**How To Launch EC2 With Bootstrap in AWS**

When you launch an AWS resource (EC2 or RDS), you will have an instance with a default configuration. Then, you have to install software and do custom configuration to bring it to a certain state. Bootstrapping enables you to script software installation and configuration and execute it while launching the instance. Being able to bootstrap EC2 is the first step to become the practitioner of the Infrastructure-as-Code philosophy in AWS.

This entry is a bonus step from [**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). In the post, we have a manual step to launch an EC2 instance and install and configure Anaconda environment. In AWS, you can do everything with a piece of code!

In the post above, we launched an EC2 instance and installed Anaconda3. Then we installed the psycopg2 package for the Postgres DB connection. Let’s automate this process.

There are two ways of bootstrapping. The first way is to add the script to the user data field in advanced details in Step3: Configure Instance Details when you launch it manually from AWS Management Console. The second way is to use AWS CLI to code everything up without going into the console.

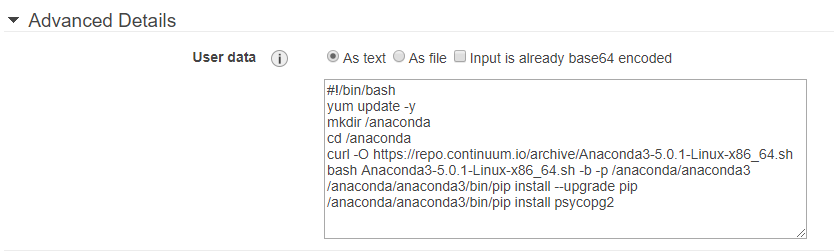
Let’s get it started! Any additional information, you can refer to the AWS documentation [here](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html).

1. **Bootstrapping EC2 with User data**

You can simply add the script when you launch the instance from the management console for advanced configuration details in step 3 as below.

For further information on launching EC2 instance, you can refer [How To Launch an EC2 Instance From AMI in AWS](https://www.mydatahack.com/how-to-launch-an-ec2-instance-from-ami-in-aws).

**Make sure to enable Auto-assign Public IP**. If you disable it, you cannot download anything and yum update won’t work.



Here is the script. AWS recommends to run yum update. I am using curl here. You can also use wget. They are basically the same. The script will be run as the root user. Therefore, you should omit sudo. The scrip below is for Amazon Linux. Other linux may use different code. I tested on Centos7 and it runs fine. Not sure with Ubuntu or other ones.

You can use -b for silent installation of Anaconda (meaning you do not need to type yes for license agreement) and -p for specifying the installation directory. You can see the conda document [here](http://stuarteberg.github.io/conda-docs/help/silent.html) for further information.

1. **Bootstrapping EC2 with AWS CLI**

If you are an AWS engineer, you cannot live without AWS Command Line Interface (CLI). It is a command line tool which enables you to do many things you would do in the management console with terminal commands. For installation and basic instruction, see [here](https://aws.amazon.com/cli/).

Once you install it, you first need to configure it. It is a relatively easy step. The reference is [here](https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-getting-started.html).

Let’s code up the launch and configuration of EC2 with AWS CLI.

In this section, we will use 3 commands for EC2, [run-instances](https://docs.aws.amazon.com/cli/latest/reference/ec2/run-instances.html) (for launching an EC2), [associate-address](https://docs.aws.amazon.com/cli/latest/reference/ec2/associate-address.html) (for associating an Elastic IP to the launched instance) and [terminate-instance](https://docs.aws.amazon.com/cli/latest/reference/ec2/terminate-instances.html) (for terminating the instance).

**run-instances**

You need to save the launch script as a text file. In the example below, you need to run the launch command from where you save the text file (e.g. cd /scripts/). The file:// prefix is necessary.

Another key point is to make sure you have the associate-public-ip-address option. Without it, nothing gets installed. Rest are perhaps self-explanatory. Just need to get the right security group, subnet, instance type, keys and so on.

Once the instance is up and running, you can ssh to the instance to see if the anaconda is installed as specified. So cool, isn’t it?

[cc lang="text" tab\_size="4" lines="-1" theme="mac-classic" line\_numbers="false"]

aws ec2 run-instances --image-id ami-942dd1f6 --count 1 \

--instance-type t2.micro --key-name mydatahack-ec2 \

--iam-instance-profile Name=ec2-admin --subnet-id subnet-93ee7bf4 \

--security-group-ids sg-7be1fb1d --associate-public-ip-address \

--user-data file://ec2\_anaconda\_install.txt

[/cc]

**associate-address**

Once your instance is ready, you can associate an Elastic IP address. Note that you cannot associate it unless you have a running instance.

[cc lang="text" tab\_size="4" lines="-1" theme="mac-classic" line\_numbers="false"]

aws ec2 associate-address --instance-id <your ec2> --allocation-id <your Elastic IP>

[/cc]

**terminate-instances**

Yeah, you can terminate the instance with a line of command!

[cc lang="text" tab\_size="4" lines="-1" theme="mac-classic" line\_numbers="false"]

aws ec2 terminate-instances --instance-ids <your ec2>

[/cc]

OK, you can launch a bootstrapped EC2 instance with a line of code now. The same thing can be done with RDS. Have a look at the post here: **How To Launch Postgres RDS With AWS Command Line Interface (CLI)**.

Good Times!