

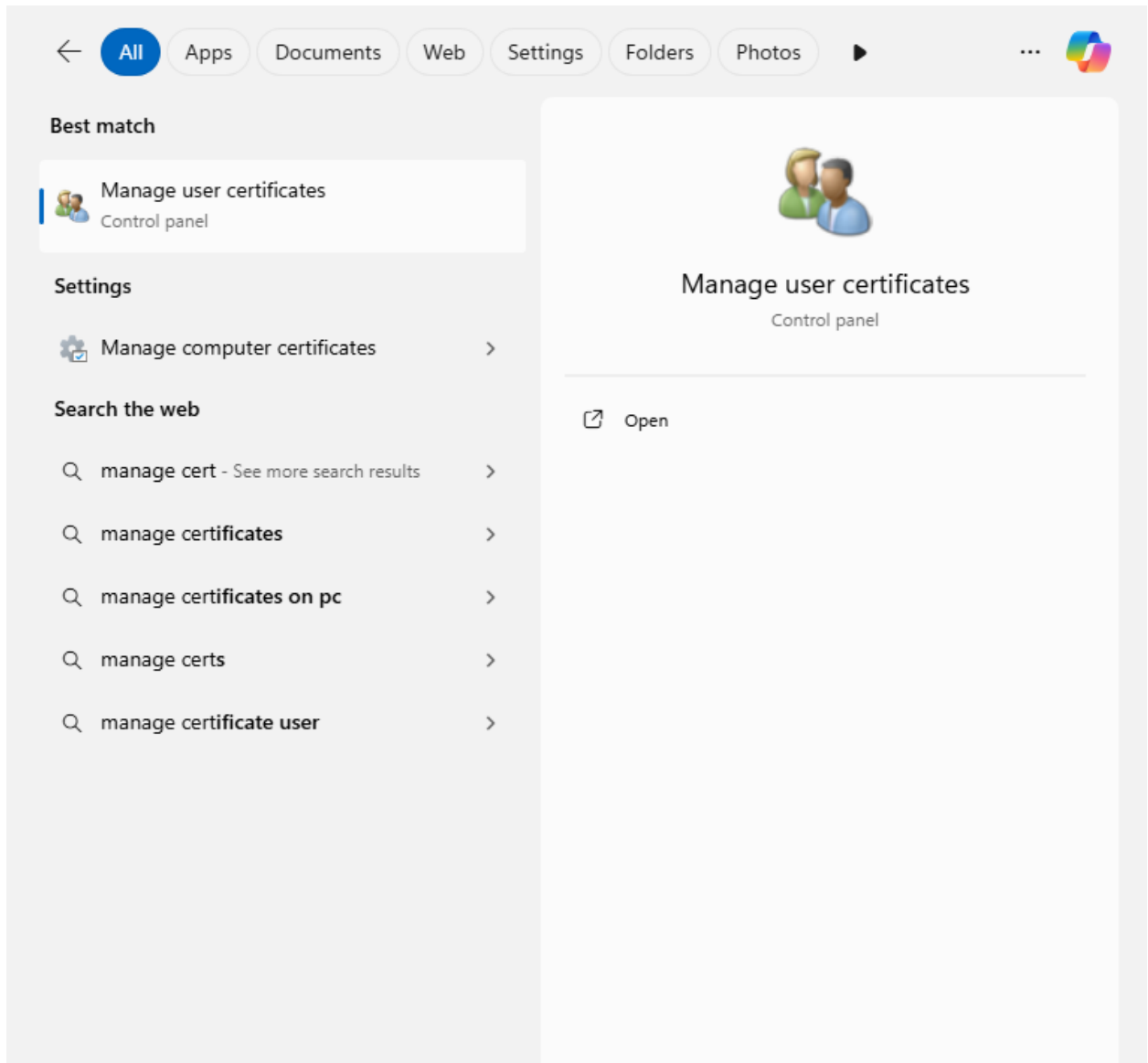
How to clone project on gitlab server

MỤC LỤC

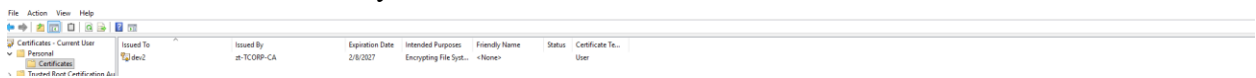
1. Check cert and export export cert	2
2. Chuyển cert sang dưới dạng .pem.....	8
3. Config gitlab to clone.....	8
4. Verify working	9

1. Check cert and export export cert

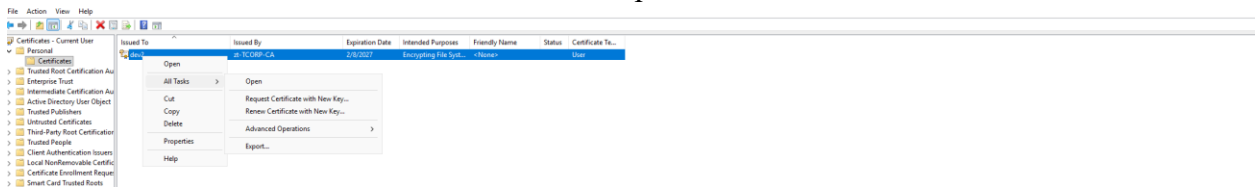
On client computer



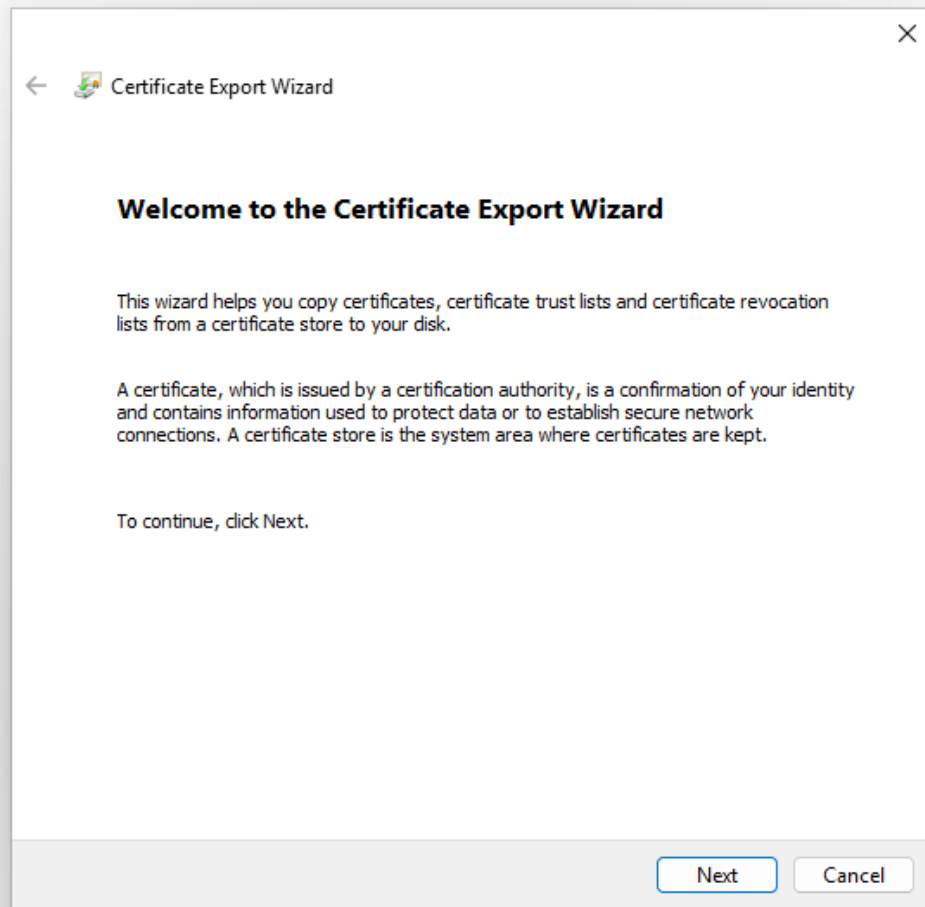
check cert user that AD verify



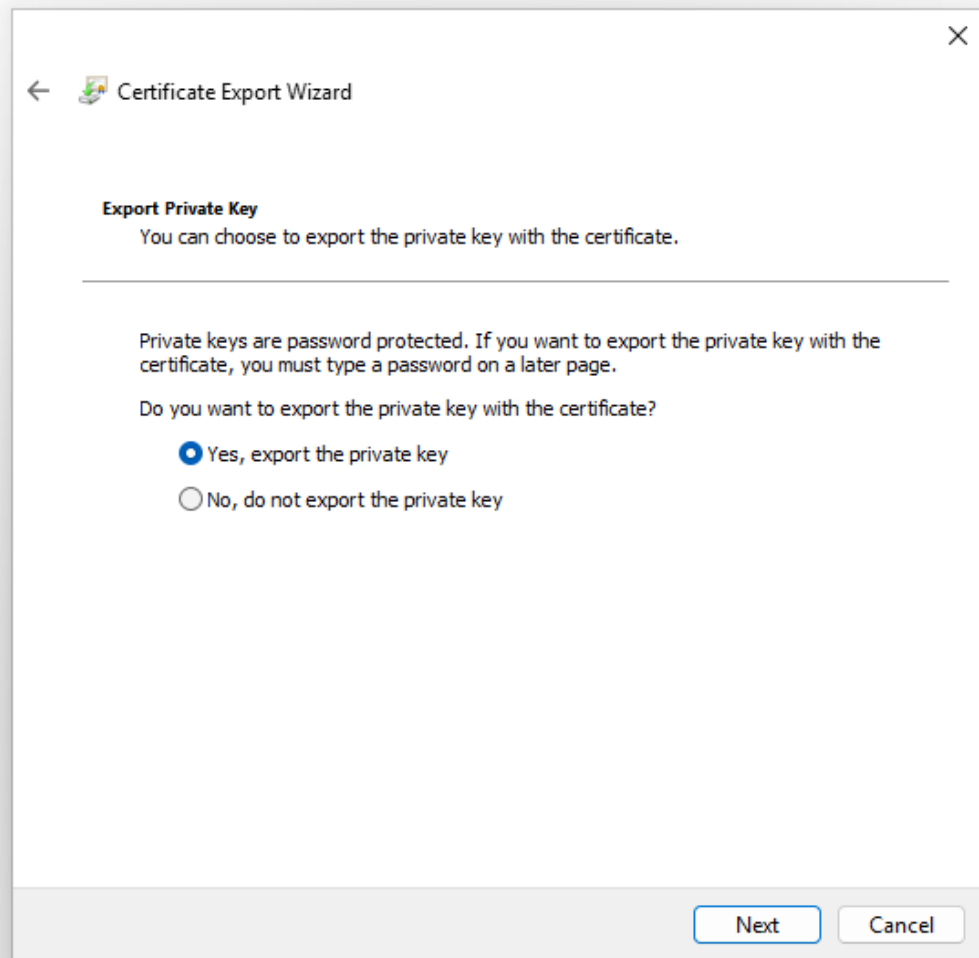
And then double cliclck on cert -> All tasks -> export



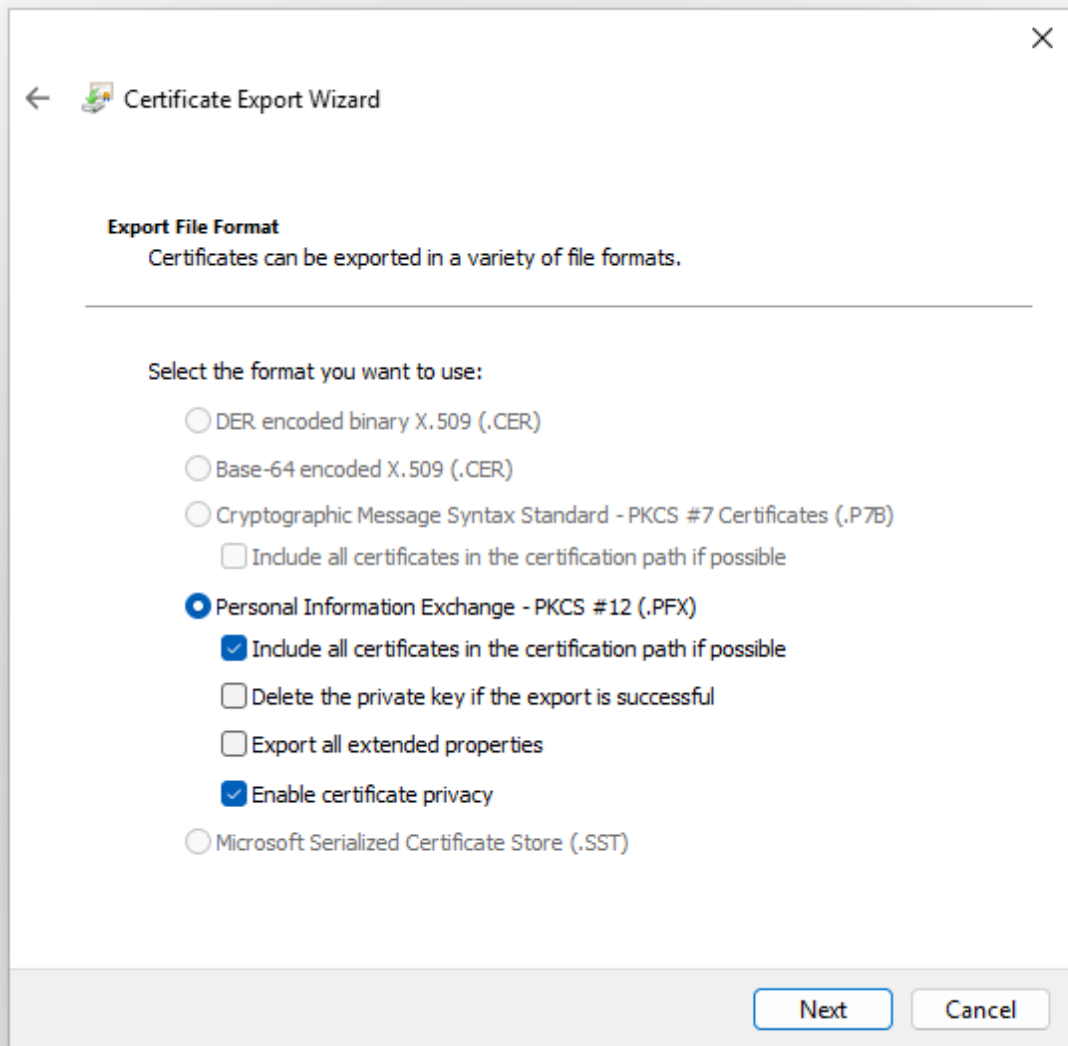
Next






Click **Yes, export the private key**



Export format choose



Setup password for file

  Certificate Export Wizard 

Security
To maintain security, you must protect the private key to a security principal or by using a password.

☐ Group or user names (recommended)

Add

Remove

☒ Password:
.....




Confirm password:
.....

Encryption: TripleDES-SHA1 ▾

Next

Cancel

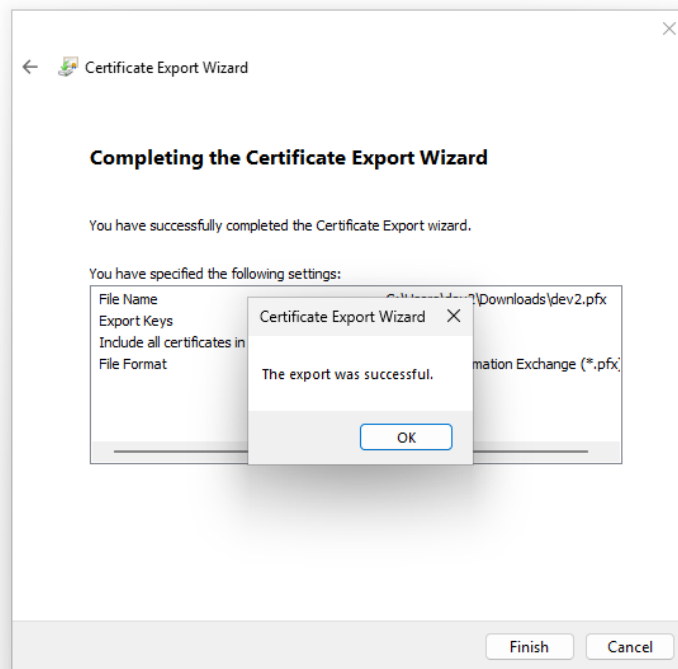
Choose path to store file

  Certificate Export Wizard 

File to Export
Specify the name of the file you want to export

File name:

Finish to export



2. Convert cert to .pem

Open Git bash on Window client and go to the path to download cert
And then run the commandline:

```
openssl pkcs12 -in dev2.pfx -out dev2-encrypted.pem
```

3. Config gitlab to clone

Open Window powershell

```
# Delete config domain-specific
```

```
git config --global --unset-all http."https://gitlab.zt.devsecops.local/".sslVerify
```

```
git config --global --unset-all http."https://gitlab.zt.devsecops.local/".sslCert
```

```
git config --global --unset-all http."https://gitlab.zt.devsecops.local/".sslKey
```

```
git config --global --unset-all
```

```
http."https://gitlab.zt.devsecops.local/".sslCertPasswordProtected
```

```
git config --global --unset-all http."https://gitlab.zt.devsecops.local/".sslBackend
```



```
# Delete config global
git config --global --unset-all http.sslCert
git config --global --unset-all http.sslKey
git config --global --unset-all http.sslCertPasswordProtected
git config --global --unset-all http.sslBackend
git config --global --unset-all http.sslVerify

# Switch to OpenSSL backend
git config --global http.sslBackend openssl

# Config cho GitLab domain
git config --global http."https://gitlab.zt.devsecops.local/".sslVerify false
git config --global http."https://gitlab.zt.devsecops.local/".sslCert
"C:/Users/dev2/Downloads/dev2-encrypted.pem"
git config --global http."https://gitlab.zt.devsecops.local/".sslKey
"C:/Users/dev2/Downloads/dev2-encrypted.pem"
```

4. Verify working

```
PS C:\Users\dev2> git config --global --list | findstr gitlab
http.https://gitlab.zt.devsecops.local/.sslverify=false
http.https://gitlab.zt.devsecops.local/.sslcert=C:/Users/dev2/Downloads/dev2-encrypted.pem
http.https://gitlab.zt.devsecops.local/.sslkey=C:/Users/dev2/Downloads/dev2-encrypted.pem
http.https://gitlab.zt.devsecops.local/.sslcertpasswordprotected=true
PS C:\Users\dev2>

PS C:\Users\dev2> git clone https://gitlab.zt.devsecops.local/dev2/test.git
Cloning into 'test'...
fatal: Invalid 'host' input argument (cannot be empty)
Password for 'cert:///C%3A/Users/dev2/Downloads/dev2-encrypted.pem':
fatal: Invalid 'host' input argument (cannot be empty)
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 3 (from 1)
Receiving objects: 100% (6/6), done.
```