Operating System Management

Assignment-1

Nectar

Name: Elvan Alandi

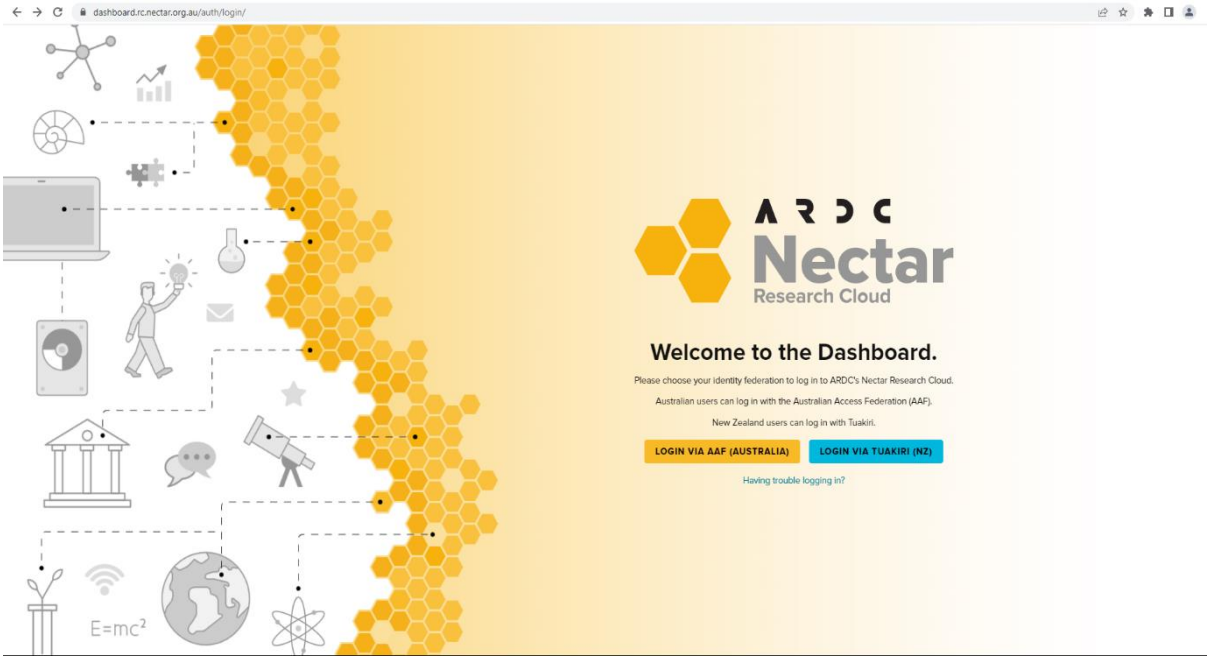
Table of Contents

Task 1. Nectar Walkthrough 3
1.1 Login to Nectar 3
1.2 Setup SSH Keys 5
1.2.1 Windows Users - PuTTY 5
1.2.2 Using Nectar Public Key (Alternative) 7
1.3 Creating Virtual Machine 8
1.4 Ping and Login 11
1.4.1 Ping 11
1.4.2 Login using Windows PuTTY 12
Task 2. Add Web server, DB server, and Website within an LXC Container 15
2.1 Adding LXC Web server 15
2.1.1 Login to Nectar 15
2.1.2 Update and upgrade packages 15
2.1.3 LXC and LXD 15
2.2 Adding Database Server 17
2.2.1 Install MySQL server 17
2.2.2 Open MySQL 17
2.2.3 Create root password 17
2.2.4 Login again using root as user 17
2.2.5 Granting the root user privileges 17
2.2.6 Create database 17
2.2.7 Create table for the website 17
2.3 Adding Simple Functional Website 18
2.3.1 Install PHP 18
2.3.2 Install php-mysqli to connect mysqli with php 18
2.3.3 Create an index.php file in /var/www/html folder 18
2.3.4 Test the web page 19
References 21

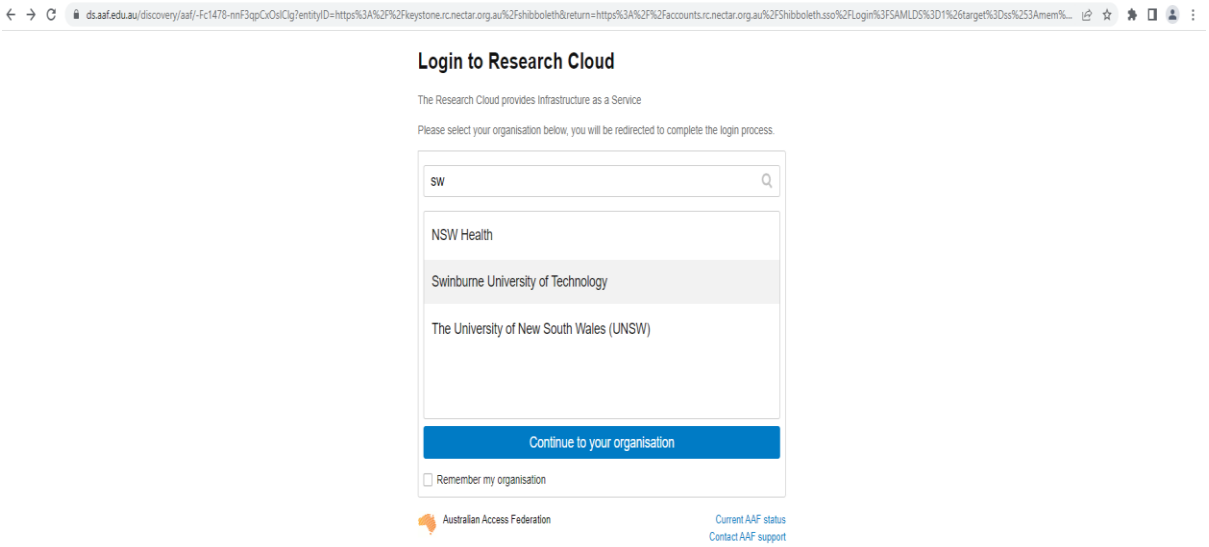
Task 1. Nectar Walkthrough

1.1 Login to Nectar

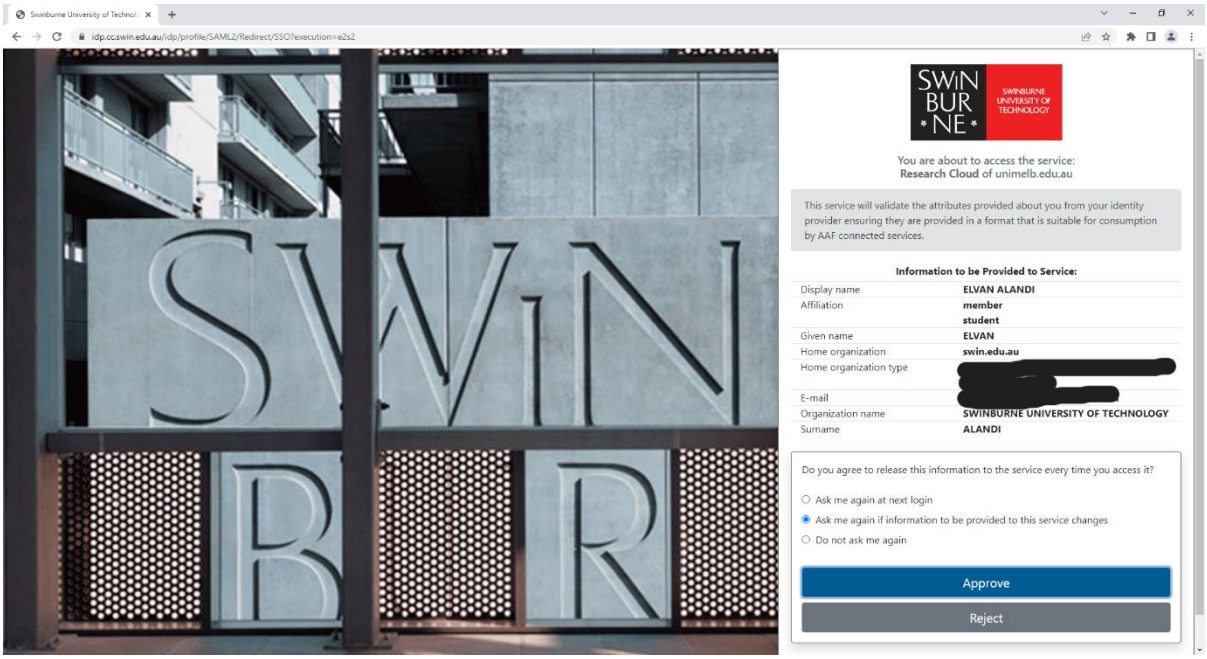
1. Go to web browser and navigate to https://dashboard.rc.nectar.org.au/auth/login
2. Click the “LOGIN VIA AAF (AUSTRALIA)”



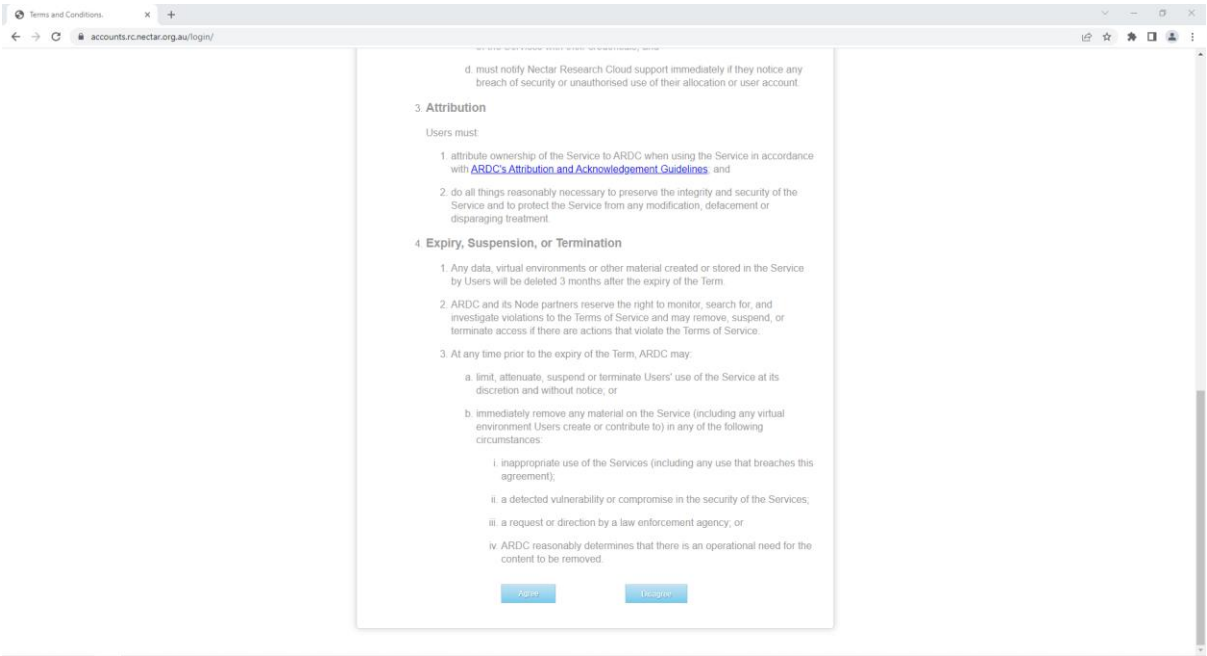
3. Search for your organisation (My organisation is Swinburne University of Technology)



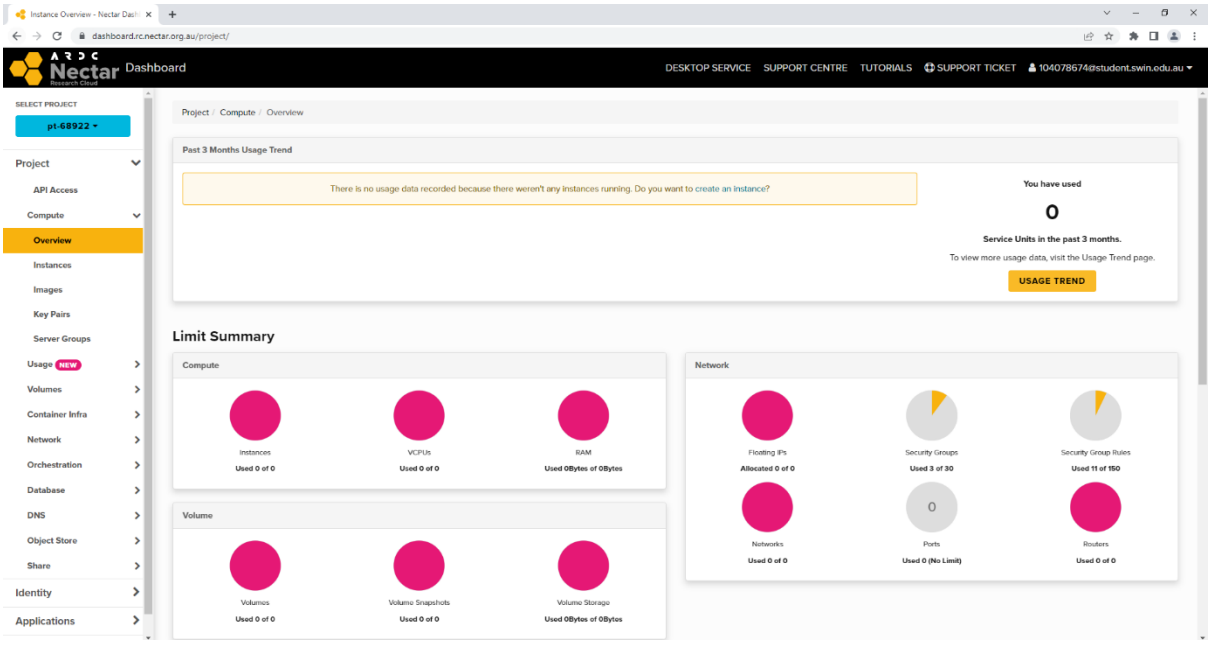
4. Login using organisation’s login credential (I have logged in before and I just need click on the approve button)



5. Login into nectar by aggreging the terms and conditions



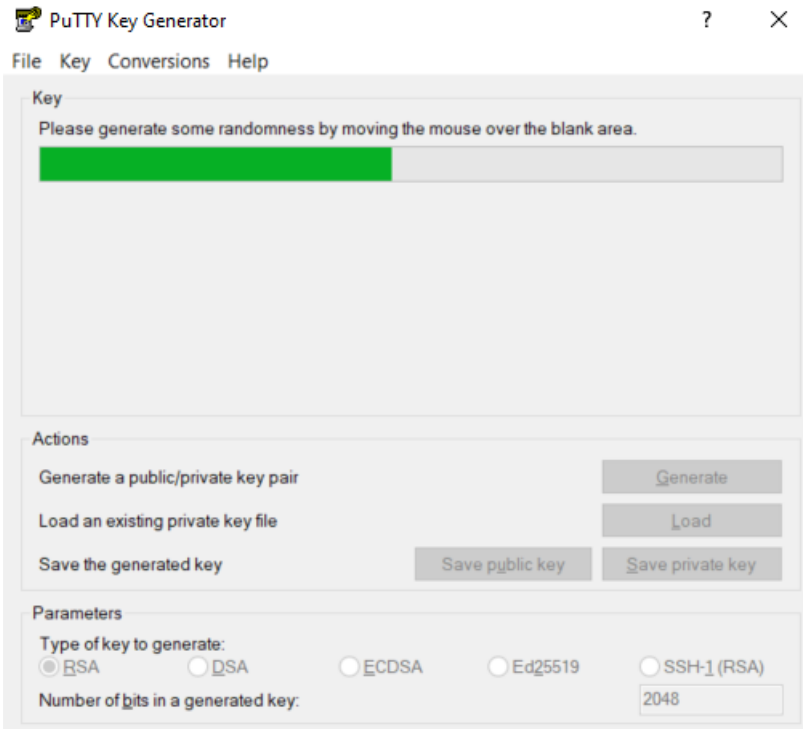
6. After that, you will see the Nectar Dashboard page



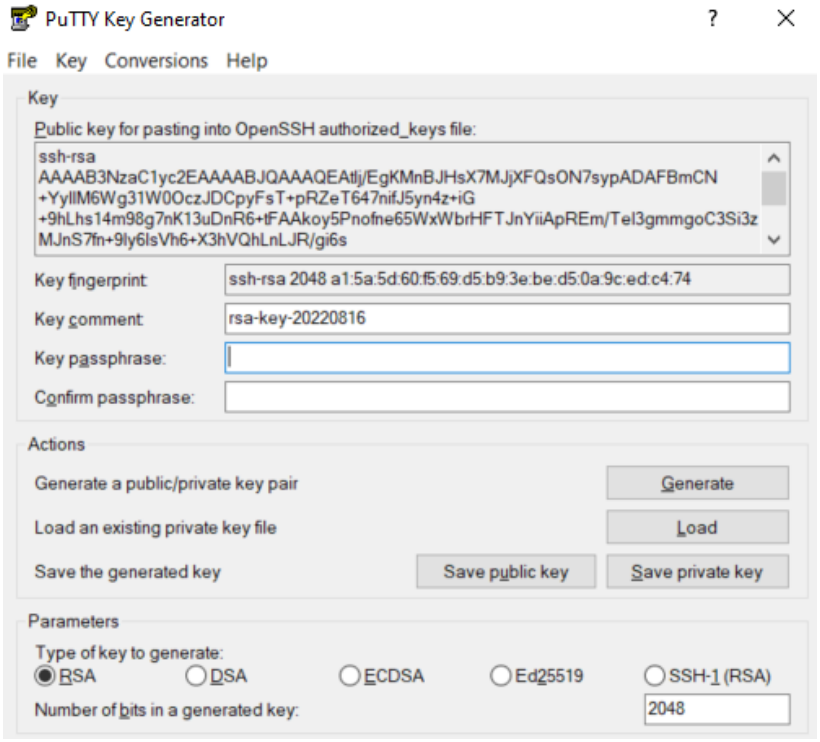
1.2 Setup SSH Keys

1.2.1 Windows Users - PuTTY

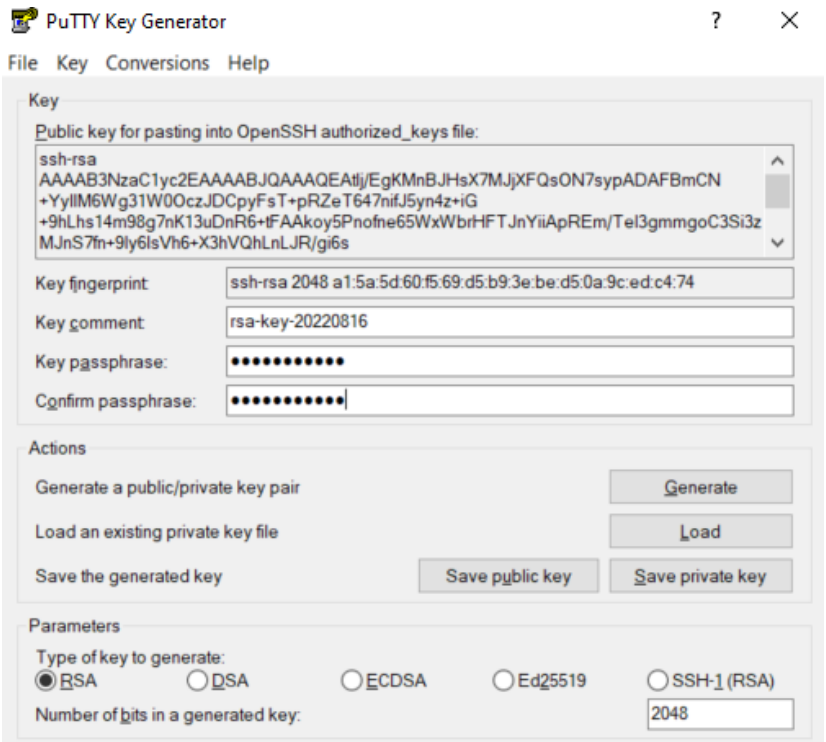
1. Launch PuttyGen.exe from the Start search field or using Win + R
2. Click on Generate button and wait until the process has been finished



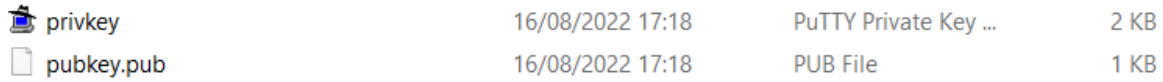
3. If the process has been finished, it will be looked like the image below. Copy the rsa key into your clipboard and paste it into the temporary file (to be pasted into the text field in the Nectar interface)



4. For a security reason, fill in the key passphrase before saving the public key and private key to the local drive.



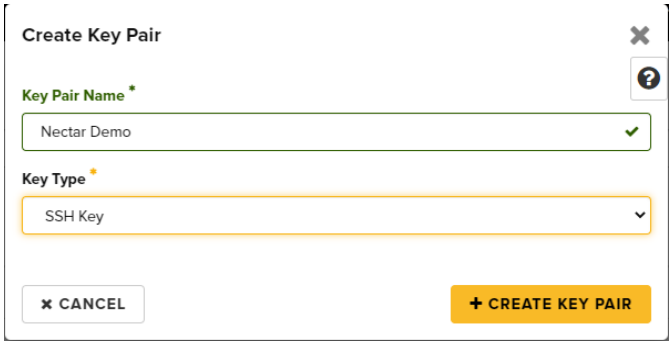
5. Save the public key and the private key to safe places on computer



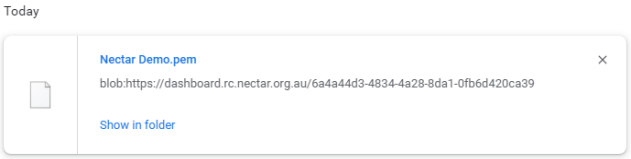
1.2.2 Using Nectar Public Key (Alternative)

1. Creating PEM file using Nectar:

- Go to Key Pairs tab
- Click on the +Create Key Pair button
- Input Key Pair Name and choose Key Type to SSH Key
- Click +Create Key Pair button in the popup box

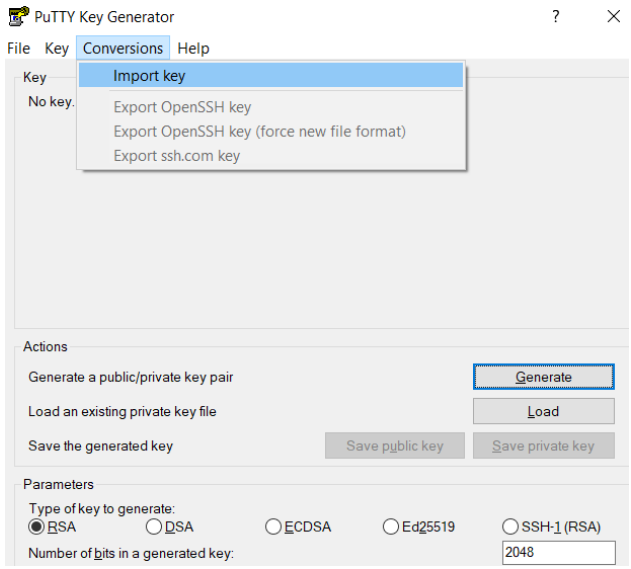


2. It will automatically download the .pem file like in the image below, save the .pem file to a safe place

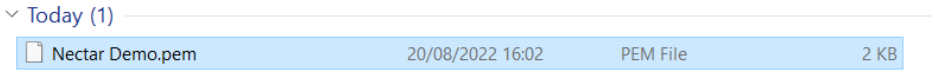


3. Open PuttyGen.exe

4. From the menu select Conversions → Import key



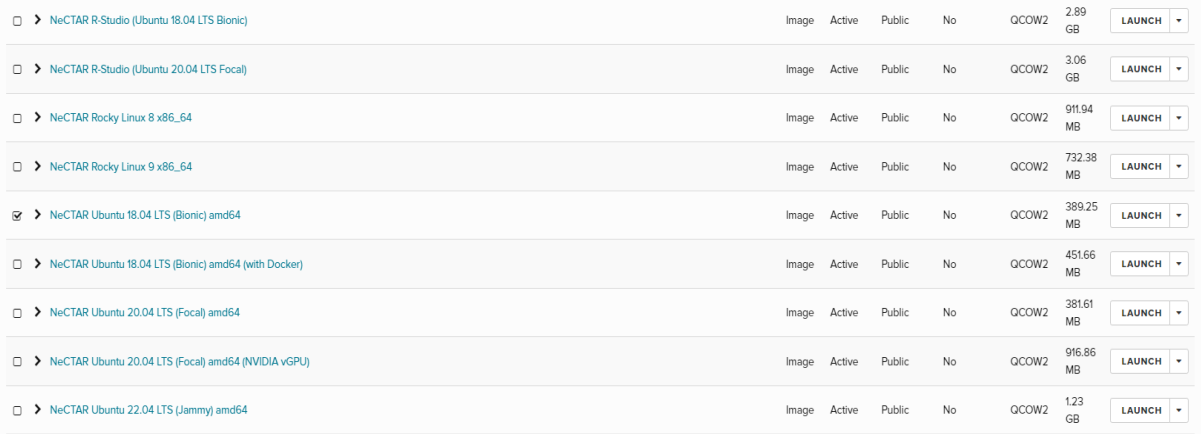
5. Choose the .pem file that has been downloaded before



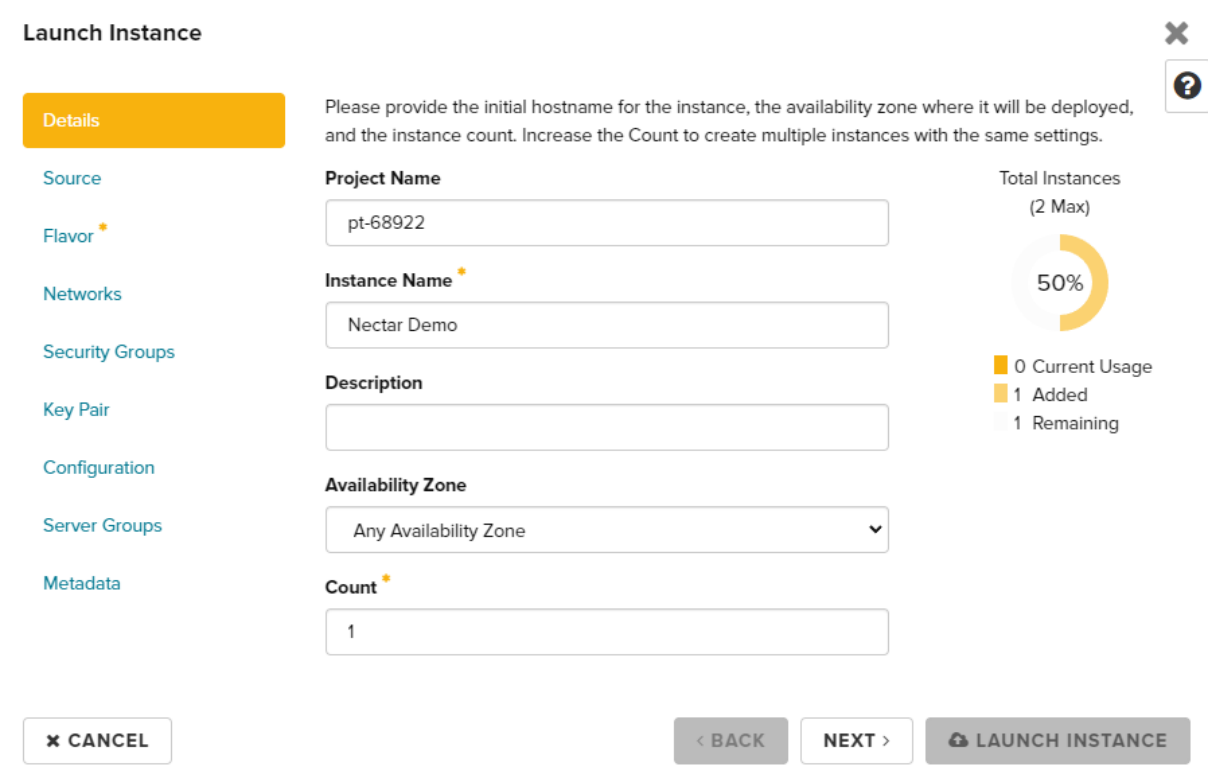
6. Repeat point 1.2.1 step 3, 4, and 5

1.3 Creating Virtual Machine

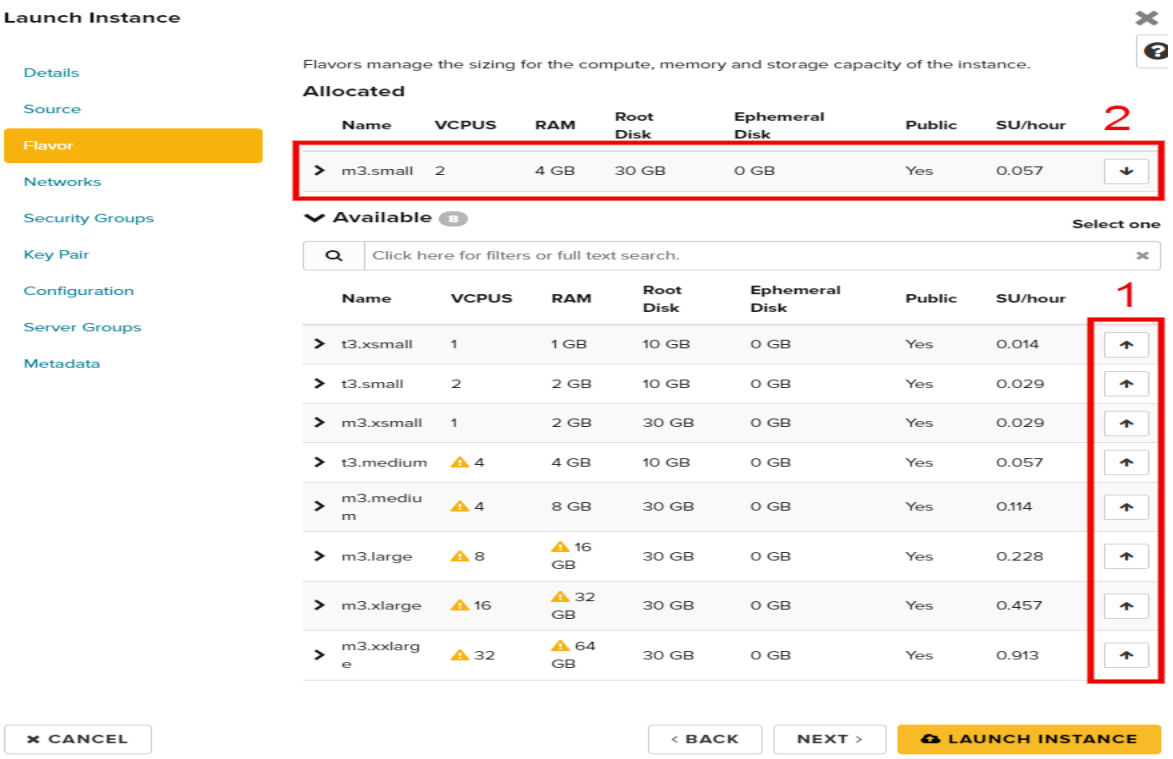
1. Go back to the NectarCloud Dashboard. On the left side panel click “Compute” and then click “images”
2. Select NeCTAR Ubuntu 18.04 LTS (Bionic) amd64 by clicking on the box (There is Ubuntu version 22.04 but I am using Ubuntu version 18.04 LTS (Bionic) amd64, it is up to your choice) and click Launch on the right side of the image you were selected



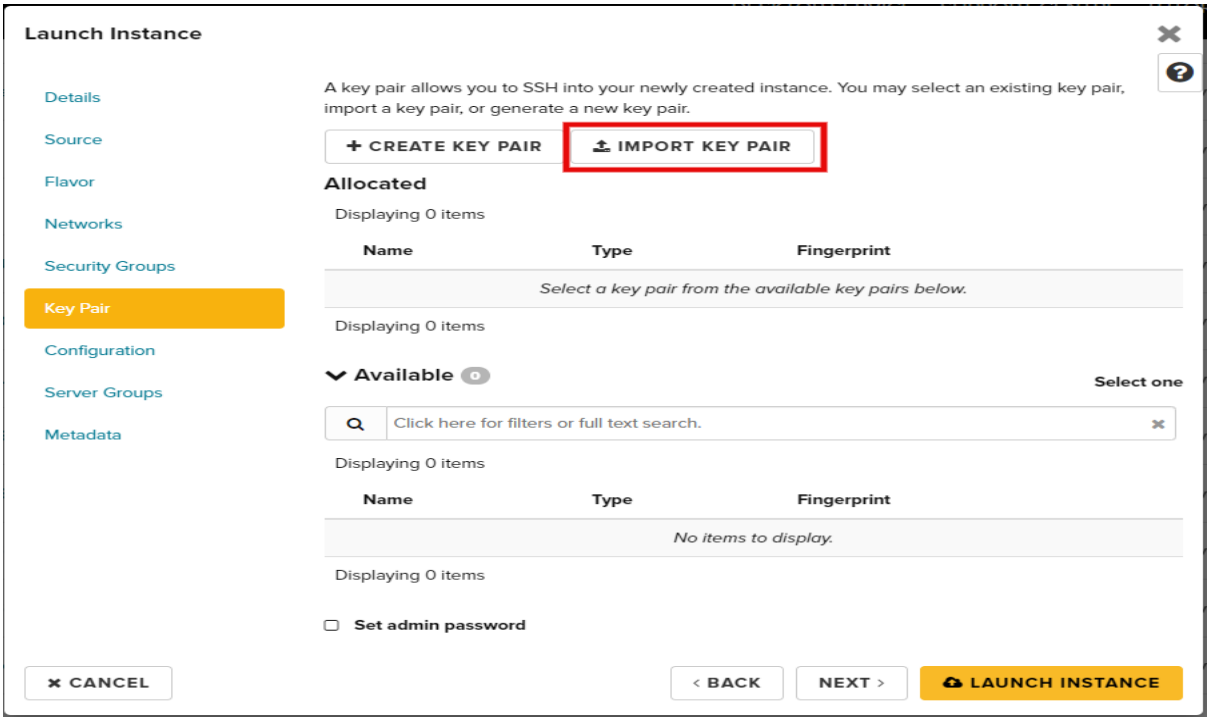
3. Give your machine an appropriate name and description. Leave the default availability zone and count



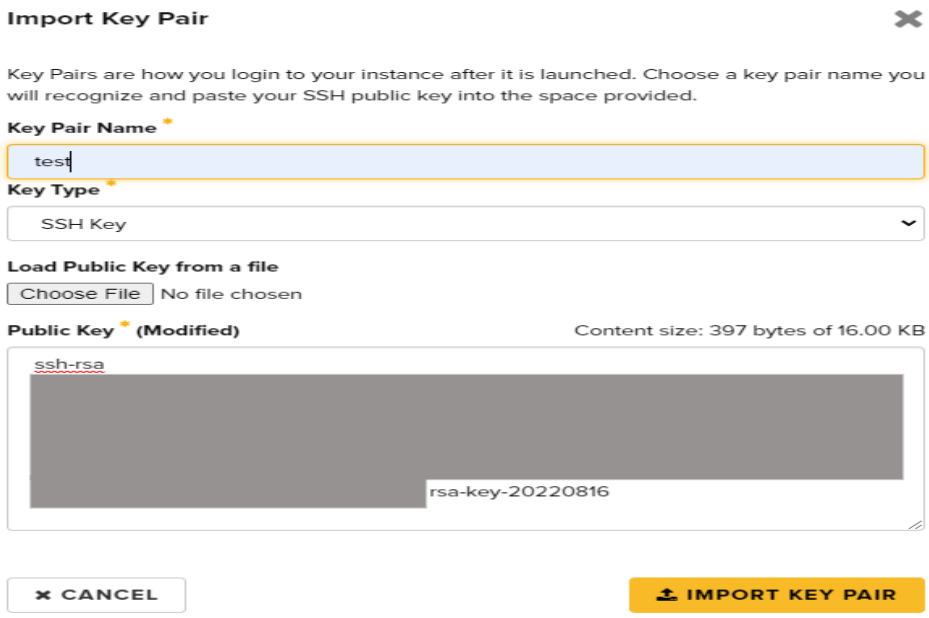
4. In the Flavor* tab, click on the up arrow (number 1 in the image) near **m3.small** (it already moved to allocation section) and it will move to the allocation section (number 2 in the image). A standard Nectar account can use up to 8GB of RAM and 2 virtual CPUs



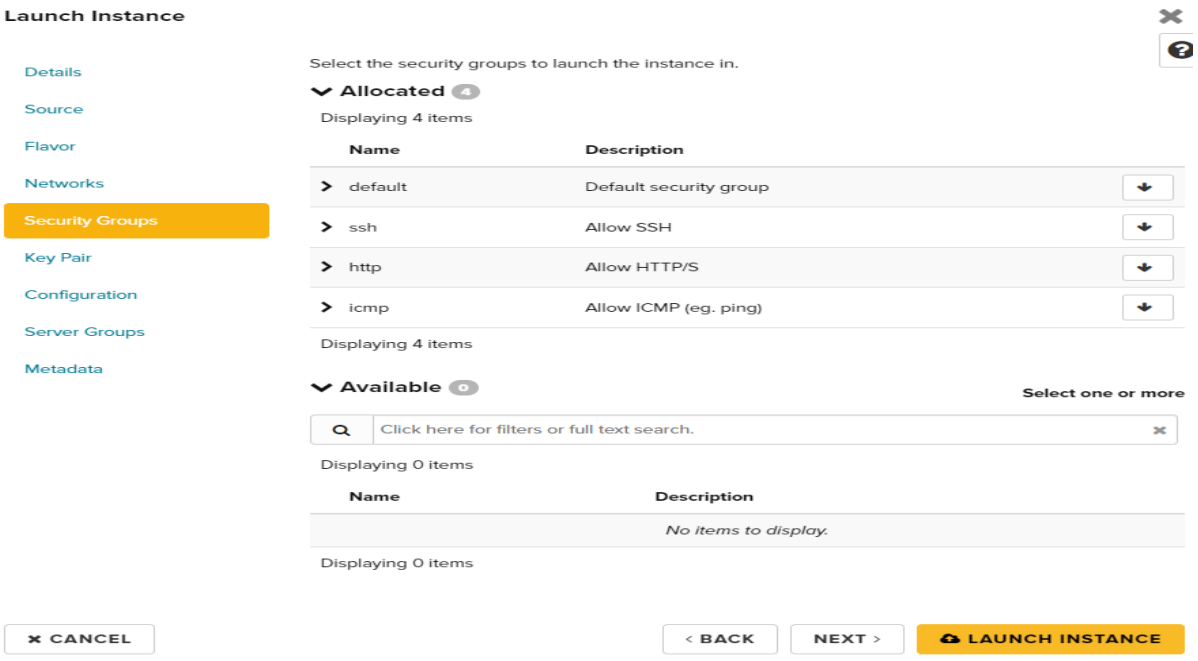
5. Click the Key Pair tab and click Import Key Pair (because we do not have allocated key)



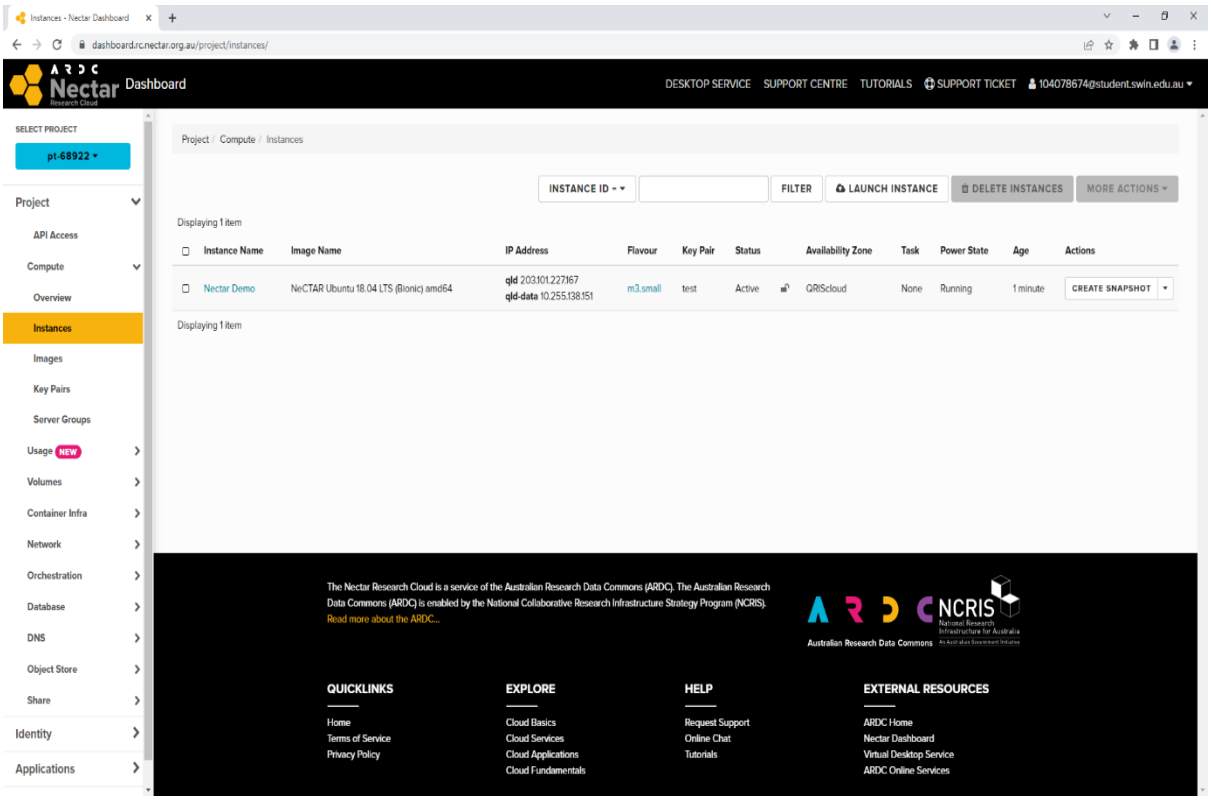
6. Give your Key Pair name and choose SSH Keys for the Key Type. After that, copy your SSH RSA key that has been generated and has been kept in the temporary file to Public Key text area, then click on the import key pair button.



7. Nectar's firewall will by default block the ports we need to use our server. Click on the "Security Groups" tab. Click the up arrows next to ssh, http and icmp so that they move to the "Allocated" section. We need ssh allowed to login to our virtual machine and we need icmp if we want to ping it (ssh, http, and icmp in the image below has already been clicked from available section to allocated section). Lastly click on "Launch Instance".



8. Click on the instance tab on the Nectar Dashboard to see the instance that has been created before.



1.4 Ping and Login

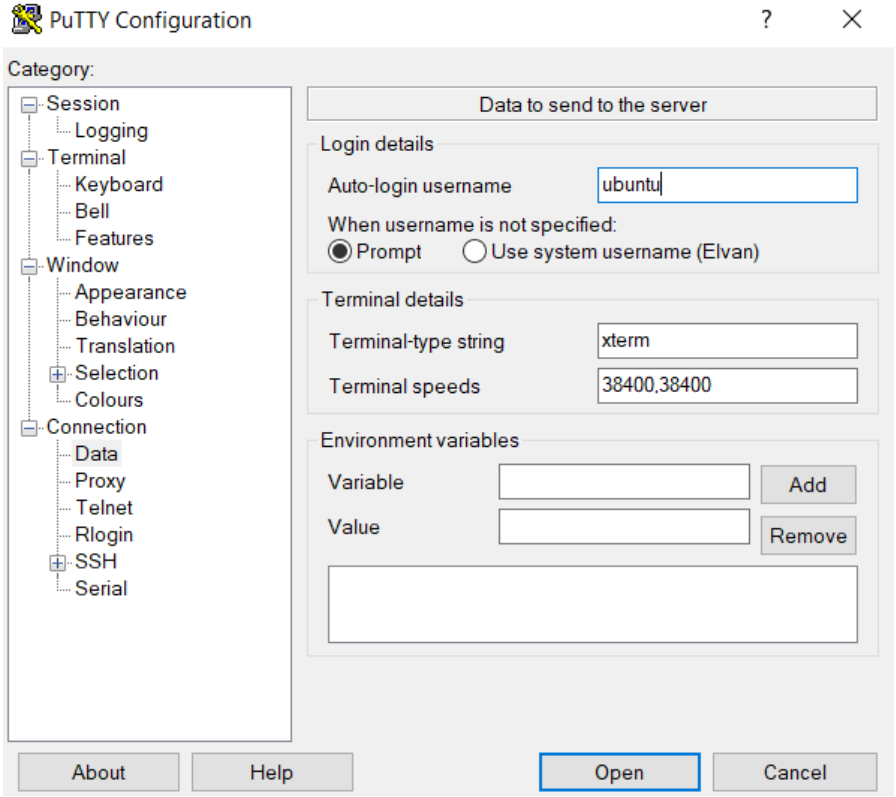
1.4.1 Ping

1. Open terminal (linux) or command prompt (windows). Use the IP address from the instance page, ping the instance to see if it was already there.

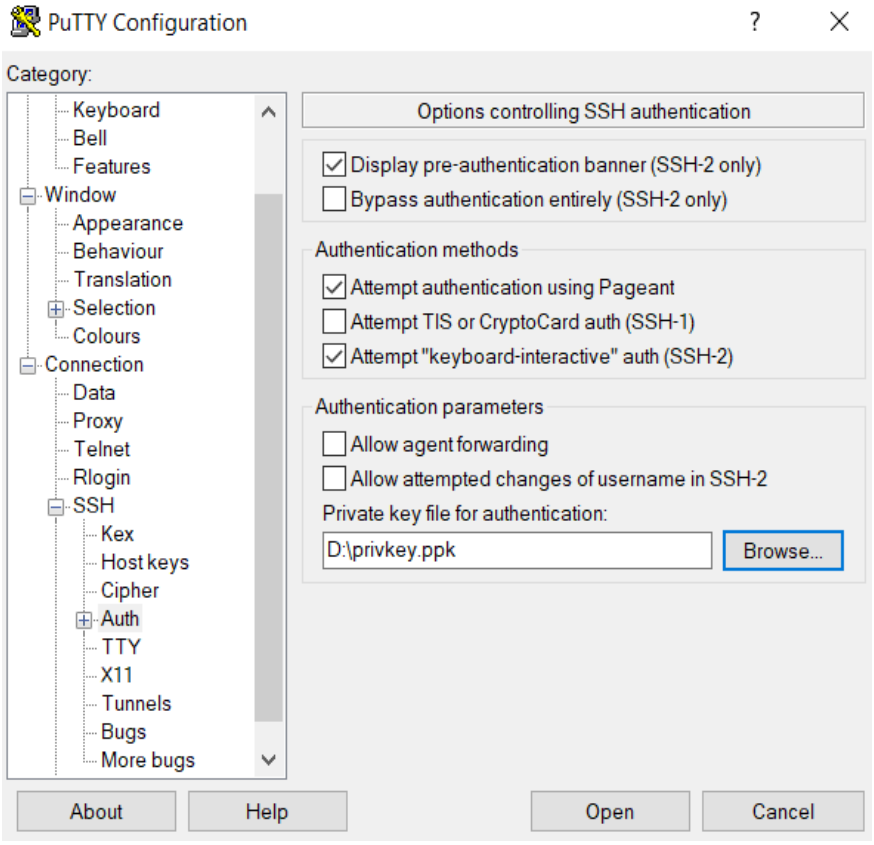
```
C:\WINDOWS\system32>ping 203.101.227.161

Pinging 203.101.227.161 with 32 bytes of data:
Reply from 203.101.227.161: bytes=32 time=42ms TTL=49
Reply from 203.101.227.161: bytes=32 time=42ms TTL=49
Reply from 203.101.227.161: bytes=32 time=43ms TTL=49
Reply from 203.101.227.161: bytes=32 time=42ms TTL=49

Ping statistics for 203.101.227.161:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 42ms, Maximum = 43ms, Average = 42ms
```

1.4.2 Login using Windows PuTTY

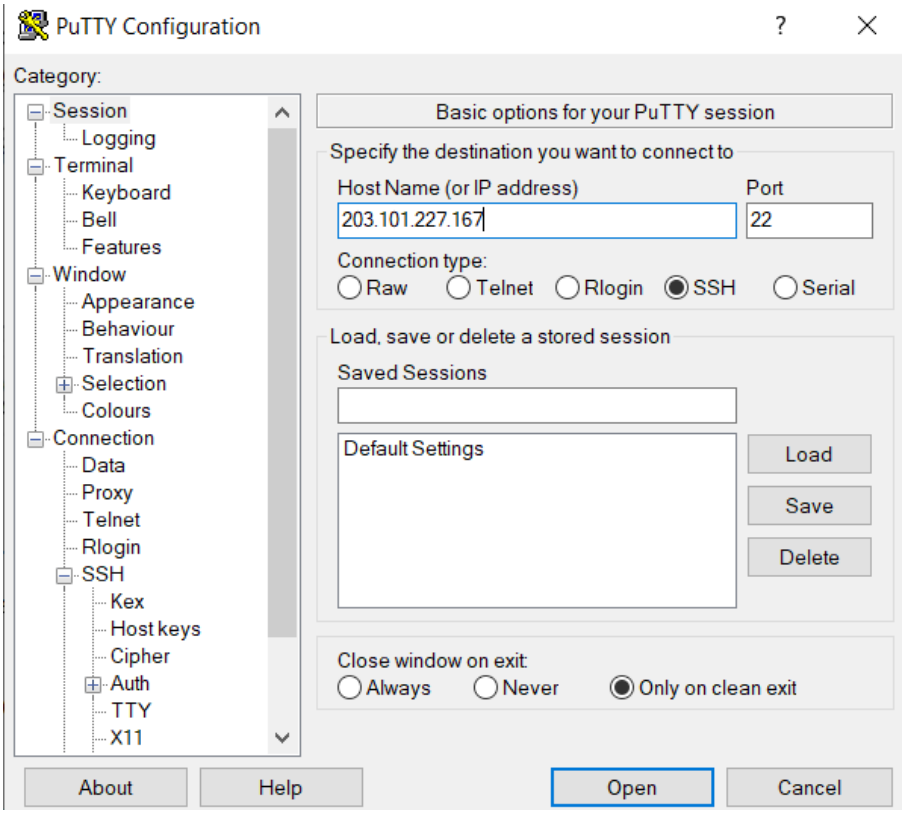
1. Open Putty, on the sidepane go to Connection → Data. In the auto-login username type ubuntu



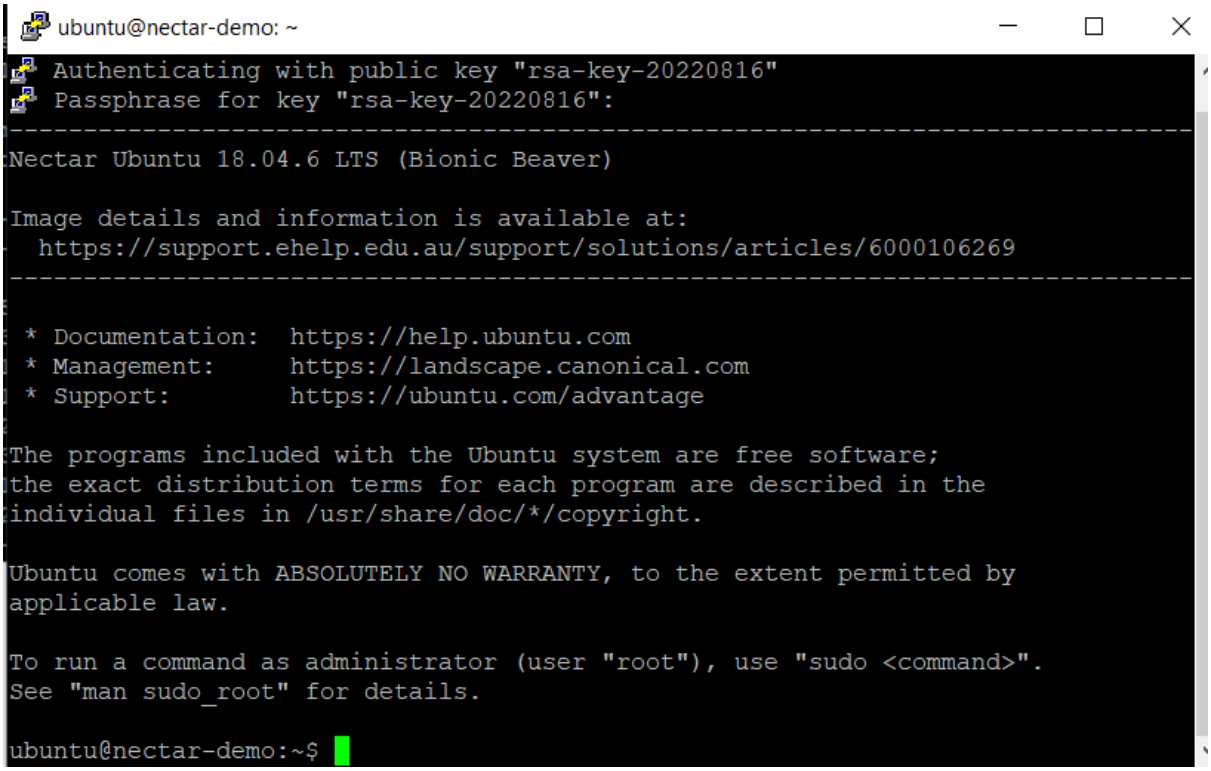
2. On the sidepane go to Connection → SSH → Auth. Click browse and select private key file (.ppk) that was generated earlier.

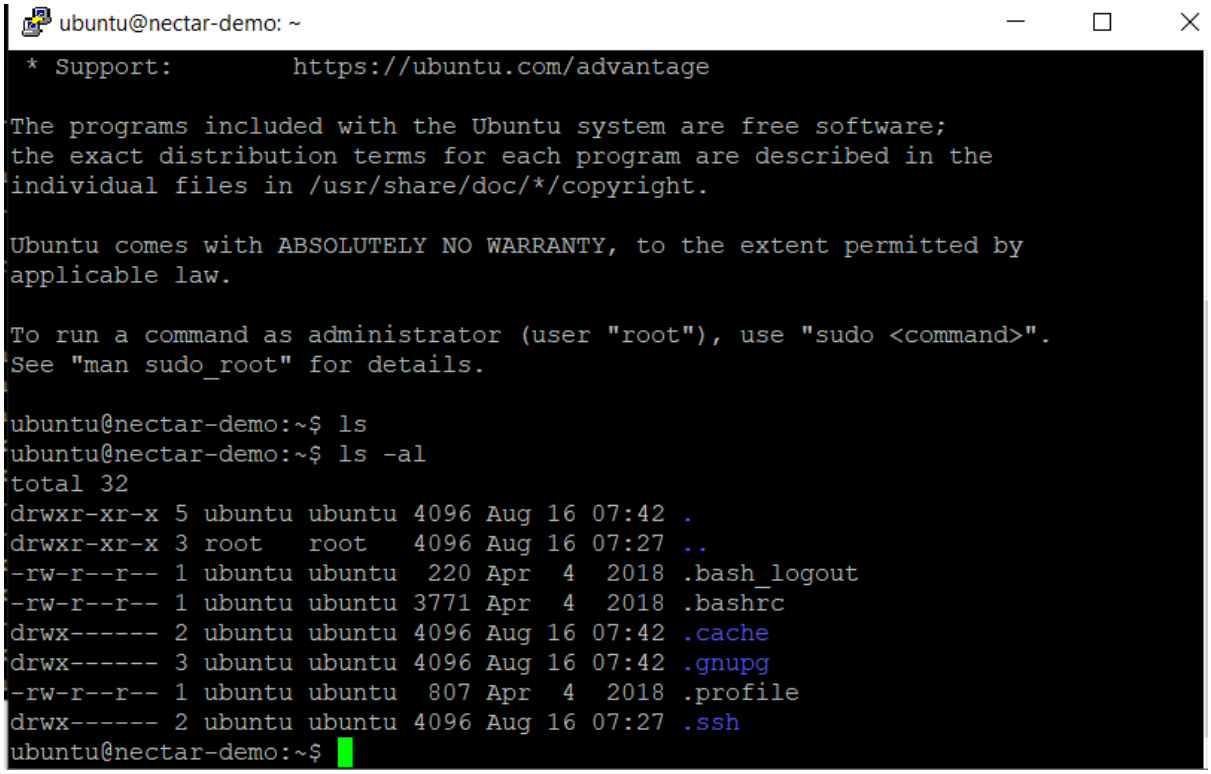


3. Click on the Session category and enter the IP address from the Nectar instances page, then click open button



4. Configuration window will be replaced by a console window that means you have logged into the Nectar instance




ubuntu@nectar-demo: ~
* Support: https://ubuntu.com/advantage
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
ubuntu@nectar-demo:~$ ls
ubuntu@nectar-demo:~$ ls -al
total 32
drwxr-xr-x 5 ubuntu ubuntu 4096 Aug 16 07:42 .
drwxr-xr-x 3 root root 4096 Aug 16 07:27 ..
-rw-r--r-- 1 ubuntu ubuntu 220 Apr 4 2018 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Apr 4 2018 .bashrc
drwx------ 2 ubuntu ubuntu 4096 Aug 16 07:42 .cache
drwx------ 3 ubuntu ubuntu 4096 Aug 16 07:42 .gnupg
-rw-r--r-- 1 ubuntu ubuntu 807 Apr 4 2018 .profile
drwx------ 2 ubuntu ubuntu 4096 Aug 16 07:27 .ssh
ubuntu@nectar-demo:~$

Task 2. Add Web server, DB server, and Website within an LXC Container

2.1 Adding LXC Web server

2.1.1 Login to Nectar

2.1.2 Update and upgrade packages

Using command “sudo apt update” and “sudo apt upgrade”

2.1.3 LXC and LXD

1. Install lxc and lxd using “sudo apt install lxc lxd”
2. Add ubuntu user to the lxd group using “sudo usermod --append --groups lxd ubuntu”
3. Setup lxd with the command below and accept all default options using “sudo lxd init”

```
ubuntu@nectar-demo:~$ sudo lxd init
Would you like to use LXD clustering? (yes/no) [default=no]:
Do you want to configure a new storage pool? (yes/no) [default=yes]:
Name of the new storage pool [default=default]:
Would you like to connect to a MAAS server? (yes/no) [default=no]:
Would you like to create a new local network bridge? (yes/no) [default=yes]:
What should the new bridge be called? [default=lxdbr0]:
What IPv4 address should be used? (CIDR subnet notation, “auto” or “none”) [defa
ult=auto]:
What IPv6 address should be used? (CIDR subnet notation, “auto” or “none”) [defa
ult=auto]:
Would you like LXD to be available over the network? (yes/no) [default=no]:
Would you like stale cached images to be updated automatically? (yes/no) [defaul
t=yes]
Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]:
```

4. List the lxc containers on the system

```
ubuntu@nectar-demo:~$ lxc list
To start your first container, try: lxc launch ubuntu:18.04

+------+-------+------+------+------+-----------+
| NAME | STATE | IPV4 | IPV6 | TYPE | SNAPSHOTS |
+------+-------+------+------+------+-----------+
```

5. For the assignment I am going to use Ubuntu because I have better understanding in the bash code than Alpine's code which is used in the lab

```
ubuntu@nectar-demo:/$ lxc image list images: | grep -i -E 'ubuntu.*amd64'
| ubuntu/16.04 (7 more)                | bda83bc541f3 | yes  | Ubuntu xenial amd64 (20220819_07:43)         | x86_64 | 88.67MB  | Aug 19, 2022 at 12:00am (UTC) |
| ubuntu/16.04/cloud (3 more)          | f19210abb9a0 | yes  | Ubuntu xenial amd64 (20220819_07:43)         | x86_64 | 103.78MB | Aug 19, 2022 at 12:00am (UTC) |
| ubuntu/18.04 (7 more)                | 50760fe1ea1f | yes  | Ubuntu bionic amd64 (20220819_07:42)         | x86_64 | 106.62MB | Aug 19, 2022 at 12:00am (UTC) |
| ubuntu/18.04/cloud (3 more)          | e56003343e0f | yes  | Ubuntu bionic amd64 (20220819_07:42)         | x86_64 | 113.33MB | Aug 19, 2022 at 12:00am (UTC) |
| ubuntu/focal (7 more)                | 2b6253c70ef1 | yes  | Ubuntu focal amd64 (20220819_07:43)          | x86_64 | 111.21MB | Aug 19, 2022 at 12:00am (UTC) |
| ubuntu/focal/cloud (3 more)          | 65fb3e405ee8 | yes  | Ubuntu focal amd64 (20220819_07:42)          | x86_64 | 122.21MB | Aug 19, 2022 at 12:00am (UTC) |
| ubuntu/jammy (7 more)                | 57e87a75b472 | yes  | Ubuntu jammy amd64 (20220819_07:42)          | x86_64 | 113.25MB | Aug 19, 2022 at 12:00am (UTC) |
| ubuntu/jammy/cloud (3 more)          | 65e2f0c3fd58 | yes  | Ubuntu jammy amd64 (20220819_07:43)          | x86_64 | 130.03MB | Aug 19, 2022 at 12:00am (UTC) |
| ubuntu/kinetic (3 more)              | 0f509ed4b5d0 | yes  | Ubuntu kinetic amd64 (20220819_07:43)        | x86_64 | 112.49MB | Aug 19, 2022 at 12:00am (UTC) |
| ubuntu/kinetic/cloud (1 more)        | 86cf80d18f7d | yes  | Ubuntu kinetic amd64 (20220819_07:44)        | x86_64 | 129.15MB | Aug 19, 2022 at 12:00am (UTC) |
|                                      | 06809e1b255c | yes  | Ubuntu focal amd64 (20220817_07:43)          | x86_64 | 111.21MB | Aug 17, 2022 at 12:00am (UTC) |
|                                      | 0d7e0fd2e25b | yes  | Ubuntu focal amd64 (20220817_07:42)          | x86_64 | 122.20MB | Aug 17, 2022 at 12:00am (UTC) |
|                                      | 195fb2252864 | yes  | Ubuntu focal amd64 (20220818_07:42)          | x86_64 | 122.21MB | Aug 18, 2022 at 12:00am (UTC) |
|                                      | 39318784fd83 | yes  | Ubuntu jammy amd64 (20220817_07:43)          | x86_64 | 113.25MB | Aug 17, 2022 at 12:00am (UTC) |
|                                      | 4a32b3d8ef36 | yes  | Ubuntu kinetic amd64 (20220818_07:42)        | x86_64 | 129.12MB | Aug 18, 2022 at 12:00am (UTC) |
|                                      | 581d02ce0264 | yes  | Ubuntu xenial amd64 (20220817_07:43)         | x86_64 | 88.67MB  | Aug 17, 2022 at 12:00am (UTC) |
|                                      | 5d6999599e75 | yes  | Ubuntu focal amd64 (20220818_07:42)          | x86_64 | 111.21MB | Aug 18, 2022 at 12:00am (UTC) |
|                                      | 6964b0c12354 | yes  | Ubuntu xenial amd64 (20220818_07:42)         | x86_64 | 88.67MB  | Aug 18, 2022 at 12:00am (UTC) |
|                                      | 6f3071a4d391 | yes  | Ubuntu xenial amd64 (20220818_07:43)         | x86_64 | 103.78MB | Aug 18, 2022 at 12:00am (UTC) |
|                                      | 83710d10e23f | yes  | Ubuntu kinetic amd64 (20220818_07:43)        | x86_64 | 112.45MB | Aug 18, 2022 at 12:00am (UTC) |
|                                      | 8f471cf21e8a | yes  | Ubuntu xenial amd64 (20220817_07:43)         | x86_64 | 103.78MB | Aug 17, 2022 at 12:00am (UTC) |
|                                      | 916046a11f94 | yes  | Ubuntu jammy amd64 (20220818_07:42)          | x86_64 | 130.03MB | Aug 18, 2022 at 12:00am (UTC) |
|                                      | a5de8b0432d7 | yes  | Ubuntu bionic amd64 (20220817_07:42)         | x86_64 | 106.62MB | Aug 17, 2022 at 12:00am (UTC) |
|                                      | b6bb9184ceef | yes  | Ubuntu bionic amd64 (20220818_07:42)         | x86_64 | 113.33MB | Aug 18, 2022 at 12:00am (UTC) |
|                                      | ba6d550e0a57 | yes  | Ubuntu bionic amd64 (20220818_07:43)         | x86_64 | 106.62MB | Aug 18, 2022 at 12:00am (UTC) |
|                                      | d05dc16f39e2 | yes  | Ubuntu jammy amd64 (20220817_07:42)          | x86_64 | 130.03MB | Aug 17, 2022 at 12:00am (UTC) |
|                                      | d5c71f9daa2a | yes  | Ubuntu kinetic amd64 (20220817_07:43)        | x86_64 | 129.11MB | Aug 17, 2022 at 12:00am (UTC) |
|                                      | d74c5871c3c5 | yes  | Ubuntu bionic amd64 (20220817_07:43)         | x86_64 | 113.33MB | Aug 17, 2022 at 12:00am (UTC) |
|                                      | db9265a7a0f7 | yes  | Ubuntu kinetic amd64 (20220817_07:42)        | x86_64 | 112.44MB | Aug 17, 2022 at 12:00am (UTC) |
|                                      | f41c8c8377b2 | yes  | Ubuntu jammy amd64 (20220818_07:42)          | x86_64 | 113.25MB | Aug 18, 2022 at 12:00am (UTC) |
```

6. Download and launch image (I am using ubuntu 18.04) using "lxc launch images ubuntu/18.04 web"

```
ubuntu@nectar-demo:/home$ lxc launch images:ubuntu/18.04 web
Creating web
Starting web
```

7. Check the container using command "lxd list"

```
ubuntu@nectar-demo:/$ lxc list
+------+---------+----------------------+----------------------------------------------+------------+-----------+
| NAME |  STATE  |         IPV4         |                     IPV6                     |    TYPE    | SNAPSHOTS |
+------+---------+----------------------+----------------------------------------------+------------+-----------+
| web  | RUNNING | 10.116.245.126 (eth0) | fd42:a9fc:94d4:4ae1:216:3eff:fef4:a502 (eth0) | PERSISTENT | 0         |
+------+---------+----------------------+----------------------------------------------+------------+-----------+
```

8. Log into the container using command "lxc exec web bash". In my case, web is my container name.

```
ubuntu@nectar-demo:/$ lxc exec web bash
root@web:~#
```

9. Install Apache2 Web Server using command "sudo apt-get install apache2"

10. To start the Apache Web Server use command "sudo systemctl start apache2"

11. Bind a port to the host to redirect traffic to the container using iptables:

"sudo -E bash -c 'iptables -t nat -I PREROUTING -i eth0 -p TCP -d $PUBLIC_IP --dport $PORT -j DNAT --to-destination $CONTAINER_IP:$PORT -m comment --comment "bind port to lxc ubuntu container"'

$PORT = use 80 for web server

$PUBLIC_IP = your VM IP which is Nectar public IP

$CONTAINER_IP = your container IP from "lxc list"

2.2 Adding Database Server

2.2.1 Install MySQL server

Type command “sudo apt-get install mysql-server” and start using “sudo systemctl start mysql”

2.2.2 Open MySQL

Type command “sudo mysql”

2.2.3 Create root password

1. Type command “ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY root;” (the password is ‘root’)
2. Type “exit” to exit from mysql

2.2.4 Login again using root as user

```
root@web:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.39-0ubuntu0.18.04.2 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

2.2.5 Granting the root user privileges

Type command “GRANT PRIVILEGE ON database.table TO 'username'@'host';”

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost';
```

2.2.6 Create database

Type command “CREATE DATABASE database_name;”

```
mysql> CREATE DATABASE test;
Query OK, 1 row affected (0.00 sec)
```

2.2.7 Create table for the website

Type “CREATE TABLE table_name (column_name datatype, column_name2 datatype, …);”. ALTER TABLE in the image below is used for changing column name

```
mysql> CREATE TABLE Users( user_id INT NOT NULL AUTO_INCREMENT, Name varchar(255), PRIMARY KEY(user_id) );
Query OK, 0 rows affected (0.01 sec)

mysql> ALTER TABLE Users CHANGE Name name varchar(255);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

2.3 Adding Simple Functional Website

2.3.1 Install PHP

1. Type “sudo apt-get install php”
2. Check it using “php --version”

2.3.2 Install php-mysqli to connect mysqli with php

Type “sudo apt-get install php-mysqli”

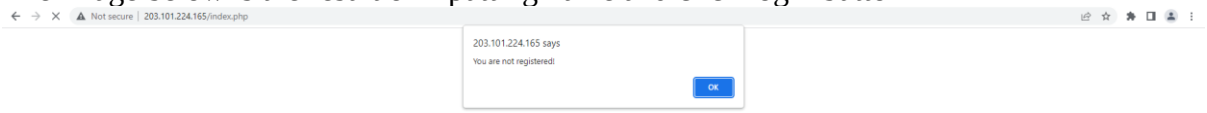
2.3.3 Create an index.php file in /var/www/html folder

```
<!DOCTYPE html>
<?php
try {
        if ( $_SERVER['REQUEST_METHOD'] == 'POST' ) {
                $servername = "localhost";
                $username = "root";
                $password = "root";
                $database = "test";
                $conn = mysqli_connect($servername, $username, $password,
$database);

                if (!$conn) {
                        die("Connection failed: ". mysqli_connect_error());
                }

                $name = $_POST["fullname"];
                if($_POST["login"]) {
                        $sql = "SELECT * FROM Users WHERE name='$name'";

                        $result = $conn->query($sql);

                        if ($result->num_rows > 0) {
                                while($row = $result->fetch_assoc()) {
                                        echo "welcome ".$row["name"]."<br>";
                                }
                        } else {
                                echo "<script>alert('You are not
registered!')</script>";
                        }
                } else {
                        $sql = "INSERT INTO Users (user_id, name) VALUES
(null,'$name');";

                        $result = $conn->query($sql);

                        if($result === TRUE) {
                                echo "<script>alert('Registered
successfully!')</script>";
                        } else {
```

```
                                echo "Error: ". $sql . "<br>" .
mysqli_error($conn);
                        }
                }
        }
        mysqli_close($conn);
} catch (Exception $e) {
        echo $e;
}
?>
<html lang="en">
  <head>
    <meta charset=UTF-8">
    <meta name="description" content="Testing Web Page in LXC">
    <meta name="keywords" content="Web, Test">
    <meta name="author" content="Elvan Alandi">
    <title>Testing Website</title>
  </head>
  <body>
        <h1>Login Using Name</h1>
        <form action="index.php" method="post">
                <label for="fullname">Full Name : </label>
                <input type="text" id="fullname" name="fullname"><br><br>
                <input type="submit" name="login" value="Login">
                <input type="submit" name="register" value="Register">
        </form>
  </body>
</html>
```

2.3.4 Test the web page

The image below is the result of inputting name and click login button



The image below is the result of inputting name and click register button

I input “test” in the full name input box

This is the result of successful login and the ‘test’ name inserted into the database.


```
mysql> select * from Users;
+---------+-------+
| user_id | name  |
+---------+-------+
|       1 | elvan |
|       2 | test  |
+---------+-------+
2 rows in set (0.00 sec)
```

References

[1] "What is PEM Format?", *knowledge.digicert.com*. [Online]. Available: https://knowledge.digicert.com/quovadis/ssl-certificates/ssl-general-topics/what-is-pem-format.html#:~:text=PEM%20or%20Privacy%20Enhanced%20Mail,very%20distinct%20headers%20and%20footers. [Accessed: 17- Aug- 2022].

[2] "What are PEM files?", *s3.amazonaws.com*. [Online]. Available: https://s3.amazonaws.com/smhelpcenter/smhelp941/classic/Content/security/concepts/what_are_pem_files.htm. [Accessed: 17- Aug- 2022].

[3] "Certificate and key format for importing - AWS Certificate Manager", *docs.aws.amazon.com*. [Online]. Available: https://docs.aws.amazon.com/acm/latest/userguide/import-certificate-format.html#:~:text=PEM%20stands%20for%20Privacy%20Enhanced,a%20PEM%E2%80%93formatted%20file%20is%20. [Accessed: 17- Aug- 2022].

[4] "Privacy-Enhanced Mail - Wikipedia", *en.wikipedia.org*. [Online]. Available: https://en.wikipedia.org/wiki/Privacy-Enhanced_Mail. [Accessed: 17- Aug- 2022].

[5] N. Cheema, "The Difference Between .PEM And .PPK And How To Generate .PPK File For SSH From Putty", *c-sharpcorner.com*, 2020. [Online]. Available: https://www.c-sharpcorner.com/article/difference-between-pem-and-ppk/. [Accessed: 18- Aug- 2022].

[6] "42.4. Pluggable Authentication Modules (PAM)", *web.mit.edu*. [Online]. Available: http://web.mit.edu/rhel-doc/5/RHEL-5-manual/Deployment_Guide-en-US/ch-pam.html. [Accessed: 18- Aug- 2022].

[7] "What is a Container? - Docker", *Docker*. [Online]. Available: https://www.docker.com/resources/what-container/. [Accessed: 18- Aug- 2022].

[8] D. Both, "A Linux user's guide to Logical Volume Management", *opensource.com*, 2016. [Online]. Available: https://opensource.com/business/16/9/linux-users-guide-lvm. [Accessed: 20- Aug- 2022].