# Jupyter Notebook In The Cloud (Amazon EC2)

## Motivation

If your computation needs overgrow a memory and disk size of your laptop, you need to look around for another solution: using [Google Colab](https://colab.research.google.com/), running your package as [a docker image](http://simecek.xyz/shredded_code/2017/02/17/data-science-amazon-vm-with-start/stop-functionality/) elsewhere etc.

Here, we will focus on one possibility - running Jupyter Notebook on a virtual machine in the cloud. Specifically, we will speak about Amazon EC2. However, with slight modifications, it applies to Google Cloud or Microsoft Azure as well.

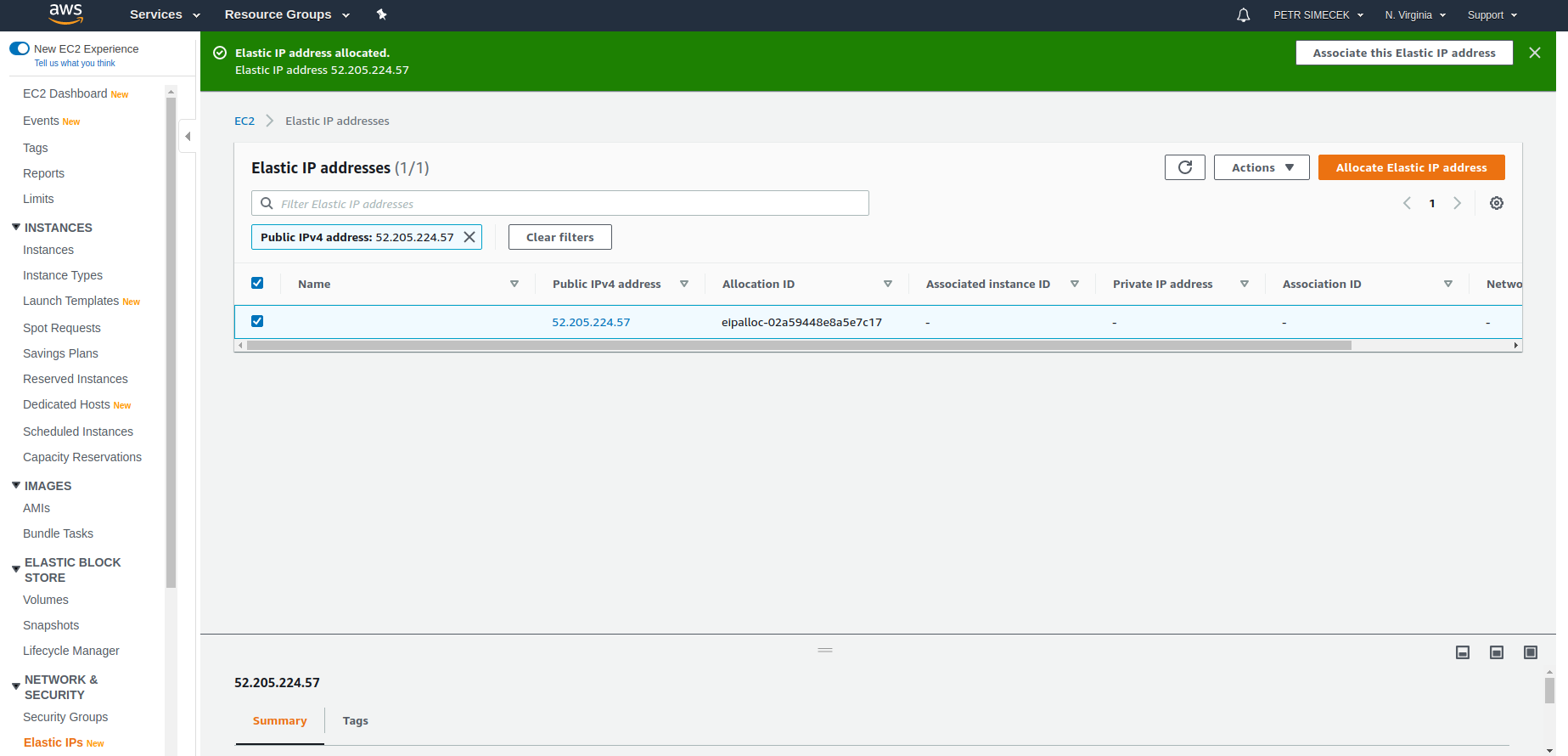
In the cloud, you can get any memory or number of CPUs you want. But because you pay for them, you want to turn your virtual machine down as much as possible. This guide should make this switching on/off process easy for you.

I expect you to know AWS EC2 and Jupyter Notebook. Ideally, you already have the virtual machine running and you are looking for tricks to simplify the on/off process.

## Trick #1: Get Elastic IP address

Typically, the machine gets a new IP address after you restart it. This might be impractical, to avoid that you need to get a new Elastic IP and link it to your machine.

**Create Elastic IP:** log into Amazon EC2, choose NETWORK & SECURITY / Elastic IP from the left menu (you should get to the page below), click on “Allocate Elastic IP address” button, keep default parameters -> click on allocate. New elastic IP appears (52.205.224.57 on the page below), not yet assigned to any instance.



**Link Elastic IP to your instance:** Now, select your newly created Elastic IP and click on Actions button, select “Associate Address” and choose your instance. It should start working immediately (you might want to reboot the instance, just to be extra sure).

## Trick #2: Set a password for your Jupyter Notebook

Unless you set the password, you will need to copy a new token every time Jupyter NB restarts.

Log in to your virtual machine and run jupyter notebook password (you might need to activate your environment first). Restart the Jupyter and test that it asks for the password.

## Trick #3: Start and stop your virtual machine from the command line

The syntax is the following:

aws ec2 start-instances --instance-ids "YOUR\_INSTANCE\_ID"

aws ec2 stop-instances --instance-ids "YOUR\_INSTANCE\_ID"

I highly recommend adding these as aliases to your .bashrc file. For example, those are mine:

alias aws-start='aws ec2 start-instances --instance-ids "i-0b95ee7b6dd3d4398"'

alias aws-stop='aws ec2 stop-instances --instance-ids "i-0b95ee7b6dd3d4398"'

alias aws-info='aws ec2 describe-instance-status --instance-ids "i-0b95ee7b6dd3d4398"'

alias aws-ssh='ssh -i ~/.ssh/petrs.pem ubuntu@52.205.224.57'

## Trick #4: Running Jupyter Notebook as a service

I.e. Jupyter notebook starts automatically when the instance starts. This is the most difficult part and I highly recommend reading [this guide](https://naysan.ca/2019/09/07/jupyter-notebook-as-a-service-on-ubuntu-18-04-with-python-3/) before going further.

**Create a file ‘jupyter.service’**, this is example of mine (using pyenv environments)

[Unit]

Description=Jupyter Notebook

[Service]

Type=simple

PIDFile=/run/jupyter.pid

ExecStart=/bin/bash -c "export PATH=\"/home/ubuntu/.pyenv/bin:$PATH\";eval \"$(pyenv init -)\";eval \"$(pyenv virtualenv-init -)\";pyenv activate venv3.6.9;jupyter-notebook --notebook-dir=/home/ubuntu --ip=0.0.0.0"

User=ubuntu

Group=ubuntu

WorkingDirectory=/home/ubuntu

Restart=always

RestartSec=10

[Install]

WantedBy=multi-user.target

**Then run the following commands** in the shell (on your AWS virtual machine)

sudo cp jupyter.service /etc/systemd/system/

sudo systemctl enable jupyter.service

sudo systemctl daemon-reload

sudo systemctl start jupyter.service

**Time to pray.** If everything works, your Jupyter Notebook is running and you can access it at http://YOUR\_ELASTIC\_IP:8888

You can also check the service as follows:

systemctl status jupyter.service

## Trick #5: Shutting down the instance automatically

If you are worried that you forgot to shut down the instance, you can set up a cron job to shut it down for you.

**Open cron table.** You need sudo privileges to shut down the machine, so the right command is sudo crontab -e

**Add a cron job.** Add one line at the end of the file and save. I recommend switching off the machine daily (at 6pm CET) at so, my line is the following

0 17 \* \* \* /sbin/shutdown -h "now"

If you want to switch off the machine at a different time or only for some days of the week, see the [cron documentation](https://www.ostechnix.com/a-beginners-guide-to-cron-jobs/). If you DO NOW want the machine to stop on a given day, you can always access the table by sudo crontab -e and comment out the line (until you finish your job).