**How To Launch an EC2 Instance From AMI in AWS**

Let’s launch an Linux EC2 Instance From AMI. In this example, I am launching Linux Centos 7 from Amazone Machine Image. You can choose whichever OS you want for the use case we’ve been working on [here](https://www.mydatahack.com/how-to-create-your-personal-data-science-computing-environment-in-aws).

This is a step in [**How To Create Your Personal Data Science Computing Environment In AWS**](https://www.mydatahack.com/how-to-create-your-personal-data-science-computing-environment-in-aws).

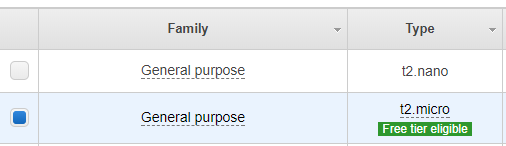
**Steps**

1. **Go to EC2 Dashboard and click Launch Instance.**
2. **Go to Comminity AMIs and Search for Amazone Linux.**

You will see many options. I went with amzn-ami-hvm-2017.09.1.20180115-x86\_64-gp2. It is a 64 bit and EBS as a root device. You can choose whichever you like.

1. **Choose Instance Type.**

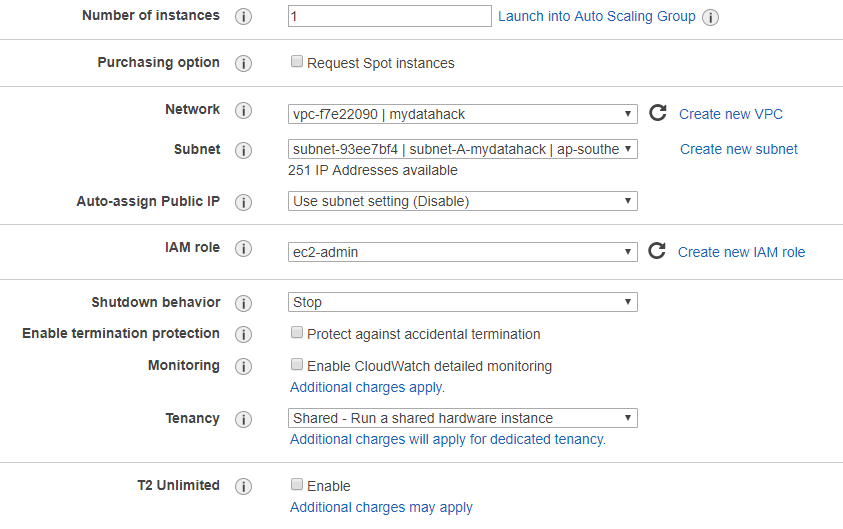
You can choose the appropriate instance type. I am choosing t2.micro as it is eligible for free tier. For our use case (see [here](https://www.mydatahack.com/how-to-create-your-personal-data-science-computing-environment-in-aws)), t2.micro is not good enough as this box is for training machine learning algorithm. For the purpose of working out infrastructure set up, let’s start with t2.micro. For further information about EC2 instance type, check [here](https://aws.amazon.com/ec2/instance-types/).



1. **Configure Instance Details**

You need to choose the correct VPC and Subnet. You also need to create the IAM role for EC2 instance with the AdministratorAccess policy. For further explanation about IAM Role, you can check out [here](https://www.mydatahack.com/how-to-create-admin-user-in-aws). This can be done from Create new IAM role option.

Disable auto-assign public IP as we are using Elastic IP for this instance.



1. **Add Storage**

You can choose whatever storage option you want. I am going with default of 8GB SSD.

1. **Add Tags**

This is optional. For our use case, it is not necessary. In an enterprise environment, tagging is absolutely necessary to manage resources.

1. **Attach the existing security group**

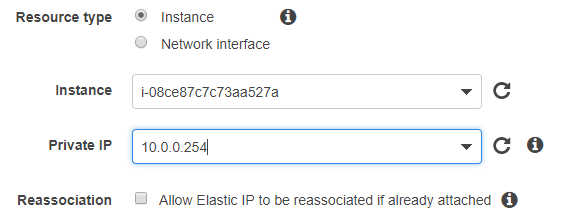
We have created a security group here (). Let’s attach it.

1. **Create a key pair**

For SSH access to the instance, you need to create public and private key. Make sure to keep the private key file securely.

1. **Create Elastic IP address and associate it with EC2 Instance**

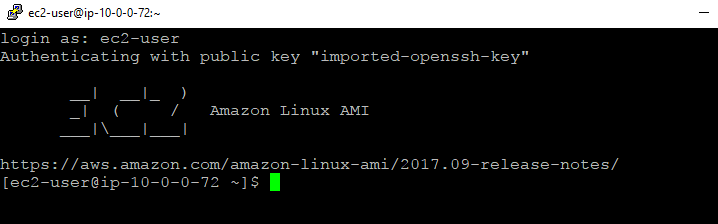
First of all, you need to create an [Elastic IP address](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-e) from VPC dashboard. This step should be straight forward. Then, associate it with the EC2 you just launched.



1. **Try SSH access from desktop**

Use the private key you downloaded and connect to the instance with SSH. For the host IP address, you can use the elastic IP you associated with the instance in the previous step. This should be straight forward. If you have any trouble, AWS has an extensive trouble shoot guide [here](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/TroubleshootingInstancesConnecting.html).

For Amazon Linux, the user name is ec2-user for the root account.



1. **Attach EBS Volume**

This is optional. Basically, you can create EBS and attach to the instance. For the details, go to this: **[How To Attach EBS Volume to EC2 Linux Instance In AWS](https://www.mydatahack.com/how-to-attach-ebs-volume-to-ec2-linux-instance-in-aws)**.

Great! We now have a Linux server running.

Let’s go back to <strong><a href="https://www.mydatahack.com/how-to-create-your-personal-data-science-computing-environment-in-aws" target="\_blank">How To Create Your Personal Data Science Computing Environment In AWS</a></strong> to complete the rest of the steps!