**Homework 4: Virtual Private Cloud**

Due: Sunday October 7 11:59PM

**Create a Virtual Private Cloud (10 points)**

Screenshot of your VPCs list (make sure the VPC name and CIDR range are visible):

|  |
| --- |
|  |

**Create A Public Subnet (15 points)**

Screenshot of your Subnets list (make sure the CIDR range is visible): (14 points)

|  |
| --- |
|  |

Is “Auto-assign public IPv4 address” showing “Yes”? (1 point)

|  |
| --- |
|  |

**Create an EC2 instance in your Public Subnet (15 points)**

Screenshot of your Instance Description (the details within the “Description” tab of your EC2 Instance - you do not need to go any lower than the “Owner” field) (4 points)

|  |
| --- |
|  |

Screenshot of your Security Group Inbound rules: (4 points)

|  |
| --- |
|  |

Screenshot of your SSH attempt: (4 points)

|  |
| --- |
|  |

What was the outcome of your SSH request? Why? (3 points)

|  |
| --- |
| The request timed out. We have not yet set up an internet gateway, so there is no NAT for the instance’s public ip, it will not resolve to anything, and cannot connect. It needs to resolve to the private ip, but can’t.  Additionally, a route table needs to exist, to route traffic to the internet gateway before that will work. |

**Create Internet Gateway (10 points)**

Screenshot of your Internet Gateways list: (3 points)

|  |
| --- |
|  |

Screenshot of your SSH attempt: (4 points)

|  |
| --- |
|  |

What was the outcome of your SSH request? (3 points)

|  |
| --- |
| Still a timeout, even with the internet gateway there is no route table definition. The default route table allows access between instances in each subnet it’s associated to, but does not grant internet access, so the public IP still cannot resolve. |

**Create A Route Table (10 points)**

Screenshot of your Route Table Routes tab once it is configured: (1 point)

|  |
| --- |
|  |

Screenshot of your Route Table Subnet Associations tab once it is configured: (7 points)

|  |
| --- |
|  |

Screenshot of your ssh connection to EC2 instance:

|  |
| --- |
|  |

Screenshot of your ping request from the EC2 instance to Google (ping [www.google.com](http://www.google.com/))

(2 points)

|  |
| --- |
|  |

**Create A Private Subnet (10 points)**

Screenshot of your Subnets list (make sure the CIDR range is visible): (9 points)

|  |
| --- |
|  |

Is “Auto-assign public IPv4 address” showing “Yes”? (1 point)

|  |
| --- |
|  |

**Create an EC2 instance in your Private Subnet (15 points)**

Screenshot of your Private Instance Description (the details within the “Description” tab of your EC2 Instance - you do not need to go any lower than the “Owner” field) (5 points)

|  |
| --- |
|  |

Screenshot of your Security Group Inbound rules: (5 points)

|  |
| --- |
|  |

Screenshot of SSH connection from EC2 instance in Public subnet to EC2 instance in Private Subnet:

|  |
| --- |
| Using bastion host described below |

Screenshot of the last 20 lines of output from running ‘update’: (5 points)

|  |
| --- |
|  |

**Create a NAT Gateway (15 points)**

Name of the subnet your NAT Gateway is created in: (5 points)

|  |
| --- |
| (the public subnet) |

Screen shot of your Route Tables list. Make sure that the default Route Table is selected in the list, and the Routes tab is visible in your screen shot. (5 points)

|  |
| --- |
|  |

Screenshot of the results of your update command: (5 points)

|  |
| --- |
|  |

**Bonus: Set up a Network ACL (10 points)**

The Network ACL is another layer of security that are used at the network level. Set up a NACL for a server intended to run a web server. Provide screenshots of both your inbound and outbound configuration settings.

**Bonus: Create a Bastion Host (10 points)**

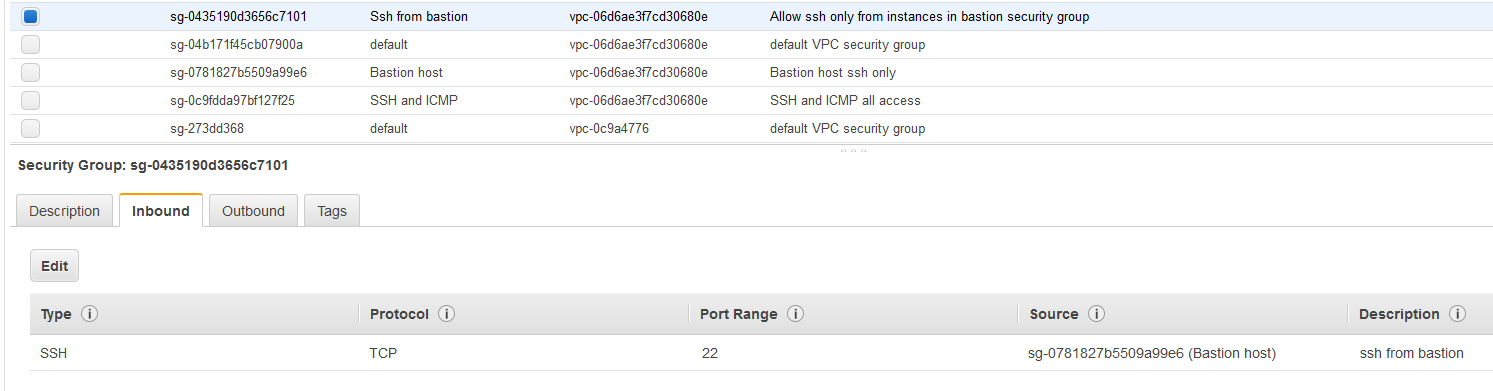
A Bastion Host is a specially configured EC2 instance which enables communications between your EC2 instances. As a bonus exercise, 1) describe how a Bastion Host works. 2) document and provide evidence (supporting screenshots) of a Bastion Host setup.

A bastion host is used from a public subnet (or the internet) and allows only incoming connections from a public subnet into a private subnet. It does this through specifically allowing access from the private subnet instances, which can be abstrated through security groups. It’s the opposite flow of a NAT gateway, which allows only outgoing connections from a public subnet, and used from a private subnet.

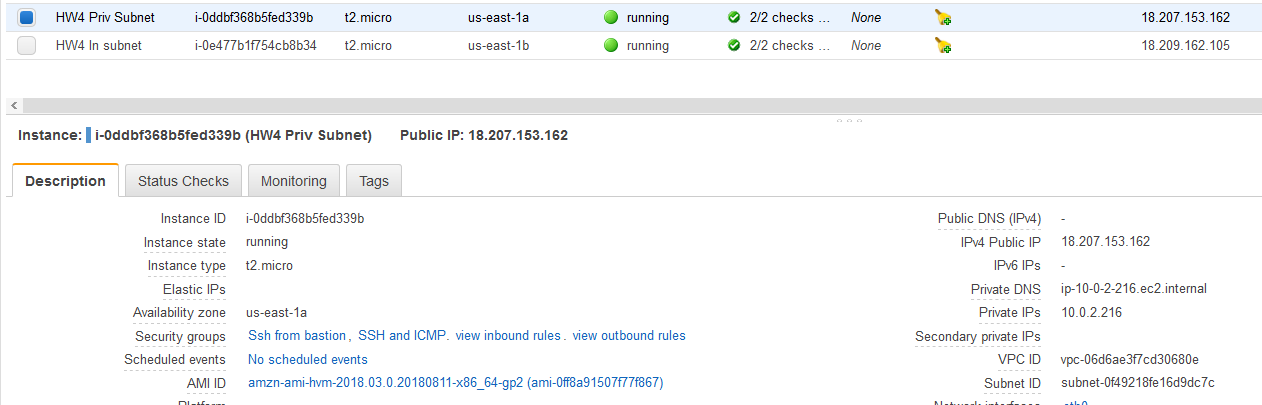
Setup:

Security groups:

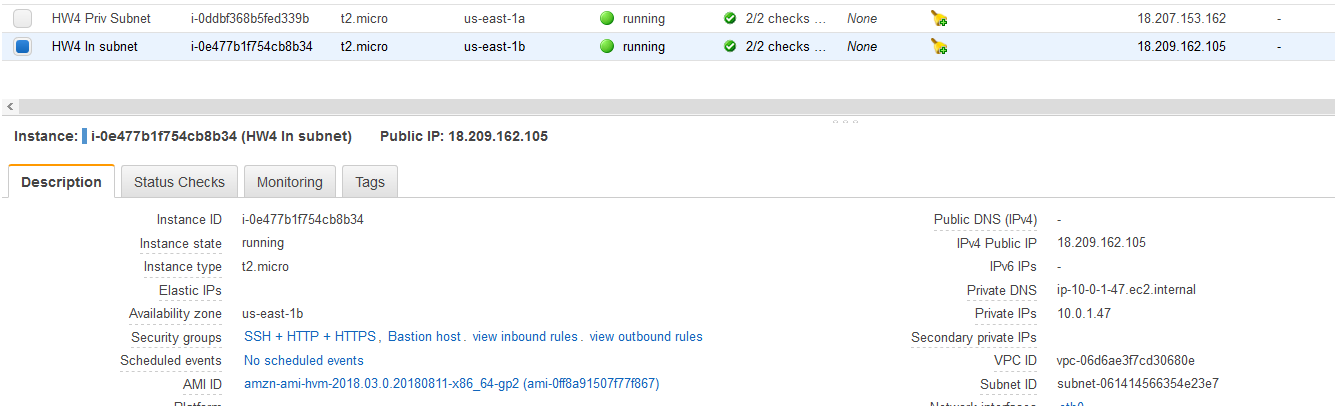
* A bastion host group for all bastion hosts in the vpc (only allows ssh)
* An ‘ssh from bastion’ group for all instances that should be connected to through each bastion host



Add ‘ssh from bastion’ security group to the instance in the private subnet

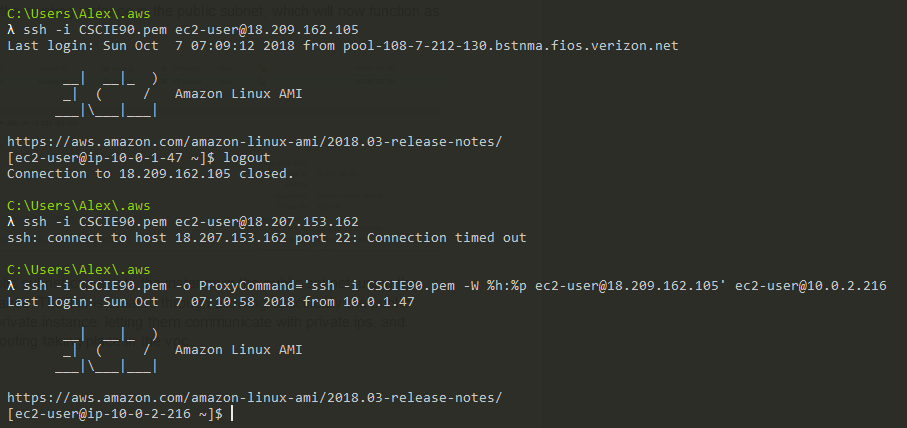


Add ‘bastion host’ group to the existing instance in the public subnet, which will now function as the bastion host



Now the instance in the public subnet can ssh to the instance in the public subnet, using the private ip of the private instance. The permissions are in place, and the instances can see each other within the vpc on the private ips, if they are known.

Next, we use ssh proxycommand to set up a stdin pipe from the local client to the bastion host, where the specified command is run on the bastion host, using the local client’s environment context. This way, the private key can remain on the local client, and piped to the command execution on the bastion host only for the duration of the command being run, without being saved to disk on the bastion host.



The login to the bastion host using it’s public ip succeeds, since it’s subnet is in the route table. The login to the private instance fails, since it’s subnet is not in the routing table (private subnet).

The proxy command to ssh into the private instance from the bastion host, after ssh-ing into the bastion host succeeds, since the bastion host can see the private instance on it’s private ip due to the security group setting.