Setting up MiPinG

This document describes how to generally setup MiPinG and how to use it to get personality predictions.

Version history

|  |  |  |
| --- | --- | --- |
| Date | Author | Content |
| 2020-10-01 | Henning Usselmann | Initial creation |

Inhalt

[Prerequisites (installing Linux in AWS) 1](#_Toc53591030)

[Installing Miping onto Linux 9](#_Toc53591031)

[Configuration 12](#_Toc53591032)

[Start the webserver 14](#_Toc53591033)

[Access website 15](#_Toc53591034)

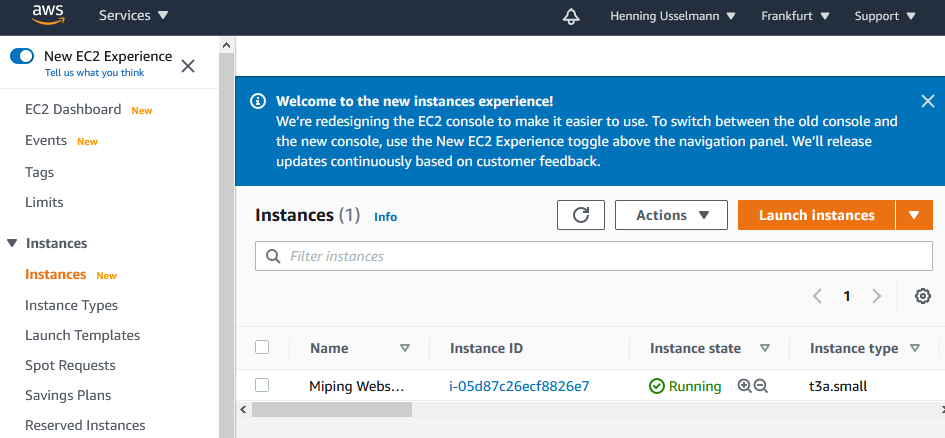
[Access the API directly 17](#_Toc53591035)

MiPinG needs 2 GB of memory, around 12 GB of disk space (including operating system) and needs Python to properly work. You definitely need a pair of **Twitter API Developer keys** to locally install MiPinG.

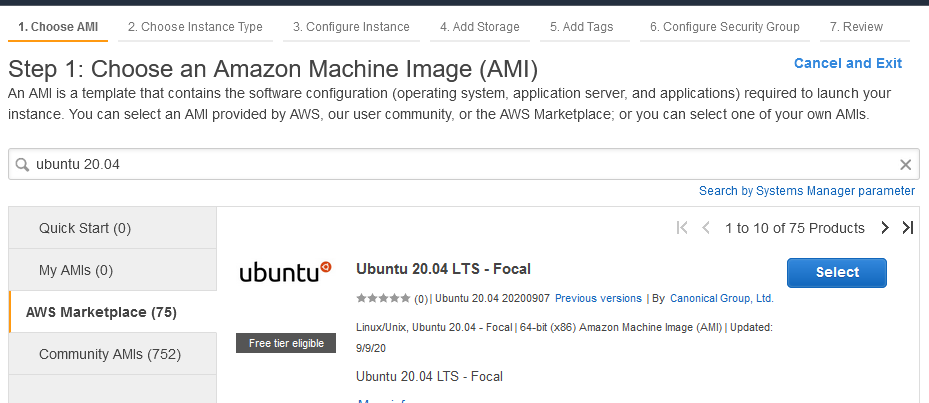
# Prerequisites (installing Linux in AWS)

You need a unix system to install and use MiPinG. We will show the process with an Amazon AWS server. You can set this server up from any computer you like.

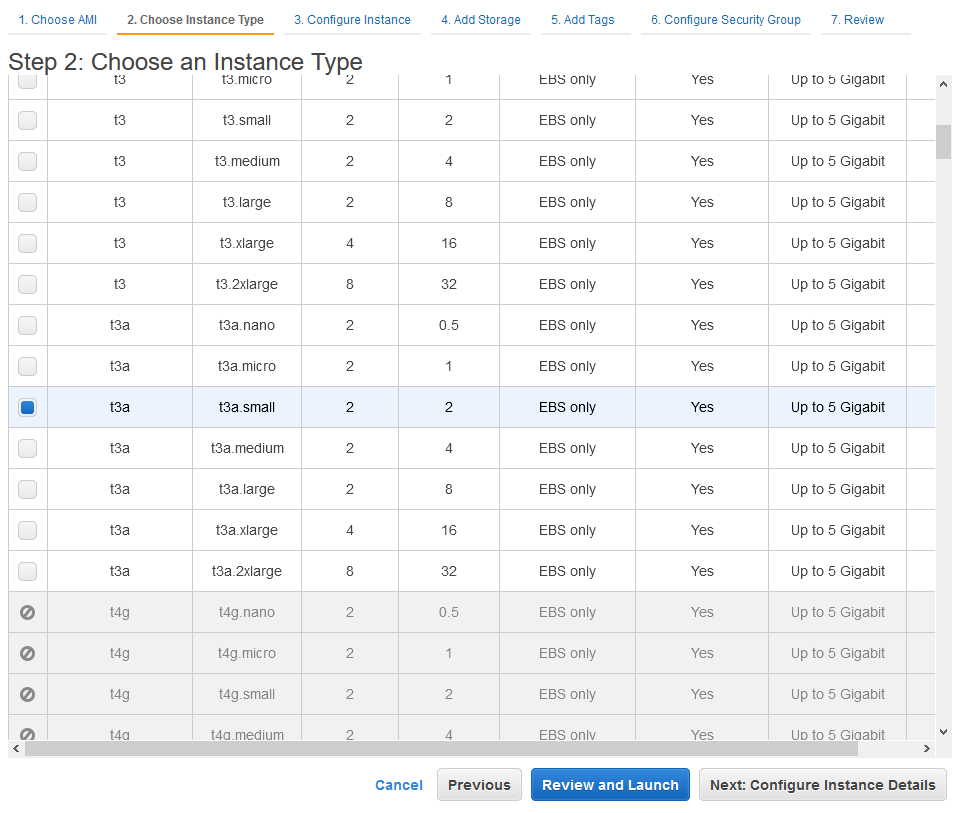
1. Create an account at Amazon AWS <https://aws.amazon.com/>
2. Go to EC2 dashboard 🡪 Instances 🡪 Launch Instance



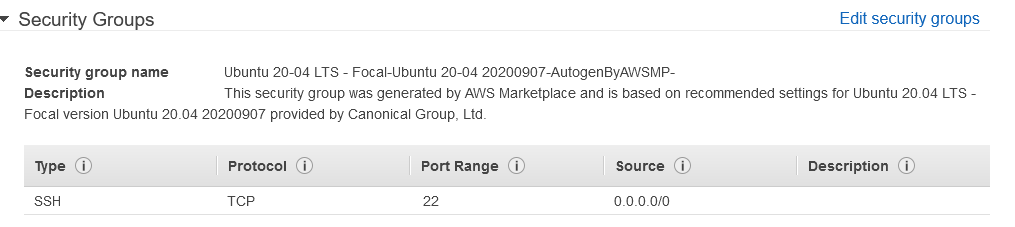
1. We will select Ubuntu’s 20.04 LTS – Focal version – search for “ubuntu 20.04” in the AWS marketplace. You might need to accept the terms of use.



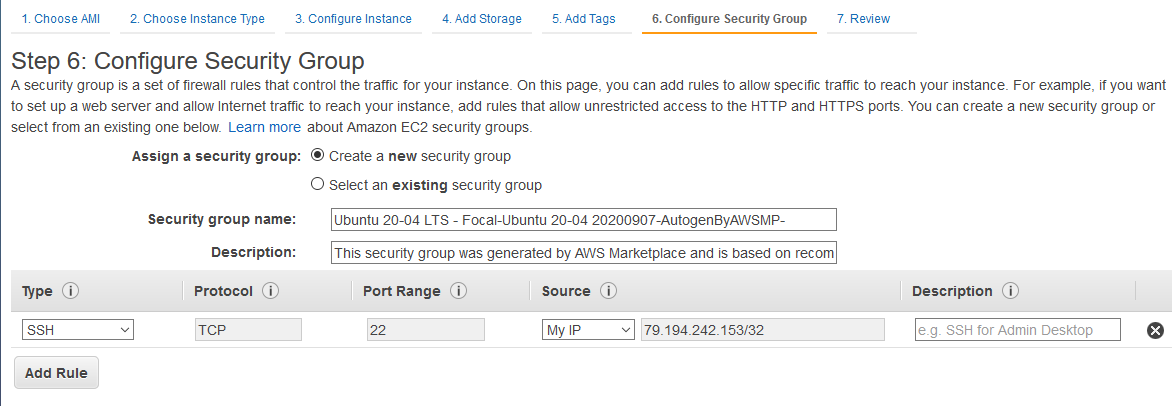
1. MiPinG needs at least 2 GB of memory, therefore we choose “t3a.small” as Instance type. Go directly to review and launch.



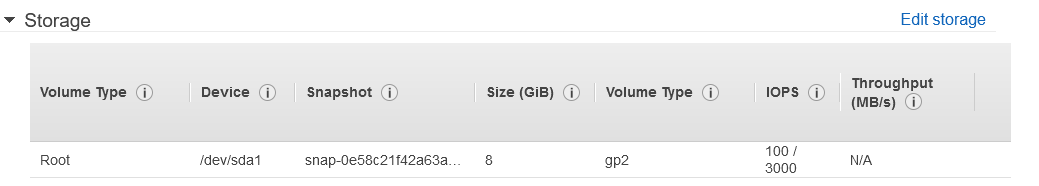
1. Edit security groups



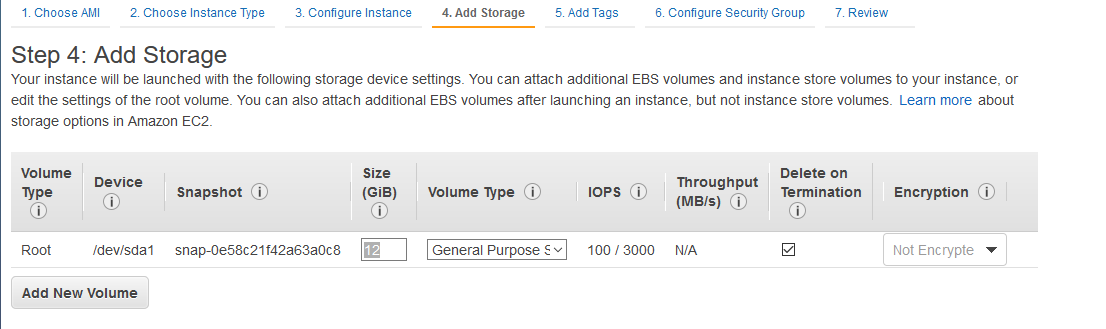
1. Create a new security group to allow and limit access to the machine (Firewall settings). It is recommended to choose “My IP” for SSH access. This has to be kept in might, in case you want to have access from different networks. Give the rule a meaningful description.



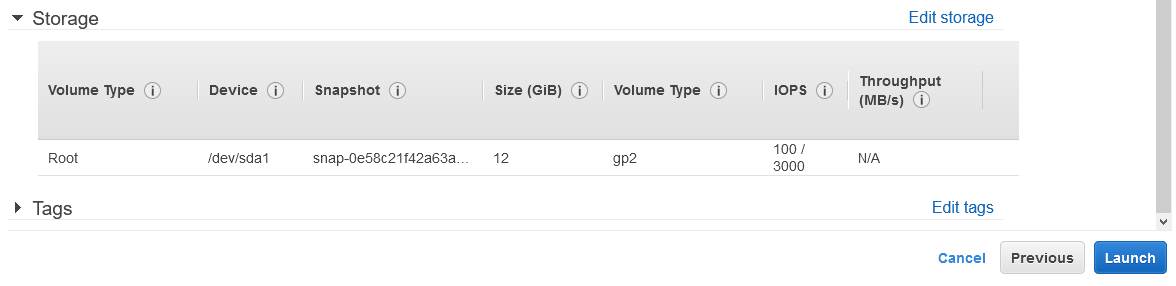
1. Go again to review and launch and edit storage



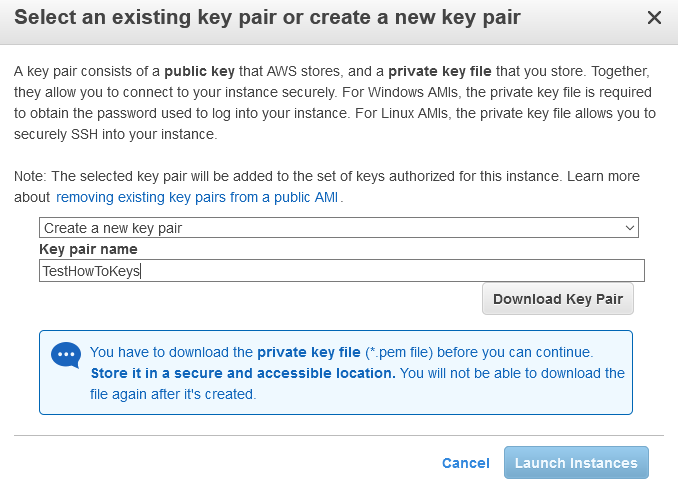
1. Increase the storage to at least 12 GB



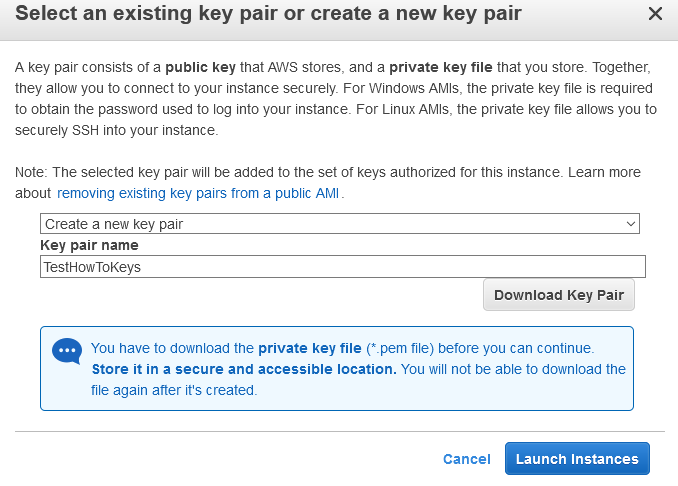
1. Press launch



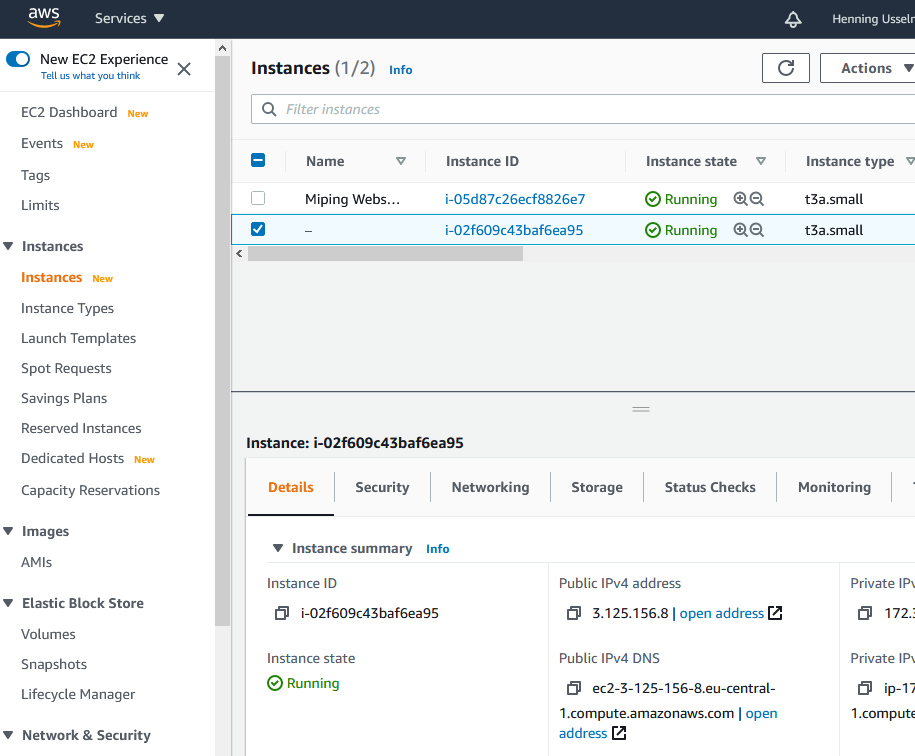
1. You have to provide or create access keys. We will create new ones and download the key pair. Save it on your local device, e.g. in C:\tmp\TestHowToKeys.pem



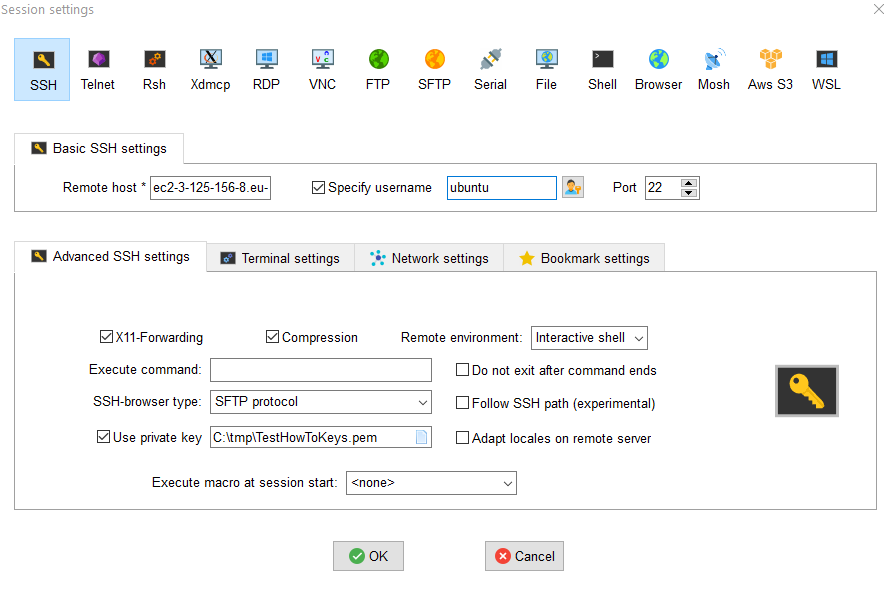
1. Once downloaded you are able to click “launch instances”



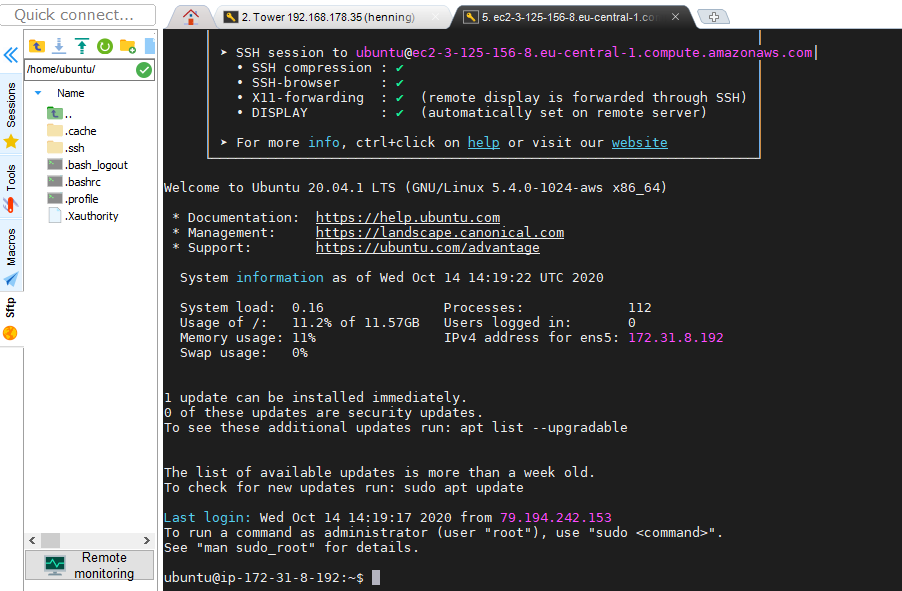
1. Go back to the EC2 dashboard and click instances. You will see your new instance and its public DNS address. Copy the address. In our case: ec2-3-125-156-8.eu-central-1.compute.amazonaws.com.



1. Download any SSH client you like. I recommend MobaXterm (<https://mobaxterm.mobatek.net/> ). Launch this client and create a new session. Choose SSH, enter the public DNS address into the remote host field, click advanced SSH settings, and click use private key. Enter the path to your previously downloaded key. Specify the username as “Ubuntu”. Click OK.



1. The login to the server should be successful and you should see the following output.

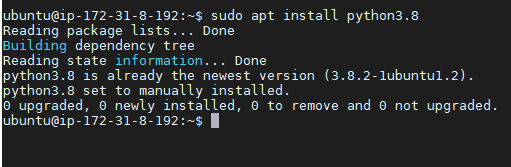


The following setup should be the same, regardless of whether you are using a server or local computer.

# Installing Miping onto Linux

1. Install Python via

sudo apt install python3.8s



1. Create a virtual environment (for enclosing our activities in a Python virtual environment)

**sudo apt-get update**

**sudo apt-get install python3.8-venv**

Confirm with “Y”.

**python3.8 -m venv tutorial-env**

1. Activate virtual environment

**source tutorial-env/bin/activate**

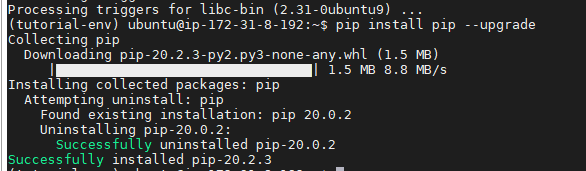
You see (tutorial-env) at the beginning of the line.



1. Install and upgrade pip

**sudo apt install python3-pip**

**pip install pip --upgrade**



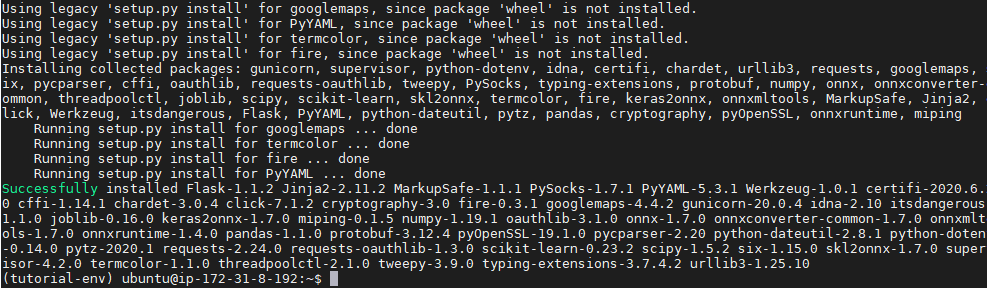
1. If you want to use the MiPinG website, you need to install the webserver nginx

**sudo apt install nginx**

1. Actually install MiPinG via the pip package manager

**pip install miping**

This will download all dependencies:



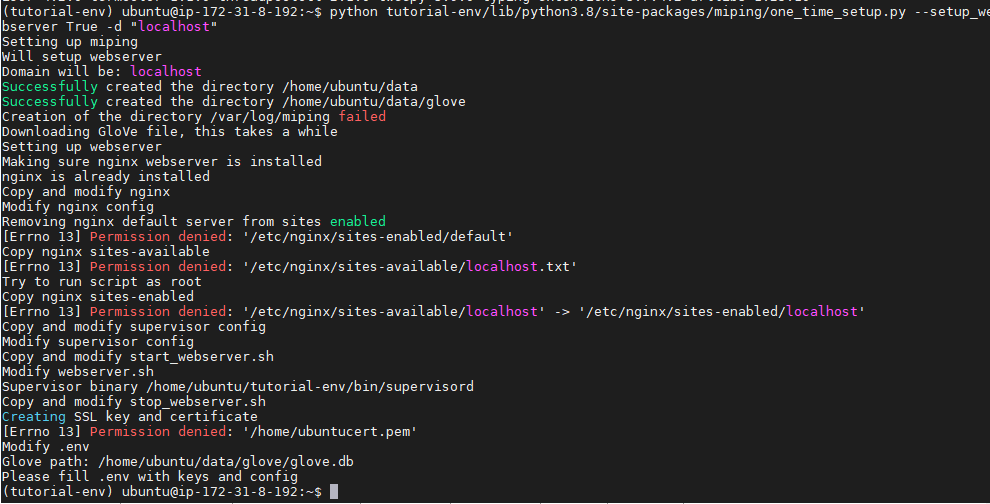
1. Run the ontime configuration script. We need to run it 3 times. It sets up our local webserver and does all configurations for us.

It is important to run it first as the local user, and then as root user and finally again as local user.

**python tutorial-env/lib/python3.8/site-packages/miping/one\_time\_setup.py --setup\_webserver True -d "localhost"**

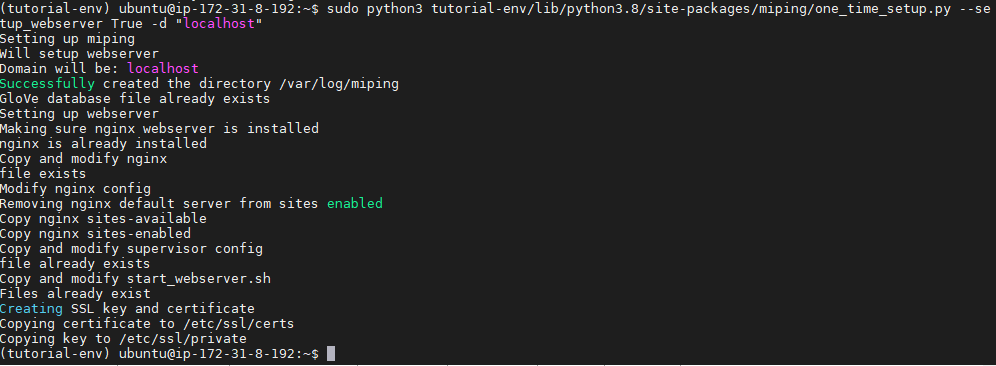
The first run will download and extract the GloVe database file from (<https://miping-glove.s3.eu-central-1.amazonaws.com/glove.zip> ) this will take a while, since this file is big. It is expected that some errors are shown, this is due to the different user permissions.

Output:



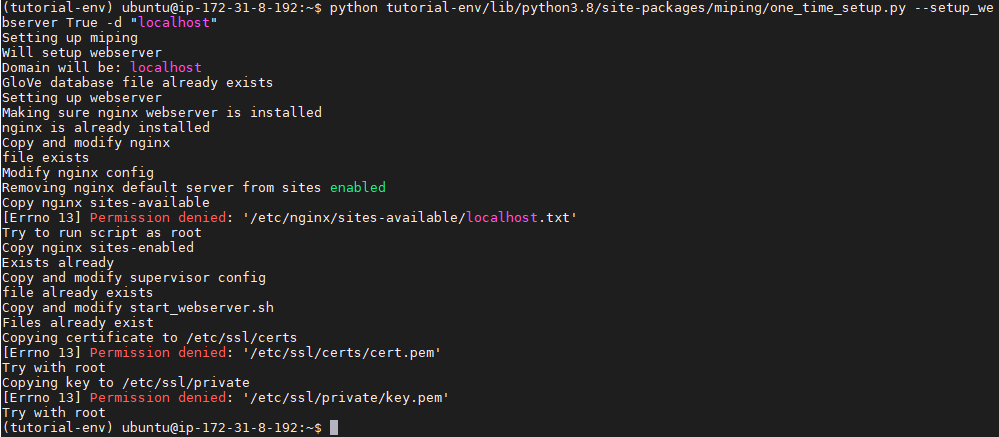
**sudo python3 tutorial-env/lib/python3.8/site-packages/miping/one\_time\_setup.py --setup\_webserver True -d "localhost"**

Output:



**python tutorial-env/lib/python3.8/site-packages/miping/one\_time\_setup.py --setup\_webserver True -d "localhost"**

Output: the errors are expected



# Configuration

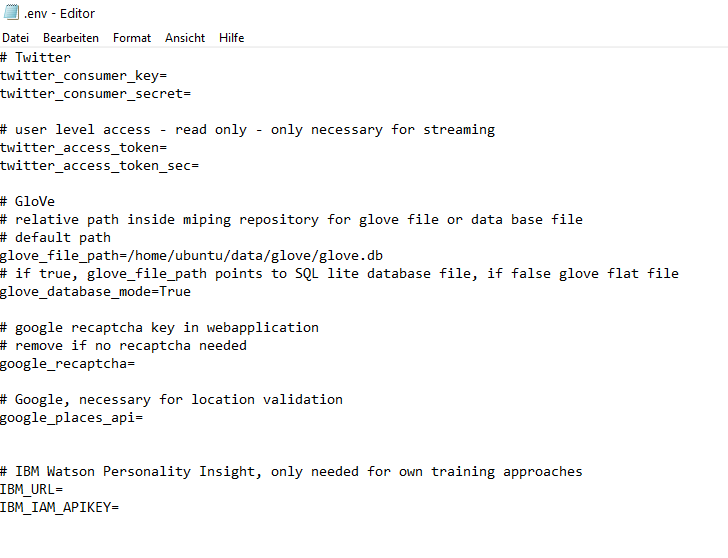
By performing

**Ls –a**

You can see the “.env” file.



Open it with your favourite editor:



You need to at least a twitter\_consumer\_key and twitter\_consumer\_secret. You have to apply for this at <https://developer.twitter.com/> .

This will look something like this:



You can also see, that the onetime script already entered the correct glove\_file\_path for the GloVe data base file. If you manually downloaded it, you would have to enter the correct path here.

To activate reCaptcha just provide your individual key here. You need to exchange the Javascript file at the following location. Just swap script.js with script-captcha.js. You also need to put your Public reCaptcha key in the respective form in index.html. Afterwards restart the webserver nginx.

~/tutorial-env/lib/python3.8/site-packages/miping/webapp/webfiles/www/js



The other parameters can be left empty, as they are only relevant for training.

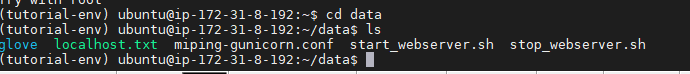
# Start the webserver

You only need to execute one script.

**Cd data**

**Ls**

The setup script downloaded all necessary files:



To start or stop the webserver just execute the respective script.

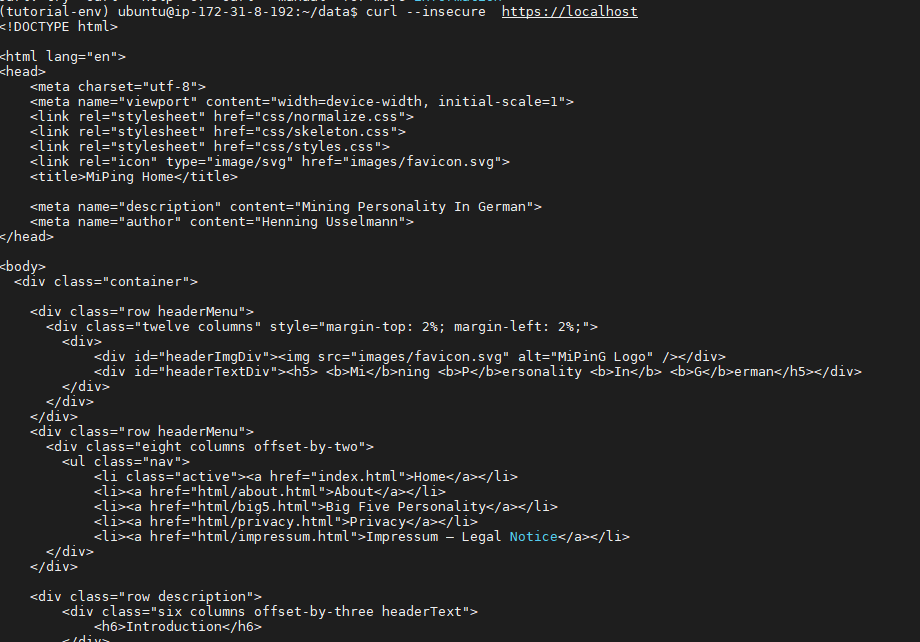
**sudo sh start\_webserver.sh**

**sudo sh stop\_webserver.sh**

By entering

**curl --insecure** [**https://localhost**](https://localhost)

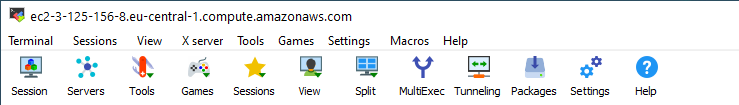
you should get the homepage of MiPinG back.



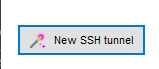
# Access website

Since we configured MiPinG to host the website locally, we need to setup a tunnel via MobaXTerm. On your local machine you could just enter “https://localhost” into your browser.

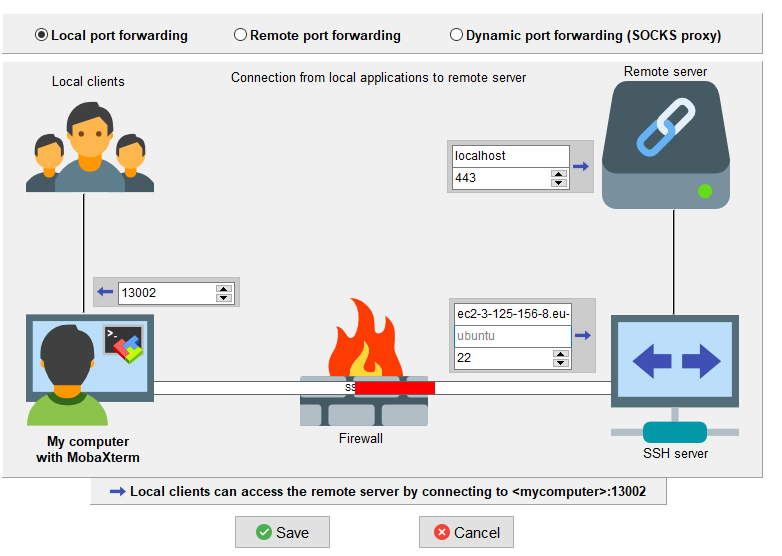
1. Click “Tunneling”



1. New SSH Tunnel



1. Local port forwarding – setup any port you like on the left side. On the right side choose “localhost” and port “443” (HTTPS). Below enter you public DNS from the AWS server, Ubuntu user name and port 22.

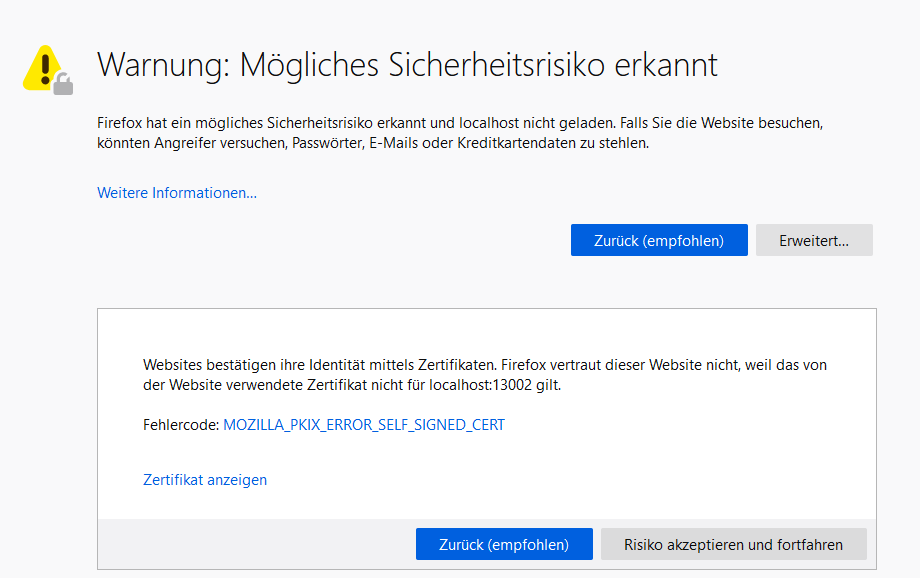


1. Select SSH key for access (same as for SSH session) and run it

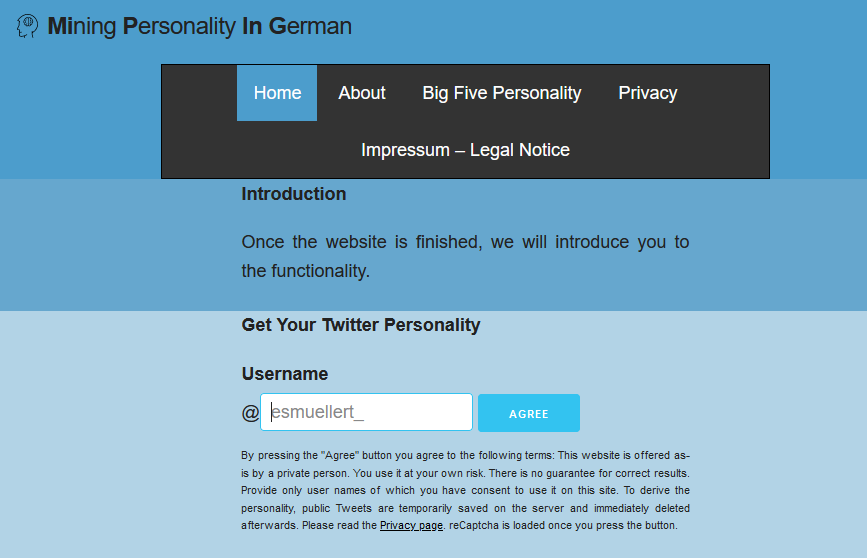


1. Using your favourite browser, you can now go to <https://localhost:13002/>

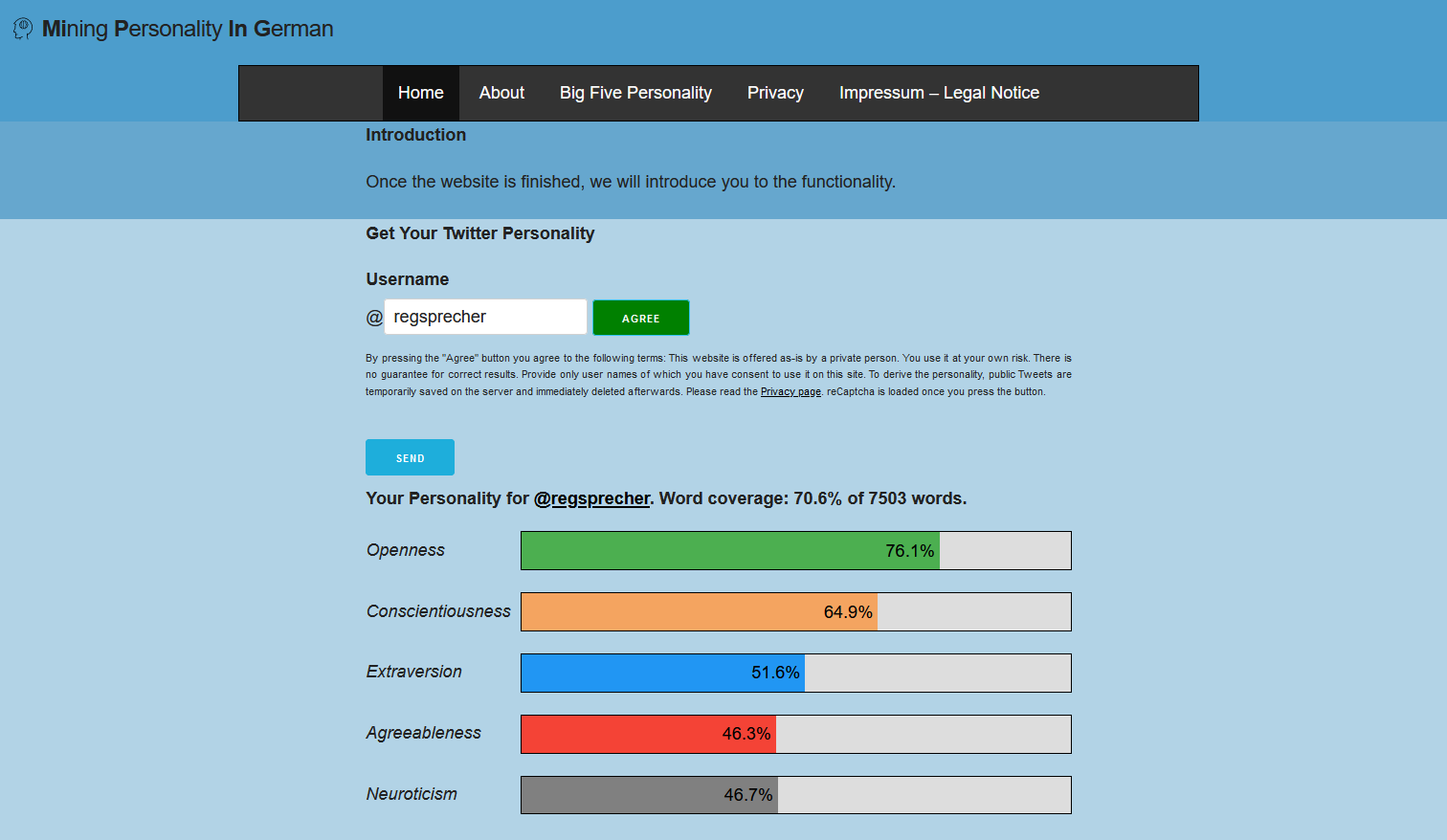
Since the SSL certificate is self-signed during the onetime configuration, a warning message is shown. Depending on your browser there are different ways of ignoring this issue. For Firefox you can accept the risk and continue.



1. MiPinG is ready to use.



Enter a public Twitter name and you will get the personality results:



# Access the API directly

If you are interested in an automatic approach of using MiPinG you can setup the whole server in the same way. Start the server the same way.

You can make a get request to the following API Endpoint:

**curl --insecure https://localhost/api/test**

If the result is positive, you can start querying the API (if you did not activate reCaptcha).

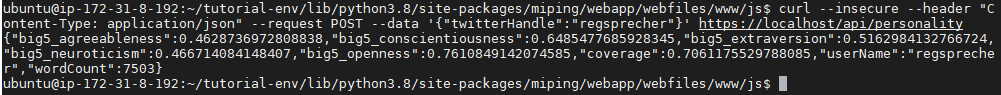
You need to provide the username as the following formatted JSON: {"twitterHandle":"regsprecher"}

A full working example would be the following:

**curl --insecure --header "Content-Type: application/json" --request POST --data '{"twitterHandle":"regsprecher"}'** [**https://localhost/api/personality**](https://localhost/api/personality)

Resulting in the following response:

{"big5\_agreeableness":0.4628736972808838,"big5\_conscientiousness":0.6485477685928345,"big5\_extraversion":0.5162984132766724,"big5\_neuroticism":0.466714084148407,"big5\_openness":0.7610849142074585,"coverage":0.7061175529788085,"userName":"regsprecher","wordCount":7503}



Pretty JSON:

{

"big5\_agreeableness": 0.4628736972808838,

"big5\_conscientiousness": 0.6485477685928345,

"big5\_extraversion": 0.5162984132766724,

"big5\_neuroticism": 0.466714084148407,

"big5\_openness": 0.7610849142074585,

"coverage": 0.7061175529788085,

"userName": "regsprecher",

"wordCount": 7503

}

Via this, you could automatically process more data.