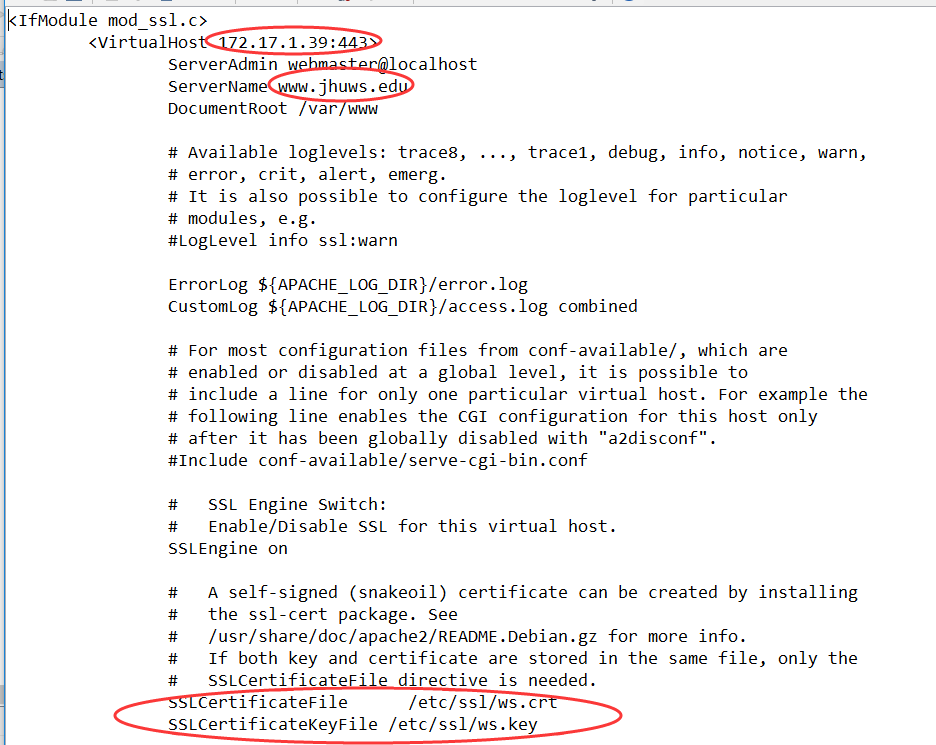
* “sudo su”
* “apt-get update”
* “sudo apt-get install mysql-server”
* “sudo apt-get install apache2”
* “sudo apt-get install php5 libapache2-mod-php5”
* “sudo apt-get install php5-mysql php5-curl php5-gd php5-intl php-pear php5-imagick php5-imap php5-mcrypt php5-memcache php5-ming php5-ps php5-pspell php5-recode php5-snmp php5-sqlite php5-tidy php5-xmlrpc php5-xsl;”

### Enable SSL connection

Use the following command to configure the configuration file of Apache.

* “sudo cp /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-enabled/default-ssl.conf”

And then edit this defaulr-ssl.conf file, which is in the sites-enable folder. Modify these locations:



You can choose any server name as you like, as well as the SSLCertificateFile and SSLCertficateKeyFile. But you should do the following configuration accord these setting you have done in this step.

The “172.17.1.19” is the IP address of this web server, which is the WS node in this topology.

You should also use this command” sudo a2enmod ssl” to enable the SSL module of Apache2 if there prompt the problem that you try to connect 443 port to our web server while it refuses.

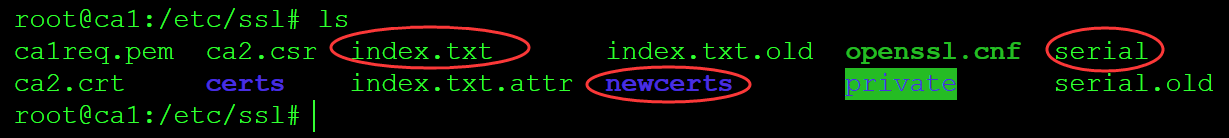
Then restart the apache2 service using this command”service apache2 restart”.

## Build our hierarchy of certificate authority

Firstly, we need to build the root certificate authority (CA), which is the ca1 node in our topology.

Login on to the ca1 node, go into the “/etc/ssl” folder as the root identity, and enter these commands to set up the basic folder of OpenSSL:

* “touch /etc/ssl/index.txt”
* “echo 01 > /etc/ssl/serial”
* “mkdir newcerts”



Then we need to generate the key for the root certificate authority (CA).

Use these commands to generate a key.

* “openssl genrsa -des3 -out /etc/ssl/private/ca1key.pem 2048”

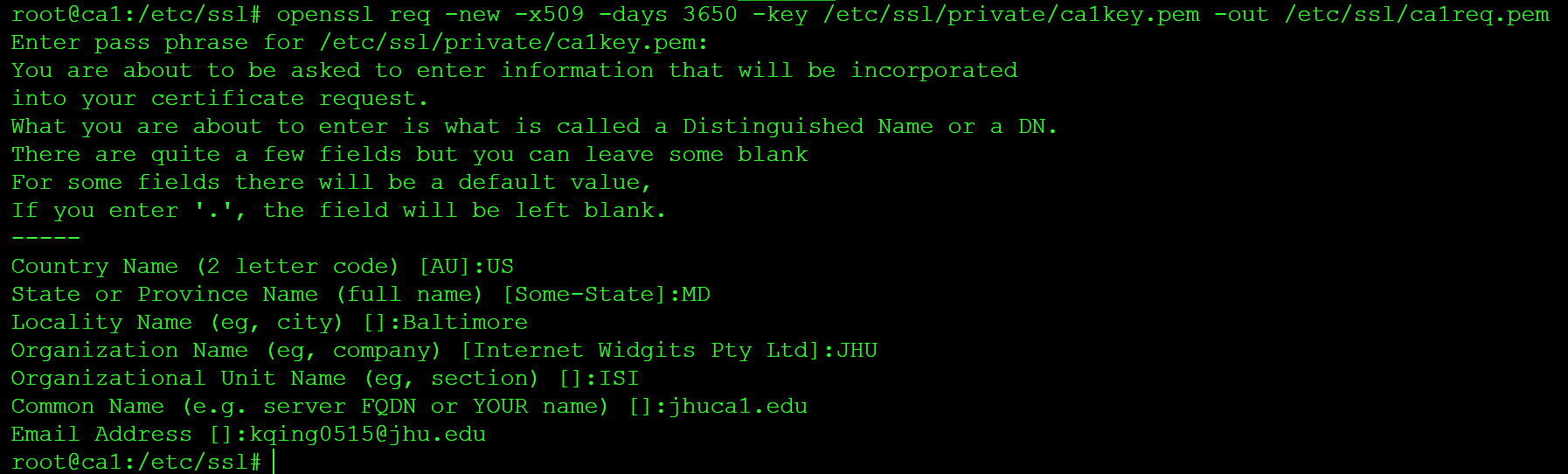
It will require you to enter a password, set up a password in this step and remember this password.

Then we need to sign the certificate for itself.

Use this command to sign the certificate for itself.

* “openssl req -new -x509 -days 3650 -key /etc/ssl/private/ca1key.pem -out /etc/ssl/ca1req.pem”

We need to enter the basic information when we sign the certificate; this information will be included in the digital certificate we signed.

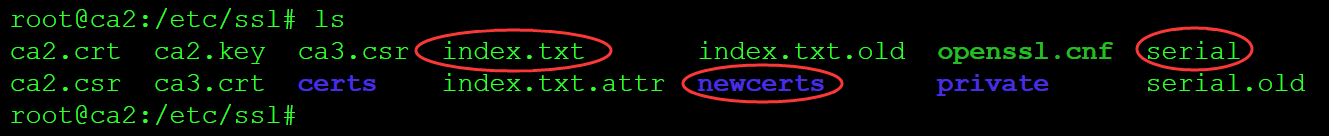


Then we move to ca2 node, which is the intermediate certificate authority (CA) in our topology.

Login to ca2 node; go into “/etc/ssl” folder as the root identity.

Use these commands to create basic folders for the OpenSSL.

* “touch /etc/ssl/index.txt”
* “echo 01 > /etc/ssl/serial”
* “mkdir newcerts”



Then we need to generate a key for intermediate certificate authority (CA), use these commands to generate a key.

* “openssl genrsa -des3 -out /etc/ssl/ca2.key 2048”
* “openssl rsa -in /etc/ssl/ca2.key -out /etc/ssl/ca2.key”

Then we need to create a request document for intermediate certificate authority (CA). Use this command to create this file.

* “openssl req -new -days 3650 -key /etc/ssl/ca2.key -out /etc/ssl/ca2.csr”

This step will create a file with extension “.csr, which is the request document.

Then we move to the issuing certificate authority (CA), which is the ca3 node in our topology.

In the same way, we need to create a basic folder for OpenSSL.

* “touch /etc/ssl/index.txt”
* “echo 01 > /etc/ssl/serial”
* “mkdir newcerts”

Then we need to generate the key for the issuing certificate authority (CA). Use these commands to generate the key.

* “openssl genrsa -des3 -out /etc/ssl/ca3.key 2048”
* “openssl rsa -in /etc/ssl/ca3.key -out /etc/ssl/ca3.key”

Then we also need to generate a request file for the issuing certificate authority (CA). Use this command to generate it.

* “openssl req -new -days 3650 -key /etc/ssl/ca3.key -out /etc/ssl/ca3.csr”

After having done the above, we need to generate the request document of the web server. We need to move to the web server, which is the ‘ws’ node in our topology.

Create the same folder for the openssl using these commands.

* “touch /etc/ssl/index.txt”
* “echo 01 > /etc/ssl/serial”

And then generate the key for the web server. Using these commands.

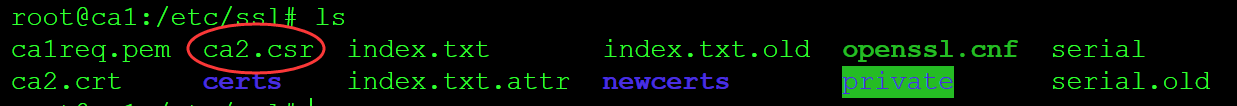
* “openssl genrsa -des3 -out /etc/ssl/ws.key 2048”
* “openssl rsa -in /etc/ssl/ws.key -out /etc/ssl/ws.key”

Generate the request document for the web server using this command.

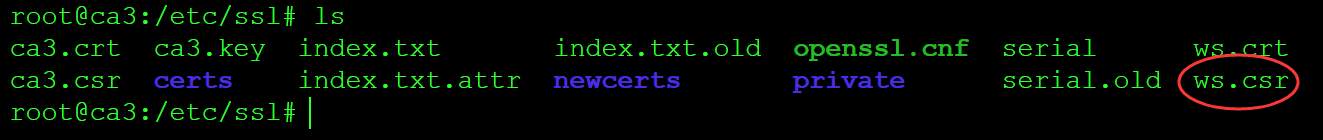
* “openssl req -new -days 3650 -key /etc/ssl/ws.key -out /etc/ssl/ws.csr”

Then we have the request document for the intermediate certificate authority (CA) on ca2 node, the request document for issuing certificate authority (CA) on ca3 node, the request document for the web server on ws node.

We need to move the request document for intermediate authority (CA) to ca1 node, which is the root certificate authority (CA), move the request document for issuing certificate authority (CA) to intermediate certificate authority (CA), and move the request document for web server to issuing certificate authority (CA).







You can use the WinSCP to make this movement or use the SCP command. If you confront with the problem of privilege, please feel free to give out the privilege using the command “chmod”.

The instruction of how to log onto GENI node using WinSCP can be found in this link:

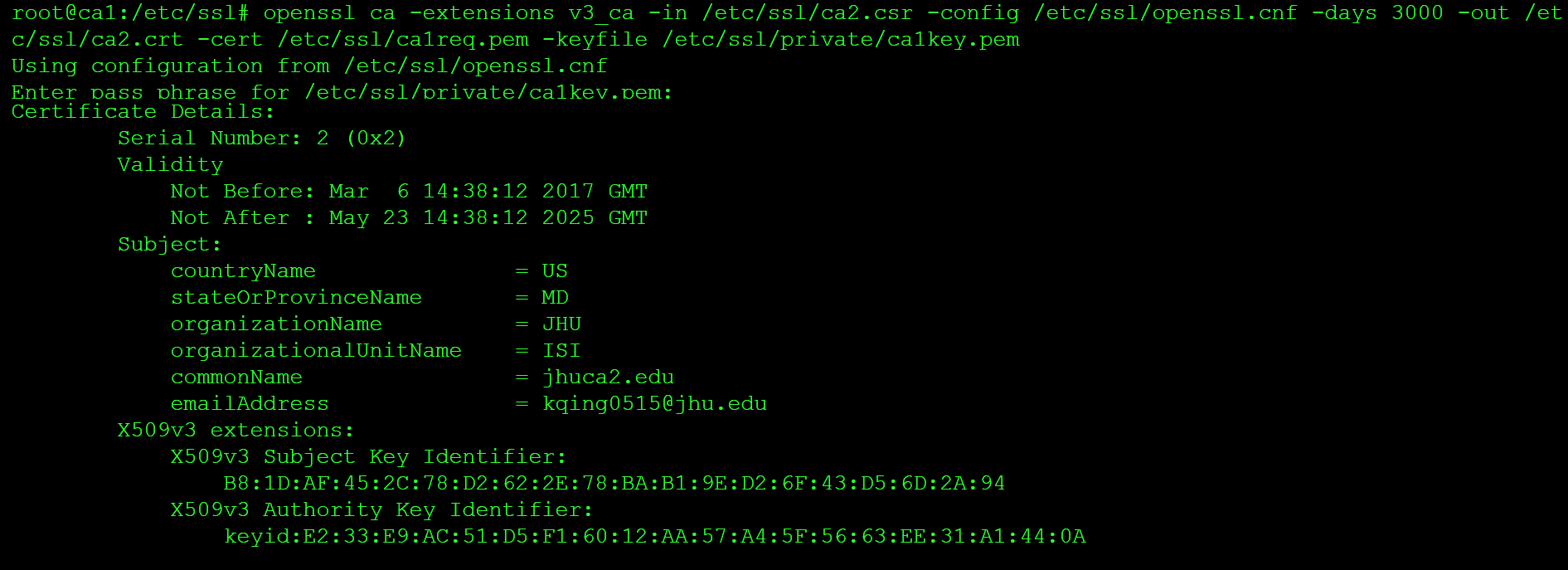
<http://mountrouidoux.people.cofc.edu/CyberPaths/winscp.html>

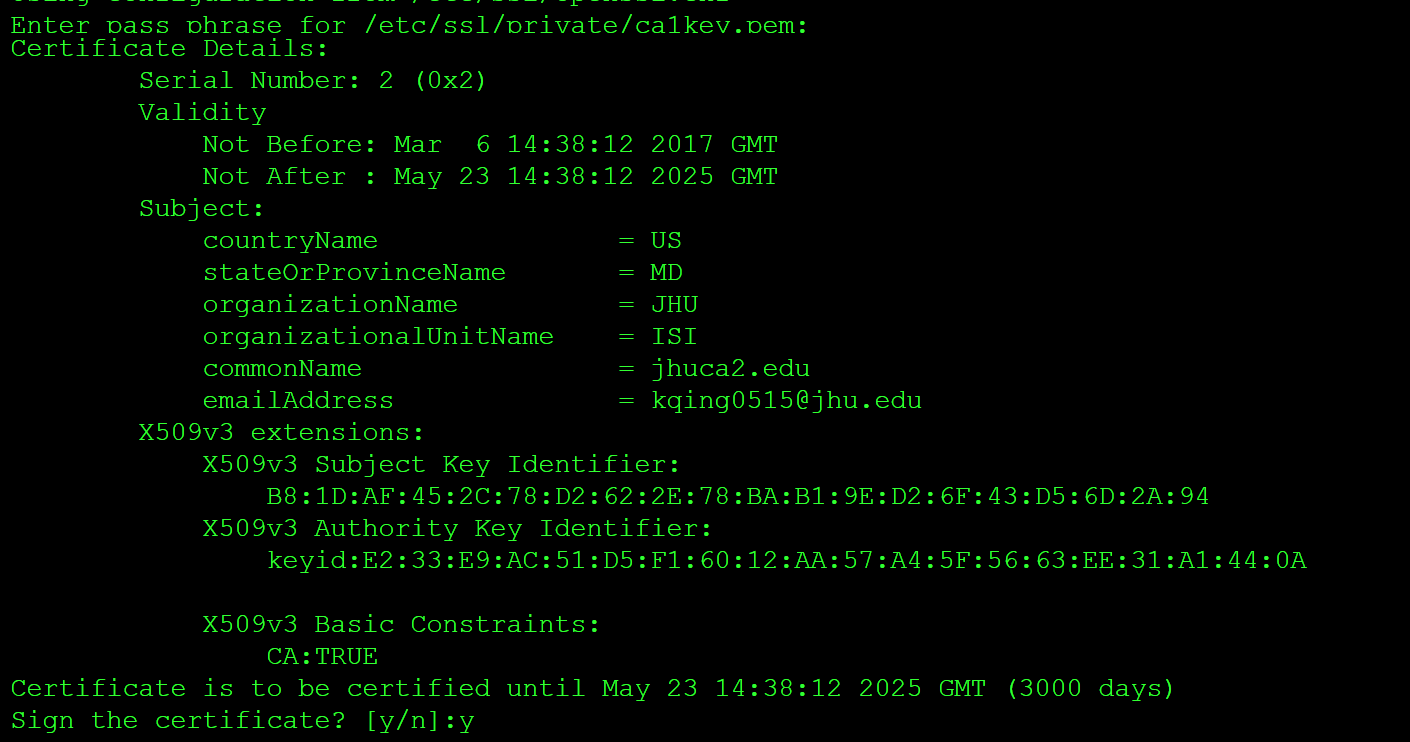
The instruction of how to use SCP command on Ubuntu to transfer the files can be found through the link below:

<https://help.ubuntu.com/community/SSH/TransferFiles>

Then on ca1 node, use this command to sign the digital certificate for the intermediate certificate authority (CA).

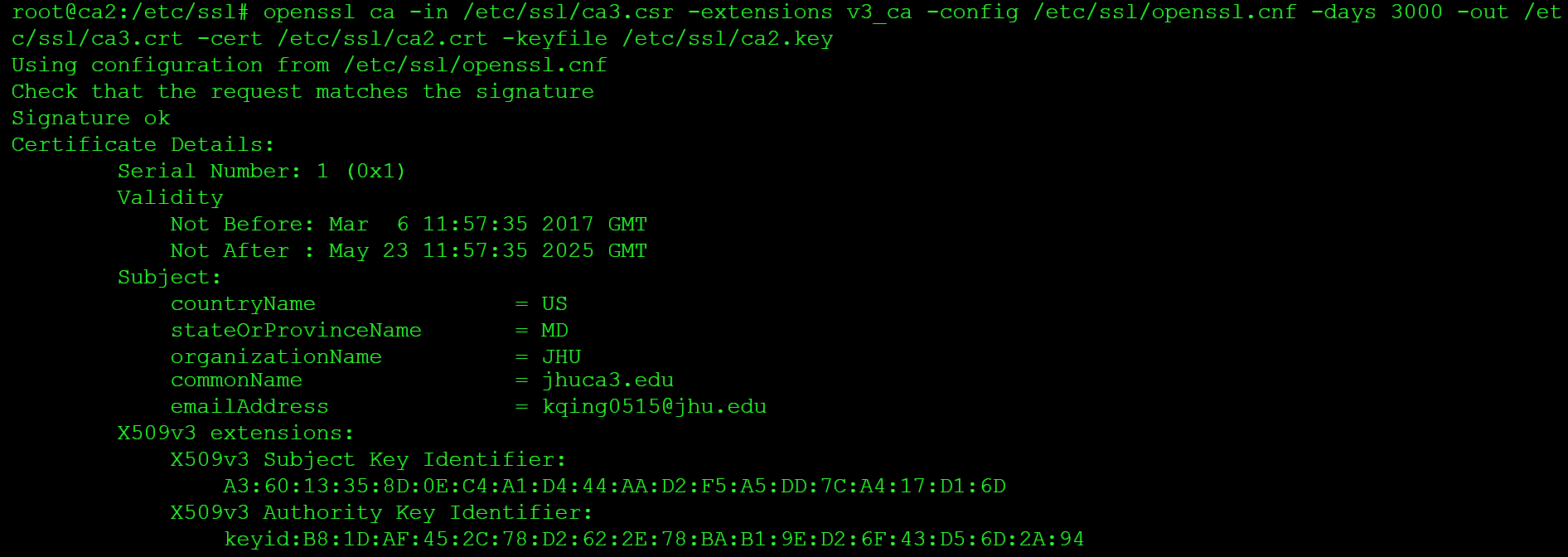
* “openssl ca -extensions v3\_ca -in /etc/ssl/ca2.csr -config /etc/ssl/openssl.cnf -days 3000 -out /etc/ssl/ca2.crt -cert /etc/ssl/ca1req.pem -keyfile /etc/ssl/private/ca1key.pem”

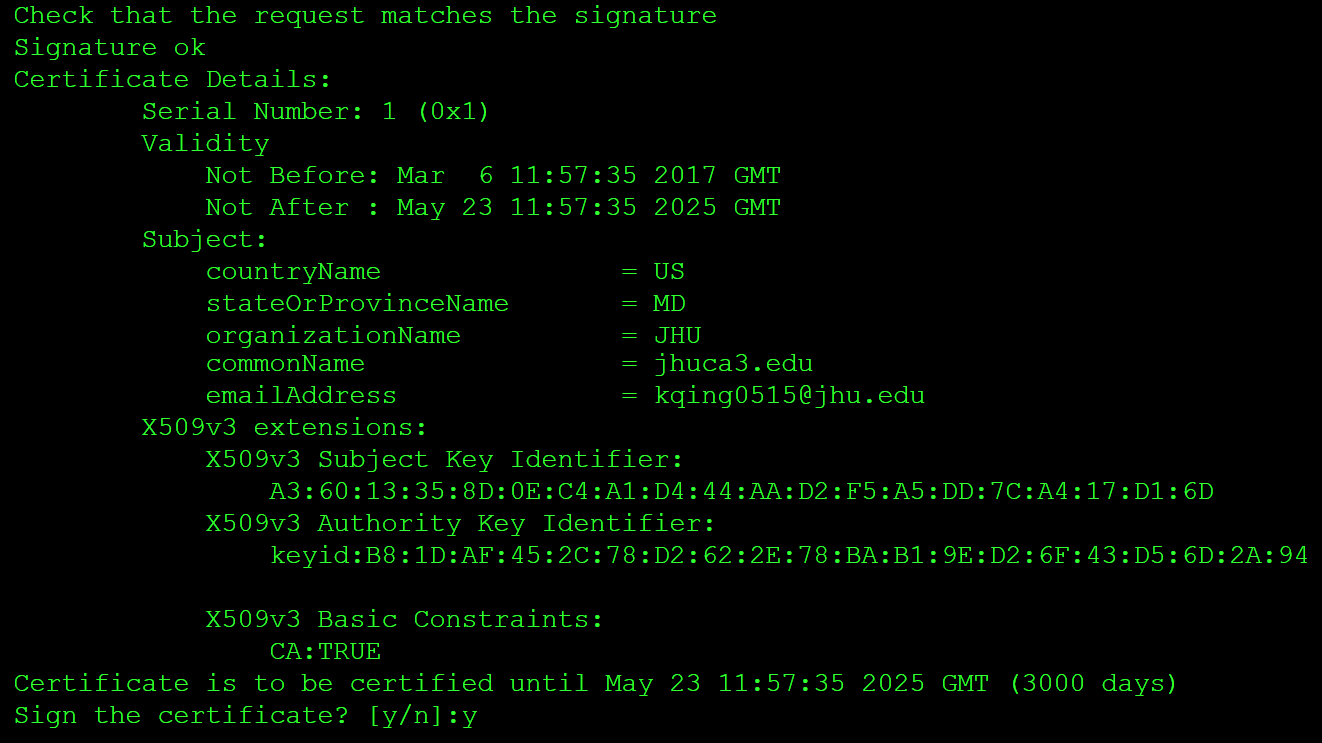




On ca2 node, use this command to sign the digital certificate for the issuing certificate authority (CA).

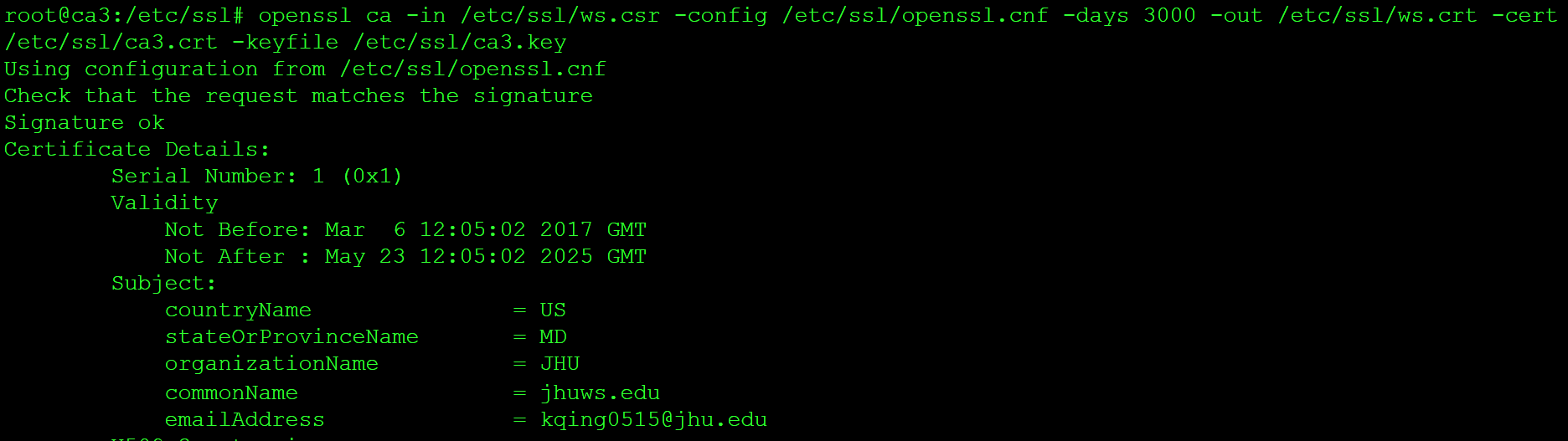
“openssl ca -in /etc/ssl/ca3.csr -extensions v3\_ca -config /etc/ssl/openssl.cnf -days 3000 -out /etc/ssl/ca3.crt -cert /etc/ssl/ca2.crt -keyfile /etc/ssl/ca2.key”

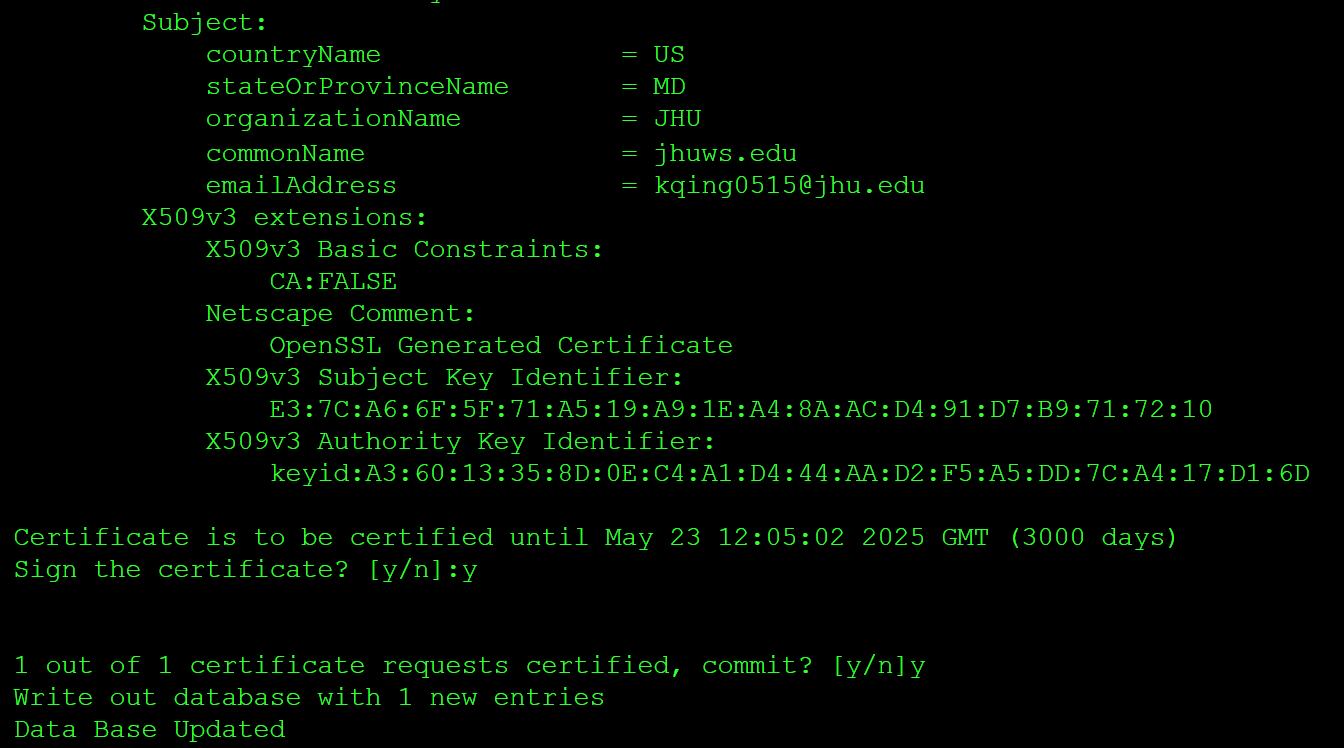




On ca3 node, use this command to sign the digital certificate for the web server.

“openssl ca -in /etc/ssl/ws.csr -config /etc/ssl/openssl.cnf -days 3000 -out /etc/ssl/ws.crt -cert /etc/ssl/ca3.crt -keyfile /etc/ssl/ca3.key”





To sum up, the commands used on each node are as follows:

Ca1 node:

* “touch /etc/ssl/index.txt”
* “echo 01 > /etc/ssl/serial”
* “mkdir newcerts”
* “openssl genrsa -des3 -out /etc/ssl/private/ca1key.pem 2048”
* “openssl req -new -x509 -days 3650 -key /etc/ssl/private/ca1key.pem -out /etc/ssl/ca1req.pem”
* “openssl ca -extensions v3\_ca -in /etc/ssl/ca2.csr -config /etc/ssl/openssl.cnf -days 3000 -out /etc/ssl/ca2.crt -cert /etc/ssl/ca1req.pem -keyfile /etc/ssl/private/ca1key.pem”

Ca2 node

* “touch /etc/ssl/index.txt”
* “echo 01 > /etc/ssl/serial”
* “mkdir newcerts”
* “openssl genrsa -des3 -out /etc/ssl/ca2.key 2048”
* “openssl rsa -in /etc/ssl/ca2.key -out /etc/ssl/ca2.key”
* “openssl req -new -days 3650 -key /etc/ssl/ca2.key -out /etc/ssl/ca2.csr”
* “openssl ca -in /etc/ssl/ca3.csr -extensions v3\_ca -config /etc/ssl/openssl.cnf -days 3000 -out /etc/ssl/ca3.crt -cert /etc/ssl/ca2.crt -keyfile /etc/ssl/ca2.key”

Ca3 node

* “touch /etc/ssl/index.txt”
* “echo 01 > /etc/ssl/serial”
* “mkdir newcerts”
* “openssl genrsa -des3 -out /etc/ssl/ca3.key 2048”
* “openssl rsa -in /etc/ssl/ca3.key -out /etc/ssl/ca3.key”
* “openssl req -new -days 3650 -key /etc/ssl/ca3.key -out /etc/ssl/ca3.csr”
* “openssl ca -in /etc/ssl/ws.csr -config /etc/ssl/openssl.cnf -days 3000 -out /etc/ssl/ws.crt -cert /etc/ssl/ca3.crt -keyfile /etc/ssl/ca3.key”

Ws node

* “touch /etc/ssl/index.txt”
* “echo 01 > /etc/ssl/serial”
* “openssl genrsa -des3 -out /etc/ssl/ws.key 2048”
* “openssl rsa -in /etc/ssl/ws.key -out /etc/ssl/ws.key”
* “openssl req -new -days 3650 -key /etc/ssl/ws.key -out /etc/ssl/ws.csr”

We now have the digital certificate of the intermediate certificate authority (CA) on ca1 node, the digital certificate of issuing certificate authority (CA) on ca2 node, the digital certificate of web server on the ca3 node.

We need to move the digital certificate to the corresponding ca node. Still, use the WinSCP or SCP command to make it.

We also need to move these digital certificates to the user node.

These digital certificates include the ca1 node’s digital certificate, which is the root certificate authority. That’s because we run a certificate authority offline.

And finally, remember to install the certificate on ws node.

The commands are as follows:

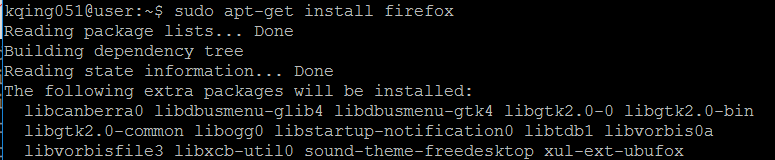
“sudo cp ws.crt /etc/ssl/certs”

“sudo cp ws.key /etc/ssl/private”

## Run and open a browser on GENI

We choose Firefox browser in this experiment. Of course, you can choose other browsers if you like. This part should be done on user node.

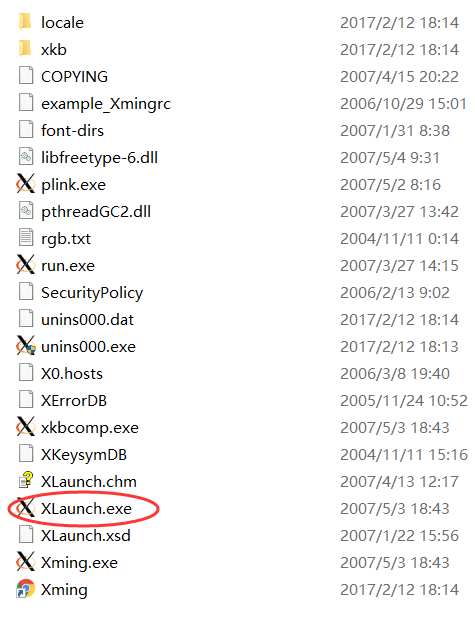
Command:”sudo apt-get install firefox”



The operation of next step will be different for Windows, Mac and Linux operation systems. For Windows and MacOS operation systems, we need to depend on third party software to enable the graphics display on GENI node.

### **For Windows operation system**:

Install the Xming software on your local operation system. Xming is an X11 display server for Microsoft Windows operating systems. Then run it to start the X server. You should see the Xming icon in the taskbar if it is running.

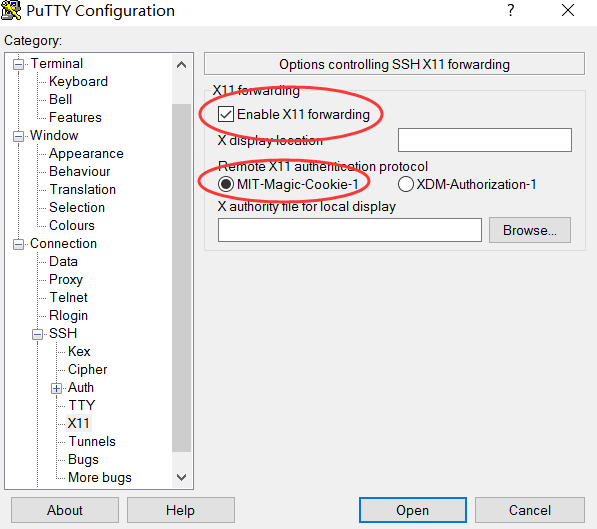


Then use PuTTY to log into the GENI node.

You can see the instruction here about how to log onto GENI node using PuTTY.

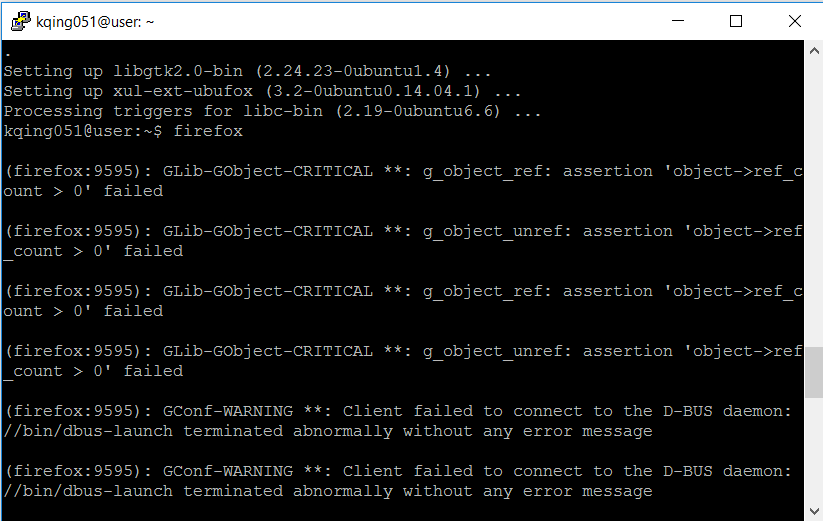
http://groups.geni.net/geni/wiki/HowTo/LoginToNodes

Remember to click the option on X11 option besides the other steps of logging onto GENI node using PuTTY.

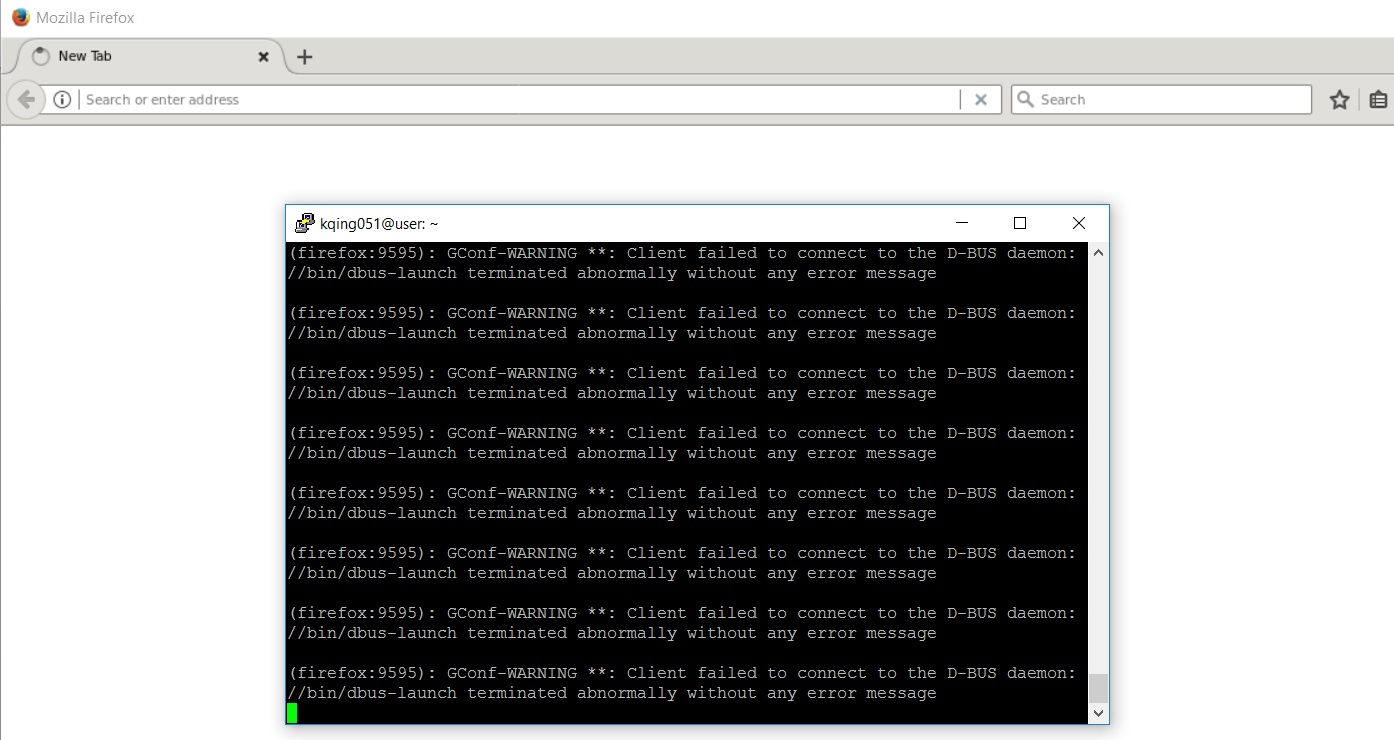


Then we can run the graphics display on GENI node on Windows operation system.

Command:”firefox”

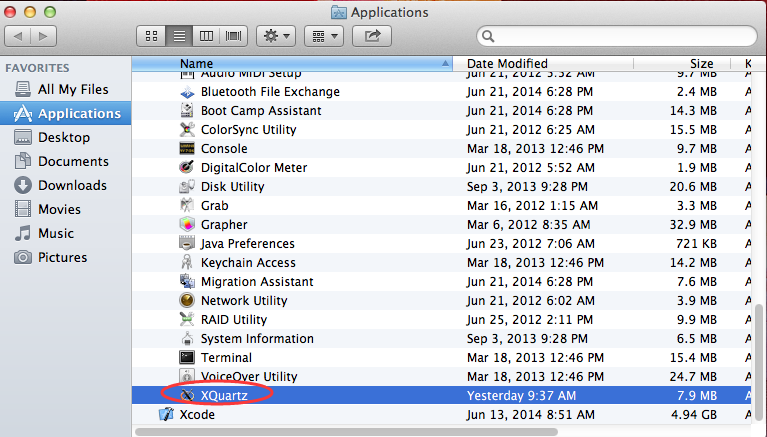


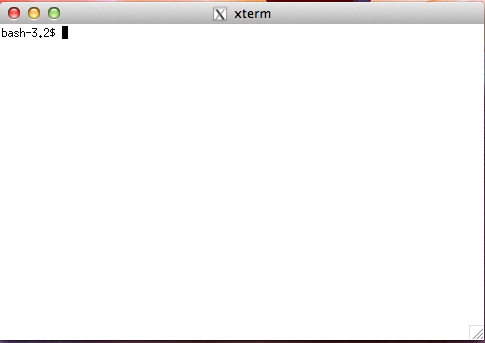
Wait for a second, and then we can see the browser GUI display is shown with the help of Xming



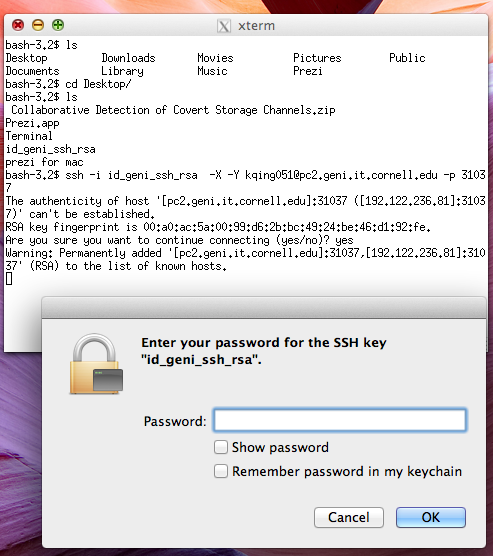
### For MacOS operation system

Install XQuartz on your Mac. XQuartz is an X server designed for MacOS.



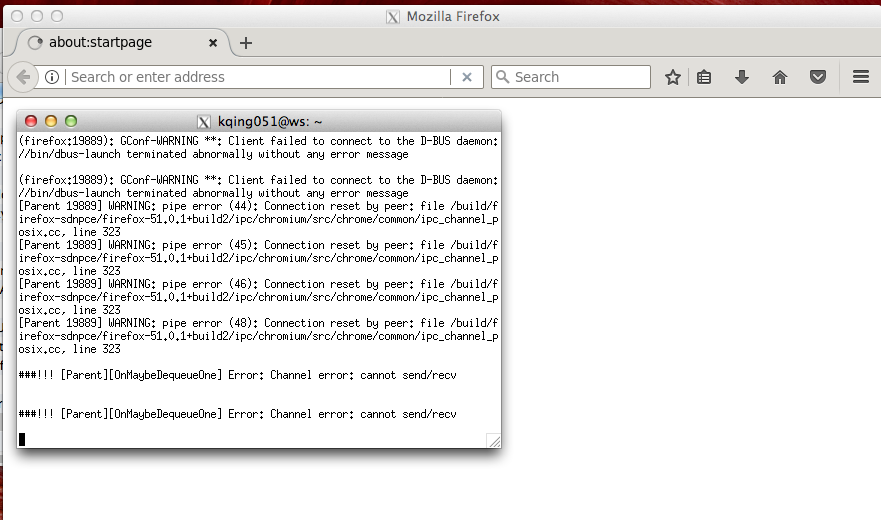
Right click on the XQuartz icon in the dock and select Applications > Terminal. Doing this should bring up a new xterm terminal windows. 

Then make an ssh connection to the GENI node on this terminal windows.



We can enable the graphics display of browser on GENI node with the help of XQuartz software.

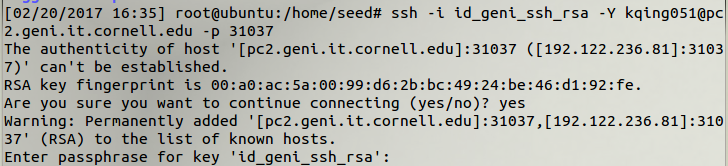
Command:”firefox”



### For other Linux operation system

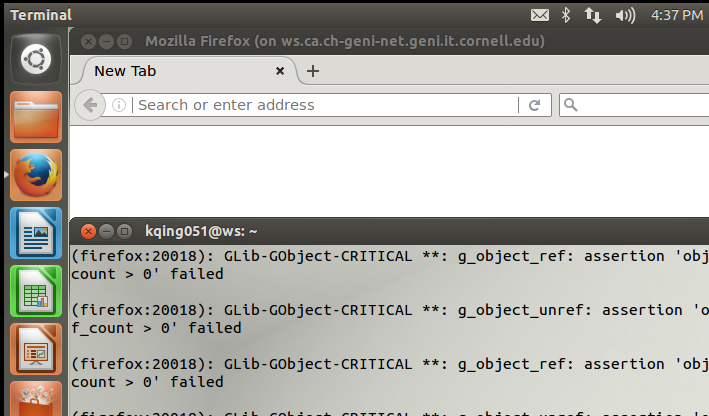
It’s much simple when you are using Linux operation system.

Just ssh into the Linux system of your choice using the -Y argument.



Then run Firefox browser.

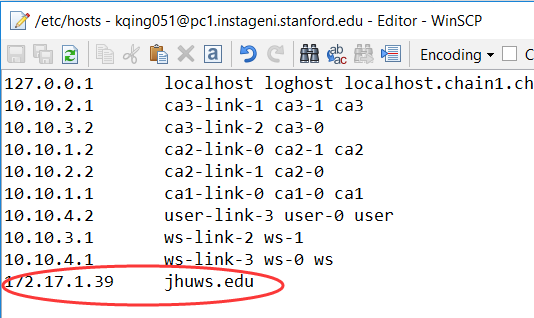
Command:”firefox”



## View results

Login in user node and install the browser.

Modify hosts file, which is in the location of “/etc”



Add the ws node IP address into it.

Use these commands to update and install Firefox browser.

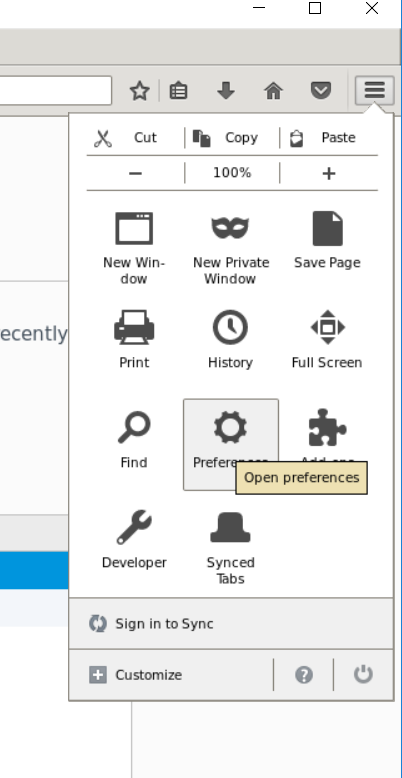
“sudo su”

“apt-get update”

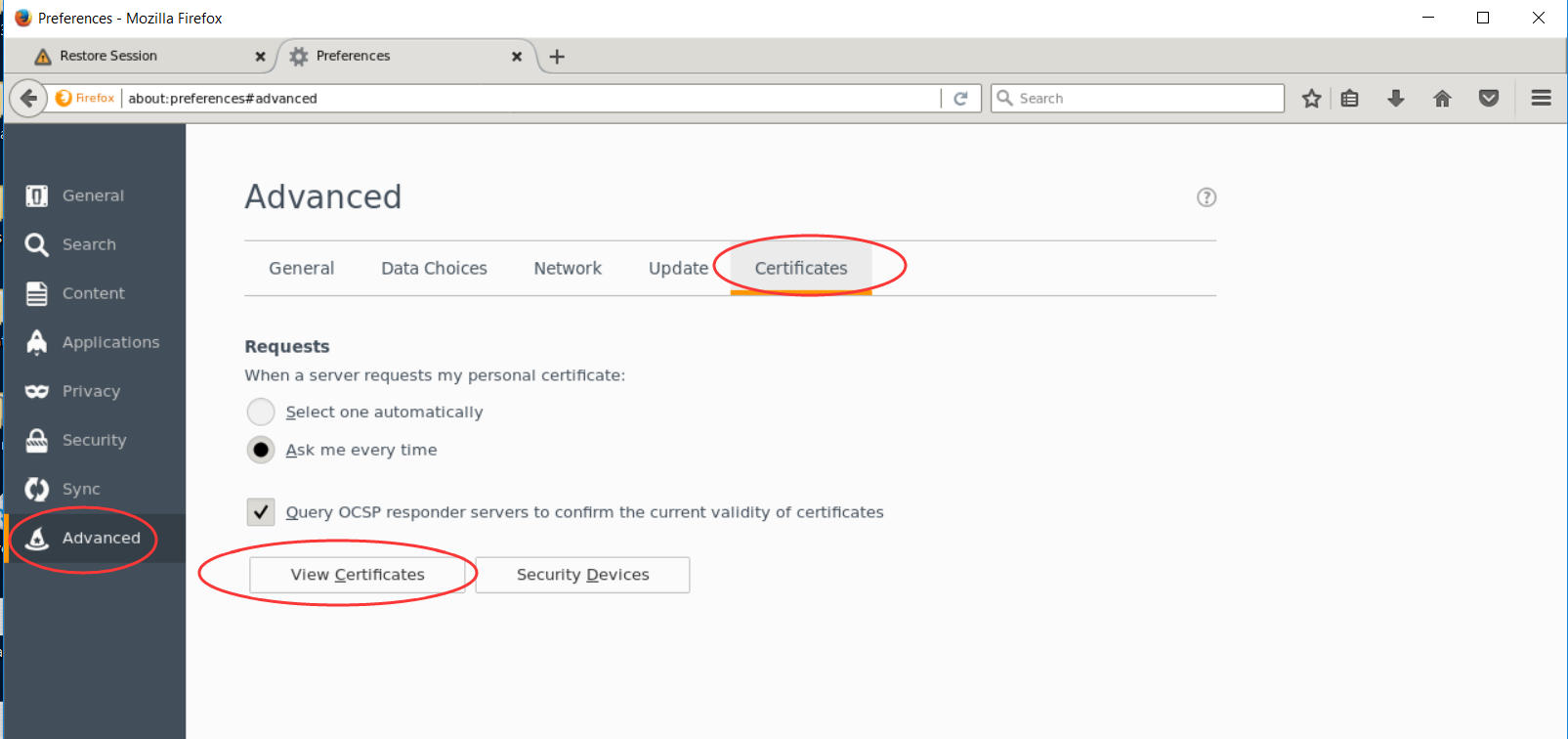
“apt-get install firefox”

Then use command “firefox” to open it. About how to run and open a browser can refer to the instructions above.

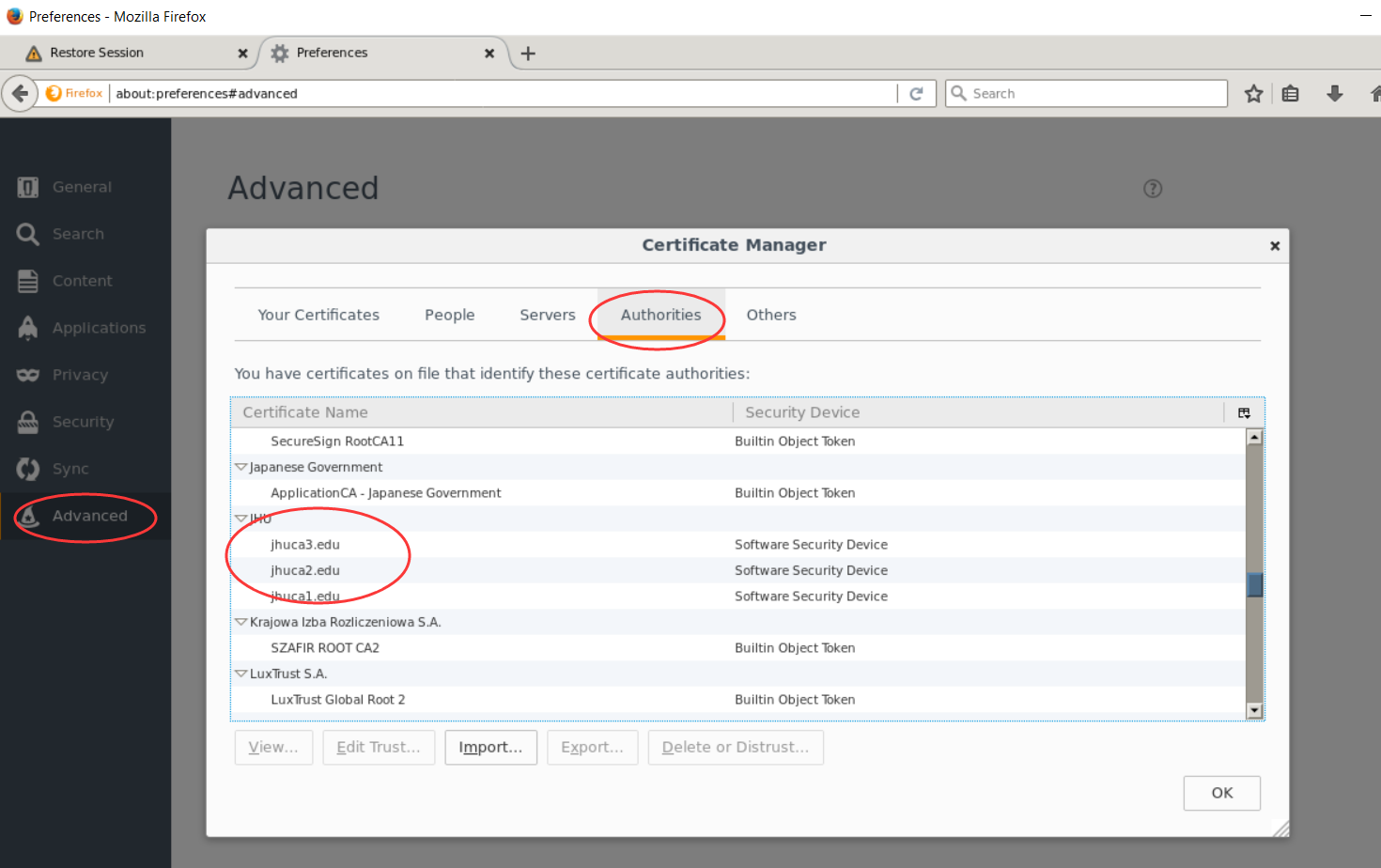
Enter here to open the option.



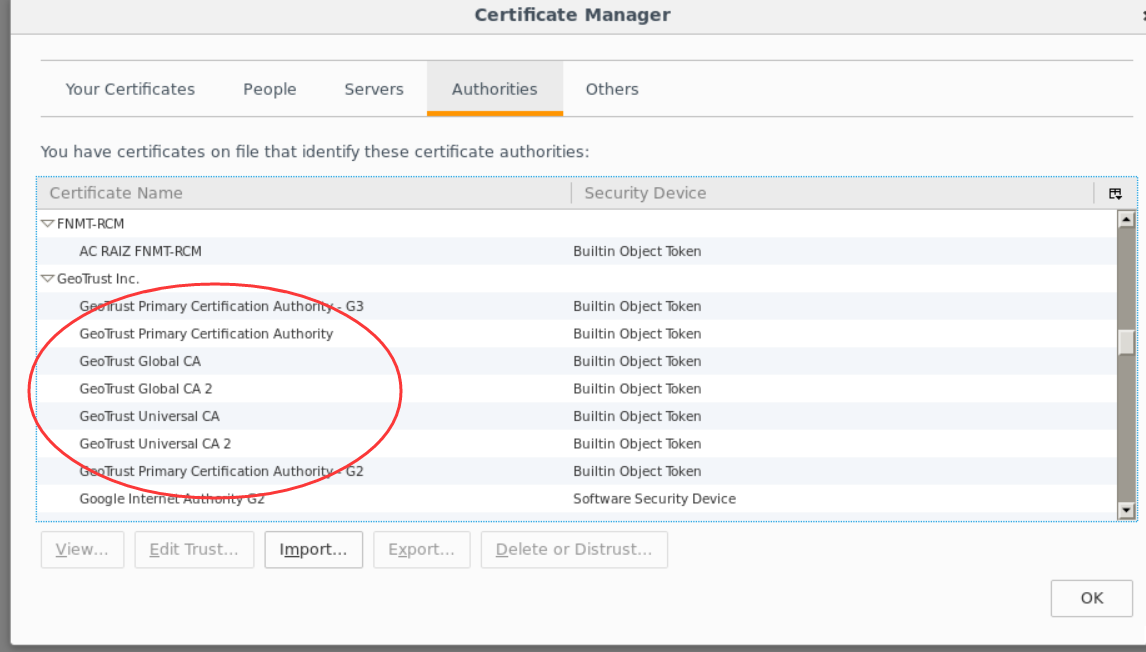
And then click “Advanced”,”Certificate”, “View Certificates” to open the window to import the digital certificate.



We import root certificate authority (CA) digital certificate, intermediate certificate authority (CA) digital certificate and issuing certificate authority (CA) digital certificate into it.



We also can see the hierarchy of other organizations’ certificate authority in this display.

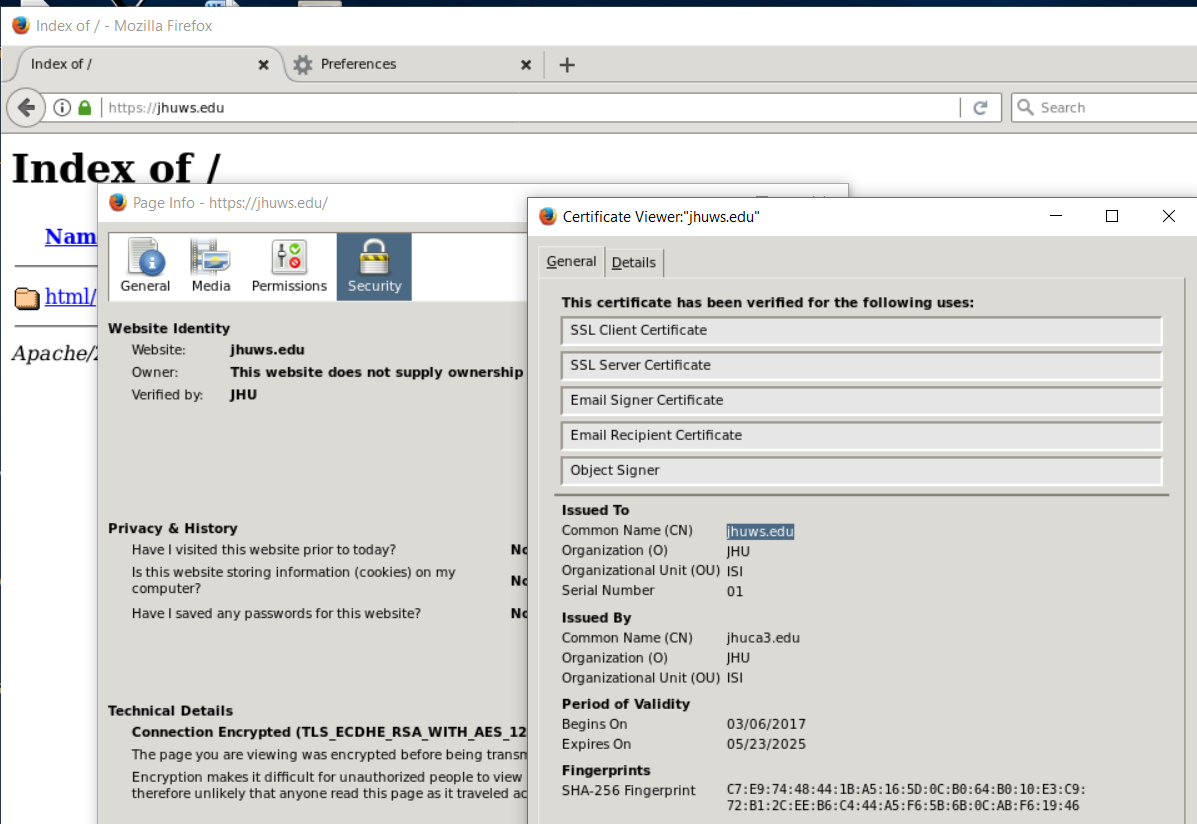


Then we can visit our website, which is supported by our web server. Be sure to add “https” before our domain name. Like in my case, enter <https://jhuws.edu> in the browser.



We can see there is a green lock exist, it means that our website is based on HTTPS visit, which means our digital certificates works well.

And we can see the details of our digital certificate by click the green clock.



Also, we can view hierarchy of our digital certificates on our laptop. But be sure to install the digital certificates of root certificate authority, intermediate certificate authority and issuing certificate authority.

Then we open the digital certificate of the web server. We can see the hierarchy of our certificate authority as follow:

