



SDI

Setting up SSL connection to
1 - OpenPages
And
2 - GSA

Eddie Hartman – GBS Senior Consultant & SDI/TDI Storyteller

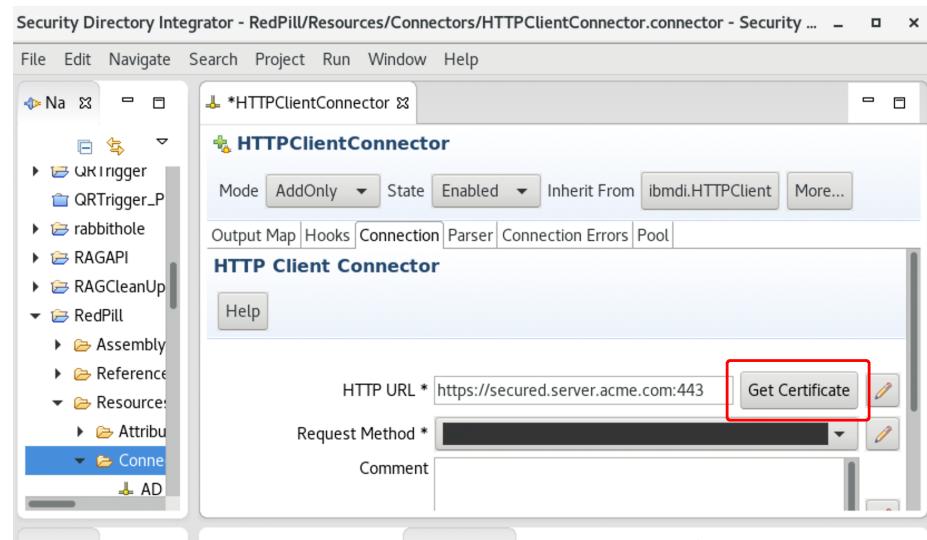
- May 20 2020



In order for SDI to make an SSL connection to OpenPages, it must have the necessary certificates in its keystore.

These include the client certificate and any other certs that trust of this one depends on.

Some servers and services provide for easy request of client certificates. With these you can simply set up an HTTP Client Connector in the SDI CE (Config Editor), configure the URL to the server (<https://...>) and press the **Get Certificate** button.



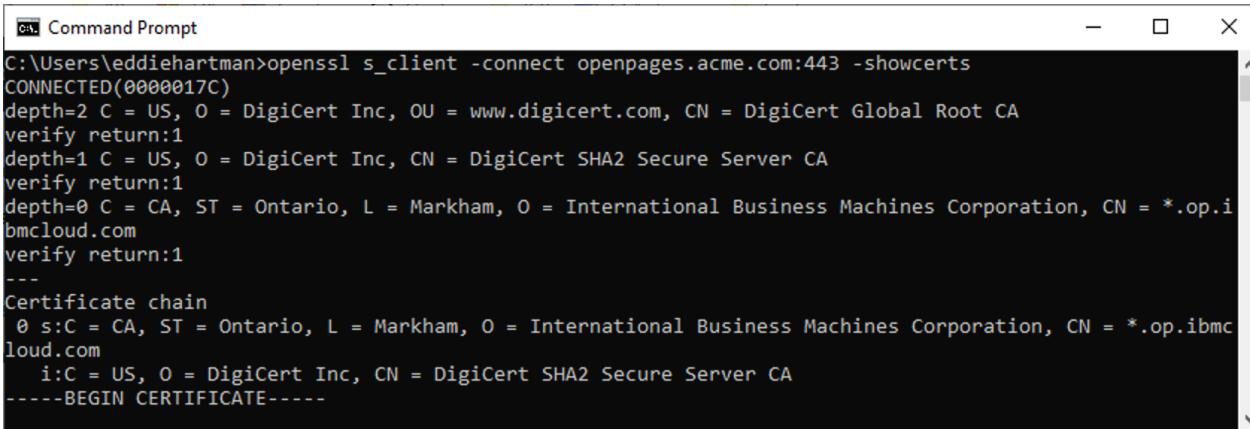
Unfortunately, OpenPages does not respond to this type of request.



1.a Requesting Certificates with openssl

Fortunately the **openssl** commandline utility provides functionality to return certificates from all kinds of services. This tool is available on various Linux distros, as well as on modern Windows 10 installations*.

Simply open a command window and fire up the **openssl** utility with the **-connect** argument followed by the hostname of the server, and including **-showcerts** to display certificates.



```
C:\Users\eddiehartman>openssl s_client -connect openpages.acme.com:443 -showcerts
CONNECTED(0000017C)
depth=2 C = US, O = DigiCert Inc, OU = www.digicert.com, CN = DigiCert Global Root CA
verify return:1
depth=1 C = US, O = DigiCert Inc, CN = DigiCert SHA2 Secure Server CA
verify return:1
depth=0 C = CA, ST = Ontario, L = Markham, O = International Business Machines Corporation, CN = *.op.ibmcloud.com
verify return:1
---
Certificate chain
  0 s:C = CA, ST = Ontario, L = Markham, O = International Business Machines Corporation, CN = *.op.ibmcloud.com
      i:C = US, O = DigiCert Inc, CN = DigiCert SHA2 Secure Server CA
-----BEGIN CERTIFICATE-----
```

This will result in a lot of output to the terminal window, including all certificates needed for a secure client connection. These will be in order from top-down starting with the client cert, possibly including an intermediate one, and ending with the root cert.



1.b Saving Certificates to Files



Each certificate must be copied, including the 'BEGIN' and 'END' lines, and saved to files labeled suitably and legibly.

The screenshot shows the Notepad++ interface with a file named 'new 1' open. The content of the file is a certificate block:

```
-----BEGIN CERTIFICATE-----
MIIRQjCCBiqAwIBAgIQBcUn2Jk9trO8mUhqYNC2jDANBgkqhkiG9w0BAQsFADB
-----
```

A 'Save As' dialog is displayed over the Notepad++ window, showing the path 'TDI on 'Mac' (X) > POC_MVP >'. The file name is 'digicert.crt' and the save type is 'All types (*.*)'. The save button is highlighted.

```
C:\ Select Command Prompt

Certificate chain
0 s:C = CA, ST = Ontario, L = Markham, O = International Business Machines Corporation, CN = *.op.ibmcloud.com
    i:C = US, O = DigiCert Inc, CN = DigiCert SHA2 Secure Server CA
-----BEGIN CERTIFICATE-----
MIIEhjCCAQgAwIBAgIQBcUn2Jk9tR08mUhQNYC2jDANBgkqhkiG9w0BAQsFADBN
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgsW5jMScwJQYDVQDExSE
aWdpQ2VydCBTSEEyIFN1Y3VyzSBTZJ2ZXIgQ0EwHhcNMTkwnZxE4MDAwMDAwWhcN
MjExMDIwMDAwMDAwIjCbgzELMKA1UEBhMCQ0ExEDAO8gNVBAgTB09udGfyaW8x
EDA0BgNVBTAhcmtoYw0NDayBgNVBAoTK01udGVybmf0aw9uYWwgQnVzalW5l
c3MgtWFjjaGluzXKmgQ9yjCG9yYXRpb24xjAyBGNVBAMMEsoub3AuawJty2vdwQu
Y29tMIIIBIAnBqkqhkiG9w0BAQEFAAOCAQ8AMIIIBcKCAQEAvyahJrqvNrvJZUuJ
3FivT-Bw+lnLybkrsQjbgwMRZTQxG+Wlvru7ncLsZ811vdN3I9sF+xwPf215qHE0
GnfMpYowoiF99AvQVA4XHL0eVBnI64FF4xrCqKtLaIwzIMNSg805wsToQsLpjRD
yjoftaBGGPsvnrlxFwFE5Yn8d1fE5tM+8XKA4ZCj7/cdehkr/S7etza1mtRPkwi
QLrzxN6pI4/Ff7FIsbgetPgt1FPQHdsLbowVaGC8fuNym
TLafwXVFzD1LefzHTWm2ee8z0ML8y6QyVgXhqE6mRuZh
CA+EhwYyDVR0jBggwFoAUD4BhHIXyDuvKoeNrji0LOHG
AdjukxhKL0TtvT7UzTbF63eMC0GA1udeQ0mMCSCESou
tgg9vcC5pYm1jB6912C5jb20wDgYDVR0PAQH/BAQDAgWg
GAQUFBwMBBggRgEEFBQcDAjBrBgnVHR8EZDBiMc+gLaR
kaWdpY2VydC5jb20vc3njYS1zaGeYLwczLmNybvDaOc2g
uzGlnaWN1cnQuY29tL3NzY2Etc2hhm1nNi5jcmwvTAYD
Bhv1sAQEWkjbAoBgrBgfEFBQcCARYcaHR0cHM6Ly93d3cu
lQUzAIIBgZngQwBAGiWfFAYIKwvBBQQUHAQEECDbuMCQGCCSg
vb2NzcCSkwdp2VydC5jb20wRgYIKwvBBQQUHAMKG0m0
/pz2J1zxJZx0LmNvb59eawdpQ2ydfNIQJTJZWN1cmvTZJ2
ITAQH/BAIwADCCAfGcIsGAQQB1nkCBAIEggHoBIIIB5AHi
izGdwCjw1mAT569+443FDNgS3BAAAFsBrLd4wAABAMA
q2T090J5C8sCnV/0FfmHgs6M1CBlgF4CIQD+oITg1Nst
/mosKA7uhMvD5SiwB2A1div+dZfpIMQ5lfvfNu/1aNR1Y2
/Bbaay3n8AAAQDAweCwRQIgLZ1dpG6T6QEsCpYr+3JCJKC21
/C1Qc1Kxb4hPr3fKmfJJfhG3+Q2TFeukrqtq7t5VmzUBES
H2Kj-KMDa5OK+2msxtT/TM51taGoAAAAbAay3ysAAQDQ
/KIFQVKSDnbFIH9BzlwPj08FpEIrtnNNHrAiAmMyoYNHiJ
/fhovX+wKlaIdDQ83AFYUbpov18ls0/XhvUsyPsdGdrm8
/Bbaay3rKAAQDAwegwRghAnjDe/MC6spiyQ1oWOnndoWHA
/8aiEafuk+spoez/4jpBkiWrzsv2/OnlhWL32/YfbwL9j
/LBQADggtBAL+2b9z5Ds2+9+/*0$Baaqjosc0Q56CJSK
81ovTag3BsmlArW0msuQOCjrn9N10d9RRvksJnY1bx0x28
/iVprL7jzDuk6V2D2KfkMsQsiHMHC1MiH8H0xEjsf5+b+UZ
/9BSJCRTrdkKfmfxndguswzQtpPAMWMY0rkneElktTvah/X
ya0b0Jj9G4Lw3C/4CA6zfroStBINH3vg6fbB398dAry0
k+HBMPV1Q8zTXPSL3RoKGQcNjzC+y2uc=
-----  

giCert Inc, CN = DigiCert SHA2 Secure Server CA  

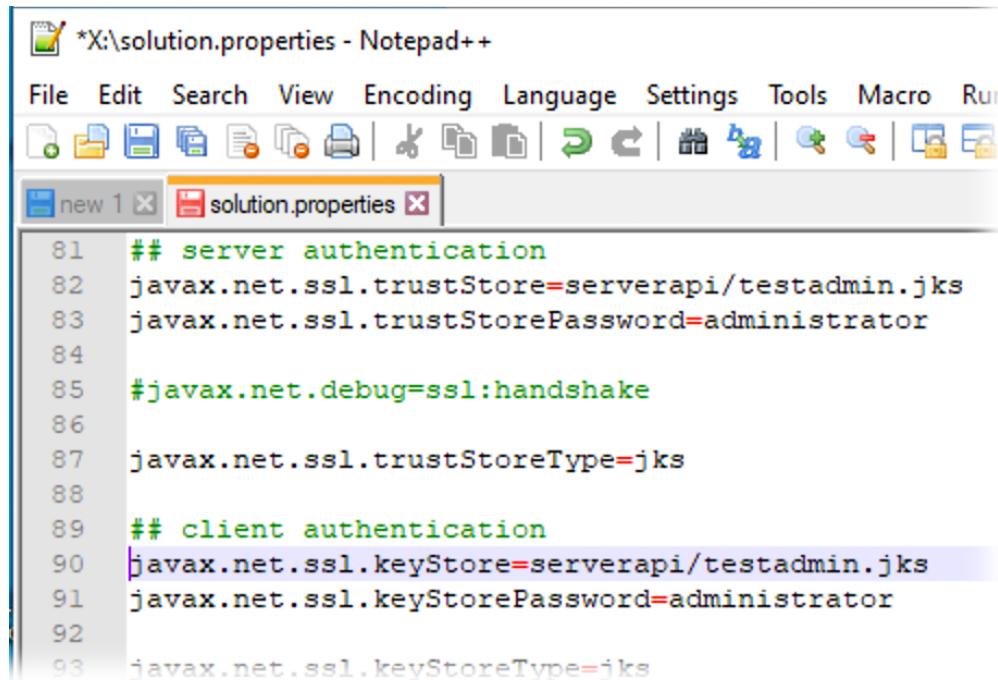
giCert Inc, OU = www.digicert.com, CN = DigiCert Global Root CA
-----  

QAF2j627KdciiQ4tyS+8kTANBgkqhkiG9w0BAQsFADbh
VMBMGA1UEChMMRGlnaUNlcnQgsW5jMRkwFwDVQOLExB3
-----  

d3cuZGlnaWN1cnQuY29tMSAwHgYDVQDExdEaWdpQ2VydCBhbG9iYWwgUm9vdCBD
```

1.c Destination = The SDI Keystore

The SDI Keystore is defined in the **solution.properties** file found in the *Solution Directory* of the SDI installation.



The screenshot shows a Notepad++ window with the title bar "X:\solution.properties - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, and Run. The toolbar has various icons for file operations like Open, Save, Find, and Copy. Below the toolbar, there are tabs for "new 1" and "solution.properties". The code editor displays the following properties:

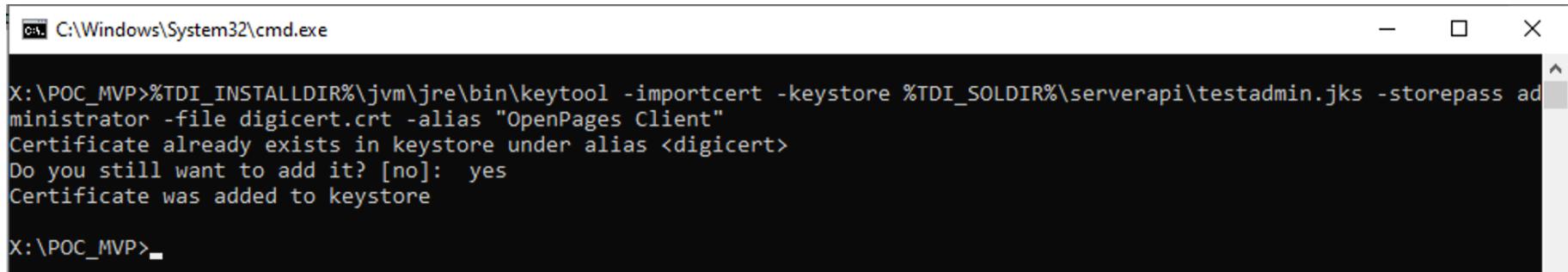
```
81 ## server authentication
82 javax.net.ssl.trustStore=serverapi/testadmin.jks
83 javax.net.ssl.trustStorePassword=administrator
84
85 #javax.net.debug=ssl:handshake
86
87 javax.net.ssl.trustStoreType=jks
88
89 ## client authentication
90 javax.net.ssl.keyStore=serverapi/testadmin.jks
91 javax.net.ssl.keyStorePassword=administrator
92
93 javax.net.ssl.keyStoreType=jks
```

By default this is the *testadmin.jks* file in the *serverapi* sub-folder of the *Solution Directory*, and with the password 'administrator'.

NOTE: To locate your *Solution Directory*, examine the **defaultSolDir** batchfile/script in the *bin* sub-folder of the SDI installation directory (V7.2).

1.d Importing Certificates using keytool

Use the **keytool** utility found in a Java JVM installation to install the certificates. You can use your default Java installation, or the one in the *jvm/jre/bin* sub-folder of your SDI installation.



```
C:\Windows\System32\cmd.exe
X:\POC_MVP>%TDI_INSTALLDIR%\jvm\jre\bin\keytool -importcert -keystore %TDI_SOLDIR%\serverapi\testadmin.jks -storepass ad
ministrator -file digicert.crt -alias "OpenPages Client"
Certificate already exists in keystore under alias <digicert>
Do you still want to add it? [no]: yes
Certificate was added to keystore
X:\POC_MVP>
```

Execute the **keytool** utility with the following arguments:

- | | |
|-------------|--|
| -importcert | <i>command to import certificate</i> |
| -keystore | <i>filepath to the keystore</i> |
| -storepass | <i>password for the keystore</i> |
| -file | <i>certificate file to import</i> |
| -alias | <i>descriptive alias for this cert</i> |



Whenever you make changes to your keystore or truststore – which by default are the same file (testadmin.jks) – you must restart SDI for the changes to take effect.

Once restarted you can test connectivity to ensure an encrypted connection can be established.



To enable password-less file transfer using the **scp** utility requires two steps:

1. The generation of a public/private keypair
This we do using a utility like **ssh-keygen**

2. Configuring the target GSA account with the public key
This means updating a file in the `.ssh` sub-folder of the GSA account's home folder and ensuring the *ACLs* of the folder and files are correct.

NOTE: These instructions work for Centos, which is the OS of the servers operated by Cloud Ops. The approach will likely work for other Linux distros as well, since the **scp** utility is available for all. However, achieving password-less file transfer using **winscp** on Windows has eluded me (so far).



2.a Using ssh-keygen To Generate Keychain



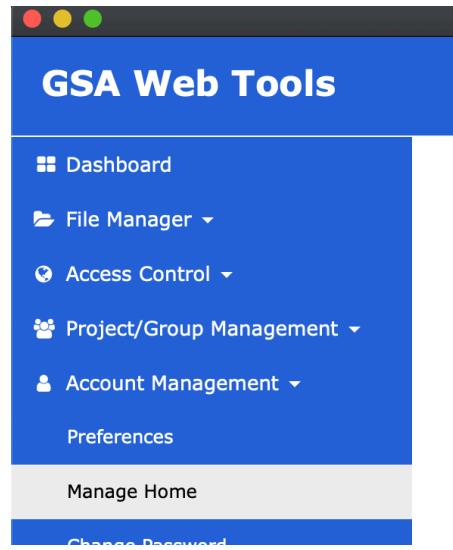
The **ssh-keygen** utility lets you easily create the public/private keypair you need. Simply run the commandline from a terminal. If you do not specify a filepath it will update the keychain of the current user (i.e. `~/.ssh/id_rsa` & `~/.ssh/id_rsa.pub`).

```
eddie@centos-7:~/tmp
File Edit View Search Terminal Help
[eddie@centos-7 tmp]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/eddie/.ssh/id_rsa): ./id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ./id_rsa.
Your public key has been saved in ./id_rsa.pub.
The key fingerprint is:
SHA256:kX5yioiwJelvQ8B4QloxNtBCFJPFWCC8IhYZJhY3dDI eddie@centos-7.shared
The key's randomart image is:
+---[RSA 2048]---+
|0#^E.o
|B@+**
|...= o o
|o+ . . .
|B .. S o
|.=. . . =
|.... . . .
|
+---[SHA256]---+
[eddie@centos-7 tmp]$
```



2.b GSA Account Must Have A Home Folder

The **id_rsa.pub** file contains the public key. Copy the contents of this file to your copy buffer. It must be appended to the end of a file named **authorized_keys** which is located in the **.ssh** sub-folder of the home directory of the GSA account being used.



For example, if the GSA account is 'itrmop' then you can use the appropriate GSA Web Tools to authenticate and configure the the home directory for your account (for example, this one for POKGSA:

<https://itrmop@pokgsa.ibm.com/gsadoc/> - the Web Tools link is at the top left).

2.c Home Folder Must Have .ssh Sub-Folder



Once the Home folder is in place then the next step is to ensure it has the `.ssh` sub-folder with the correct ACLs set. You can set these using the GSA Web Tools. Access should be *User: all, Group: R, World: R*

The screenshot shows the GSA Web Tools interface for modifying ACLs. The left sidebar has a 'Modify ACL' button highlighted. The main area shows an 'Advanced Modify ACLs' form with the path `/gsa/pokgsa/home/i/t/itrmop/.ssh` entered. The table below lists ACL entries:

Type	Name	Read	Write	Execute	Control	Remove User/Group
Owner ID	itrmop	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	n/a
Owner Group	itrmop	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	n/a
Other	World	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	n/a
Mask	Mask_Obj	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	n/a
New Users	Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	n/a
New Groups	Click on selection	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	n/a

2.d Inside .ssh Must Be The File **authorized_keys**



For a server to accept SSH connections, it must have a list of all accepted client keys. This is the **authorized_keys** file found in the **.ssh** sub-folder of the home directory.

Append this file with the contents of the **id_rsa.pub** file created in step **b** above. The single line of text should start with: **ssh-rsa**

Ensure this file has similar ACLs as the **ssh** sub-folder itself, except None for *World*.

Path: /gsa/pokgsa/home/i/t/itrmop/.ssh/authorized_keys [Change Path](#)

Type	Name	Read	Write	Execute	Control	Remove User/Group
Owner ID	itrmop	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	n/a
Owner Group	itrmop	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	n/a
Other	World	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	n/a
Mask	Mask_Obj	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	n/a
New Users	<input type="text"/> Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	n/a
New Groups	<input type="text"/> Click on selection	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	n/a

[Submit](#) [Cancel](#)



2.e Test Password-Less File Transfer

To test if setup is complete, try to transfer a file from your local system to the home folder of the GSA account using the **scp** utility.

```
eddie@centos-7:/media/psf/Home/My_Documents/TDI/POC_MVP - □ ×  
File Edit View Search Terminal Help  
[eddie@centos-7 POC_MVP]$ scp -i id_rsa test.csv itrermop@pokgsa.ibm.com:/pokgsa-h  
3/06/itrmop/  
test.csv 100% 16 0.1KB/s 00:00  
[eddie@centos-7 POC_MVP]$ █
```

Use the **-i** argument to indicate where the key files are located. The above example is run from a directory containing these key files. Then use a tool like **FileZilla** or **sftp** to list files in that directory and confirm that the file was successfully transferred.

```
eddie@centos-7:/media/psf/Home/My_Documents/TDI/POC_MVP - □ ×  
File Edit View Search Terminal Help  
[eddie@centos-7 POC_MVP]$ sftp itrermop@pokgsa.ibm.com:/pokgsa-h3/06/itrmop  
Connected to pokgsa.ibm.com.  
Changing to: /pokgsa-h3/06/itrmop  
sftp> ls  
test.csv web  
sftp> █
```

