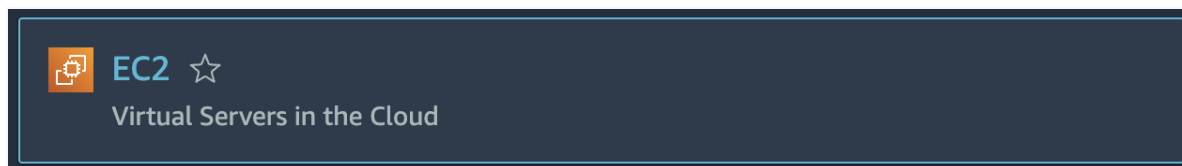


Introduction

In this Lab Step you will launch an instance in a private subnet created earlier. Although the instance does not have database server software installed on it, you can think of it as a database server, which is often shielded from direct access from the internet for security reasons. Once the instance is up and running, you will learn how to SSH into it from your local host, via a bastion host in the public subnet.

Instructions

1. In the AWS Management Console search bar, enter *EC2*, and click the **EC2** result under **Services**:



2. To see available instances, click **Instances** in the left-hand menu:



3. Click **Launch instances**:



4. In the **Name and tags** section, enter *private* under **Name**.


5. In the **Application and OS Images** section, select the **Amazon Linux 2 AMI (HVM) - Kernel 5.10** option under **Quick Start**:


▼ Application and OS Images (Amazon Machine Image) [Info](#)


An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below


 Search our full catalog including 1000s of application and OS images


Quick Start


**Amazon Linux**
aws


**Ubuntu**
ubuntu

**Windows**
Microsoft

**Red Hat**
Red Hat

**SUSE Linux**
SUSE




[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type

Free tier eligible ▼

ami-0ca285d4c2cda3300 (64-bit (x86)) / ami-0f48d15c9efb5f63d (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20220426.0 x86_64 HVM gp2

Architecture

AMI ID

64-bit (x86) ▼

ami-0ca285d4c2cda3300

6. In the **Instance Type** section, you should not change any options. Simply make sure the default **t2.micro** is selected:

Instance type

t2.micro
Family: t2 1 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0116 USD per Hour
On-Demand Windows pricing: 0.0162 USD per Hour

Free tier eligible

▼

[Compare instance types](#)

7. In the **Key pair** section, select the `338139317922` keypair:

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Select ▲

Q

Proceed without a key pair (Not recommended) Default value

226375879776
Type: rsa

↻

[Create new key pair](#)

Edit

Note: Your keypair may differ from the screenshot.

Reminder: The PEM or PPK formatted key pair can be downloaded directly from the **Your lab data** section of the Cloud Academy Lab page at any time.

8. In the **Network settings** section, click **Edit**, and configure the following instance details:

- **VPC:** Select the **cloudacademy-labs** VPC
- **Subnet:** Select the **Private-A** subnet
- **Auto-assign Public IP:** Make sure this is **disabled**
- **Firewall:** Select **Create security group**
- **Security group name:** Enter *SG-Private*
- **Description:** Enter *Security group for private subnet instances. Accept SSH inbound requests from Bastion host only.*
- **Type:** SSH
- **Protocol:** TCP
- **Port:** 22
- **Source type:** Custom
- **Source:** SG-bastion
 - *Tip:* If you don't recall the name of your bastion host's security group, leave the **Source** as **Custom**, and start typing "*bastion*". It will find the security group for you. (Example: SG-bastion)
- Click **Add security group rule**
- **Type:** HTTPS
- **Protocol:** TCP
- **Port:** 443
- **Source type:** Custom
- **Source:** *10.0.20.0/24 (Public VPC CIDR)*
- *Note:* If you also needed Windows access, you would add another rule: Type RDP; Protocol TCP; Port 3389; Source *SG-bastion*

Security group name - required

SG-Private

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and . _ - / () # , @ [] + = & ; { } ! \$ *

Description - required [Info](#)

Security group for private subnet instances. Accept SSH inbound requests from Bast

Inbound security groups rules

▼ Security group rule 1 (TCP, 22, sg-00c9d2dbd46b1b8b6)

[Remove](#)**Type** [Info](#)

ssh ▼

Protocol [Info](#)

TCP

Port range [Info](#)

22

Source type [Info](#)

Custom ▼

Source [Info](#)

sg-00c9d2dbd46b1b8b6 ✕

Description - optional [Info](#)*e.g. SSH for admin desktop*

▼ Security group rule 2 (TCP, 443, 10.0.20.0/24)

[Remove](#)**Type** [Info](#)

HTTPS ▼

Protocol [Info](#)

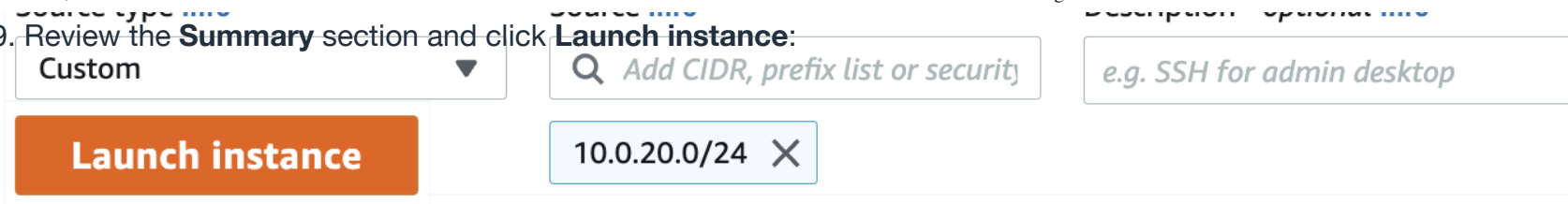
TCP

Port range [Info](#)

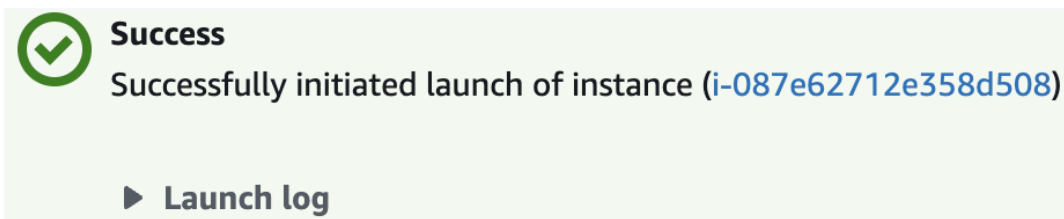
443

Source type [Info](#)**Source** [Info](#)**Description - optional** [Info](#)

9. Review the **Summary** section and click





A confirmation page will let you know that your instance is launching:



10. Click **View all instances** at the bottom of the confirmation page. On the **Instances** screen, you can view the status of your instance. It will take a short time for your instance to be launched. When you launch an instance, its initial state is **pending**, but it will transition to **running** within about 30 seconds.

11. Select the **private** instance. In the **Security** tab, click the actionable security group link (for example, **SG-Private**):

Details	Security	Networking	Storage	Status checks
▼ Security details				
IAM Role —		Owner ID  429347966565		
Security groups  sg-0d571cc746035023f (SG-Private)				

12. Switch to the **Inbound rules** tab of the security group and locate the SSH rule. If the **Source** is not the security group of your bastion host, click **Edit inbound rules** and enter *bastion* for the **Source**. The lookup will find the **SG-Bastion** security group. Make sure it is the source and then click **Save**.

In production, you may want to restrict **Outbound** access too, but for the purposes of this Lab, you can leave **All traffic** to all **Destinations** (0.0.0.0/0).

In an earlier Lab Step, you launched a bastion host which used its own security group. However, at the time you configured the security group, there was not a security group for your private instance on the private subnet. Because there is now, you can further tighten down the bastion host's security group. You'll do that next before using SSH to connect to your bastion host and private instance.

13. From the [VPC Dashboard](#), click **Security Groups**. Make note of the **Group ID** of the **SG-Private** security group. (For example, sg-9c7406e6)

14. Select the **SG-bastion** security group, switch to the **Outbound rules** tab, and click **Edit outbound rules**. Now that you have a private security group, you can restrict **Outbound rules** to instances using **SG-Private**. Configure the following:

- **Type:** SSH
- **Protocol:** TCP
- **Port:** 22
- **Destination:** Select **Custom** and then enter the security **Security group ID** of **SG-Private**

Warning: Make sure to delete the existing SG rule, and add a new one.

Click **Save rules** when ready.

Next, you will SSH into your bastion host, and enable ssh-agent forwarding so you can SSH (jump) to the private instance in your private subnet. The instructions below assume your local machine is Linux. However, you could use an SSH client such as PuTTY and download a PPK key file from the Cloud Academy Lab page (**Credentials** section).

15. This instruction is split into two different sections:

- One for those with local operating systems that are **Linux/Mac** based
- One for those with local operating systems that are **Windows** based

Complete the instructions below (a,b, c, ...) for the one that applies to you, not both.

Linux/Mac instructions

a. Download the PEM SSH key file from the **Credentials** section of the Cloud Academy lab page.

b. Make sure the permissions are correct on the PEM key file. From a terminal window in the directory you downloaded it to:



```
chmod 400 PEMfilename.pem
```

Since copying SSH private keys to a bastion instance is a security risk, you will enable SSH agent forwarding next. The `ssh-add` command can add private keys to the keychain application. Essentially, the private key will be used without having to copy it to the bastion host.

Note: Realize that attempting to SSH directly from your local host to the private IP of the instance in your private subnet will fail. (If not convinced, take a moment to try to. The connection will be refused.)

c. Enter the following command to add private keys to the authentication agent:



```
ssh-add -k PEMfilename.pem
```

You should see output similar to **Identity added: PEMfilename.pem (PEMfilename.pem)**.

d. Verify the key was added:



```
ssh-add -L
```

You should see output similar to **ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBA . . . NI3YfMHq7I 788348490820.pem**.

e. SSH into your bastion host using the authentication agent you just added:



```
ssh -A ec2-user@BastionHostPublicIP
```

Important! You must use the `-A` option shown above to enable the forwarding of the authentication agent. If you copy/paste the ssh command from the EC2 Dashboard's Connect button, you will not be able to connect to an instance on the private subnet.

Note: Your bastion host public IP can be found in the **Details** tab of the instance, under **Public IPv4 address**.

You have completed the Linux instructions. Skip the Windows instructions section below and proceed to instruction number 16.

Windows instructions

Windows requires an SSH client such as PuTTY, which also includes Pageant. Pageant is an SSH authentication agent that enables you to SSH into a private Linux instance via a bastion host, without copying the SSH keys to the bastion host.

a. If you do not have PuTTY and Pageant installed on your system, download and install the latest PuTTY MSI (Windows Installer) at <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>:

Package files

You probably want one of these. They include versions of all the PuTTY utilities.

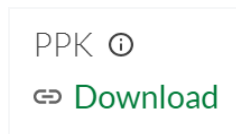
(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

MSI ('Windows Installer')

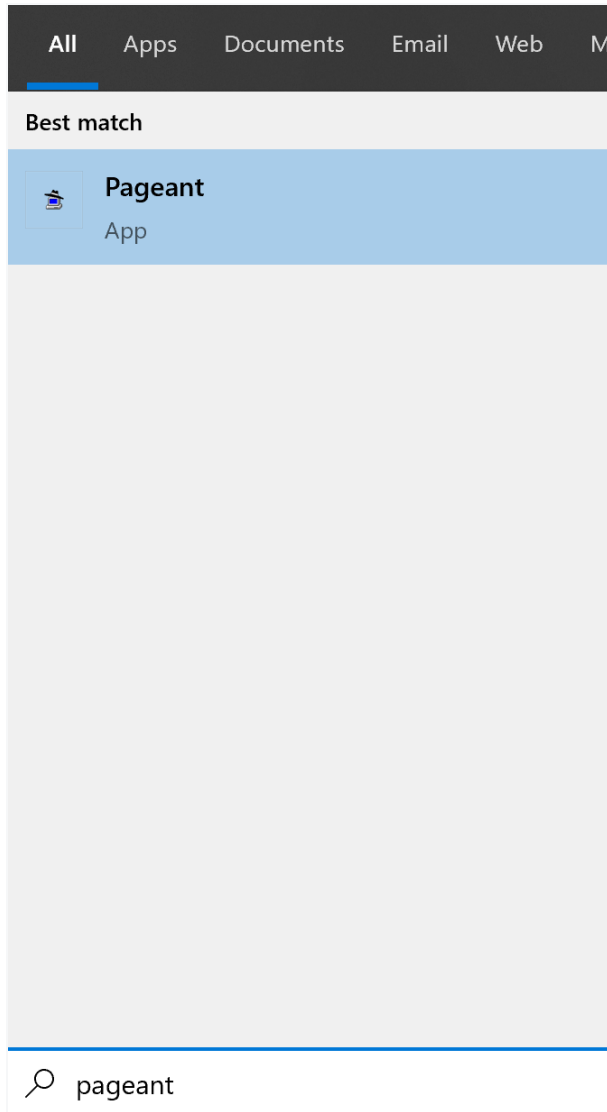
32-bit:	putty-0.73-installer.msi	(or by FTP)	(signature)
64-bit:	putty-64bit-0.73-installer.msi	(or by FTP)	(signature)

A free and useful SSH utility is called PuTTY. PuTTY supports SSH connections as well as key generation and conversion. The PuTTY package also includes Pageant. Pageant is an SSH authentication agent that enables agent forwarding on Windows.

b. Download the **PPK** key **Credentials** section in the upper-left corner of this lab:

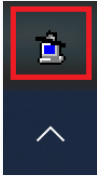


c. In the Windows search box enter *pageant* and click the **Pageant App** result:

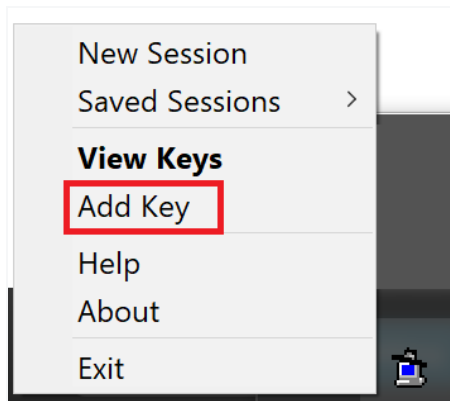


Alternatively, you can enter *pageant* into a command prompt terminal to start Pageant.

Note that Pageant runs as a Windows service. It should be displayed in your Windows task tray, but could be "hidden". Hence, you may need to "show hidden icons". Pageant is displayed as a terminal with a hat on it:



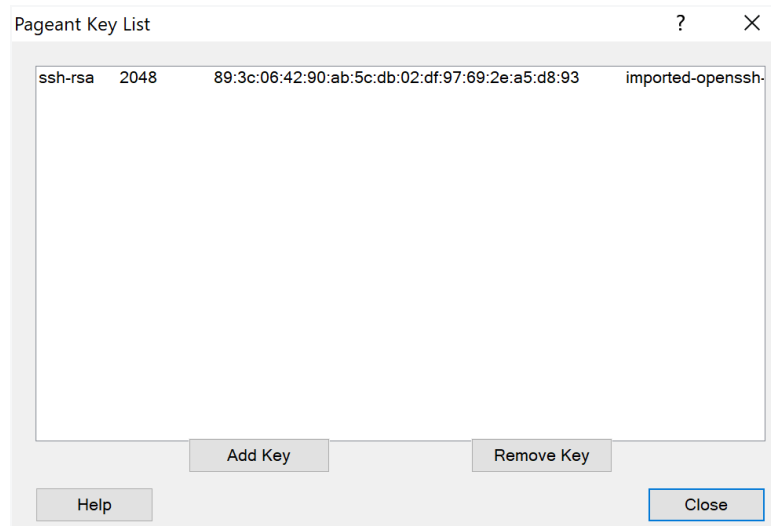
d. Right-click the Pageant icon and select **Add Key**:



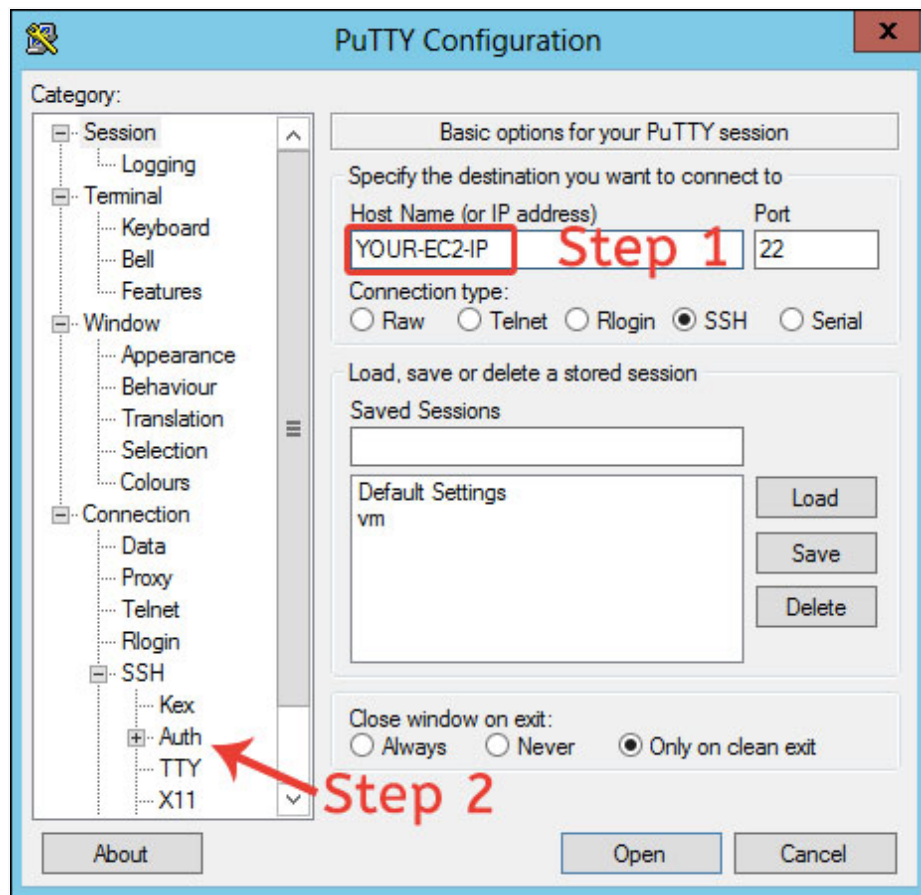
e. Browse to the PPK key file you downloaded earlier and click **Open**:



With the key added to Pageant, it is now available for use with agent forwarding. You can verify the key is added by right-clicking the Pageant icon and selecting **View Keys**:

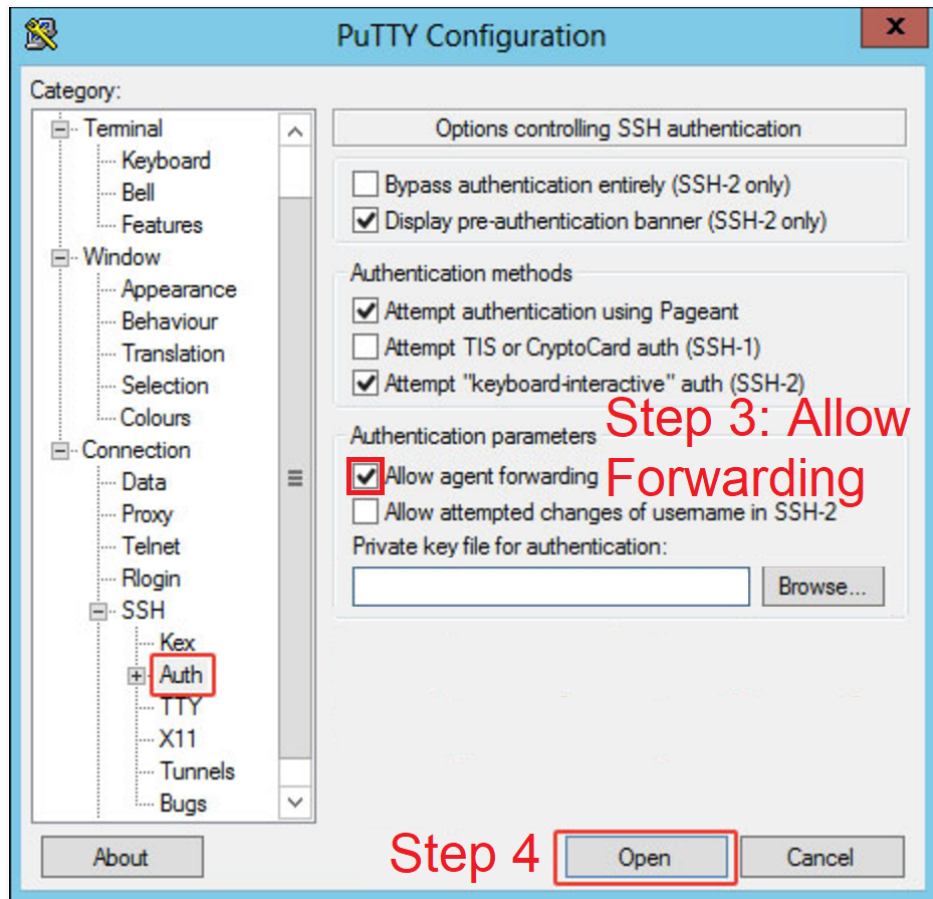


f. Open PuTTY and in the **Host Name (or IP address) field**, enter the username *ec2-user* followed by @ and the Public IP address of the bastion host. The complete field value resembles *ec2-user@12.34.56.78.:*



Note: Your bastion host public IP can be found in the **Details** tab of the instance, under **Public IPv4 address**.

g. Navigate to the **Connection > SSH > Auth** section. Select the **Allow agent forwarding** and click **Open**:



You **do not** need to browse to the key because the key is already available through Pageant which has the key added and available for forwarding.

h. Wait several seconds for the authentication prompt and click **Yes** in the **PuTTY Security Alert** to acknowledge you trust the host:



With agent forwarding enabled you will be able to connect to the Linux instance running in the private subnet without having a copy of the private key on the bastion host.

i. To verify the SSH connection was made using agent forwarding, enter:

 [Copy code](#)

```
echo $SSH_AUTH_SOCK
```

If there is no output, the connection was made without using agent forwarding. If that is the case, perform the instructions again to ensure each one is performed correctly.

You have now completed the Windows instructions. Proceed to instruction number 16 below.

16. SSH into the private instance running in your private subnet:



```
ssh ec2-user@PrivateInstancePrivateIPAddress
```

Reminder: You can get the private IP address of the instance running in your private subnet from the running instance's **Description tab** > **Private IP** field.

Enter yes to the **Are you sure you want to continue connecting (yes/no)?** prompt:

```
The authenticity of host '10.0.10.162 (10.0.10.162)' can't be established.  
ECDSA key fingerprint is SHA256:F31HpCPzQY9mEP02famwUztGMe9mDT5KHZZrRBy7t6U.  
ECDSA key fingerprint is MD5:8f:97:e9:91:05:67:bf:94:27:c8:a4:1e:3a:c1:8a:38.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '10.0.10.162' (ECDSA) to the list of known hosts.  
  
  ____|  _||_  )  
  _||_ (  _||_ /  Amazon Linux 2 AMI  
  _||_ \_||_||_||  
  
https://aws.amazon.com/amazon-linux-2/
```

Congratulations! You successfully created and configured a bastion host in order to SSH into a private instance on a private subnet.

17. Enter the following command to install operating system updates:



```
sudo yum update
```

```
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Could not retrieve mirrorlist http://amazonlinux.us-west-2.amazonaws.com/2/core/latest/x86_64/mirror.list error was
12: Timeout on http://amazonlinux.us-west-2.amazonaws.com/2/core/latest/x86_64/mirror.list: (28, 'Connection timed out after 30001
milliseconds')
```

Although the private instance security group is configured correctly, and you should have outbound access to the internet, it still timed out. The time out is caused by the private NACL denying inbound HTTP traffic. You will need Network Address Translation (NAT) to allow your private instance *outgoing* connectivity to the Internet, while at the same time *blocking* inbound traffic from the Internet. Once NAT is in place, you should be able to get package updates.

Note: For convenience sake, leave the SSH connection to your private instance open. You will return to it in a future Lab Step.

Summary

In this Lab Step you launched a basic instance that mimics a database server on a private subnet. It has a private IP address, which in and of itself makes it more secure. You learned how you can still SSH to the private instance by going through a bastion host. Further, rather than copying private SSH keys to the bastion host (a security risk), you updated your own authentication chain. That way, you can still SSH from the bastion host with SSH keys that are not physically copied to the bastion host. Although SSH access to the private instance worked, you discovered that your private instances still can't access the internet for the purposes of operating system updates however.

See the [Securely Connect to Linux Instances Running in a Private Amazon VPC](#) blog for more information. The blog contains great information whether you are connecting from a local Linux (Mac) or Windows host.