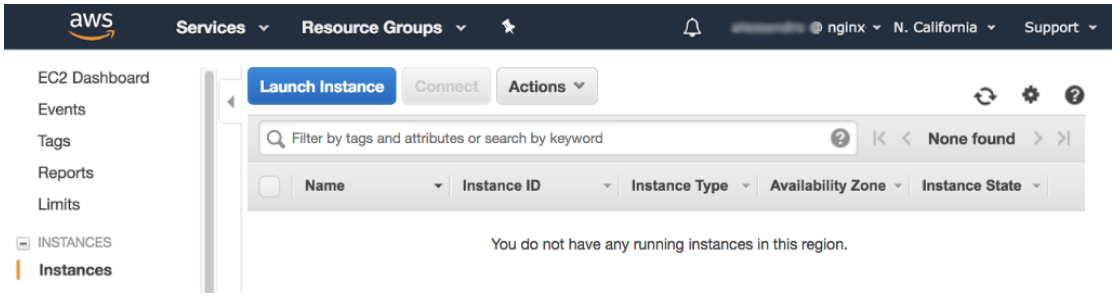
Experiment No: 07

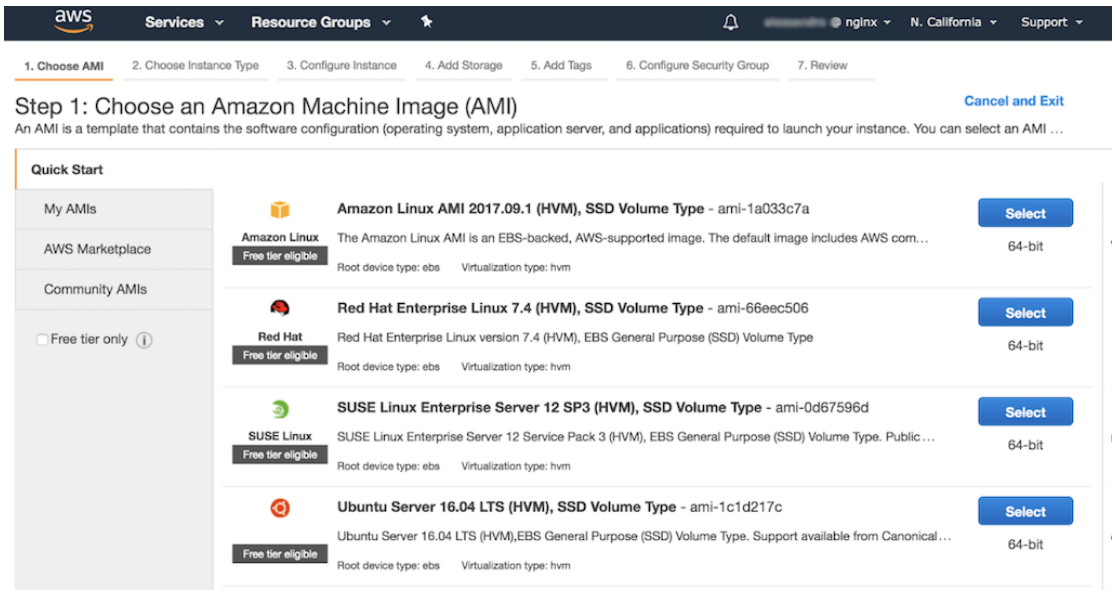
Name: Suraj P. Patil | Roll: 4120
Class: B.Tech | URN: 20131086
Div: B | Date: 07-10-2023

Creating an Amazon EC2 Instance

1. Log into the EC2 dashboard in the AWS Management Console (**https://console.aws.amazon.com/ec2/**).
2. In the left navigation bar, select **Instances**, then click the Launch Instance button.



3. In the **Step 1: Choose an Amazon Machine Image (AMI)** window, click the Select button for the Linux distribution of your choice.

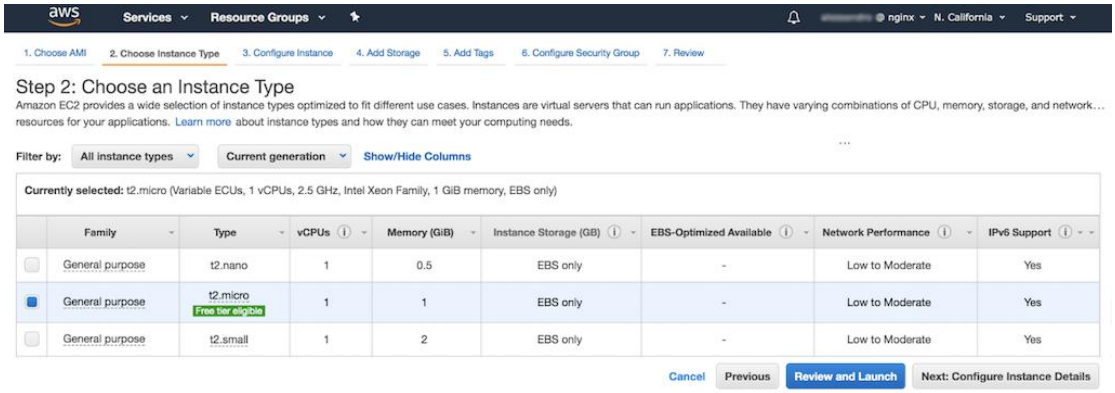


4. In the **Step 2: Choose an Instance Type** window, click the radio button for the appropriate instance type. In the screenshot, we are selecting a **t2.micro** instance, which is

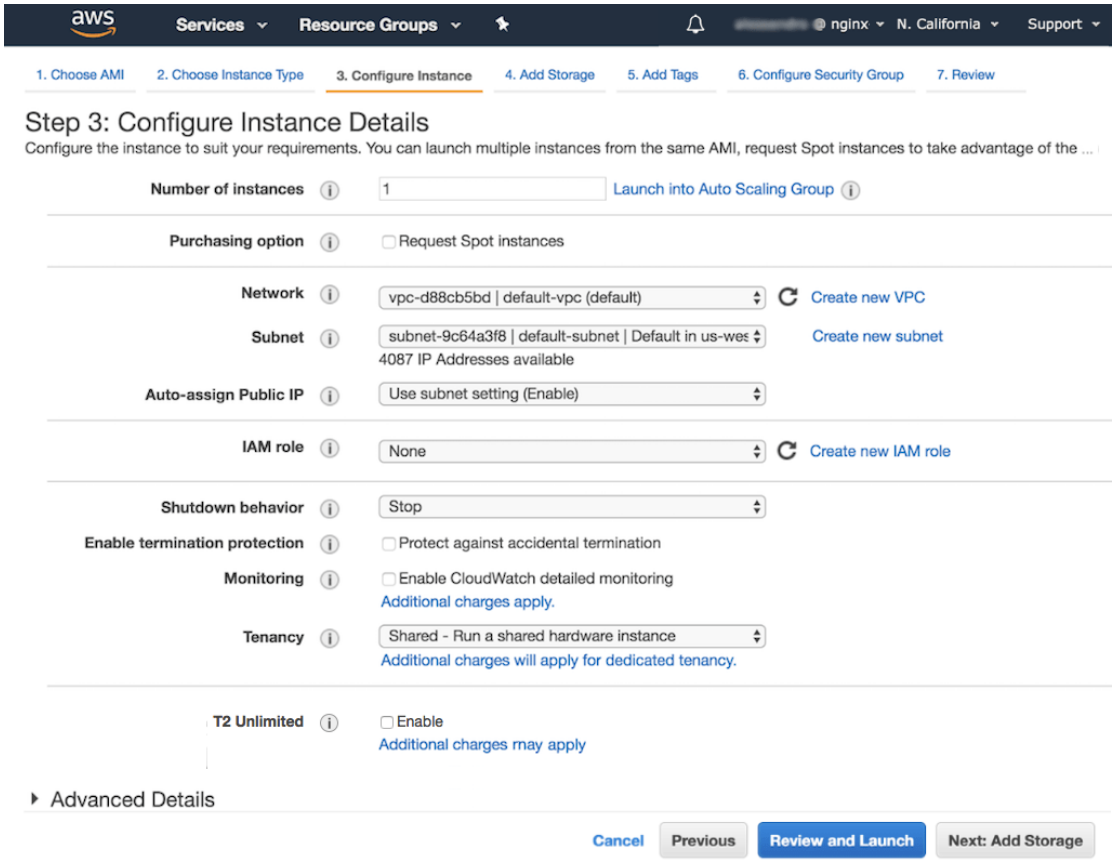
normally selected by default and is sufficient for demo purposes.

Note: At the time of publication of this guide, AWS gives you 750 hours of free usage per month with this instance type during the first year of your AWS account. Keep in mind, however, that if they run 24 hours a day, the sets of instances specified in the NGINX deployment guides use up the 750 hours in just a few days (just over 5 days for 6 instances, and just under 4 days for 8 instances).

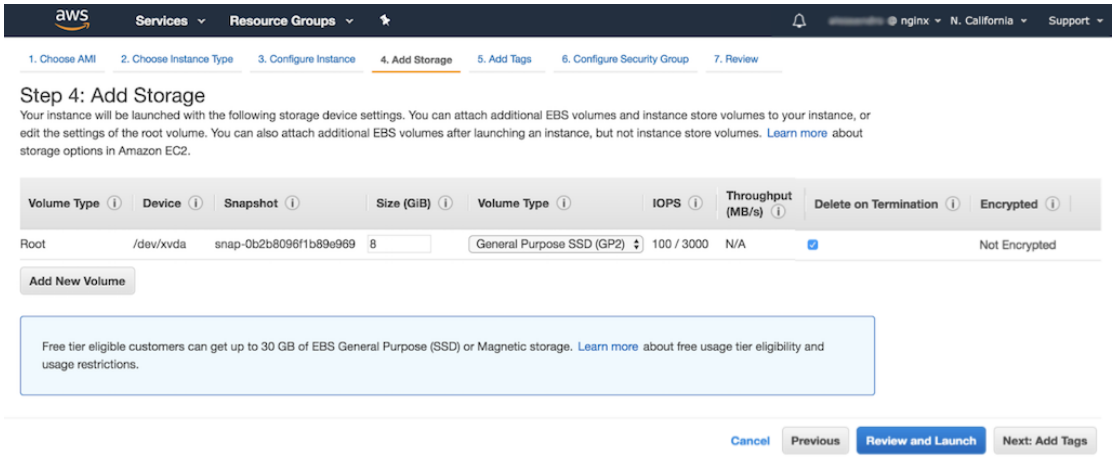
Click the **Next: Configure Instance Details** button to continue to the next step.



5. In the **Step 3: Configure Instance Details** window, select the default subnet for your VPC in the **Subnet** field, then click the **Next: Add Storage** button.



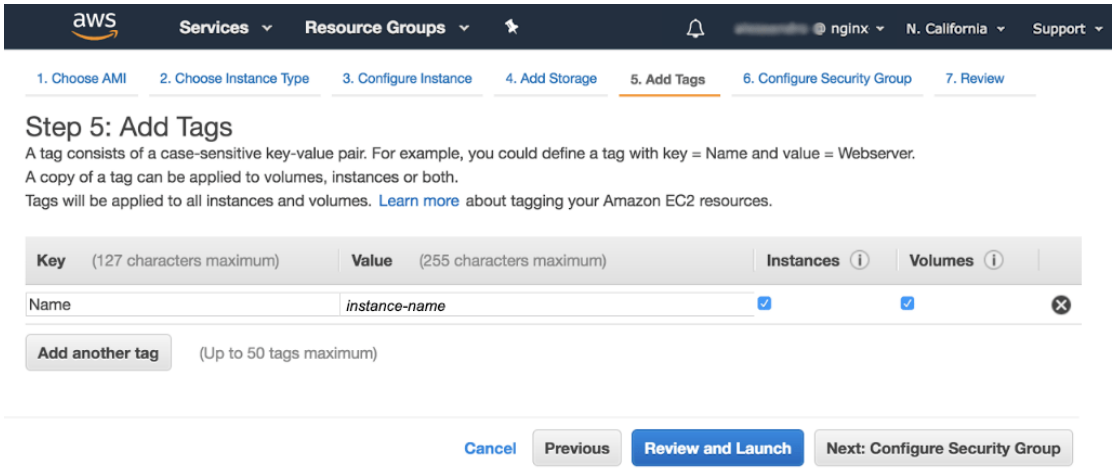
6. In the **Step 4: Add Storage** window, leave the defaults unchanged. Click the **Next: Add Tags** button.



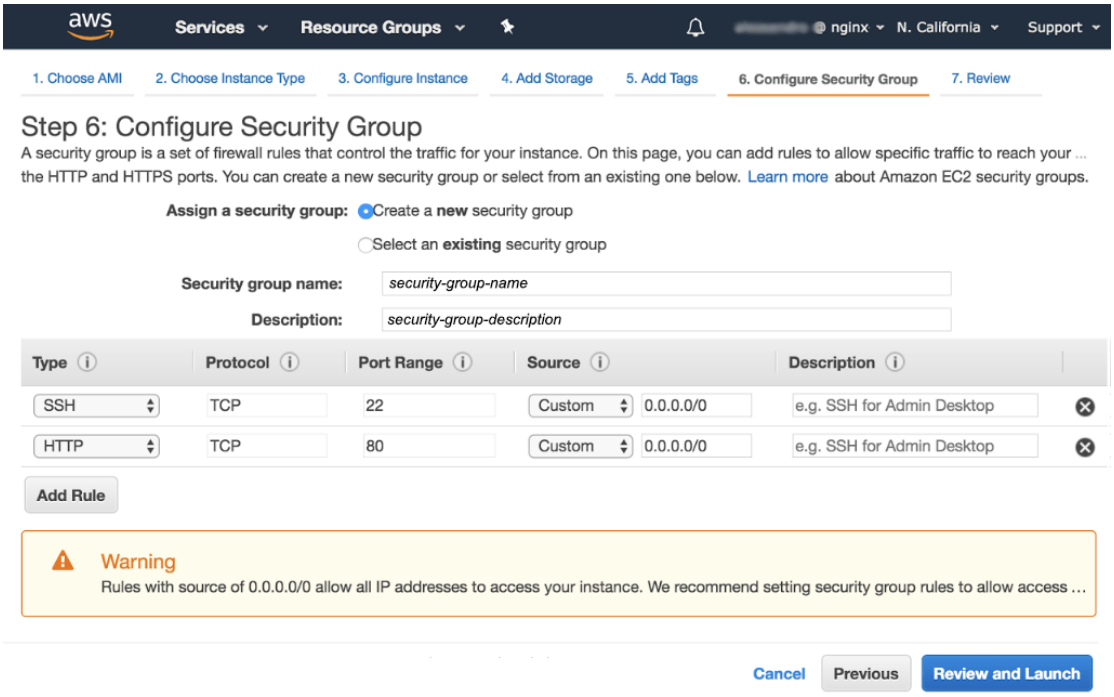
7. In the **Step 5: Add Tags** window, click the **Add Tag** button. Type **Name** in the **Key** field, and in the **Value** field type the instance name (the screenshot shows the result). This name is what will appear in the **Name** column of the summary table on the **Instances** tab of the EC2 dashboard (see the screenshot in Step 12, which shows one instance).

If you are following these instructions as directed by an NGINX deployment guide, the **Creating EC2 Instances and Installing the NGINX Software** section of the deployment guide specifies the instance names to use.

Click the **Next: Configure Security Group** button to continue to the next step.



8. In the **Step 6: Configure Security Group** window, select or enter the following values in the indicated fields:
    - **Assign a security group** –
        - If you are setting up a deployment with multiple instances (one in an NGINX deployment guide, for instance), and this is the first instance you are creating, select **Create a new security group**.
        - For subsequent instances, select **Select an existing security group** instead (it makes sense for all instances in a deployment to use the same security group).
    - **Security group name** – Name of the group. If you are following these instructions as directed by an NGINX deployment guide, the **Prerequisites and Required AWS Configuration** section of the deployment guide specifies the group name to use.
    - **Description** – Description of the group; the group name is often used.

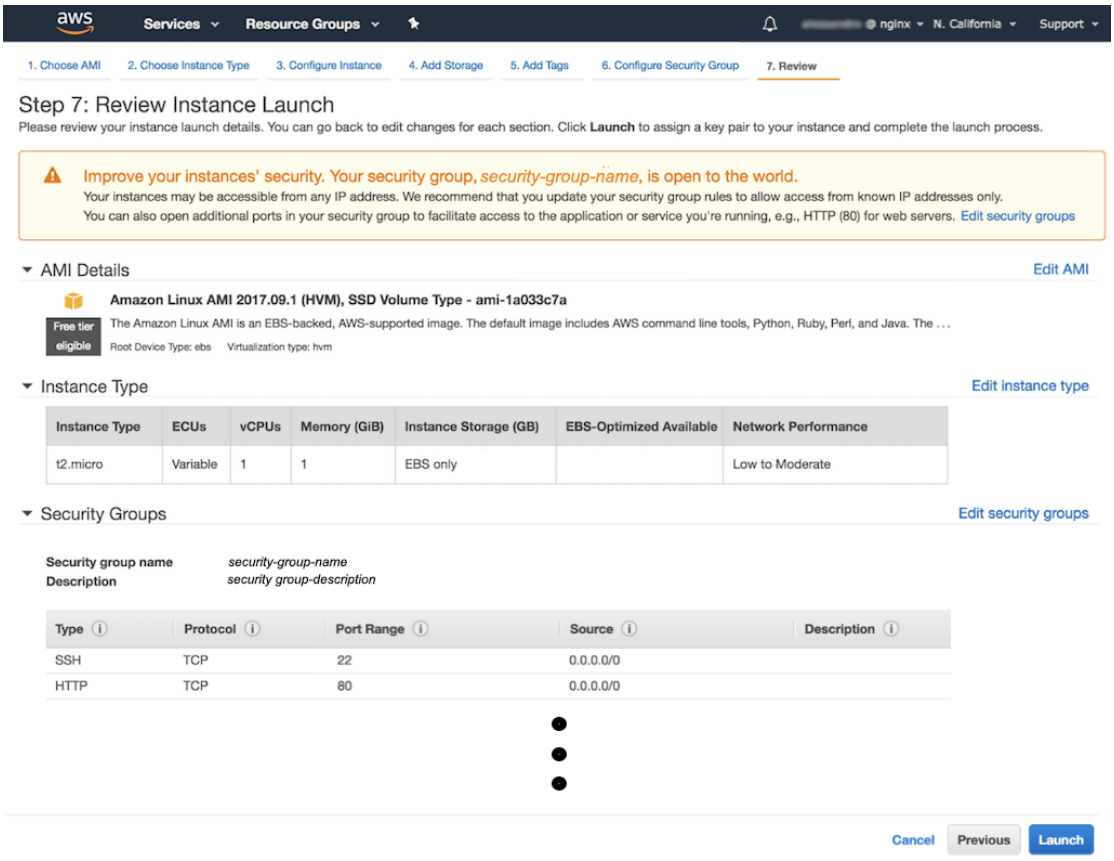


9. In the table, modify the default rule for SSH connections, if necessary, by selecting or setting the following values. They allow inbound SSH connections from all sources (any IP address):
    - **Type – SSH**
    - **Protocol – TCP**
    - **Port Range – 22**
    - **Source – Custom 0.0.0.0/0**
    - **Description – Accept SSH connections from all sources**
10. Create a rule that allows inbound HTTP connections from all sources, by clicking the **Add Rule** button and selecting or setting the following values in the new row:
    - **Type – HTTP**
    - **Protocol – TCP**
    - **Port Range – 80**
    - **Source – Custom 0.0.0.0/0**
    - **Description – Accept unencrypted HTTP connections from all sources**

    If appropriate, repeat this step to create a rule for HTTPS traffic.

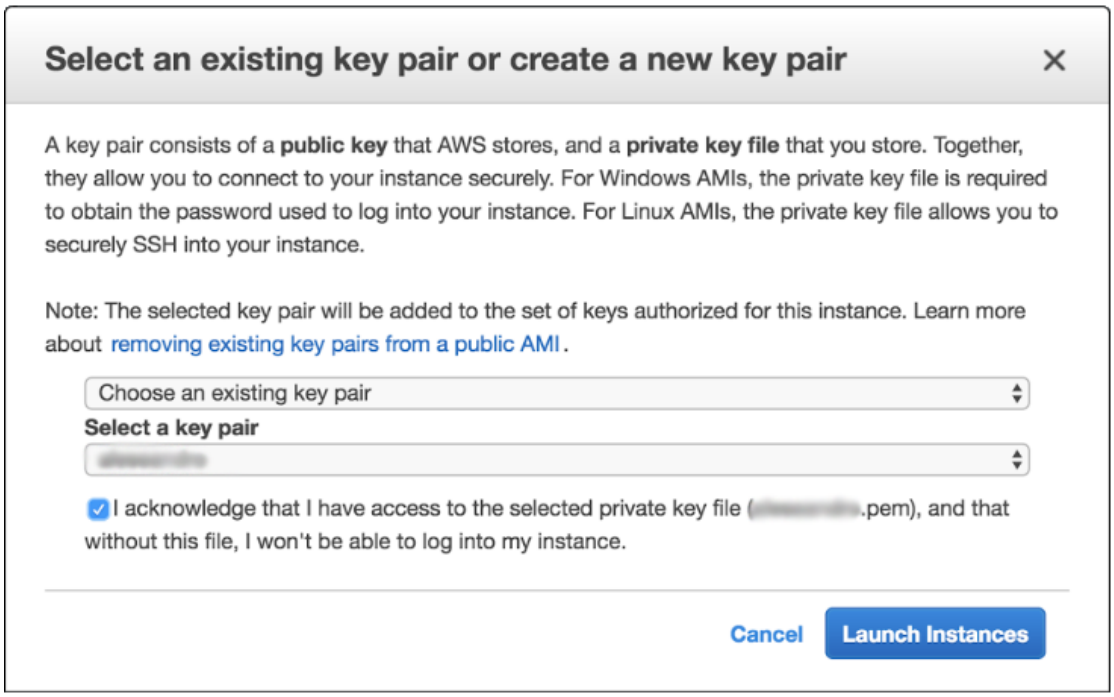
    When you've created all desired rules, click the Review and Launch button.

11. In the **Step 7: Review Instance Launch** window, verify the settings are correct. If so, click the **Launch** button in the lower-right corner of the window. To change settings, click the **Previous** button to go back to earlier windows.



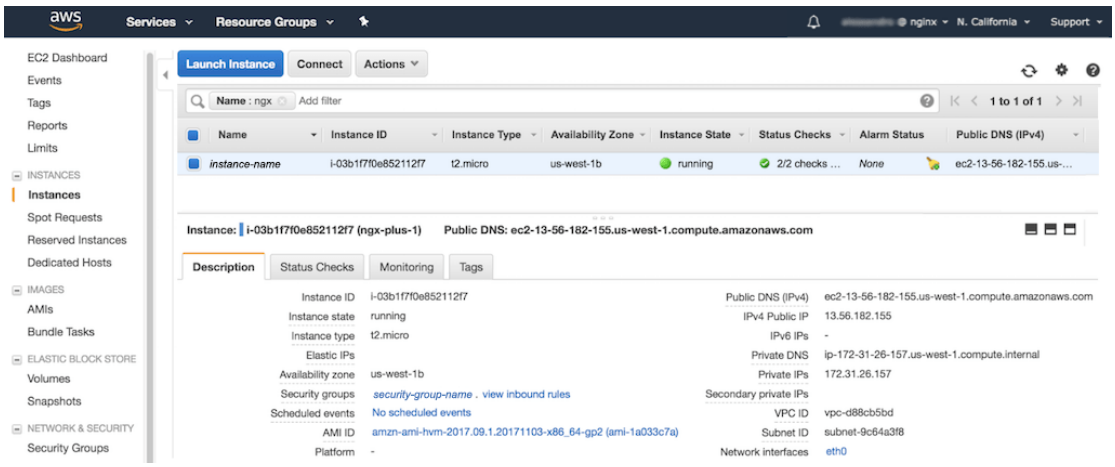
12. When you click the **Launch** button, a window pops up asking you to select an existing key pair or create a new key pair. Take the appropriate action for your use case, then click the **Launch Instances** button.

Note: It's a best practice – and essential in a production environment – to create a separate key for each EC2 instance, so that if a key is compromised only the single associated instance becomes vulnerable.



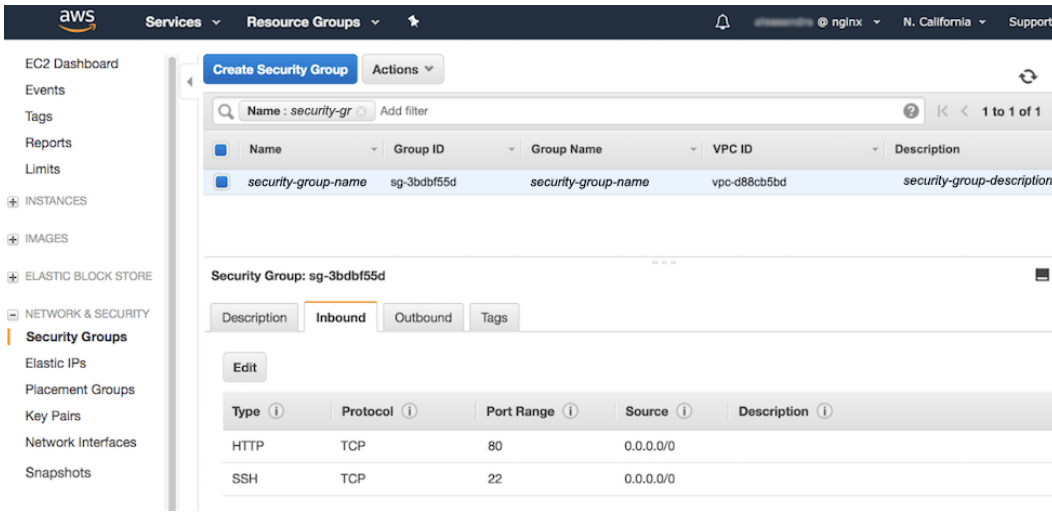
A **Launch Status** window pops up to confirm that your launch is underway. To confirm the details of your instance when the launch completes, click the View Instances button on that page.

The instances you have created so far are listed on the **Instances** dashboard. The following screenshot shows a single instance.



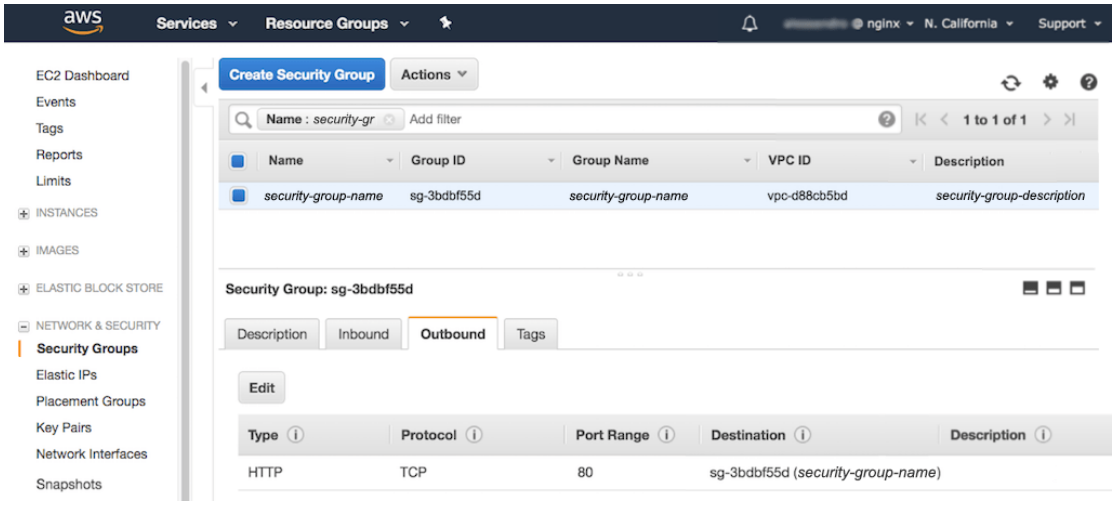
13. Finalize your security group rules. You need to do this only for the first instance in a given set, because all instances in a set can use the same security group.
    - In the left navigation bar, select **Security Groups**.

- Select the security group by clicking its radio button in the leftmost column of the table. A panel opens in the lower part of the window displaying details about the group.
- Open the **Inbound** tab and verify that the rules you created in Steps 9 and 10 are listed.



- Open the **Outbound** tab and click the **Edit** button to create a rule for outbound traffic. The set of rules depends on which ports you have used for traffic handled by the NGINX Plus instances:
    - If, for example, you have used port 80 both for client traffic and for health checks from a load balancer (for example, AWS Network Load Balancer), you need only one rule.
    - If you have configured separate ports for different purposes, or ports other than 80 (such as 443 for HTTPS), make the appropriate adjustments.

  In the **Destination** field, type the security group's ID, which appears in the **Group ID** column in the upper table (here it's **sg-3bdbf55d**).

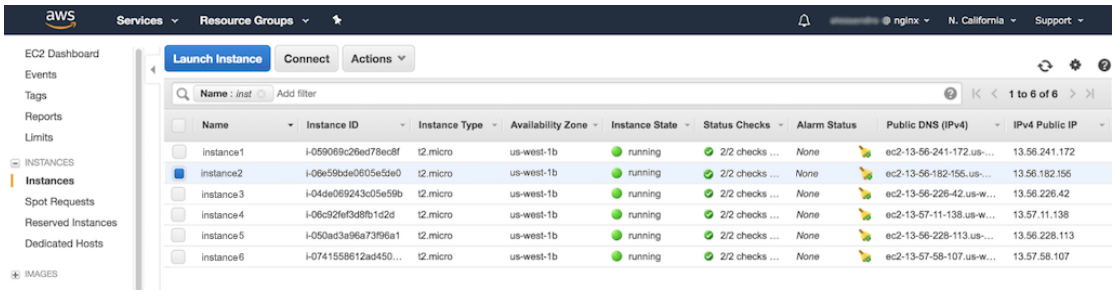
14. 

15. To install NGINX software on the instance, connect to it, and follow the instructions in the NGINX Plus Admin Guide for [NGINX Open Source](/nginx/admin-guide/installing-nginx/installing-nginx-open-source/#prebuilt and NGINX Plus.

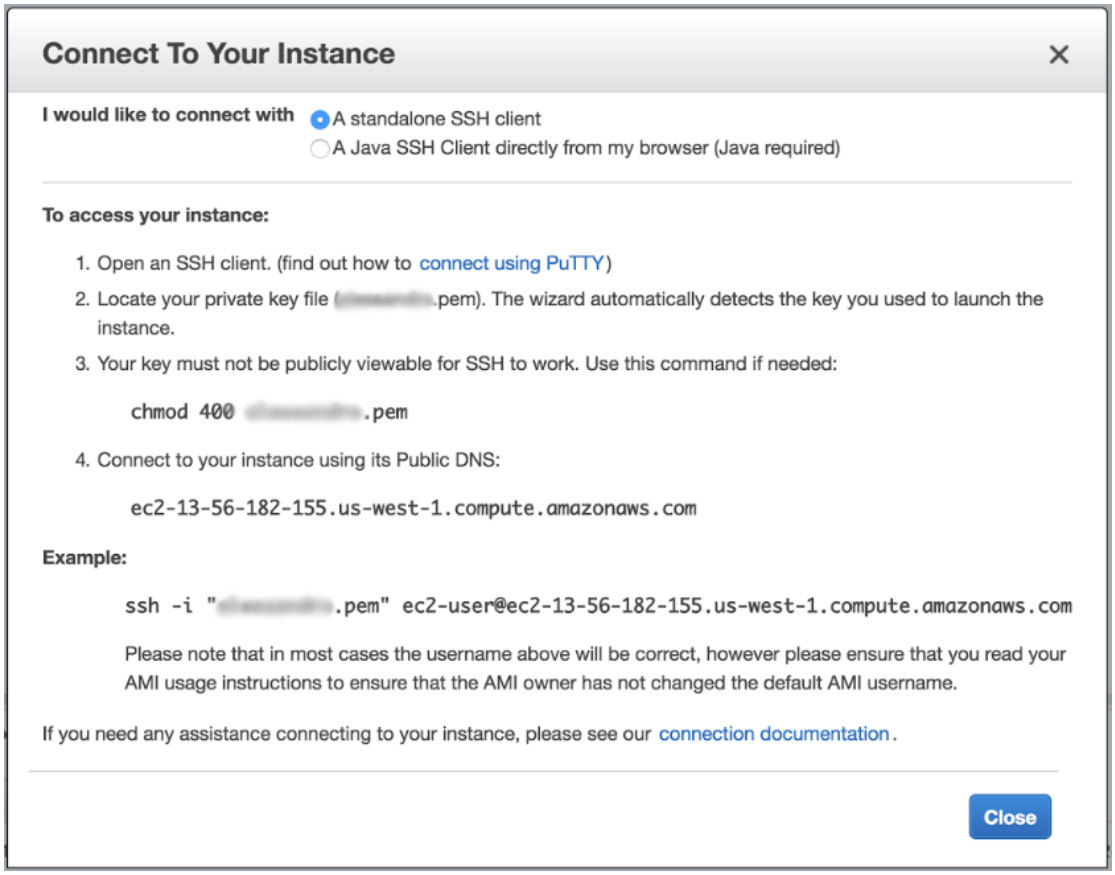
Connecting to an EC2 Instance

To install and configure NGINX Open Source or NGINX Plus on an instance, you need to open a terminal window and connect to the instance over SSH.

1. Navigate to the **Instances** tab on the EC2 Dashboard if you are not there already.
2. Click the row for an instance to select it. In the screenshot **instance2** is selected.



3. Click the **Connect** button above the list of instances. The **Connect To Your Instance** window pops up.
4. Follow the instructions in the pop-up window, which are customized for the selected instance (here **instance2**) to provide the name of the key file in the steps and in the sample `ssh` command.



Installing NGINX Software
Once you have established a connection with an instance, you can install the NGINX software on it. Follow the instructions in the NGINX Plus Admin Guide for NGINX Open Source and NGINX Plus. The Admin Guide also provides instructions for many maintenance tasks.

Automating Installation with a Configuration Manager
You can automate the installation of NGINX Open Source and NGINX Plus. Instructions for Ansible are provided below. For Chef and Puppet, see these articles on the NGINX, Inc. blog:

- Installing NGINX and NGINX Plus with Chef
- Deploying NGINX Plus for High Availability with Chef
- Installing NGINX and NGINX Plus with Puppet

Automating Installation with Ansible
NGINX, Inc. publishes a unified Ansible role for NGINX Open Source and NGINX Plus on Ansible Galaxy and GitHub. Perform these steps to install and run it.

1. Connect to the EC2 instance.

2. Install Ansible. These commands are appropriate for Debian and Ubuntu systems: