

# Experiment No: 07

Name: Suraj P. Patil

Roll: 4120

Class: B.Tech

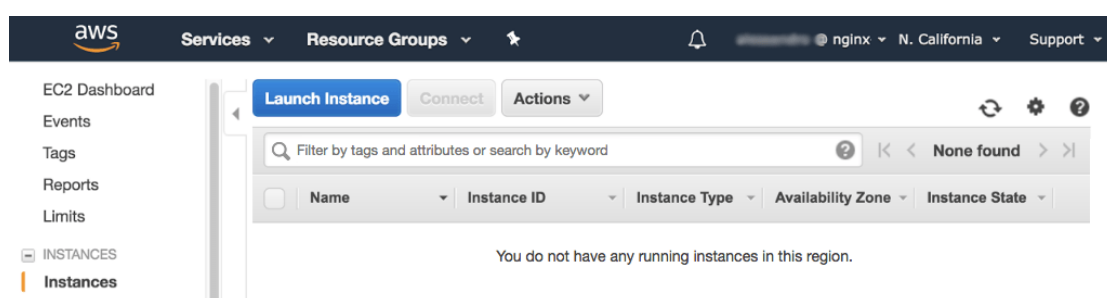
URN: 20131086

Div: B

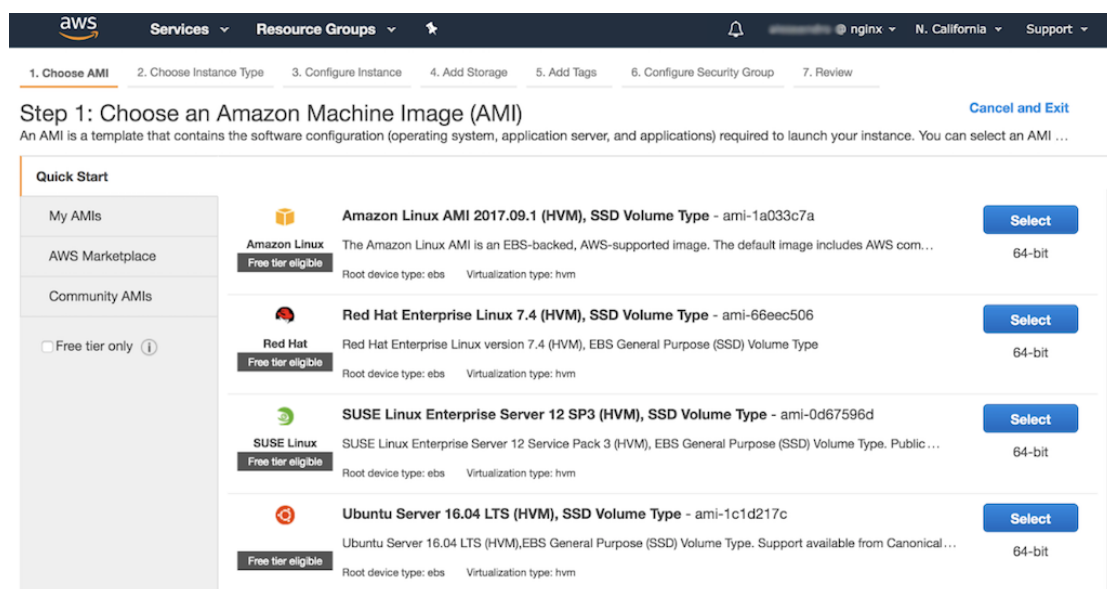
Date: 07-10-2023

## 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.

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and network... resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

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.

**Step 3: Configure Instance Details**  
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the ...

**Number of instances** ⓘ 1 [Launch into Auto Scaling Group](#) ⓘ

**Purchasing option** ⓘ ☐ Request Spot instances

**Network** ⓘ vpc-d88cb5bd | default-vpc (default) [Create new VPC](#)

**Subnet** ⓘ subnet-9c64a3f8 | default-subnet | Default in us-west-2 [Create new subnet](#)  
4087 IP Addresses available

**Auto-assign Public IP** ⓘ Use subnet setting (Enable)

**IAM role** ⓘ None [Create new IAM role](#)

**Shutdown behavior** ⓘ Stop

**Enable termination protection** ⓘ ☐ Protect against accidental termination

**Monitoring** ⓘ ☐ Enable CloudWatch detailed monitoring  
[Additional charges apply.](#)

**Tenancy** ⓘ Shared - Run a shared hardware instance  
[Additional charges will apply for dedicated tenancy.](#)

**T2 Unlimited** ⓘ ☐ Enable  
[Additional charges may apply](#)

▶ **Advanced Details**

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

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

**Step 4: Add Storage**  
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	snap-0b2b8096f1b89e969	8	General Purpose SSD (GP2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

[Add New Volume](#)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

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.

The screenshot shows the AWS Management Console interface for the 'Add Tags' step of an EC2 instance creation. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information. A progress bar at the top indicates seven steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags (current step), 6. Configure Security Group, and 7. Review. The main content area is titled 'Step 5: Add Tags' and includes explanatory text about tags. Below this is a table for adding tags with columns for 'Key', 'Value', 'Instances', and 'Volumes'. A single tag is added with the key 'Name' and value 'instance-name'. At the bottom, there are buttons for 'Cancel', 'Previous', 'Review and Launch', and 'Next: Configure Security Group'.

Key	Value	Instances	Volumes
Name	instance-name	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Buttons: Cancel, Previous, Review and Launch, Next: Configure Security Group

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.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your ... the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

**Warning**

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access ...

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.

aws Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

**⚠ Improve your instances' security. Your security group, *security-group-name*, is open to the world.**

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

▼ AMI Details [Edit AMI](#)

**Amazon Linux AMI 2017.09.1 (HVM), SSD Volume Type - ami-1a033c7a**

**Free tier eligible** The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The ...

Root Device Type: ebs Virtualization type: hvm

▼ Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only		Low to Moderate

▼ Security Groups [Edit security groups](#)

Security group name	<i>security-group-name</i>			
Description	<i>security group-description</i>			
Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ
SSH	TCP	22	0.0.0.0/0	
HTTP	TCP	80	0.0.0.0/0	

●  
●  
●

[Cancel](#) [Previous](#) [Launch](#)

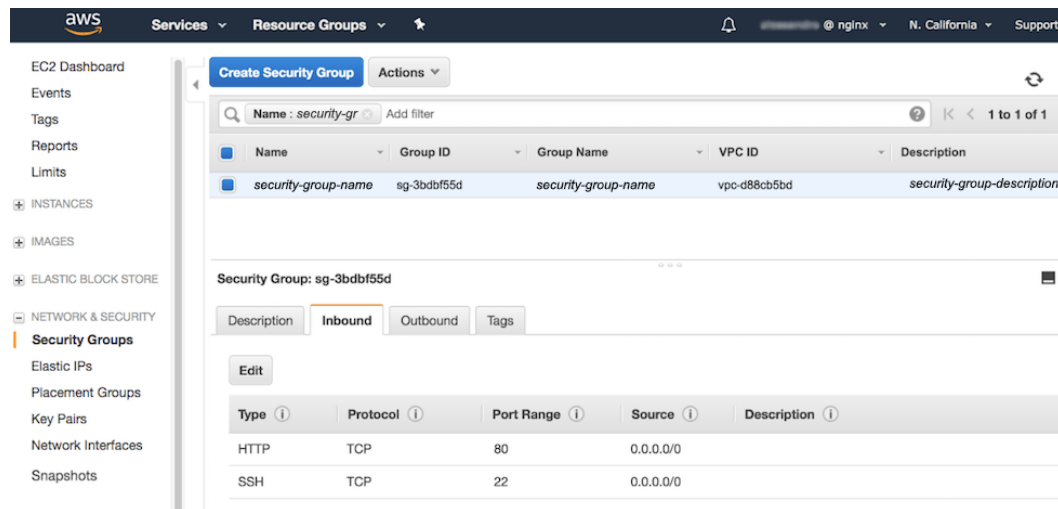
12. When you click the **Launch** button, a window pops up asking you to select an existing key pair or create a new key pair. Take the appropriate action for your use case, then click the **Launch Instances** button.

**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.





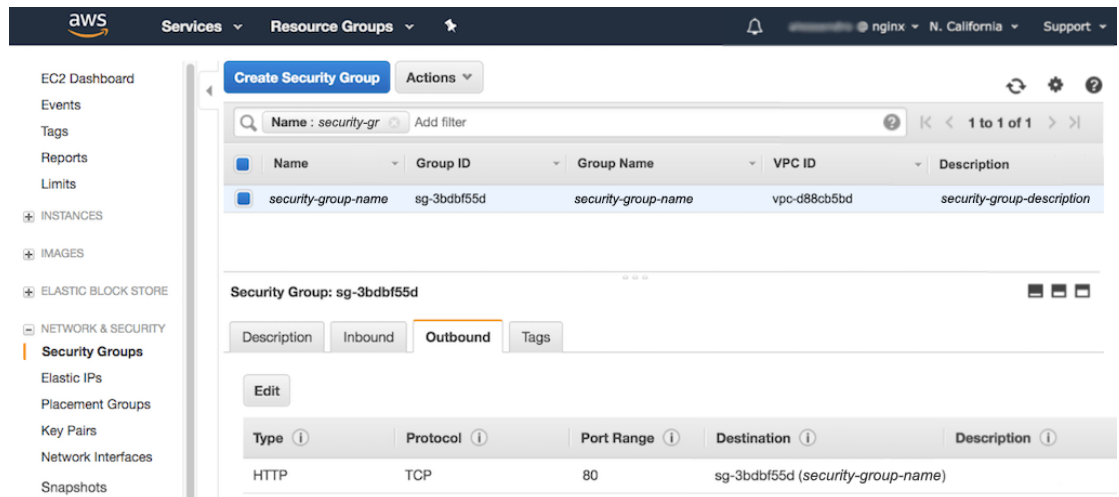
- Select the security group by clicking its radio button in the leftmost column of the table. A panel opens in the lower part of the window displaying details about the group.
- Open the **Inbound** tab and verify that the rules you created in Steps 9 and 10 are listed.



- 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**).



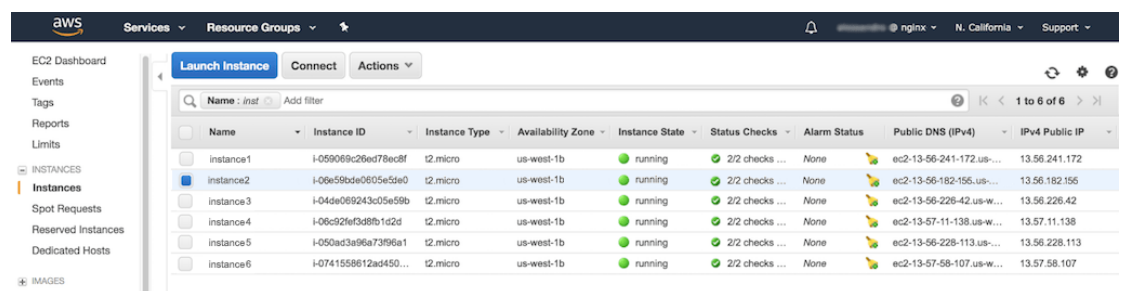


- 14.
15. To install NGINX software on the instance, connect to it, and follow the instructions in the NGINX Plus Admin Guide for [NGINX Open Source](/nginx/admin-guide/installing-nginx/installing-nginx-open-source/#prebuilt and NGINX Plus.

### 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.

### Connect To Your Instance

I would like to connect with

☒ A standalone SSH client  
☐ A Java SSH Client directly from my browser (Java required)

To access your instance:

1. Open an SSH client. (find out how to [connect using PuTTY](#))

2. Locate your private key file ( `XXXXXXXXXX.pem`). The wizard automatically detects the key you used to launch the instance.

3. Your key must not be publicly viewable for SSH to work. Use this command if needed:  

```
chmod 400 XXXXXXXX.pem
```

4. Connect to your instance using its Public DNS:  

```
ec2-13-56-182-155.us-west-1.compute.amazonaws.com
```

Example:

```
ssh -i "XXXXXXXXXX.pem" ec2-user@ec2-13-56-182-155.us-west-1.compute.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our [connection documentation](#).

Close

## 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: