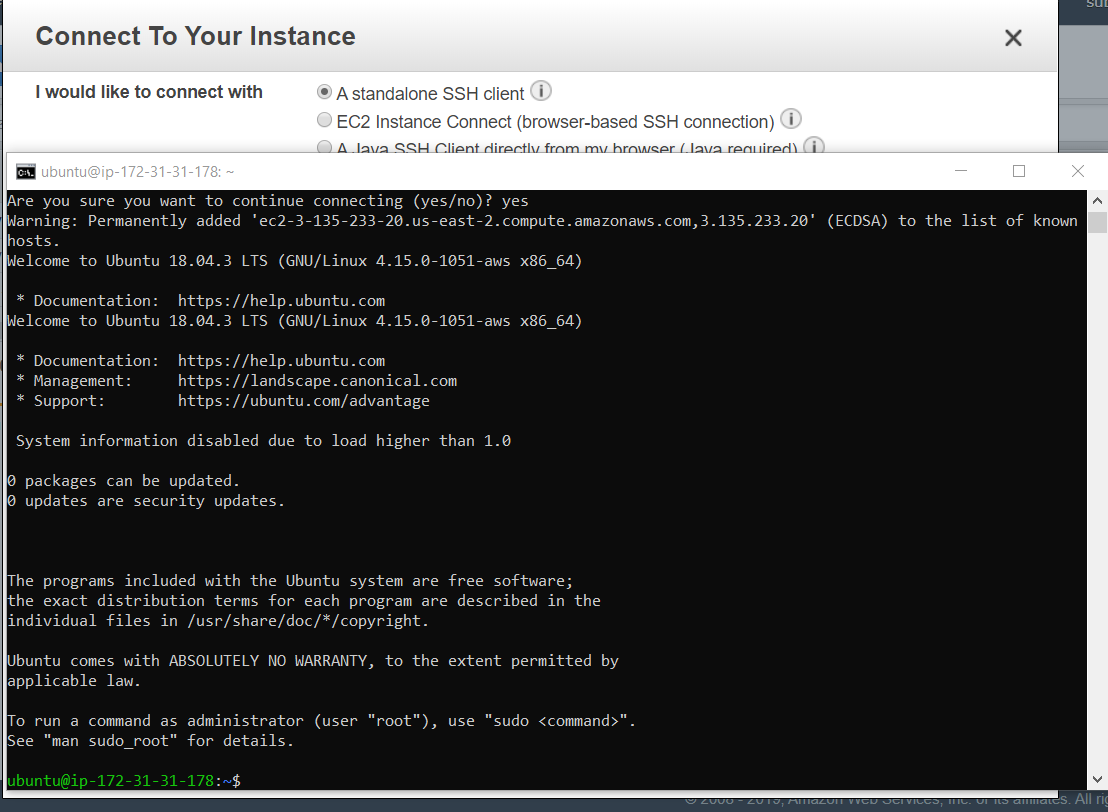
Group: Blue Hats

Assignment-7

Github Link: <https://github.com/s-c-soma/pki-example-1>

1. Ubuntu EC2 instance is up and ready for setup



Question 1: Design and build a PKI infrastructure that includes Root CA, Signing CA, and TLS Certificate,

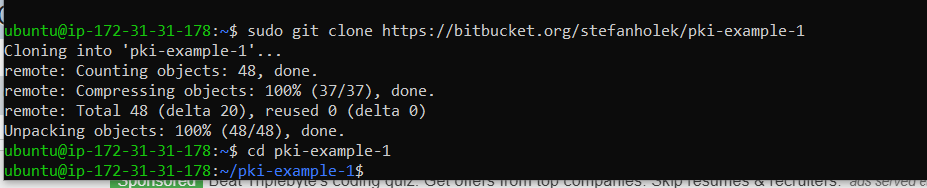
* E.g., as described here: <http://pki-tutorial.readthedocs.io/en/latest/simple/>

1. Designing and Building a PKI infrastructure

2.1 Git clone of PKI code:

**$sudo git clone** [**https://github.com/s-c-soma/pki-example-1.git**](https://github.com/s-c-soma/pki-example-1.git)

**$cd pki-example-1**

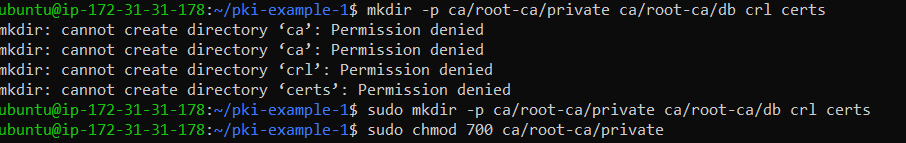


1. Root CA creation

3.1 Creating Directories: Here the ca directory holds CA resources, the crl directory holds CRLs, and the certs directory holds user certificates

**$sudo mkdir -p ca/root-ca/private ca/root-ca/db crl certs**

**$sudo chmod 700 ca/root-ca/private**



3.2 Database Creation

**$sudo cp /dev/null ca/root-ca/db/root-ca.db**

**$sudo cp /dev/null ca/root-ca/db/root-ca.db.attr**

**$sudo sh -c "echo '01' > ca/root-ca/db/root-ca.crt.srl"**

**$sudo sh -c "echo '01' > ca/root-ca/db/root-ca.crl.srl"**

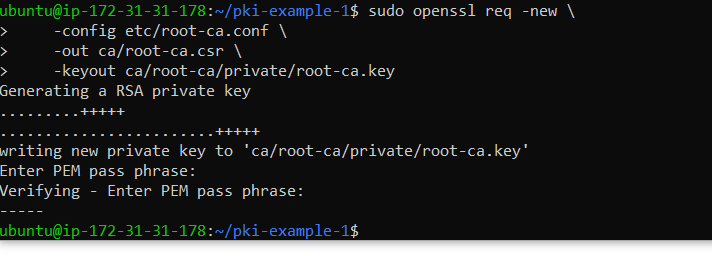
3.3 Creating CA request

**$sudo openssl req -new \**

**-config etc/root-ca.conf \**

**-out ca/root-ca.csr \**

**-keyout ca/root-ca/private/root-ca.key**



3.4 Creating CA Certificate

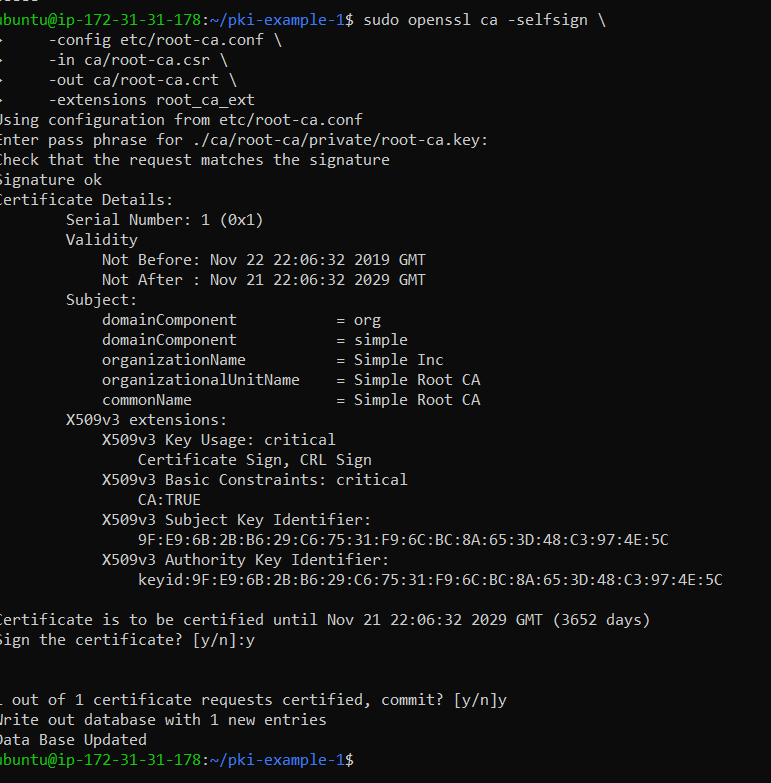
**$ sudo openssl ca -selfsign \**

**-config etc/root-ca.conf \**

**-in ca/root-ca.csr \**

**-out ca/root-ca.crt \**

**-extensions root\_ca\_ext**

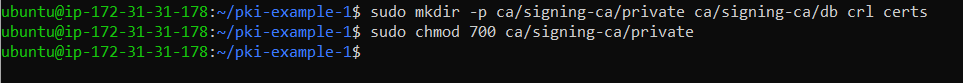


1. Singing CA Creation

4.1 Creating directories:

**$sudo mkdir -p ca/signing-ca/private ca/signing-ca/db crl certs**

**$sudo chmod 700 ca/signing-ca/private**



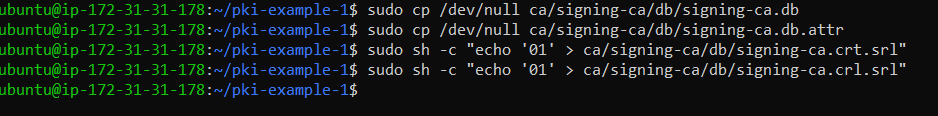
4.2 Creating database:

**$sudo cp /dev/null ca/signing-ca/db/signing-ca.db**

**$sudo cp /dev/null ca/signing-ca/db/signing-ca.db.attr**

**$sudo sh -c "echo '01' > ca/signing-ca/db/signing-ca.crt.srl"**

**$sudo sh -c "echo '01' > ca/signing-ca/db/signing-ca.crl.srl"**



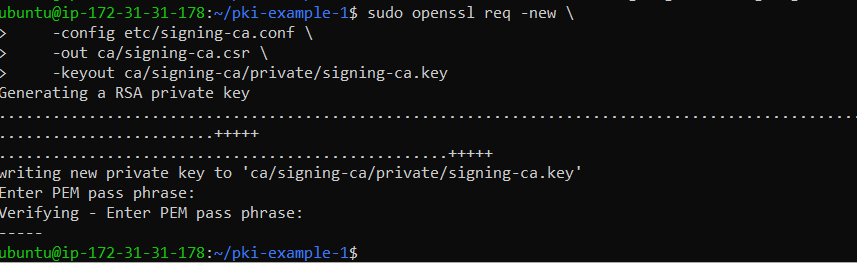
4.3 Creating CA Request:

**$sudo openssl req -new \**

**-config etc/signing-ca.conf \**

**-out ca/signing-ca.csr \**

**-keyout ca/signing-ca/private/signing-ca.key**



4.4 Creating CA Certificate:

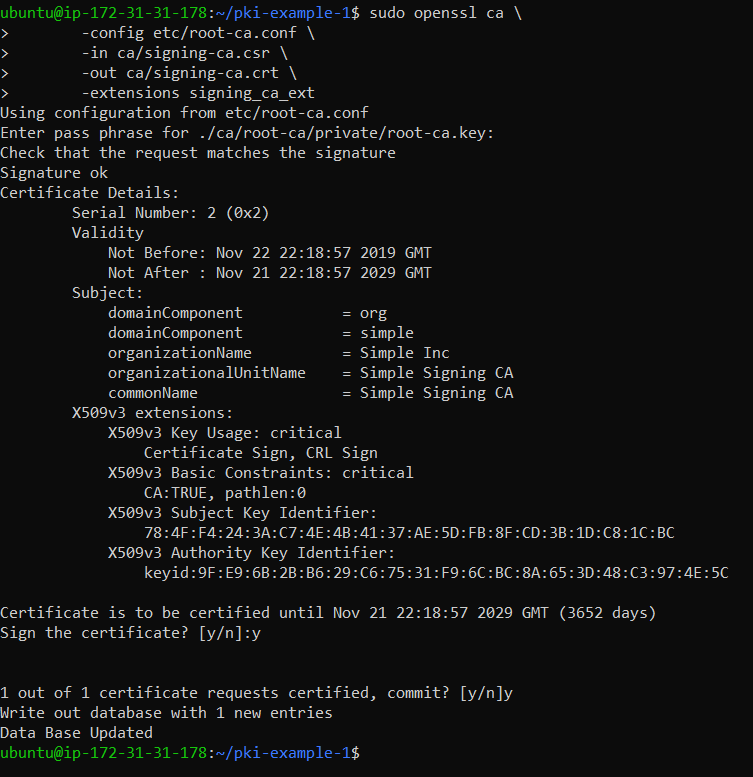
**$sudo openssl ca \**

**-config etc/root-ca.conf \**

**-in ca/signing-ca.csr \**

**-out ca/signing-ca.crt \**

**-extensions signing\_ca\_ext**



1. Operate Singing CA:

5.1 Creating Email Request:

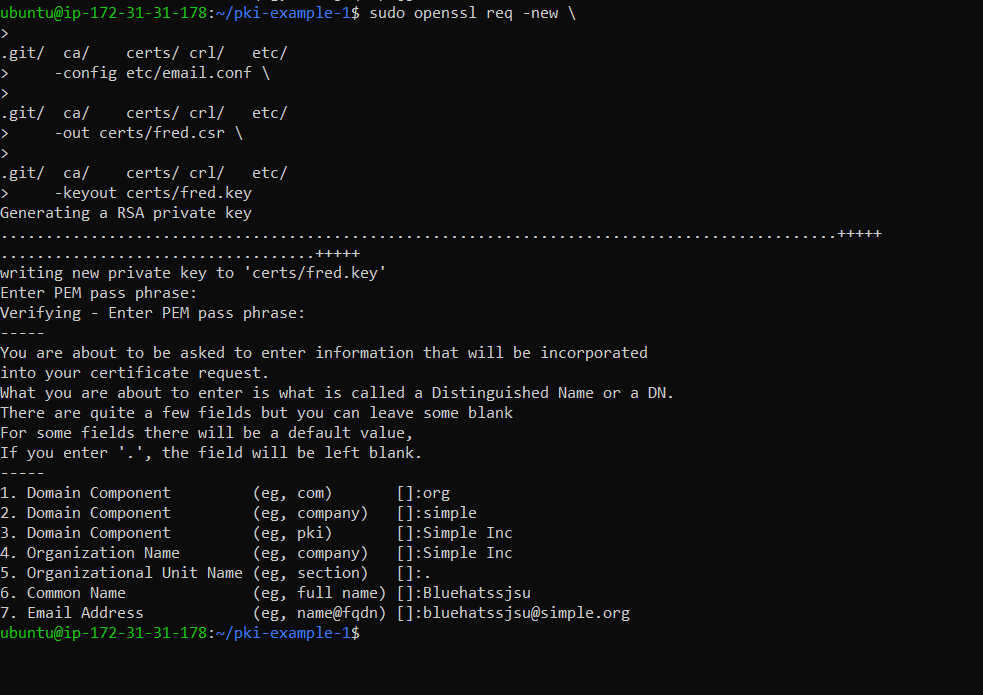
DN components: DC=org, DC=simple, O=Simple Inc, CN=Fred Flintstone, emailAddress=fred@simple.org

**$sudo openssl req -new \**

**-config etc/email.conf \**

**-out certs/fred.csr \**

**-keyout certs/fred.key**



5.2 Creating Email Certificate:

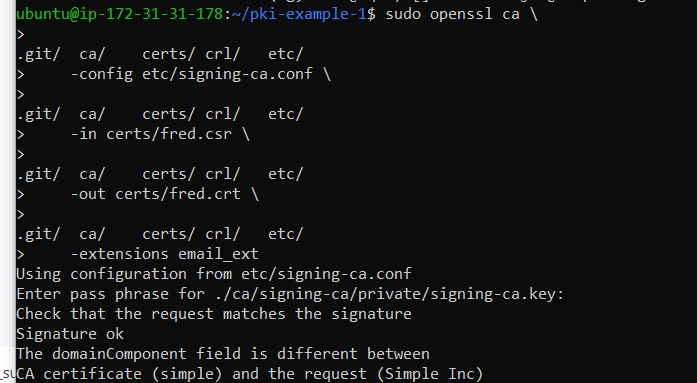
**$sudo openssl ca \**

**-config etc/signing-ca.conf \**

**-in certs/fred.csr \**

**-out certs/fred.crt \**

**-extensions email\_ext**



5.3 Creating TLS Server Request:

DN components: DC=org, DC=simple, O=Simple Inc, CN=www.simple.org

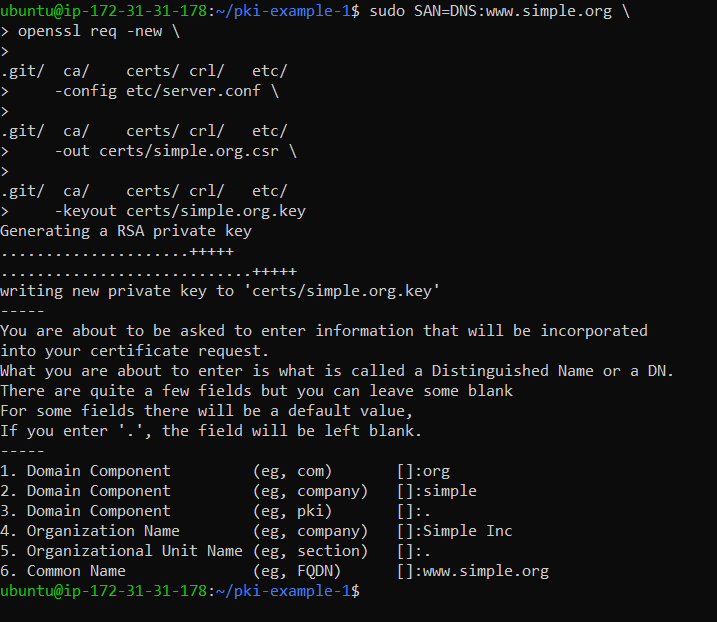
**$sudo SAN=DNS:www.simple.org \**

**openssl req -new \**

**-config etc/server.conf \**

**-out certs/simple.org.csr \**

**-keyout certs/simple.org.key**



5.4 Creating TLS Server Certificate:

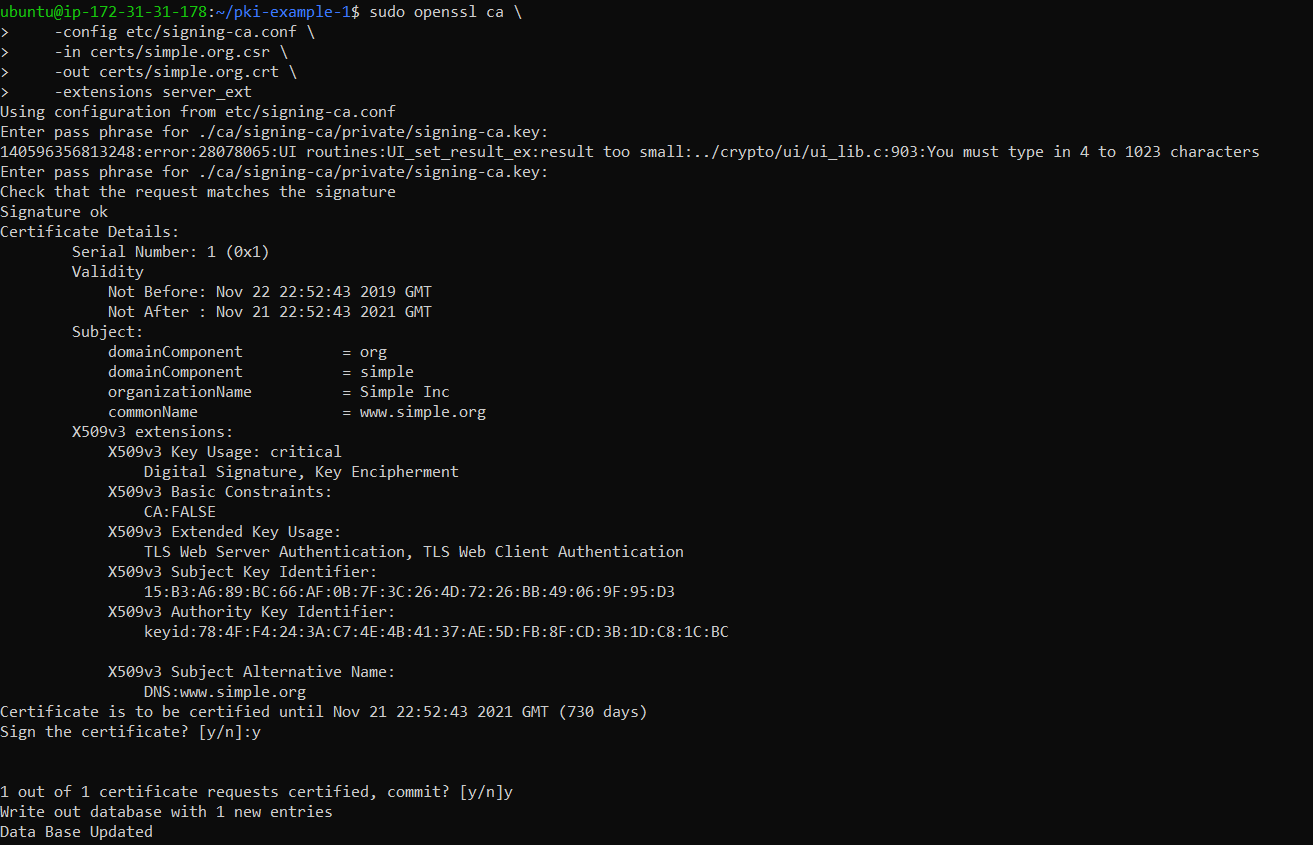
**$sudo openssl ca \**

**-config etc/signing-ca.conf \**

**-in certs/simple.org.csr \**

**-out certs/simple.org.crt \**

**-extensions server\_ext**



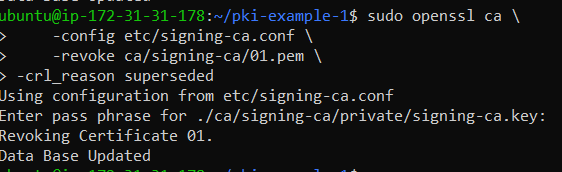
5.5 Revoking Certificate:

**$ sudo openssl ca \**

**-config etc/signing-ca.conf \**

**-revoke ca/signing-ca/01.pem \**

**-crl\_reason superseded**

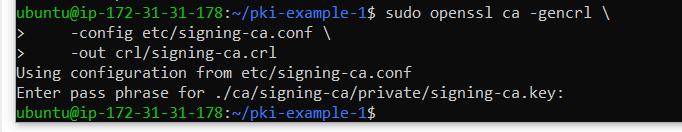


5.6 Creating URL:

**$ sudo openssl ca -gencrl \**

**-config etc/signing-ca.conf \**

**-out crl/signing-ca.crl**



1. Output Formats:

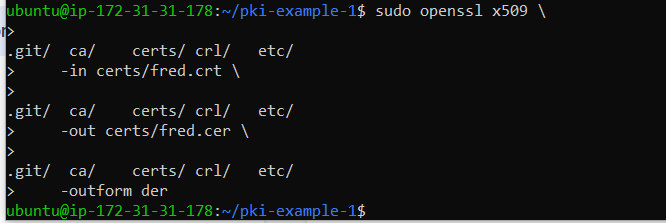
6.1 Creating DER Certificate:

**$sudo openssl x509 \**

**-in certs/fred.crt \**

**-out certs/fred.cer \**

**-outform der**



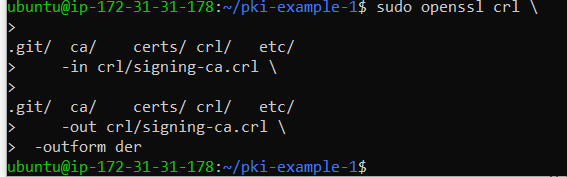
6.2 Creating DER CRL:

**$sudo openssl crl \**

**-in crl/signing-ca.crl \**

**-out crl/signing-ca.crl \**

**-outform der**



6.3 Creating PKCS#7 Bundle:

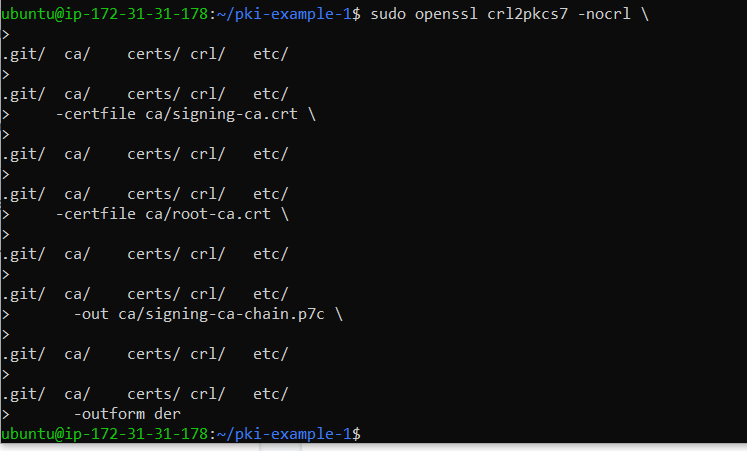
**$sudo openssl crl2pkcs7 -nocrl \**

**-certfile ca/signing-ca.crt \**

**-certfile ca/root-ca.crt \**

**-out ca/signing-ca-chain.p7c \**

**-outform der**



6.4 Creating PKCS#12 Bundle:

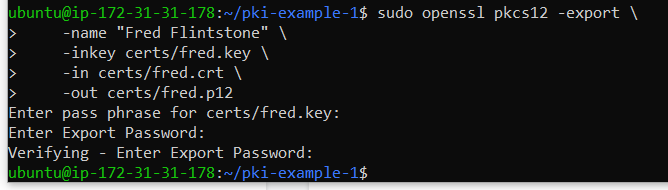
**$sudo openssl pkcs12 -export \**

**-name "Fred Flintstone" \**

**-inkey certs/fred.key \**

**-in certs/fred.crt \**

**-out certs/fred.p12**



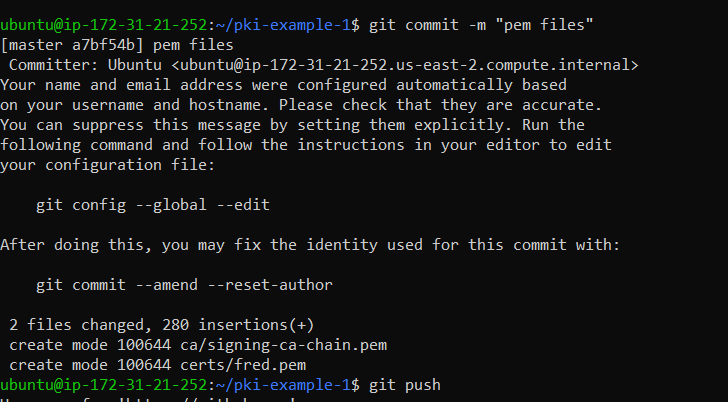
6.5 Create PEM Bundle:

**$ sudo cat ca/signing-ca.crt ca/root-ca.crt > \**

**ca/signing-ca-chain.pem**

**$ sudo cat certs/fred.key certs/fred.crt > \**

**certs/fred.pem**

****

1. View Results:

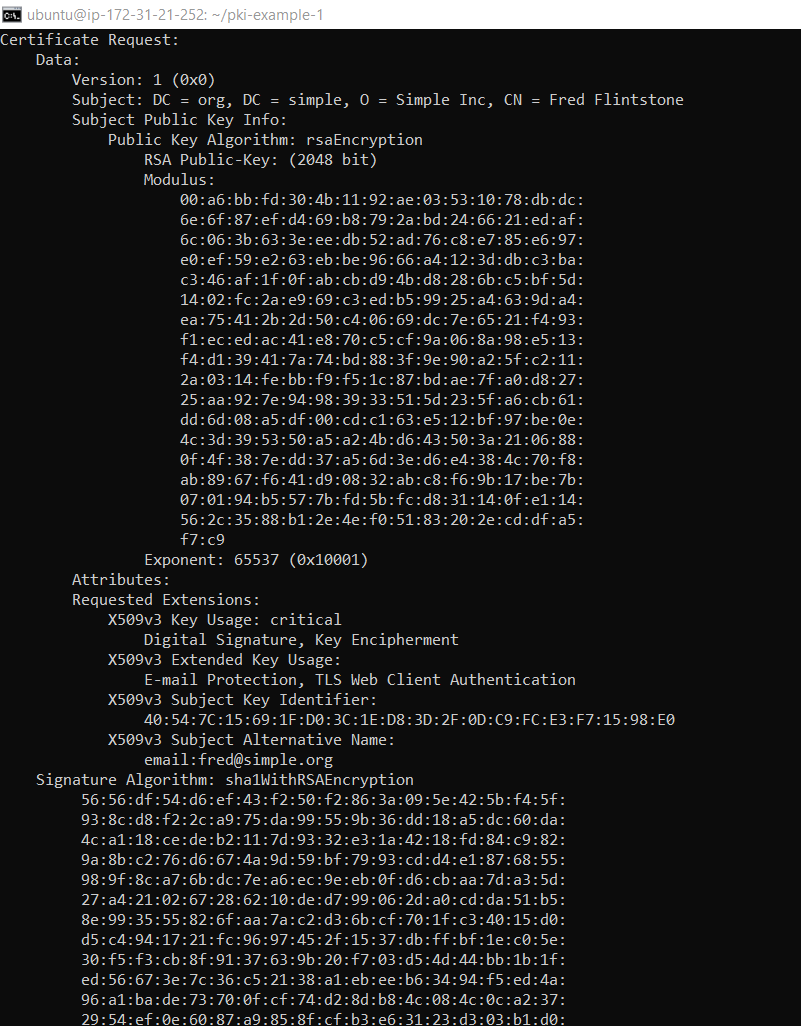
7.1 View Request:

$ sudo openssl req **\**

-in certs/fred.csr **\**

-noout **\**

-text



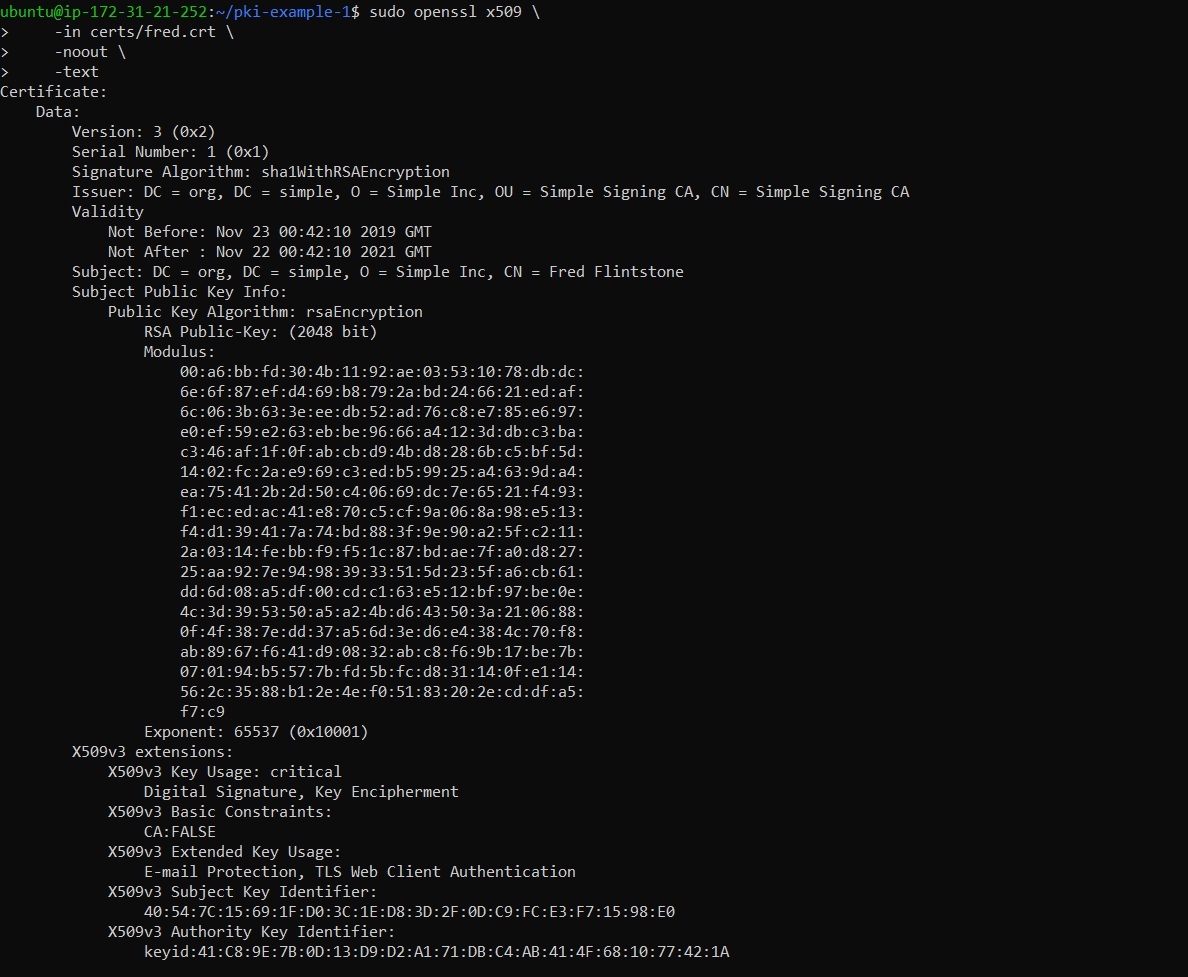
7.2 View Certificate:

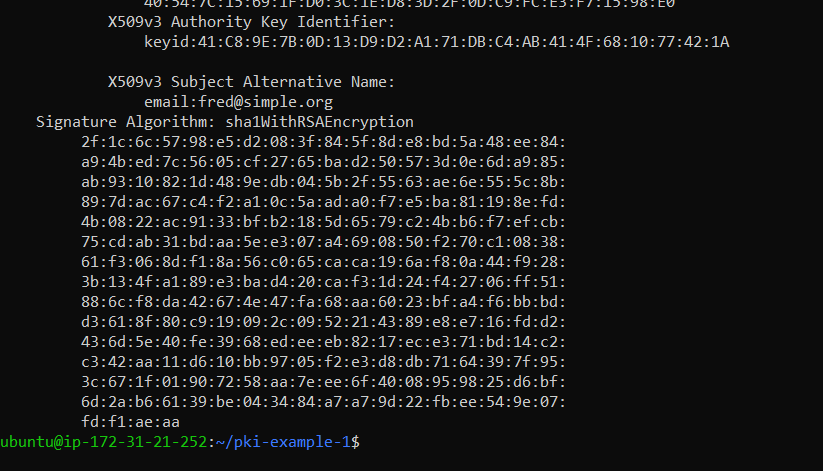
**$ sudo openssl x509 \**

**-in certs/fred.crt \**

**-noout \**

**-text**





7.3: View CRL:

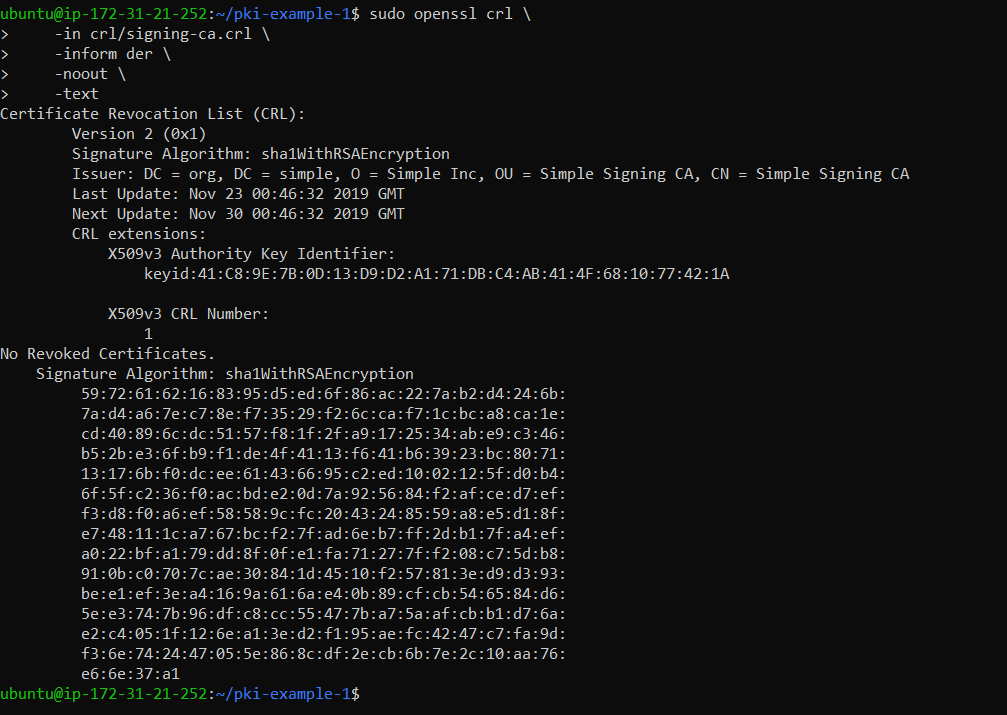
**$ sudo openssl crl \**

**-in crl/signing-ca.crl \**

**-inform der \**

**-noout \**

**-text**



7.4 View PKCS#7 Bundle:

**$ sudo openssl pkcs7 \**

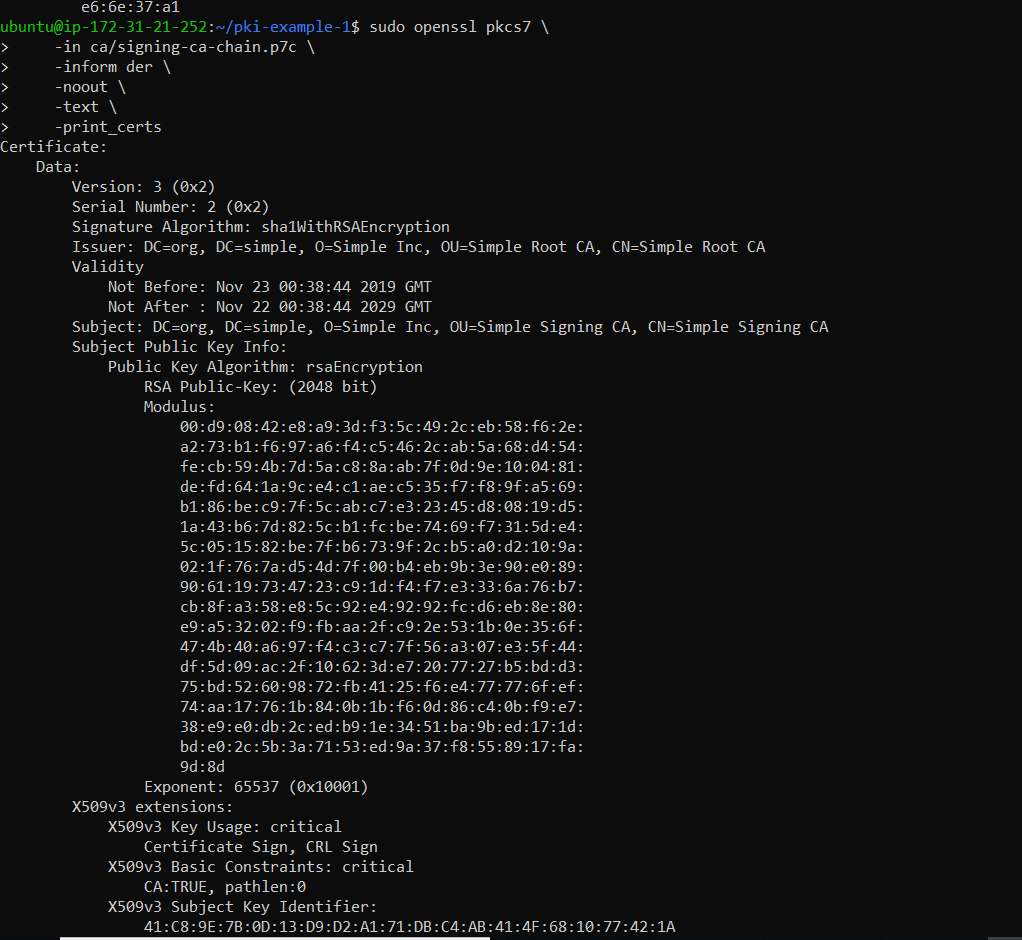
**-in ca/signing-ca-chain.p7c \**

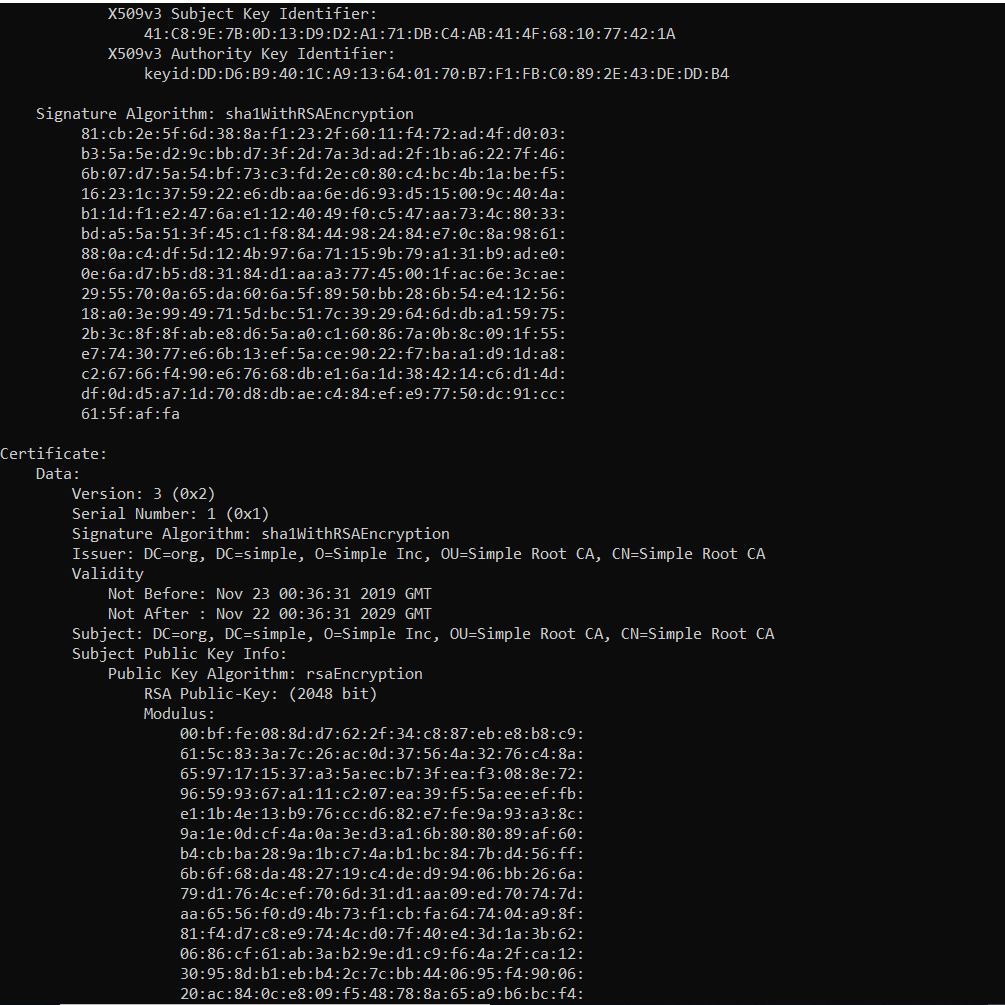
**-inform der \**

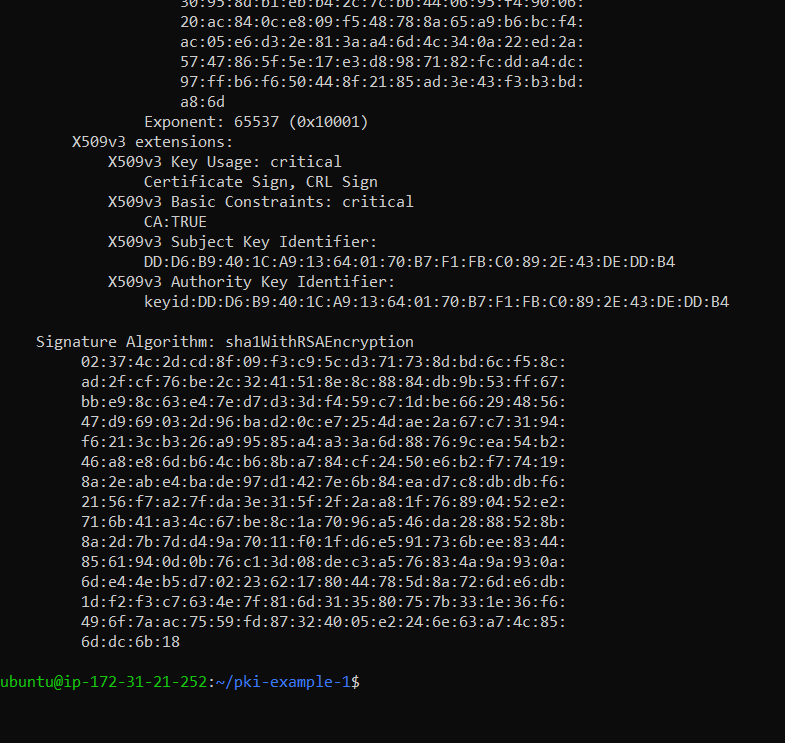
**-noout \**

**-text \**

**-print\_certs**







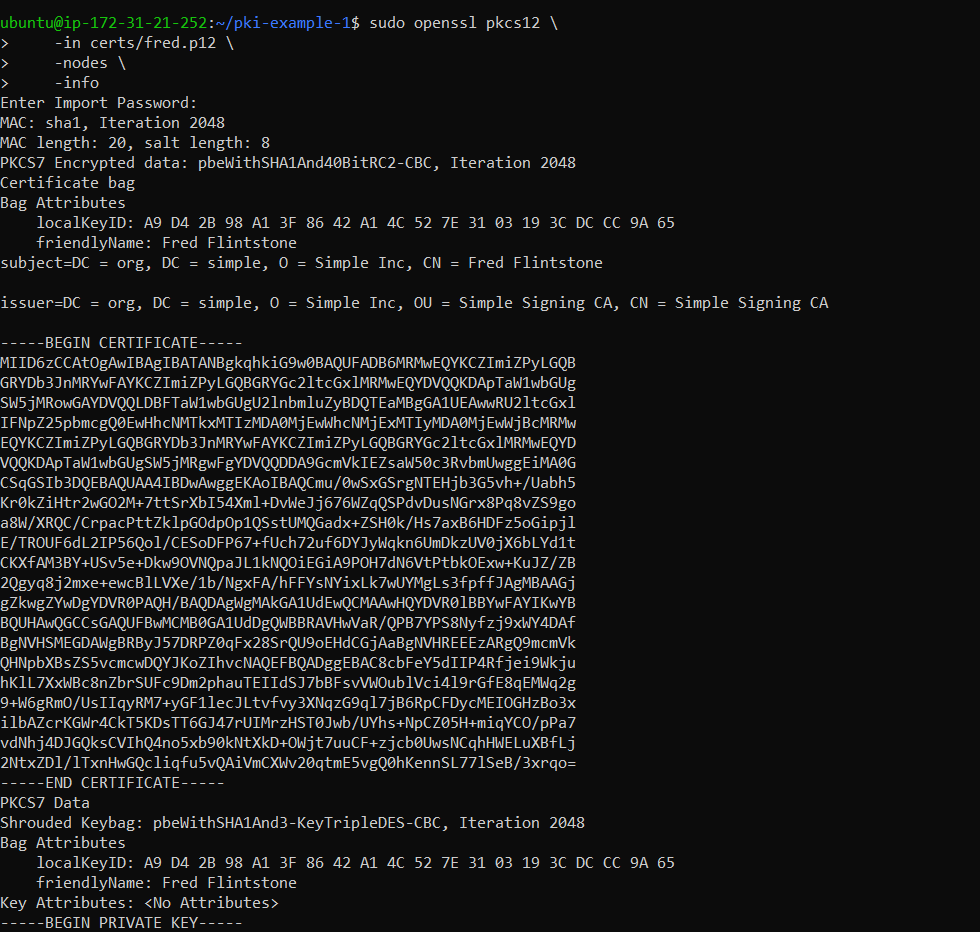
7.5 View PKCS#12 Bundle:

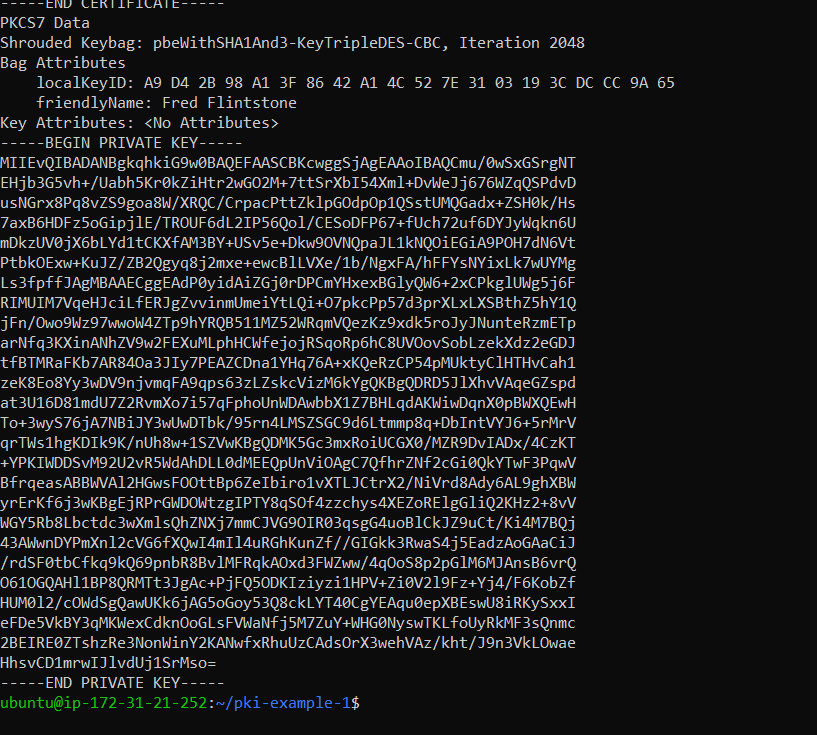
**$ sudo openssl pkcs12 \**

**-in certs/fred.p12 \**

**-nodes \**

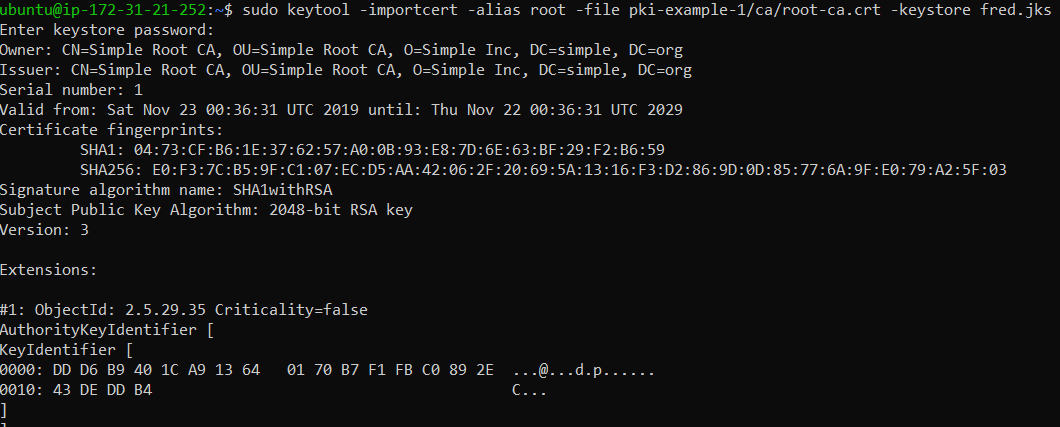
**-info**

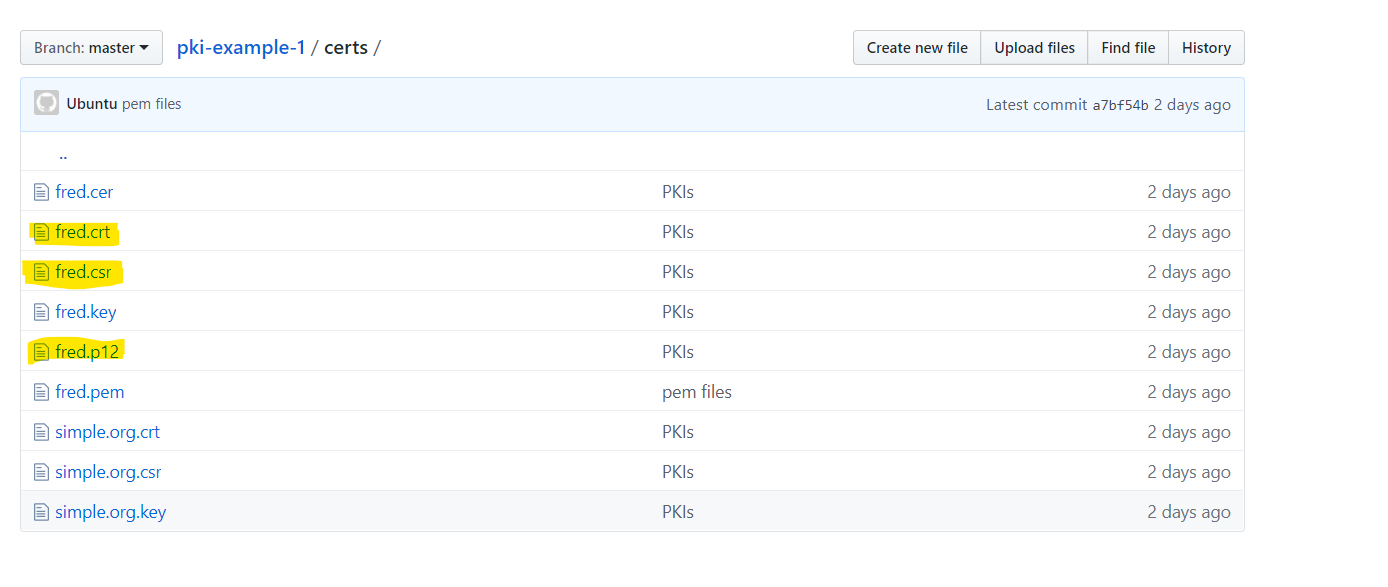




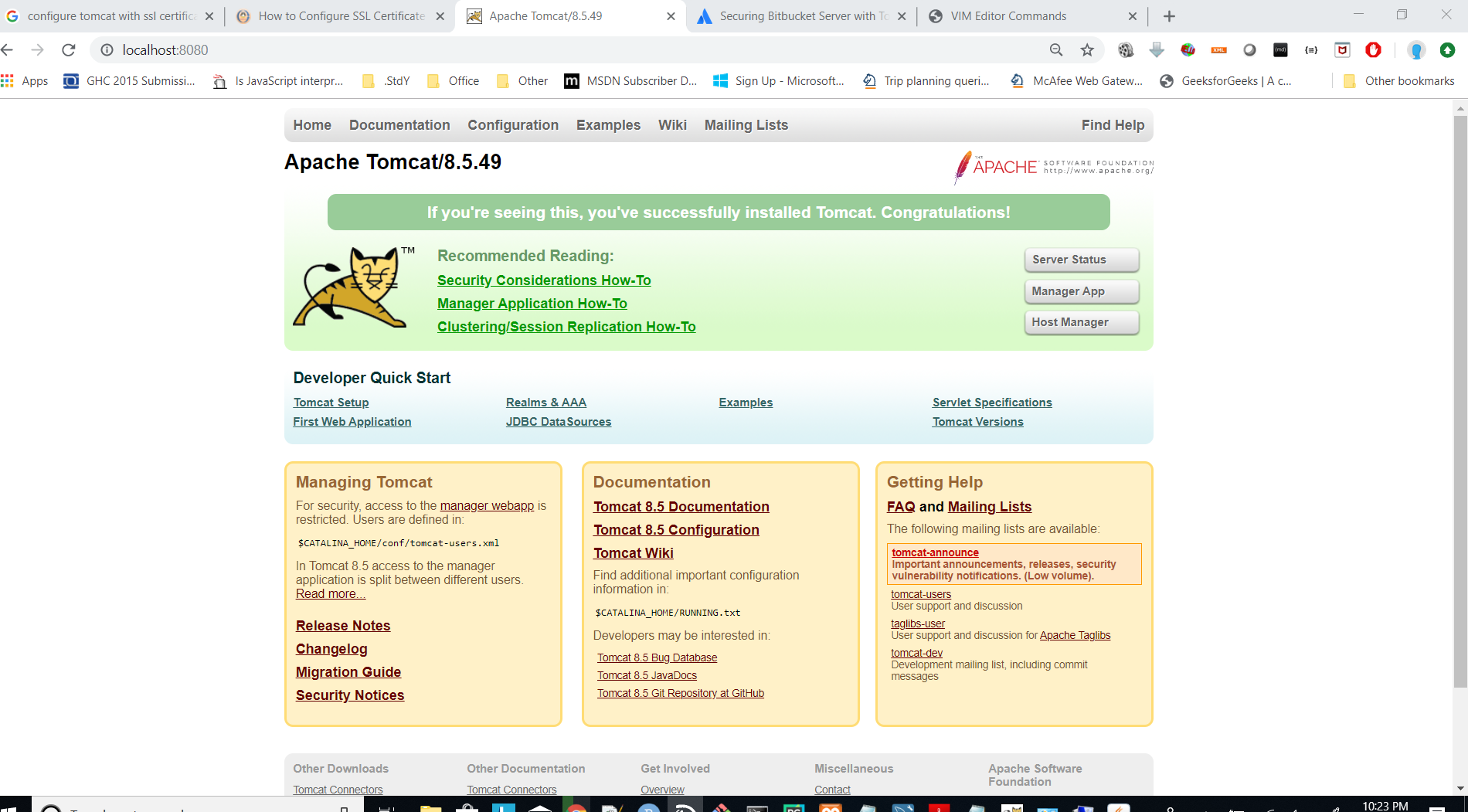
**Question 2: Use the TLS certificate to install a web server, e.g. tomcat,** <https://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html>

Preparing the Certificate Key Store: I have used the certificated to setup local tomcat. For that I have used the TLS certificate generated at the previous steps and put it in the keystore.

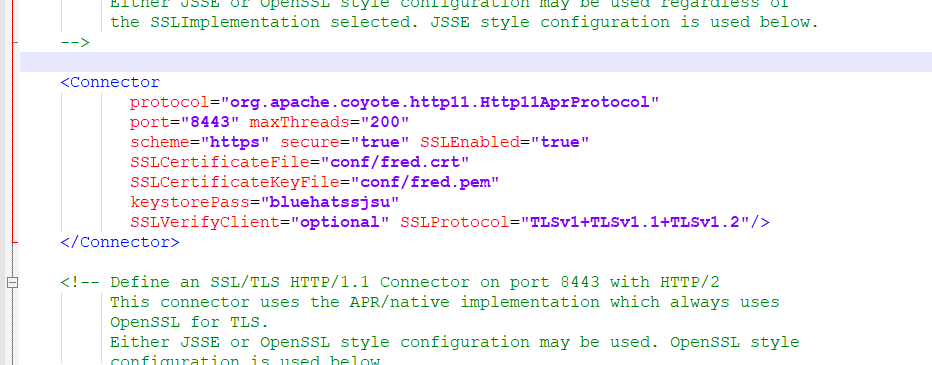




Tomcat localhost instance without TLS configuration:



After that I have modified the server.xml file to configure SSL/TLS . Configuration file is uploaded in the github.



Tomcat with TLS certificate setting and https:

