Introduction

In a previous lab step, you configured a bastion host so you could SSH into instances on your private subnet. However, you were unable to access the internet from private instances. A common use case for needing such internet access is operating system package updates. However, the installation failed.

In this lab step, now that you have a network address translation device in your VPC, you will test if the operating system updates will work from your private instances.

Instructions

1. If you are not already connected to the private instance via your bastion host, use an SSH client on your localhost to connect to your bastion host.

Hint: Refer to a previous lab step for complete instructions. Since you added private keys to the authentication earlier, you should be able to SSH directly from your localhost to the bastion host. Try the following commands on Mac OSX or Linux:

Copy code

```
ssh-add -L    # Confirm the keys are in place on your local host.
If not, add them with: $ ssh-add -K PEMfilename.pem
ssh -A ec2-user@BastionHostPublicIP    # SSH from your local host
to the bastion host in your public subnet
ssh ec2-user@PrivateInstancePrivateIPaddress   # SSH from the
bastion host into the private instance on your private subnet.
```

Note: If you are using Windows, you can use PuTTY to connect to the bastion host. You will need the PPK key pair, and the **Public IP** address of the **Bastion host**. Remember to use **ec2-user** as the username.

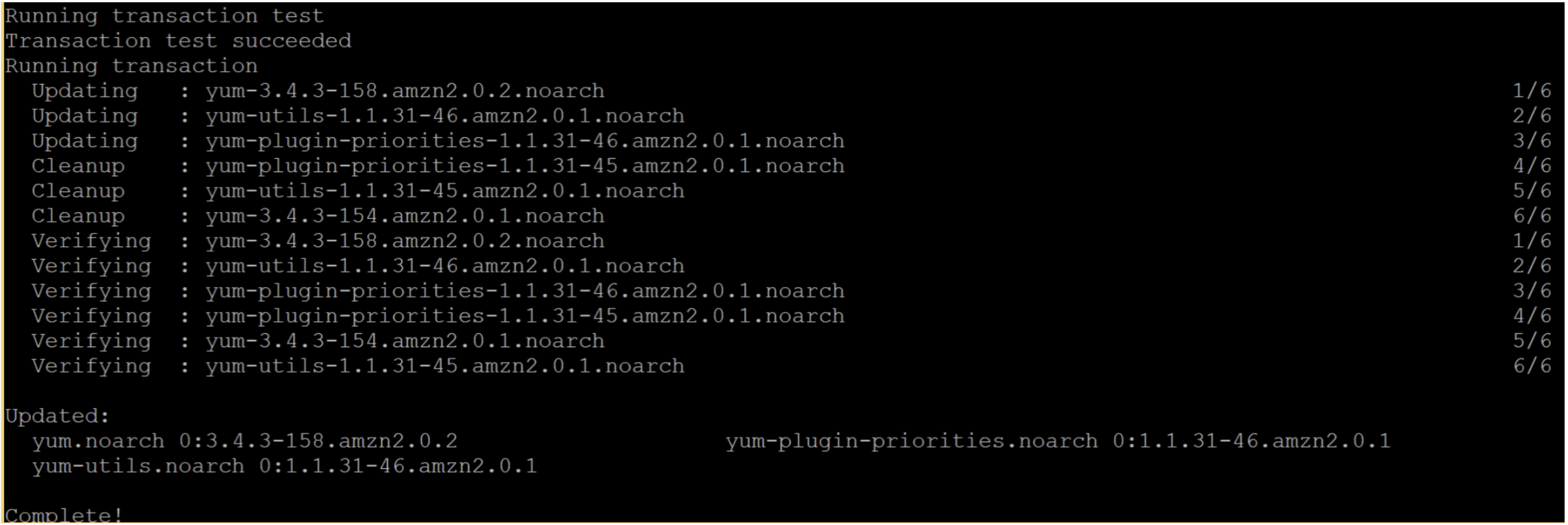
2. Enter the following command to use the yum package manager to update the operating system packages on your private instance.

Important! There are two important configurations worth mentioning again as to why this command should work in your lab environment:

- The private NACL has an Outbound Rule permitting HTTP (port 80) or HTTPS (port 443) access to anywhere on the internet (0.0.0.0/0)
- The security group for the NAT device allows HTTP/S access from any instance in the private subnet (that uses the private instance security group, which permits any destination as well)

Copy code

```
sudo yum update -y
```



```
Running transaction test
Transaction test succeeded
Running transaction
  Updating    : yum-3.4.3-158.amzn2.0.2.noarch                                    1/6
  Updating    : yum-utils-1.1.31-46.amzn2.0.1.noarch                              2/6
  Updating    : yum-plugin-priorities-1.1.31-46.amzn2.0.1.noarch                  3/6
  Cleanup     : yum-plugin-priorities-1.1.31-45.amzn2.0.1.noarch                  4/6
  Cleanup     : yum-utils-1.1.31-45.amzn2.0.1.noarch                              5/6
  Cleanup     : yum-3.4.3-154.amzn2.0.1.noarch                                    6/6
  Verifying   : yum-3.4.3-158.amzn2.0.2.noarch                                    1/6
  Verifying   : yum-utils-1.1.31-46.amzn2.0.1.noarch                              2/6
  Verifying   : yum-plugin-priorities-1.1.31-46.amzn2.0.1.noarch                  3/6
  Verifying   : yum-plugin-priorities-1.1.31-45.amzn2.0.1.noarch                  4/6
  Verifying   : yum-3.4.3-154.amzn2.0.1.noarch                                    5/6
  Verifying   : yum-utils-1.1.31-45.amzn2.0.1.noarch                              6/6

Updated:
  yum.noarch 0:3.4.3-158.amzn2.0.2                     yum-plugin-priorities.noarch 0:1.1.31-46.amzn2.0.1
  yum-utils.noarch 0:1.1.31-46.amzn2.0.1

Complete!
```

It worked!

Note: The exact package updates for your system may vary slightly, for example as the default AWS Linux AMI changes over time. You may also encounter a successful message: **No packages marked for update**, indicating the instance accessed the public internet, but no updates were required.

Summary

Thanks to previously performed configurations in several lab steps, you successfully performed an operating system update on an instance in a private subnet. To do so required access from your private instance, utilizing the route table of a private NACL and a NAT device and its security group.