

# **CSCT Cloud Server: Accessing Remote Server using Azure CLI and SSH Keys**

**for**

## **Web Development and Databases\* (UFCFES-30-1)**

**Version 2.5**

Zaheer Khan  
Owen Jones

\* This guide can also be used by other modules which require access to CSCT Cloud.

**Credits:** Contents are adapted from Version 0.99 where SSH key generation and copying to CSCTCloud was prepared by Dr. Kamran Soomro.

## Table of Contents

<b>Table of Contents</b>	<b>2</b>
<b>List of Figures</b>	<b>3</b>
<b>1. About</b>	<b>4</b>
1.1 Using your own personal machine	4
<b>2. Running Commands</b>	<b>5</b>
2.1 Choosing a terminal program	5
2.2 Installing OpenSSH and Azure CLI	6
<b>3. Generating SSH Keys</b>	<b>7</b>
3.1 Where to save your keys	7
3.2 Setting a passphrase	8
3.3 Copying your keys to H:\ drive	8
<b>4. Accessing CSCTCloud using your SSH keys</b>	<b>9</b>
4.1 Logging in to CSCTCloud for the first time	9
4.2 Copying your public SSH key to the remote server	10
4.3 Securing your public key	11
4.4 Testing your SSH keys	11

## List of Figures

Figure 1: Running PowerShell as administrator.....	5
Figure 2: Opening the C drive on a UWE lab machine.....	8
Figure 3: A terminal prompt on CSCTCloud .....	9
Figure 4: Copying your SSH public key.....	10
Figure 5: Create a directory called .ssh and open .ssh/authorized_keys for editing .....	10
Figure 6: Paste your SSH public key into the file .....	10
Figure 7: Set appropriate directory and file permissions .....	11

## 1. About

This document provides a step-by-step guide on how to create SSH keys and setup secure access to the CSCTCloud server.

CSCTCloud runs the *Ubuntu 20.04* operating system and therefore we'll use different command line/terminal tools and commands. This will be a good start for Computer Science students to see how a secure distributed system can be setup. That's great – isn't it?

Setting up above secure connection will enable students to:

1. Access MySQL server from CSCTCloud and manage their data in a dedicated database
2. Run Python Flask scripts (or C programs) on remote server, if needed
3. Connect GUI tools with CSCTCloud e.g., MySQL Workbench, VSCode

**You will need to follow these instructions if using UWE lab machines**

### 1.1 Using your own personal machine

You only need this guide if you want to use the CSCTCloud server. For example, on lab computers we'll need this guide to access CSCTCloud to manage data in MySQL Server.

If you want to access CSCTCloud by using your personal computer then you will need to install and configure additional software such as OpenSSH and Azure CLI, without these you will not be able to connect to the CSCTCloud server.

If you plan to install all required software (Python, VSCode, Workbench, MySQL server) on your personal machine then you can skip this guide. Please note all required software versions are free to download and use on your personal machines.

Discuss with your tutor what's the best approach for you.

## 2. Running Commands

### **If you're using a UWE lab machine**

On UWE lab machines you should already be able to run **ssh** and **az** commands at the Command Prompt (click on Start, then type Command Prompt), so you can skip this section.

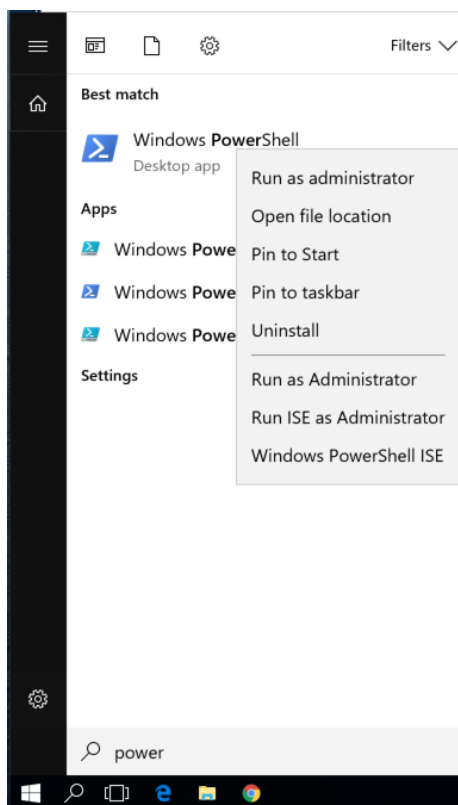
If you are using your own personal machine, then you will need to identify a terminal/command line program to use. You will also need to enable the OpenSSH Client (if using Windows) and install the Azure CLI (Command-Line Interface) tool to be able to connect to the CSCTCloud server.

### 2.1 Choosing a terminal program

#### Windows

Windows 10 includes two terminal programs: Command Prompt, and PowerShell; you can find these by searching (clicking start and then typing the name of the program).

Microsoft also have a new 'Windows Terminal' app, which can be installed from the Microsoft store: <https://www.microsoft.com/en-gb/p/windows-terminal/9n0dx20hk701>



You may sometimes need to run your command line program as an administrator. To do this find the program (click start and then type the name of the program), then right mouse click to run it as Administrator – see Figure 1.

#### macOS

macOS comes with a built-in terminal program—you can find this by searching (Cmd+Space) for 'Terminal'.

Several third-party terminal programs also exist, including iTerm2 (<https://iterm2.com/>) and Hyper (<https://hyper.is/>).

#### Linux

The built-in terminal program varies depending on which Linux distribution you're running, and most distributions will include several programs—you'll need to check which are installed on your machine and pick one.

*Figure 1: Running PowerShell as administrator*

## 2.2 Installing OpenSSH and Azure CLI

OpenSSH is a tool for remotely logging into servers using the SSH protocol. It's normally already installed on Linux/macOS, Windows users will need to enable the OpenSSH Client using the instructions at:

[https://docs.microsoft.com/en-us/windows-server/administration/openssh/openssh\\_install\\_firstuse](https://docs.microsoft.com/en-us/windows-server/administration/openssh/openssh_install_firstuse)

The Azure CLI (Command-Line Interface) tool allows you to interact with resources hosted on Microsoft's Azure cloud platform, including the CSCTCloud server.

Instructions to install the CLI tool can be found on Microsoft's website:

- Windows: <https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-windows>
- Linux: <https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-linux>
- macOS: <https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-macos>

**Assumption:** OpenSSH is already enabled on the machine you are using—if not, see section 2.

Creating SSH keys will allow you to securely connect to the CSCTCloud server using public key authentication. You can read more about SSH keys at:

- <https://www.ssh.com/ssh/key/>
- <https://www.ssh.com/ssh/openssh/>

You can generate SSH keys by running the **ssh-keygen** command in a terminal program. The following command will generate a set of public and private SSH keys:

```
ssh-keygen -t rsa
```

The command will ask you a few questions: the location in which to save the keys, and to set a security passphrase.

*In the command you're specifying the RSA digital signature algorithm. Similarly, you may specify other digital signature algorithms e.g., dsa, ecdsa, ed25519 – see more details at <https://www.ssh.com/ssh/keygen/>.*

### 3.1 Where to save your keys

Save the keys to the default suggested location for your operating system:

Operating System	Default location to save keys
Windows	C:\Users\ <username>\.ssh\id_rsa</username>
Linux	/home/<username>/.ssh/id_rsa
macOS	/Users/<username>/.ssh/id_rsa

#### **If you're using a UWE lab machine**

You need to copy your SSH keys from your current lab machine to your H: \ drive – see steps below after you've finished generating your keys.

### 3.2 Setting a passphrase

The command will ask you to enter a passphrase. You could leave the passphrase empty, but for security reasons you should enter a memorable passphrase to protect your keys from unauthorised use.

**Make sure you don't forget this passphrase; you'll need it to connect to the remote server.**

### 3.3 Copying your keys to H:\ drive

#### If you're using a UWE lab machine

The default location used to save SSH keys (on the `C:\` drive) is not shared between UWE computers, and so if you stored your keys there you would need to manually copy them between different lab machines or re-generate them and add them to the server for each machine you use.

To avoid doing this you need to instead copy your keys to your `H:\` drive and store them there, and then use these keys to connect to the server.

Find your home folder on your current machine (`C:\Users\<username>`). On UWE lab machines the `C:\` drive is hidden from view, to access it you need to click onto the address bar in a *File Explorer* window, and type "`C:`", followed by `<Enter>` as in Figure 2.

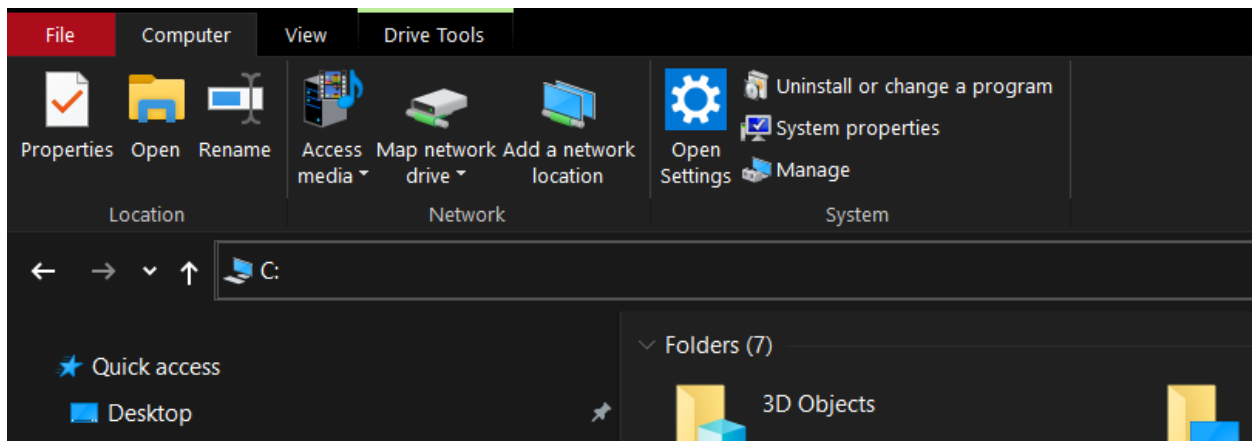


Figure 2: Opening the C drive on a UWE lab machine

Copy the `.ssh` folder, then navigate to your `H:\` drive and paste it there. You should now have a `.ssh` folder containing your SSH keys (`id_rsa` and `id_rsa.pub`) that can be accessed on any UWE lab machine.



## 4. Accessing CSCTCloud using your SSH keys

**Assumption:** Azure CLI is already installed on the machine you are using—if not, see section 2.

In order to login to CSCTCloud using public key authentication, you need to copy your public key (which you’ve just generated above) onto the remote server and properly configure it.

### 4.1 Logging in to CSCTCloud for the first time

The first time you login to CSCTCloud you will need to sign in using the Azure CLI tool and your UWE login credentials.

Run the command **az login** on command prompt or terminal to setup your account, this will open a window/tab in your default browser asking you to login to Microsoft Azure. Select your UWE account (or enter your UWE email address) and then enter your password. Once you have successfully logged in you can close the browser page and return to your terminal program. You should also be able to see Azure resource details in the command prompt window.

You can then connect to CSCTCloud using the following Azure CLI command:

```
az ssh vm --ip csctcloud.uwe.ac.uk
```

*This may ask you to install the Azure CLI ssh extension; enter `<y>` then `<Enter>` to install this, then try running the command again.*

You should now be logged in to the server and will find yourself at a terminal prompt in your newly created home directory, as in Figure 3. This means now you should be able to run commands (e.g., execute your code) on CSCTCloud.

```
PS C:\Users\oe-jones> az ssh vm --ip csctcloud.uwe.ac.uk
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1020-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon Nov  8 16:50:21 GMT 2021

System load:  0.0               Processes:            363
Usage of /home: 27.4% of 599.99GB Users logged in:      0
Memory usage:  1%               IPv4 address for eth0: 10.0.90.5
Swap usage:    0%

6 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Mon Nov  8 16:39:24 2021 from 164.11.214.93
owen.jones@uwe.ac.uk@csctcloud:~$
```

Figure 3: A terminal prompt on CSCTCloud

## 4.2 Copying your public SSH key to the remote server

You will need to copy your public key (`id_rsa.pub`) from the file location you specified when you created it (or your `H:\` drive if using a lab machine), to your clipboard. You can do this using a text editor, or from your terminal program using `cat <file location>` (e.g., `cat .ssh/id_rsa.pub`). If you're using Windows command prompt you can use the `type` command instead (e.g., `type H:\.ssh\id_rsa.pub`).

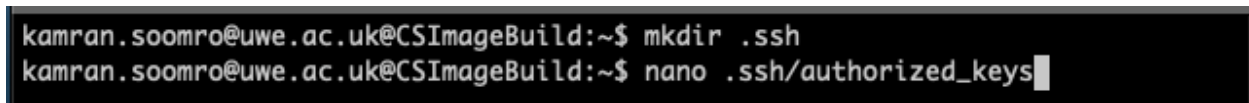
You should then see your SSH key printed in the terminal, as shown in Figure 4. Select your SSH key and copy this text using `<Ctrl+c>` or `<Cmd+c>` as normal (if using Command Prompt you might need to press `<Enter>` to copy instead).



```
k-soomro@S-FET-4Q58-01 ~  
> cat .ssh/id_rsa.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDzSG5sRckW2JgZaos9yI+0p/FpM26a1egNemfmUZPA  
pLYWi5tfbDgKUKohbFjd+O+hFNvMbhME9ntZwHB3LELiNO8Ak91d5I0WzYV0twVH9s8GubBK/ffAvNRw  
VEtbdlYVoVo6h4cDYVDKjhoXx1ZRwq0VQjN+/CaVyAiqzKi05up3wbX24szEn52AI6rp844WvlxRJp6+  
gGTBINSb5fzIQpSGZ3DONWNUat74crhcD9h+BGxJsdCWgcXR7Zjpos09wF2wvHri5MK10P85dXqADtrs  
unNl3sdUummp5fRVdTTmXmIstx4CciJm2JnLE/03RjFpdyoNLHGgHz2+d5R kamran.soomro@uwe.a  
c.uk
```

Figure 4: Copying your SSH public key

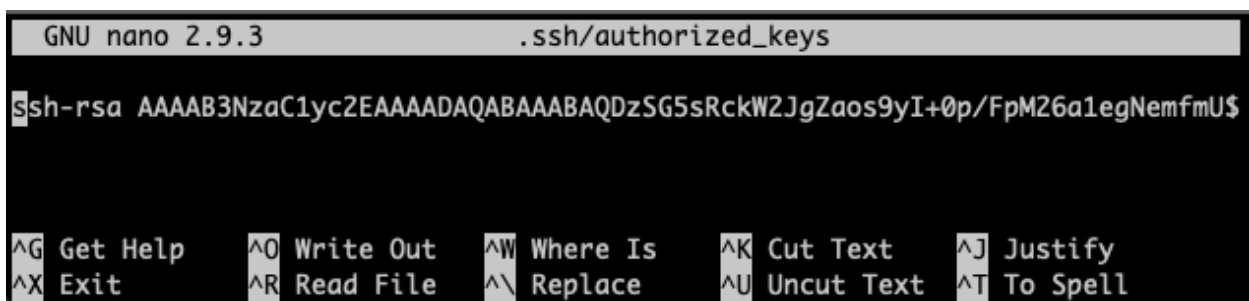
In your home directory on CSCTCloud create a new directory called `.ssh` (`mkdir .ssh`), then create a new file called `authorized_keys` within this directory using the text editing program `nano` (`nano .ssh/authorized_keys`). *Note the American spelling: 'authorized'!*



```
kamran.soomro@uwe.ac.uk@CSImageBuild:~$ mkdir .ssh  
kamran.soomro@uwe.ac.uk@CSImageBuild:~$ nano .ssh/authorized_keys
```

Figure 5: Create a directory called `.ssh` and open `.ssh/authorized_keys` for editing

Paste your public key that you copied above using `<Ctrl+v>` or `<Cmd+v>` (if using Command Prompt you might need to right click to paste instead). You can save and close the file using `<Ctrl+x>`, followed by `<y>` and `<Enter>`.



```
GNU nano 2.9.3 .ssh/authorized_keys  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDzSG5sRckW2JgZaos9yI+0p/FpM26a1egNemfmU$  
  
^G Get Help      ^O Write Out    ^W Where Is    ^K Cut Text     ^J Justify  
^X Exit          ^R Read File    ^\ Replace     ^U Uncut Text  ^T To Spell
```

Figure 6: Paste your SSH public key into the file

If you generate additional sets of SSH keys (e.g., you generate SSH keys on both a lab machine and on your own personal machine), you can add these public keys to your `authorized_keys` file also by pasting them onto a new line. Then you can use both sets of keys to access the server via SSH.

### 4.3 Securing your public key

You need to make sure the correct permissions are set on the directory and file you've just created to prevent unauthorised access. This is done by running the commands `chmod 744 .ssh` and `chmod 644 .ssh/authorized_keys` (see Figure 7).

```
kamran.soomro@uwe.ac.uk@CSImageBuild:~$ chmod 744 .ssh
kamran.soomro@uwe.ac.uk@CSImageBuild:~$ chmod 644 .ssh/authorized_keys
kamran.soomro@uwe.ac.uk@CSImageBuild:~$
```

Figure 7: Set appropriate directory and file permissions

### 4.4 Testing your SSH keys

You should now be all set to log in with your SSH keys. You can check this by logging out (by running the command `exit`), and then logging back in again using your keys.

To log back in using your SSH keys and not the Azure CLI you need to run:

```
ssh <your UWE email>@csctcloud.uwe.ac.uk
```

Substituting your full UWE email address (e.g., jane.smith@live.uwe.ac.uk) into the command. **Make sure you only use lowercase letters when entering your email address.**

When prompted, enter the passphrase you used when you generated your keys.

#### **If you're using a UWE lab machine**

By default, the SSH program looks for SSH keys stored in your local home directory (e.g., `C:\Users\<username>\.ssh\id_rsa` on Windows).

This location is not shared between machines at UWE, which is why you needed to copy your keys to your `H: \` drive when you generated them earlier.

In order to login using SSH on lab machines you will need to tell it where to find your keys, using the `-i` flag:

```
ssh -i H:\.ssh\id_rsa <your UWE email>@csctcloud.uwe.ac.uk
```

**For staff only:** You may need to change security permissions on your private key on H:\ to 'full control'. You can do it by right mouse click and from properties you should be able to see security tab from where you can select your user name and check 'full control' checkbox.

If your login was successful, you should find yourself at a terminal prompt on CSCTCloud again (Figure 2).

If you receive an error (`Permission denied (publickey)`) then you should go back and check you've copied your public key to the correct location on CSCTCloud, or if using a UWE lab machine have told SSH where to find your keys using the command with the `-i` flag above, and make sure you enter your UWE email address using lowercase letters only.