Data Modelling Tools

AUTM08016

Topic 1a
Build an AWS Cloud Platform

Dr Diarmuid Ó Briain
Version 1.0 [01 January 2024]

TUS
Ollscoil Teicneolaíochta na Sionainne:
Lár Tíre, An tIarthar Láir
Technological University of the Shannon:
Midlands Midwest

Dr Diarmuid Ó Briain

CEng, FIEI

CISSP®

Table of Contents

Table of Figures

1. Amazon AWS

Amazon Elastic Compute Cloud (EC2) delivers scalable, pay-as-you-go compute capacity in the cloud. Amazon AWS offers Auto Scaling and Elastic Load Balancing.

- Auto Scaling allows for the automatic scaling of EC2 capacity up or down according to conditions defined.
- Elastic Load Balancing automatically distributes incoming application traffic across multiple EC2 instances.

AWS has a free usage tier that can be used, for example, launch new applications, test existing applications in the cloud, or simply gain hands-on experience with AWS.

EC2 offers several free basic Amazon Machine Images (AMI), the cloud EC2 instance in this laboratory will use a Debian GNU/Linux EC2 image.

1.1 Create an AWS account

- Go to http://aws.amazon.com, and then click **Sign Up**.
- Follow the on-screen instructions.
- Set **Europe (Ireland) eu-west-1** as the preferred region.

1.2 AWS Console

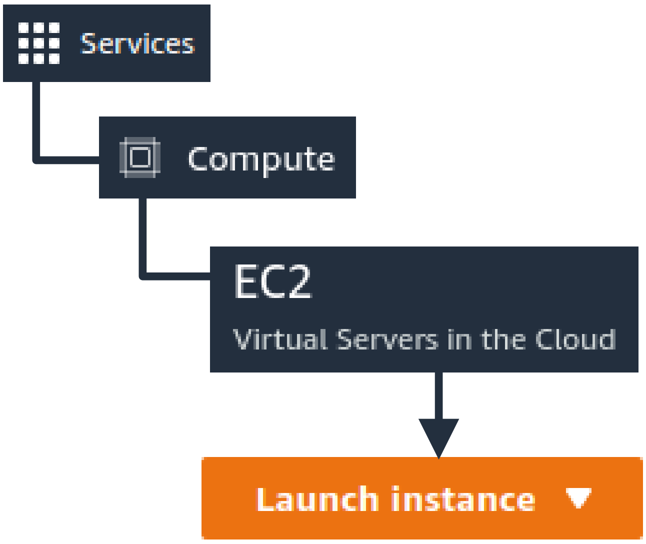
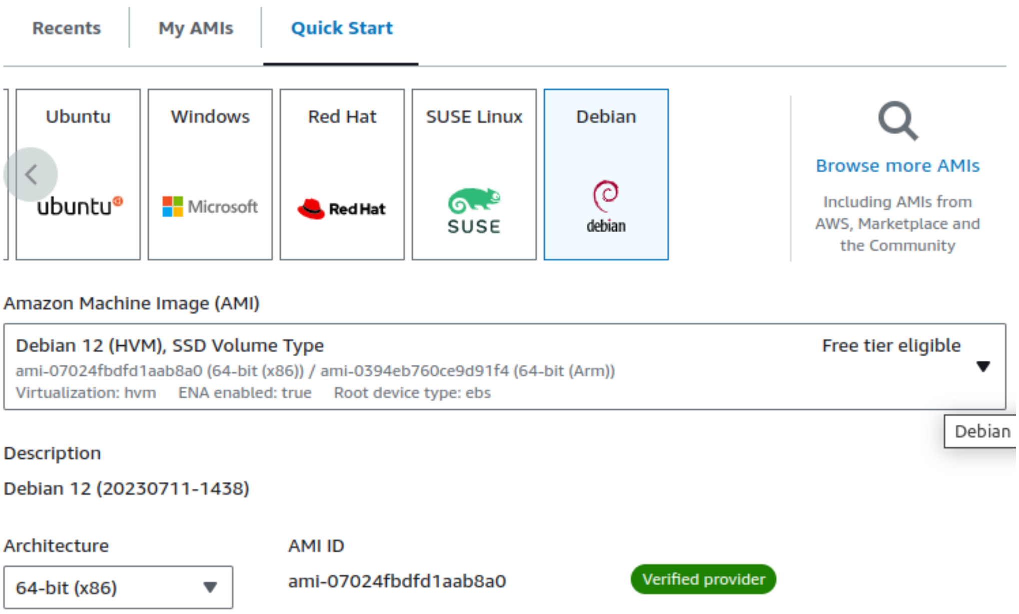


Figure 1: AWS EC2 instance

As in Figure 1, select **Services** on the top left of the console, then **Compute** → **EC2** and press the **Launch instance** icon.

1.3 Application and OS images

As there has not been an instance already created, go to the **Quick Start** option demonstrated in Figure 2, select **`debian`** and the version. The **Free tier eligible** instance is fine.



Amazon Machine Image (AMI)

Debian 12 (HVM), SSD Volume Type — Free tier eligible
ami-07024fbdfd1aab8a0 (64-bit (x86)) / ami-0394eb760ce9d91f4 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Debian

Description

Debian 12 (20230711-1438)

Architecture: 64-bit (x86)

AMI ID: ami-07024fbdfd1aab8a0

Verified provider

Figure 2: Debian Server Image

From here select the **Instance type** drop-down, notice that the instance is a **t2.micro** with 1 virtual Central Processing Unit (vCPU) and 1 GiB or 1024^3 (1,073,741,824) bytes whereas 1 GB is 1000^3 (1,000,000,000) bytes, thus the instance has 1.07 GB of memory.

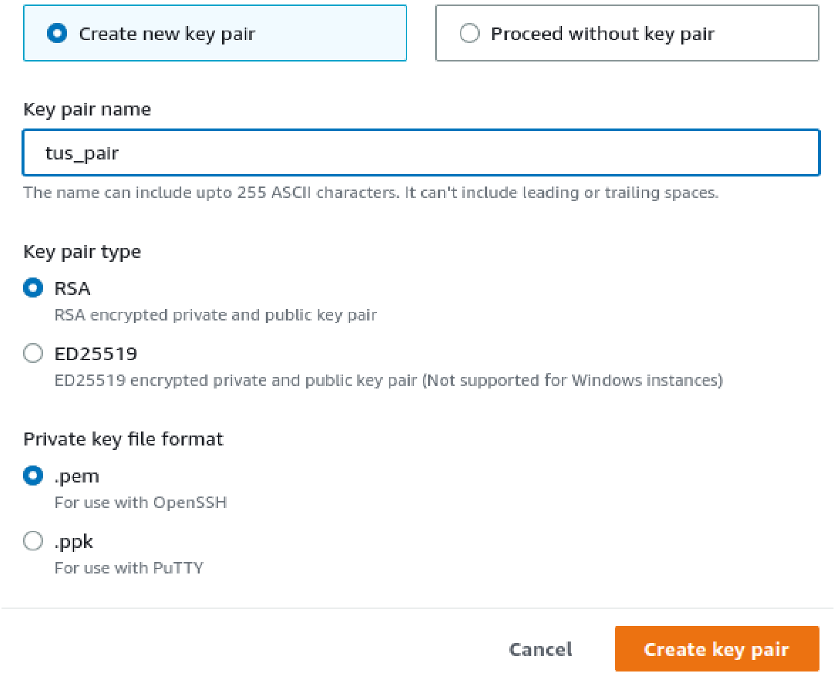
1.4 Secure Key Pair

The next drop-down is **Key pair (login)**, again as this is the first instance there is no existing pair so select **create new key pair** as shown in Figure 3. This gives an option to download a **.pem** file for use with OpenSSH or a PuTTY Private Key, **.ppk**, file for use with PuTTY. (PuTTY is the client for Secure Shell (SSH) used on Microsoft Windows). Click on **Create key pair** icon to download.

As displayed in Figure 4, a **.pem** file can easily be converted to a **.ppk** file:

- Use **PuTTYgen** and select the key to generate: **SSH-2 RSA**
- At **Load an existing private key file** select **Load**, locate the **.pem** file (select All Files(*.*)), **Open** and **OK**.
- Click **Save private key** and select **Yes** to the warning about saving without a passphrase.
- Specify the name for the new file which will have a **.ppk** extension.

It is essential that this file is maintained in a safe place. Loosing it means that access to the instances that are associated with it can no longer be accessed.



● Create new key pair ○ Proceed without key pair

Key pair name

tus_pair

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

● RSA
RSA encrypted private and public key pair

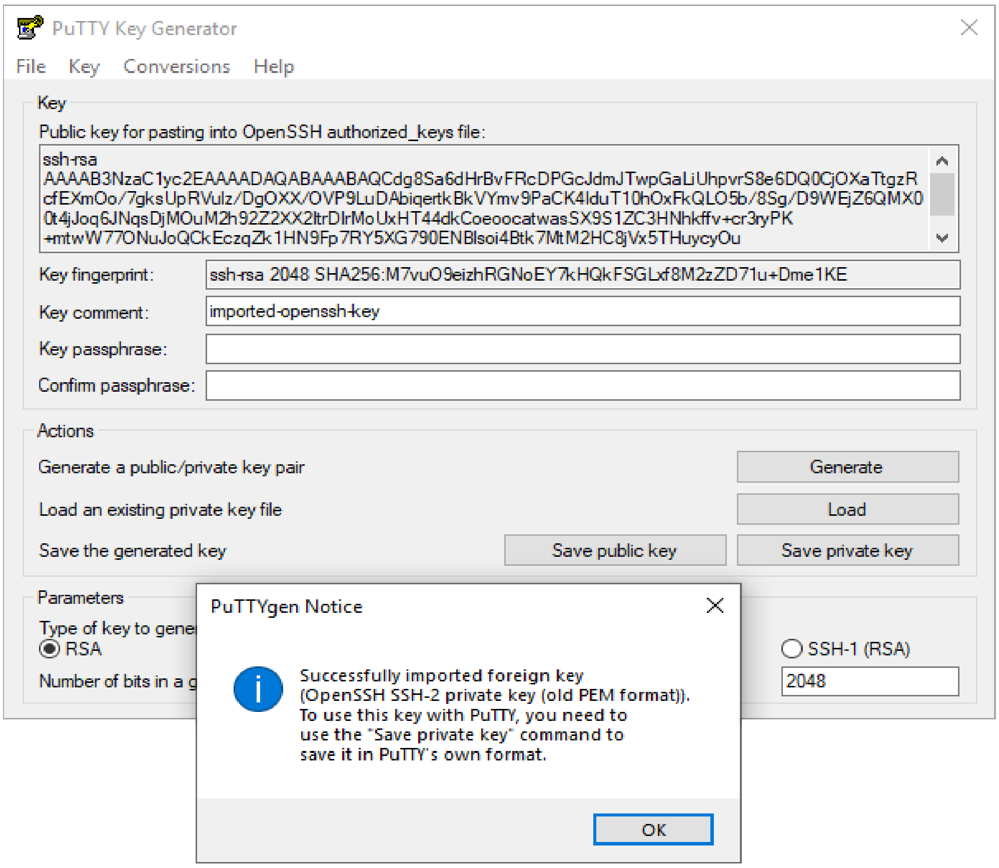
○ ED25519
ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format

● .pem
For use with OpenSSH

○ .ppk
For use with PuTTY

Figure 3: Create Key Pair



Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCdg8Sa6dHrBvFRcDPGcJdmJTwpGaLiUhpvrS8e6DQ0CjOXaTtgzR
cfEXmOo/7gksUpRVulz/DgOXX/OVP9LuDAbiqertkBkVYmv9PaCK4lduT10hOxFkQLO5b/8Sg/D9WEjZ6QMX0
0t4jJoq6JNqsDjMOuM2h92Z2XX2ltrDIrMoUxHT44dkCoeoocatwasSX9S1ZC3HNhkffv+cr3ryPK
+mtwW77ONuJoQCkEczqZk1HN9Fp7RY5XG790ENBlsoi4Btk7MtM2HC8jVx5THuycyOu
```

Key fingerprint: ssh-rsa 2048 SHA256:M7vuO9eizhRGNoEY7kHQkFSGLxf8M2zZD71u+Dme1KE

Key comment: imported-openssh-key

Key passphrase:

Confirm passphrase:

Actions

Generate a public/private key pair

Load an existing private key file

Save the generated key

Parameters

Type of key to gene
● RSA ○ SSH-1 (RSA)

Number of bits in a g 2048

PuTTYgen Notice

Successfully imported foreign key
(OpenSSH SSH-2 private key (old PEM format)).
To use this key with PuTTY, you need to
use the "Save private key" command to
save it in PuTTY's own format.

Figure 4: PuTTYGen to convert .pem to .ppk

1.5 Instance environment

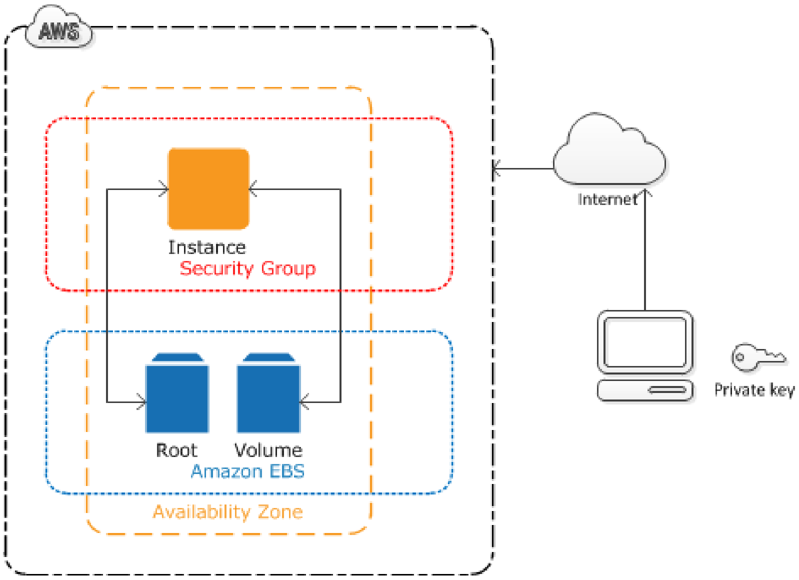
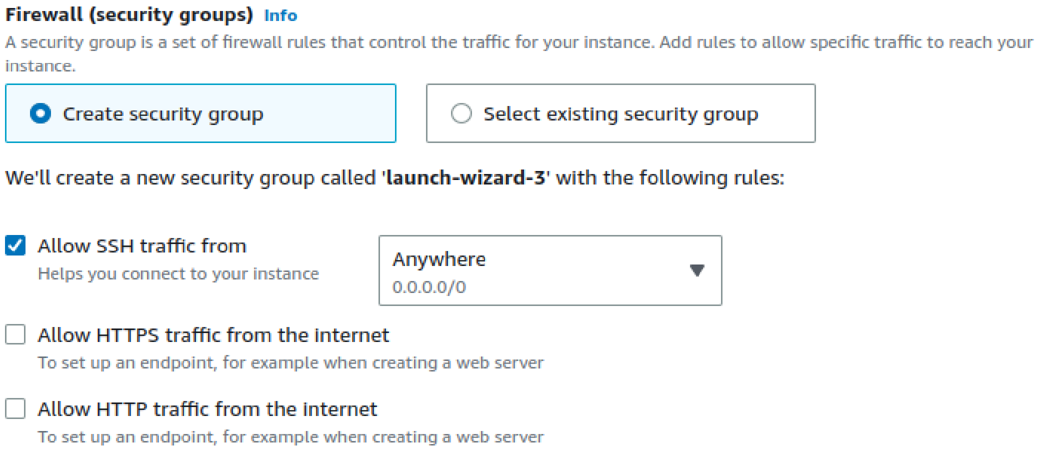


Figure 5: Instance Environment

Refer to Figure 5, each instance, such as the one being created, is protected by a **Security group** which can be accessed using SSH with a Private key (**.pem** or **.ppk**). This is a minimalist firewall configured through the AWS dashboard. The instance is also given access to an Elastic Block Store (EBS), essentially the equivalent of computer hard-drive, a virtual hard-drive.

1.6 Network settings



Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

We'll create a new security group called '**launch-wizard-3**' with the following rules:

☑ Allow SSH traffic from
Helps you connect to your instance
Anywhere ▼
0.0.0.0/0

□ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

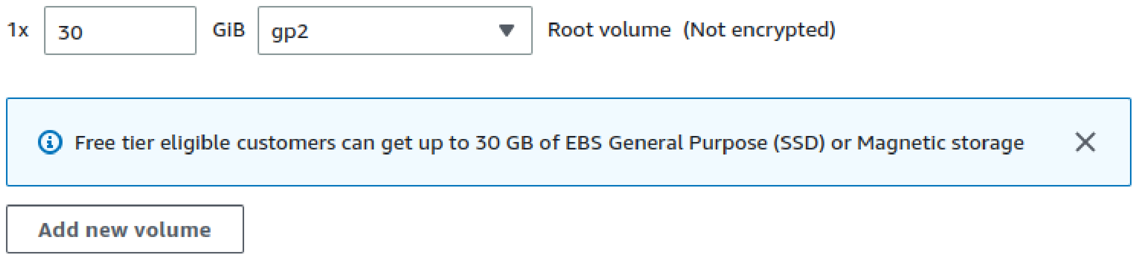
□ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

Figure 6: Security Group

From the **Network settings** drop-down select **create security group**. This offers a very basic firewall selection to allow or disallow SSH, HTTP and HTTPS traffic. Start by only allowing **SSH** traffic from **0.0.0.0/0**, essentially a wildcard representing anywhere. Obviously it is better if this is limited to the IP address or a subnet of the workstation if possible.

1.7 Configure storage

Amazon Elastic Block Store (EBS) provides block level storage volumes for use with EC2 instances. EBS volumes are network-attached, and persist independently from the life of an instance. EBS provides highly available, highly reliable, predictable storage volumes that can be attached to a running EC2 instance and exposed as a device within the instance. EBS is particularly suited for applications that require a database, file system, or access to raw block level storage.



1x 30 GiB gp2 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Figure 7: Add Storage Volume

Select the **Configure storage** drop-down and Amazon permit up to 30 GiB per account on the free tier.

1.8 Running the new instance

In the EC2 Dashboard the new instance will be visible. Select the **Instance state** drop-down and select **Start instance** if the state is not running. In this drop-down is the facility to Start or Reboot, Hibernate or Terminate an instance. The dashboard should not show ☑ **Running** and and IP address will appear in the Public IPv4 address field. Note this address: ___________________ .

1.9 Default Usernames for Amazon Machine Images (AMI)

AMI	Default Username
Amazon Linux 2	`ec2-user`
Amazon Linux	`ec2-user`
CentOS	`centos` or `ec2-user`
Debian	**`admin`**
Fedora	`fedora` or `ec2-user`
Red Hat	`ec2-user` or `root`
SUSE	`ec2-user` or `root`
Ubuntu	`ubuntu`
Oracle	`ec2-user`

Figure 8: Default Usernames for AMI

1.10 Connect to the instance with OpenSSH

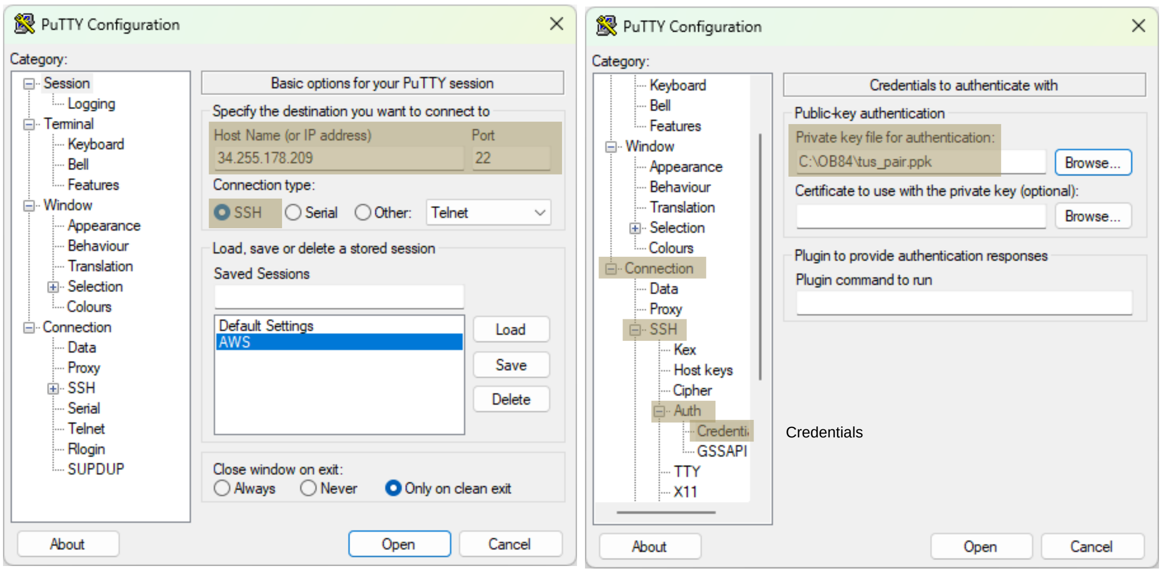
A key pair is a public and private key set that are used to authenticate instead of a password. The server, the VM, holds the public key while the private key is stored on the workstation. Accessing the VM in the future will require the **username** and the **private key**. Make sure that the key is only accessible by the owner and group and other rights are removed.

```
~$ sudo chmod 0600 tus_pair.pem
```

Connect to the IP address noted in 1.8. The default username on Amazon Debian GNU/Linux instances is `admin`.

```
~$ ssh -i tus_pair.pem admin@34.255.178.209
The authenticity of host '34.255.178.209 (34.255.178.209)' can't be
established.
ED25519 key fingerprint is
SHA256:+acoiC5mBpGZyHVaR1+1LAQSnIUScJtmMoL4bJ+Ubjo.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.255.178.209' (ED25519) to the list of known
hosts.

Linux ip-172-31-22-121 6.1.0-10-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian
6.1.37-1 (2023-07-03) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
admin@ip-172-31-22-121:~$
```

1.11 Connect to the instance with PuTTY



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 34.255.178.209

Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

Load, save or delete a stored session

Saved Sessions

Default Settings
AWS

Close window on exit: ○ Always ○ Never ● Only on clean exit

PuTTY Configuration

Credentials to authenticate with

Public-key authentication

Private key file for authentication: C:\OB84\tus_pair.ppk

Certificate to use with the private key (optional):

Plugin to provide authentication responses

Plugin command to run

Credentials

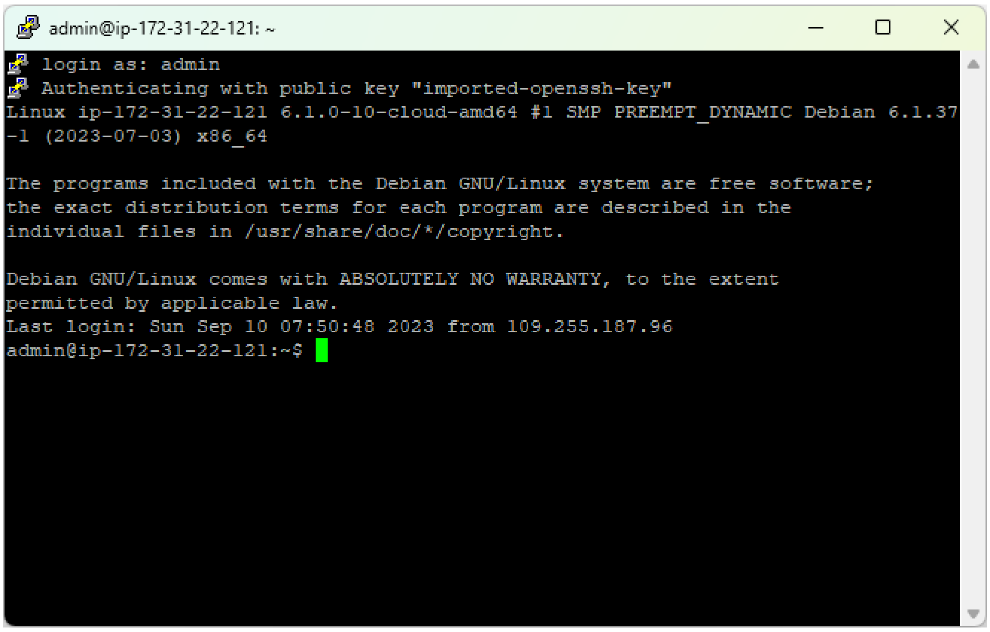
Figure 9: Connect to an EC2 AMI with PuTTY

Start **PuTTY** from the Microsoft Windows Start menu. Select **All Programs** → **PuTTY** → **PuTTY**). In the **Category** pane, select **Session** and complete the following fields:

- In the **Host Name** box, enter **`admin@<public IP address>`**.
- Under **Connection type**, select **SSH**.
- Ensure that **Port** is set to **22**.

In the **Category** pane, expand **Connection**, expand **SSH**, and then select **Auth** and **Browse** to select the **.ppk** file generated as the Private key, and then click **Open** to start the PuTTY session. Select **Yes** when prompted to accept cache of the host key.

At this stage a terminal will open with the connection to the remote EC2 AMI will open.

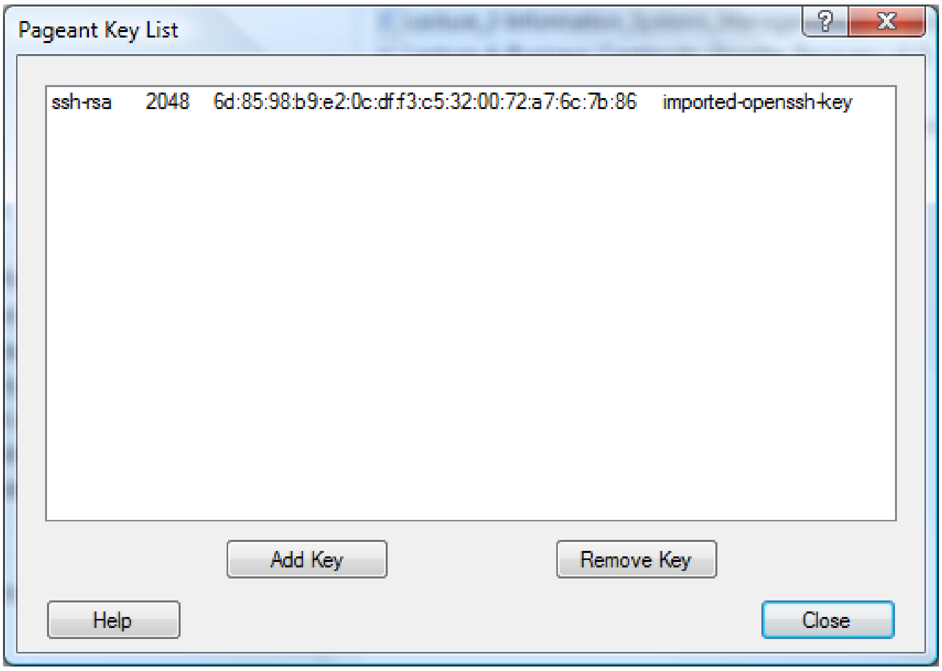


```
login as: admin
Authenticating with public key "imported-openssh-key"
Linux ip-172-31-22-121 6.1.0-10-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.37
-1 (2023-07-03) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Sep 10 07:50:48 2023 from 109.255.187.96
admin@ip-172-31-22-121:~$
```

Figure 10: PuTTY login to a Linux AMI

1.12 Storing .ppk keys in Padgent for PuTTY and PSFTP

Start **Padgent** (Start menu, click **All Programs** → **PuTTY** → **Padgent**).
It will form an icon ( ) with the running services on the toolbar.
Right click on the icon and select **View Keys**.
Browse to the **.ppk** file and click **Open**.
Keys will now be stored for future use as in Figure 11.



ssh-rsa 2048 6d:85:98:b9:e2:0c:df:f3:c5:32:00:72:a7:6c:7b:86 imported-openssh-key

Figure 11: Storing the private key in Pageant

1.13 Using the PuTTY PSFTP

Start **PSFTP** (from the Start menu, click **All Programs** → **PuTTY** → **PSFTP**).

- Type:
  ```
  psftp> open <public IP address>
  ```
- At the **login as:** prompt type the user **`debian`**
- PSFTP is now logged in to the **`debian`** home directory.

1.13.1 SFTP Using a Linux Terminal

Open a GNU/Linux Terminal and enter the following command to connect to the VM. Files can be **`put`** or **`get`** to or from the VM.

```
~$ sftp -i <my-key-pair.pem> admin@<public IP address>
```

2. Setup the new Virtual Machine

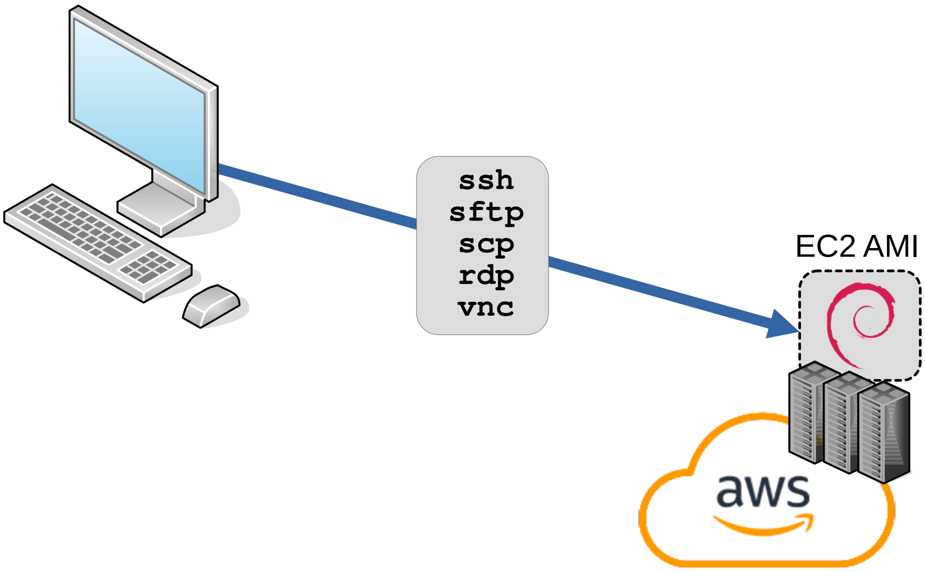


Figure 12: Workstation connecting to Amazon EC2 AMI cloud testbed VM

Using the Debian GNU/Linux Server build from the free tier Amazon EC2 AMI the remainder of this document will focus on the development of a cloud based Desktop as a laboratory for further work.

2.1 Transfer background image

Copy the course desktop backgrounds image to the testbed.

2.1.1 GNU/Linux

Use **sftp** to transfer the desktop background files as demonstrated directly from the local workstation.

```
~$ sftp -i tus_pair.pem admin@34.255.178.209

sftp> put background.png
Uploading background.png to /home/admin/background.png
background.png                              100%  171KB   1.8MB/s   00:00
sftp>

sftp> exit
~$
```

2.1.2 Windows

First on Microsoft Windows install the PuTTY packages from **https://www.putty.org** and use **WinSCP** to transfer the file.

```
C:\>pscp -i tus_pair.ppk C:\image\background.png admin@34.255.178.209:background.png
background.png           | 170 kB | 170.5 kB/s | ETA: 00:00:00 | 100%
```

2.2 Set the hostname

Change the default system hostname on the Virtual Machine (VM), logout and log back in to make it effective.

```
admin@ip-172-31-22-121:~$ sudo hostnamectl set-hostname adalabtus.ie
admin@ip-172-31-22-121:~$ exit
logout
Connection to 34.249.149.156 closed.

admin@ip-172-31-22-121:~$ ssh -i tus_pair.pem admin@34.255.178.209
admin@adalabtus:~$
```

Query the system hostname and related settings.

```
admin@adalabtus:~$ hostnamectl
 Static hostname: adalabtus.ie
       Icon name: computer-vm
         Chassis: vm ⍰
      Machine ID: 1d2a61ebac5a4a69b8a8428ad7872e44
         Boot ID: 84b0519e22574f448b2a69030f5a9fd3
  Virtualization: xen
Operating System: Debian GNU/Linux 12 (bookworm)
          Kernel: Linux 6.1.0-10-cloud-amd64
    Architecture: x86-64
 Hardware Vendor: Xen
  Hardware Model: HVM domU
Firmware Version: 4.11.amazon
```

2.3 Install the GNU/Linux Desktop

Figure 13: LXQt Desktop

Update the operating system, install the desktop, and remove software that is not required for the testbed.

```
admin@adalabtus:~$  sudo apt update && sudo apt -y upgrade
```

Install your preferred Desktop from a choice of GNOME, Xfce, KDE Plasma, Cinnamon, MATE, LXDE, or LXQt. This desktop will be passing graphics over the Internet so it is best to use a lightweight desktop. While Xfce is the lightest, LXQt is light also and makes a good compromise between light, efficient and neat.

```
admin@adalabtus:~$  sudo apt -y install lxqt
```

Select your preferred keyboard, for example **`Other`** >> **`English (UK, extended, Windows)`** >> **`OK`**.

Copy backgrounds to LXQt template.

```
admin@adalabtus:~$  sudo cp ~/background.png /usr/share/lxqt/themes/debian/
```

Remove unnecessary software.

```
admin@adalabtus:~$  sudo apt -y remove --purge libreoffice*
admin@adalabtus:~$  sudo apt -y remove --purge thunderbird
admin@adalabtus:~$  sudo apt -y remove --purge vlc-*
admin@adalabtus:~$  sudo apt -y remove --purge pulseaudio-*
admin@adalabtus:~$  sudo apt -y autoremove
```

2.4 Enable Remote Desktop Protocol on testbed

Install the Remote Desktop Protocol (RDP) on the testbed. This will allow for graphical connections from Microsoft Windows using its native RDP client or from a GNU/Linux client using an RDP application like **Remmina**.

```
admin@adalabtus:~$  sudo apt -y install xrdp
admin@adalabtus:~$  sudo systemctl enable xrdp
admin@adalabtus:~$  sudo systemctl start xrdp
```

2.5 Allow RDP on the VM

While the GNU/Linux implementation of the workstation will redirect the RDP traffic through an SSH tunnel the Windows implementation will not. Therefore if the client is Microsoft Windows it is necessary to open the RDP port 3389 to permit access. In this case however the VM is only protected by the username and password. Therefore it is recommended that the wildcard 0.0.0.0/0 is not employed and the specific IP address of the workstation or at least the subnet of the workstation is used as the source. On AWS add the new security rule.

EC2 → Instances → <Instance ID>

Security → Inbound Rules

Security groups → launch-wizard-4

Inbound rules → Edit inbound rules

Add rule

```
Type: Custom TCP
Port range: 3389     (RDP port)
Source: 0.0.0.0/0    (Wildcard, prefer specific workstation address)
Description: RDP
```

Save Rules

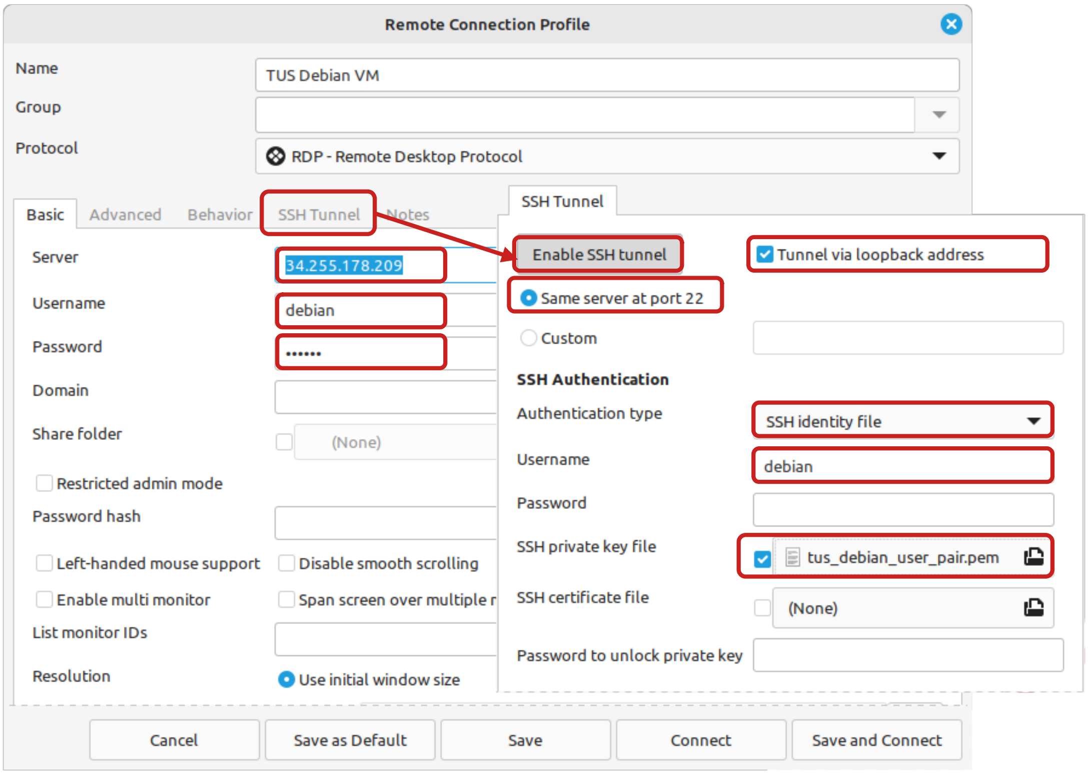
3. Testing a graphical login

Test the graphical login either using the Microsoft Windows native RDP or a suitable GNU/Linux client, in this case **Remmina** on a GNU/Linux workstation is demonstrated.

3.1 Remmina on GNU/Linux

Install **Remmina** and the **RDP plugin** on GNU/Linux.

```
user@workstation:~$ sudo apt -y install remmina remmina-plugin-rdp
```



Remote Connection Profile

Name: TUS Debian VM

Group:

Protocol: RDP - Remote Desktop Protocol

Basic | Advanced | Behavior | SSH Tunnel | Notes

Server: 34.255.178.209

Username: debian

Password: ••••••

Domain:

Share folder: □ (None)

□ Restricted admin mode

Password hash:

□ Left-handed mouse support □ Disable smooth scrolling

□ Enable multi monitor □ Span screen over multiple r

List monitor IDs:

Resolution: ● Use initial window size

SSH Tunnel

Enable SSH tunnel ☑ Tunnel via loopback address

● Same server at port 22

○ Custom

SSH Authentication

Authentication type: SSH identity file

Username: debian

Password:

SSH private key file: ☑ tus_debian_user_pair.pem

SSH certificate file: □ (None)

Password to unlock private key:

Figure 14: Configure Remmina for a RDP over SSH connection

Rimmina can establish an SSH connection to the remote VM and redirect RDP through the secure SSH tunnel thereby securing the RDP traffic. Configure as in Figure 14 and select **Save and Connect**.

3.2 Background

Change the background to that which was sent over by TFTP earlier.

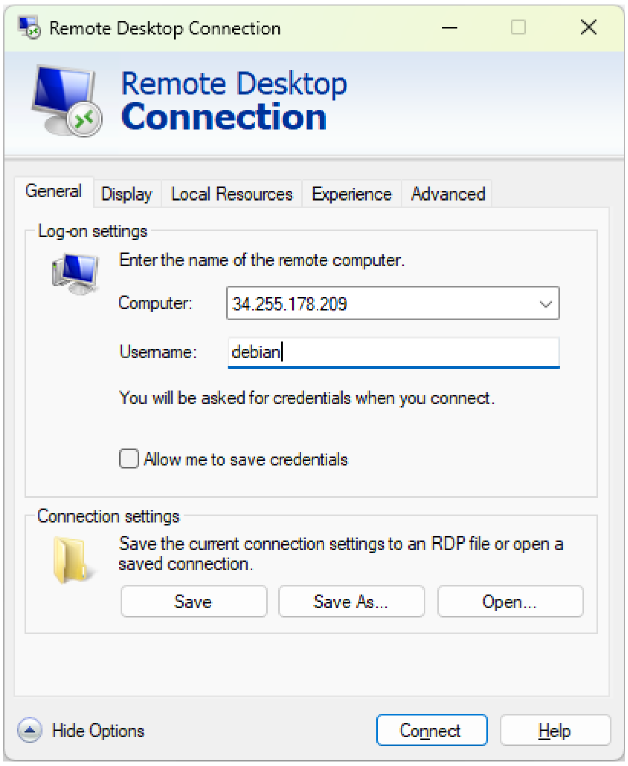
Right mouse click on the Desktop and select **`Desktop preferences`**.

Click on the **`Background`** tab.

Browse to the **`background.png`** file.

Click **`OK`**.

3.3 Microsoft Windows RDP Client



Log-on settings

Enter the name of the remote computer.

Computer: 34.255.178.209

Username: debian

You will be asked for credentials when you connect.

□ Allow me to save credentials

Connection settings

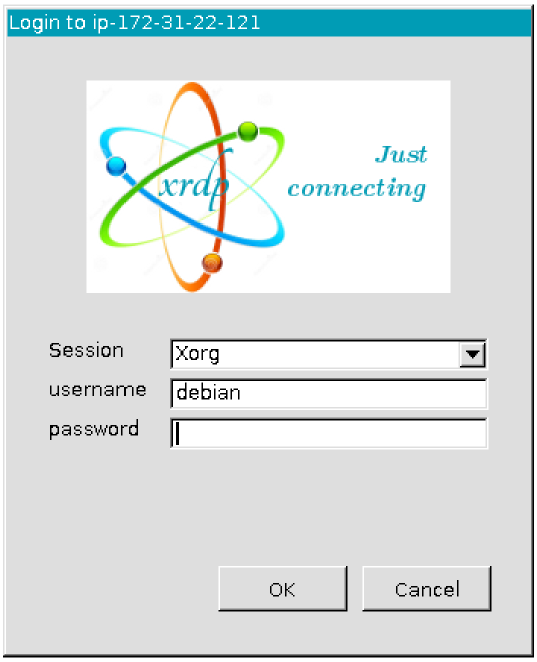
Save the current connection settings to an RDP file or open a saved connection.

Figure 15: Windows RDP Client

RDP is native to Microsoft Windows so the protocol already has a client. Type **Remote Desktop** in the search, and the RDP software can be seen as in Figure 15 Simply type in the testbed IP Address and password to connect to the testbed.

3.4 Connection to the VM graphical interface

Whichever RDP client is employed will present a login pane similar to that in Error: Reference source not found from the VM. Enter the appropriate login details, **`debian`** and **`D69a55`**, click **OK** and the desktop will appear as in Error: Reference source not found.


Login to ip-172-31-22-121

Session Xorg
username debian
password

Figure 16: Login Prompt

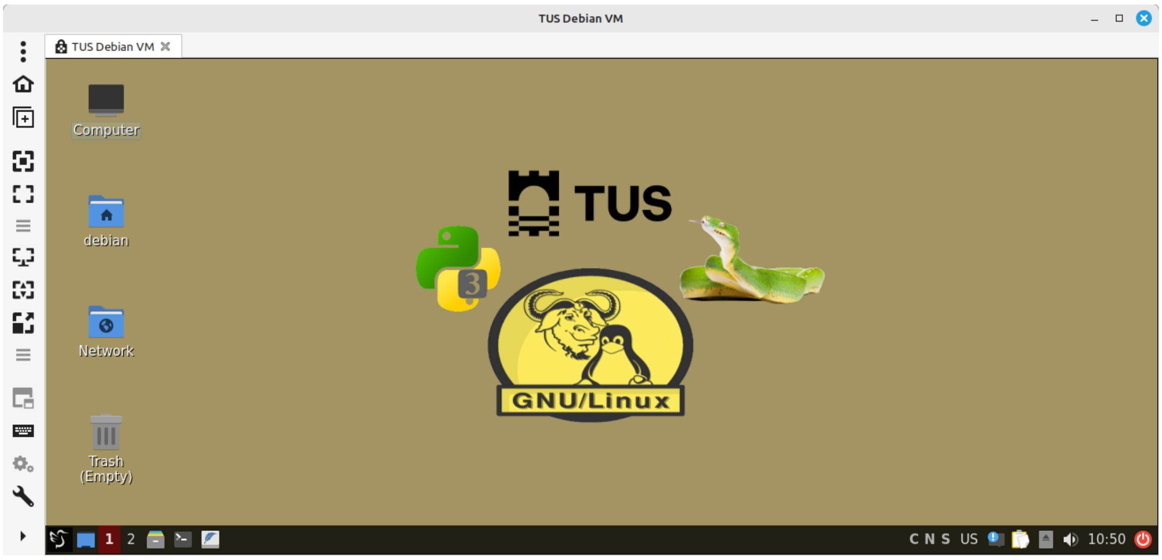


Figure 17: XRDP rendering of the testbed