Adsz.io Technologies Inc.

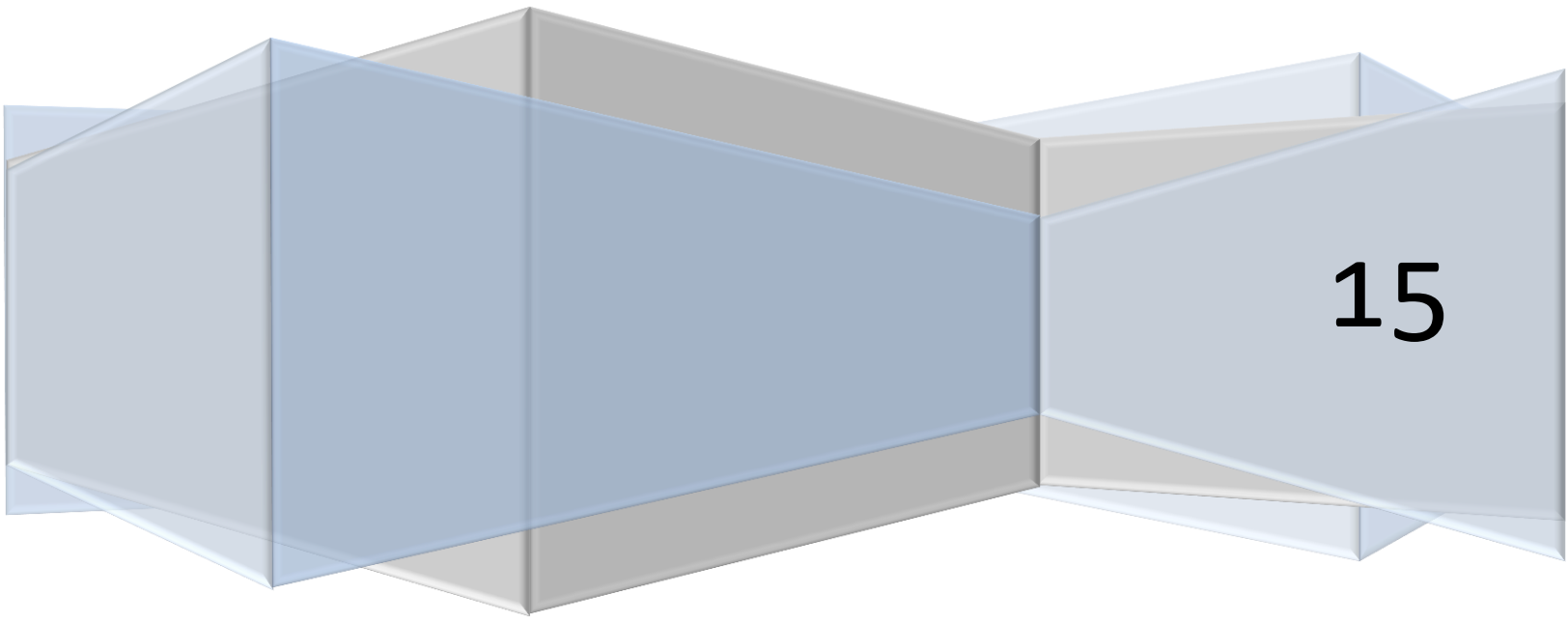
Setup Development VM in AWS

Part 1

Version 0.1

RTBKit & Graphite

Leon T.



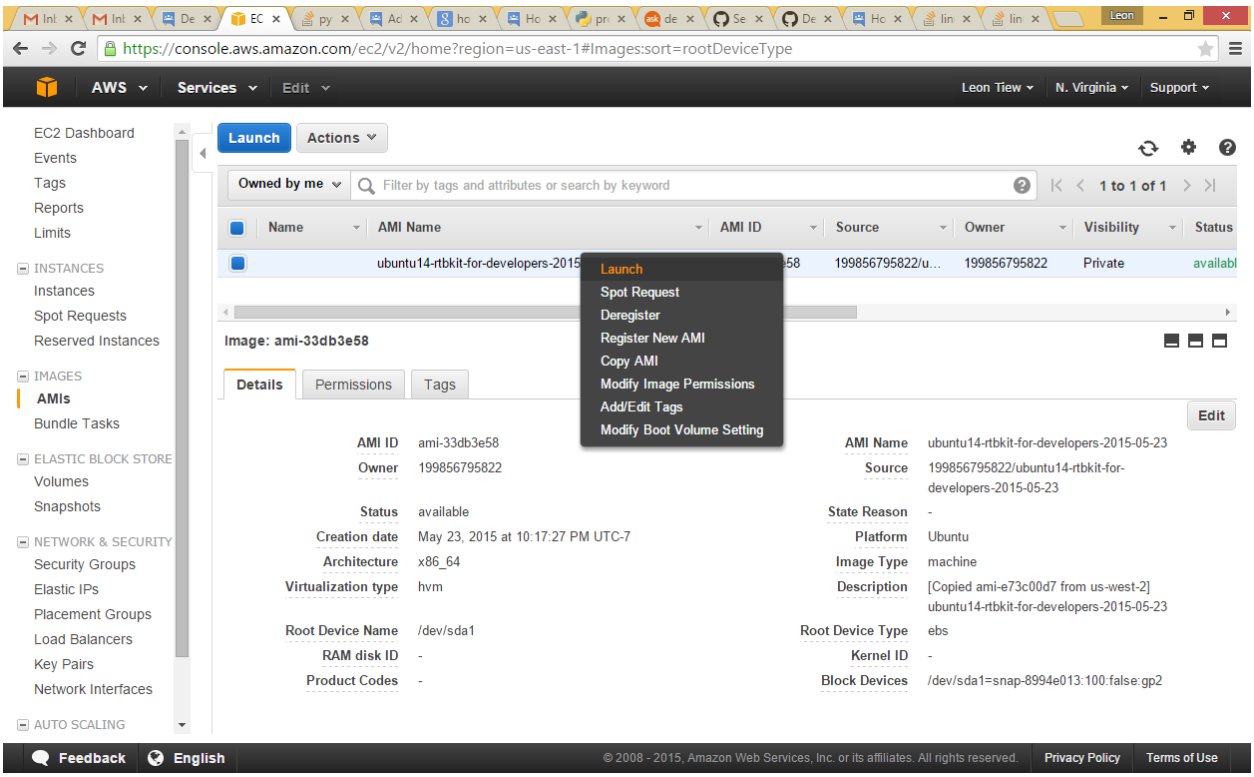
Contents

Introduction

The purpose of this document is to show you how to create an AWS EC2 using the existing RTBKit AIM I created and built from the source code. In addition, I also install RTBKit's Mock Exchange and the Graphite visualization tool.

Create a Development EC2 Instance based on an existing AIM ID ami-33db3e58 in N.Virginia (Close to Google AdX datacenter)

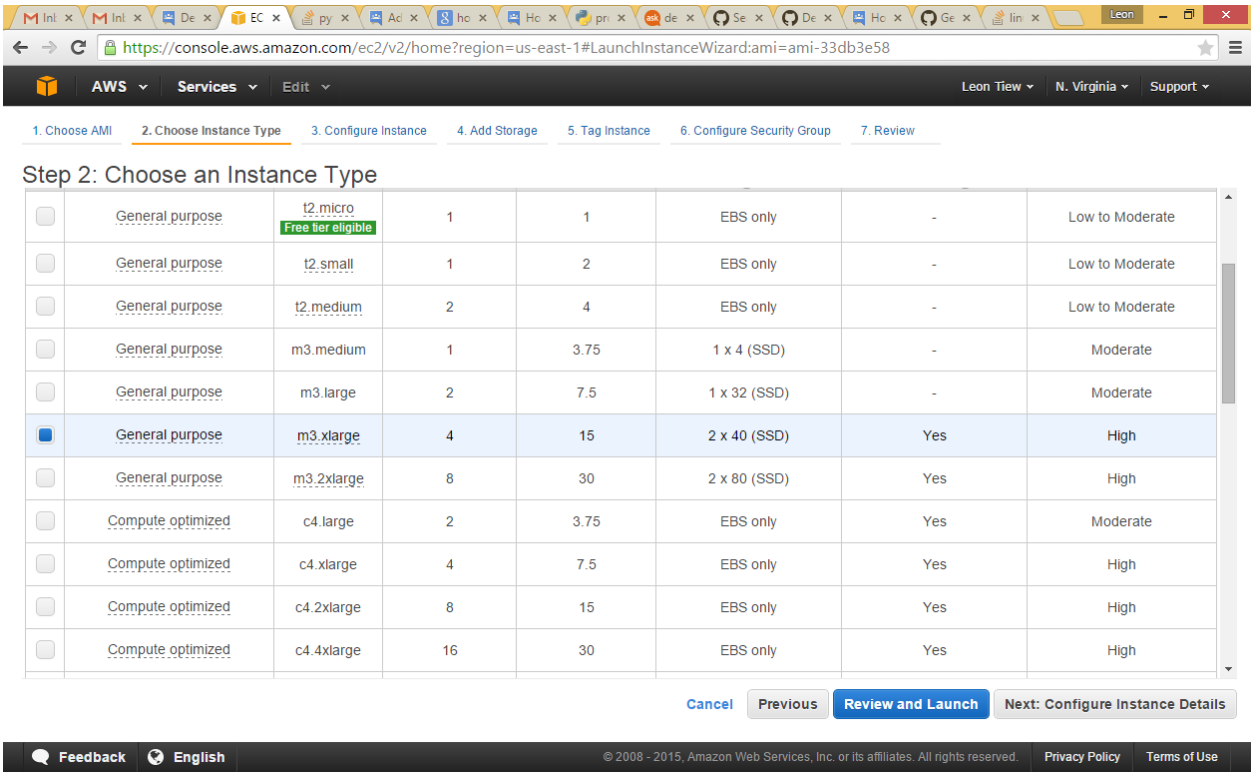
(Please provide me your AWS account and I will grant you a permission to access it)



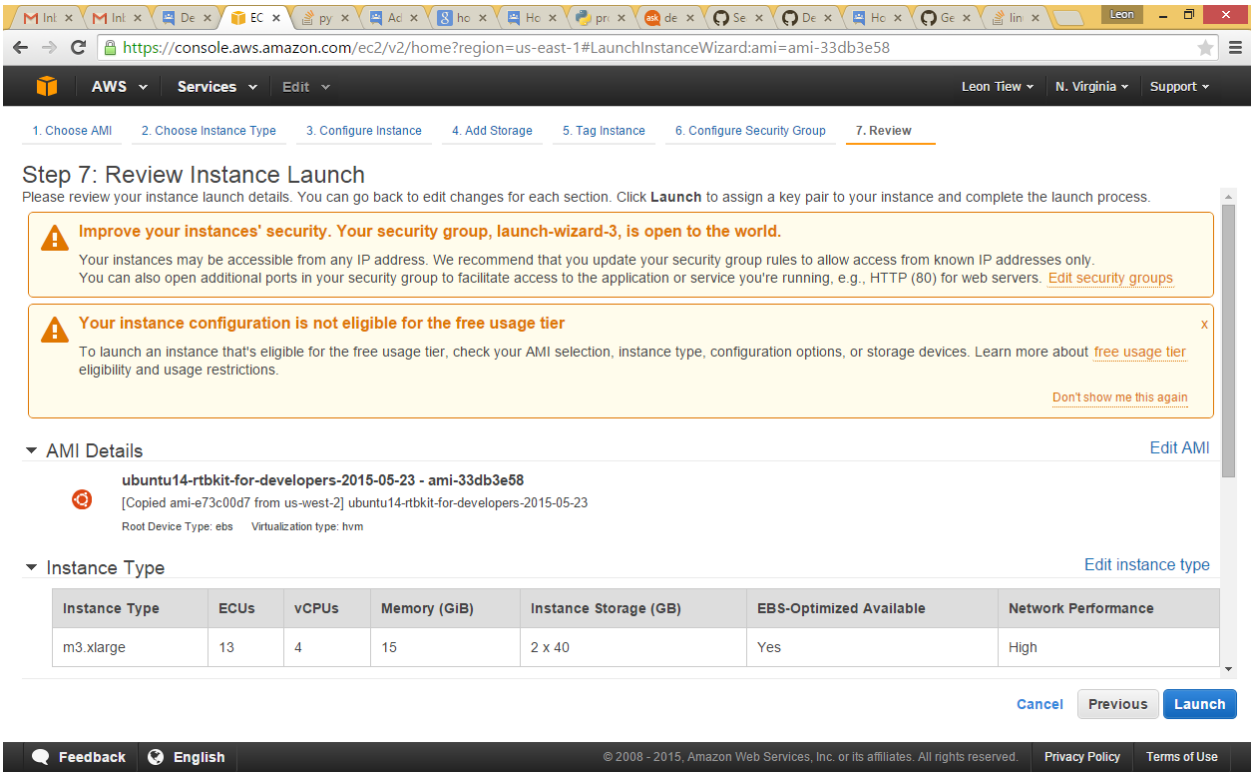
A 64-bit system is currently required because of the 4GB memory limit of 32-bit systems.

Make sure you have at least 16GB of disk for building from the source. Also, your CPU should have support for SSE4.2 as it greatly improves performance.

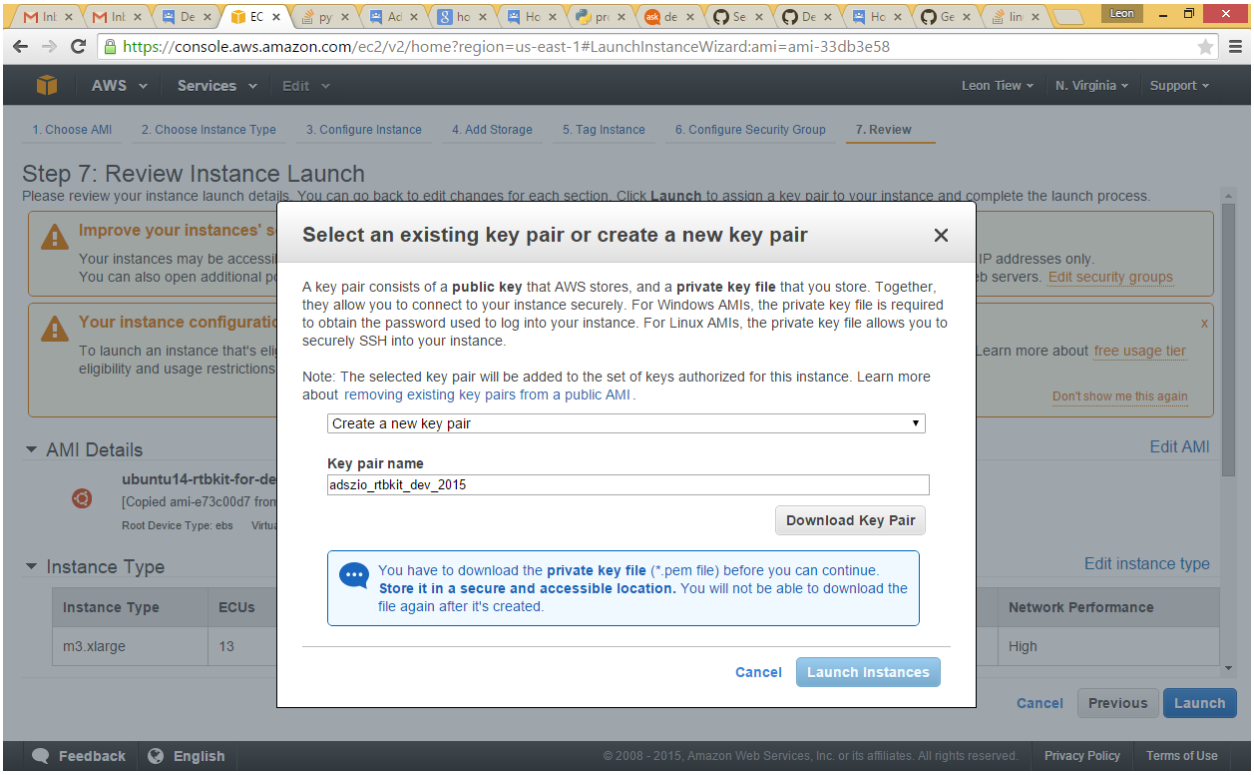
Select an m3.xlarge instance



Double check and launch your instance



Remember to download your key pair. Caution! This is the only time AWS provides you the right to download the key pair. So save you .pem file in a safe place.



SSH in your AWS EC2 Instance

Need to generate your PPK file

You will need PuTTY Key Generator to generate your .ppk file

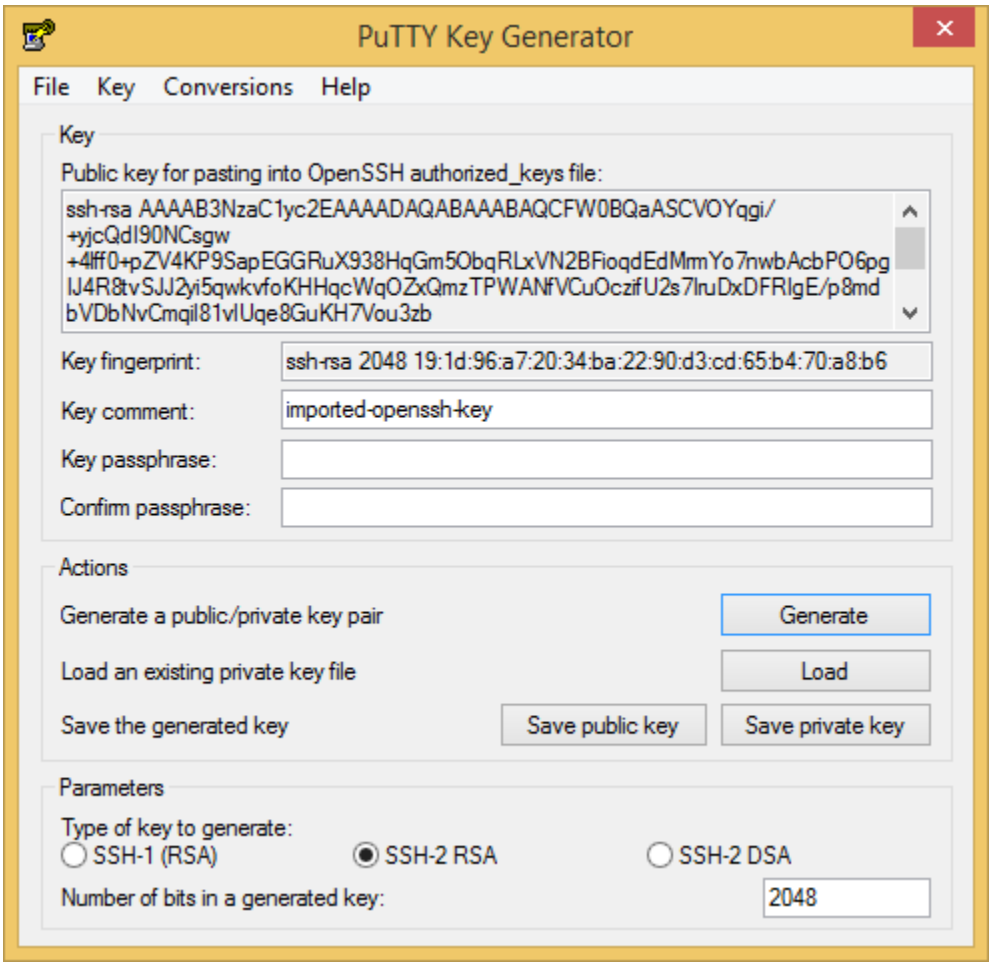
http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Create PPK file from PEM – Using Putty to connect to EC2

http://cloudarch.co.uk/2011/09/create-ppk-file-from-pem-using-putty-to-connect-to-ec2/#.VW0sUs9Viko

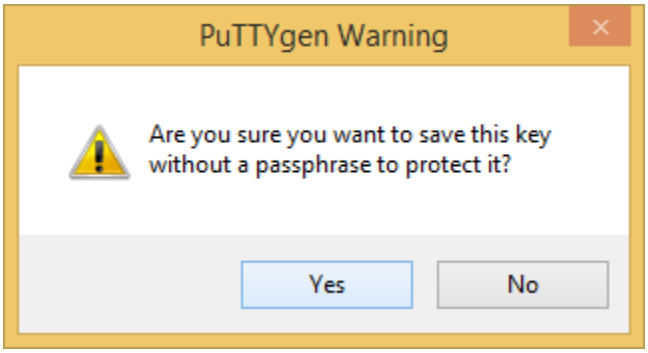
First import you .pem file

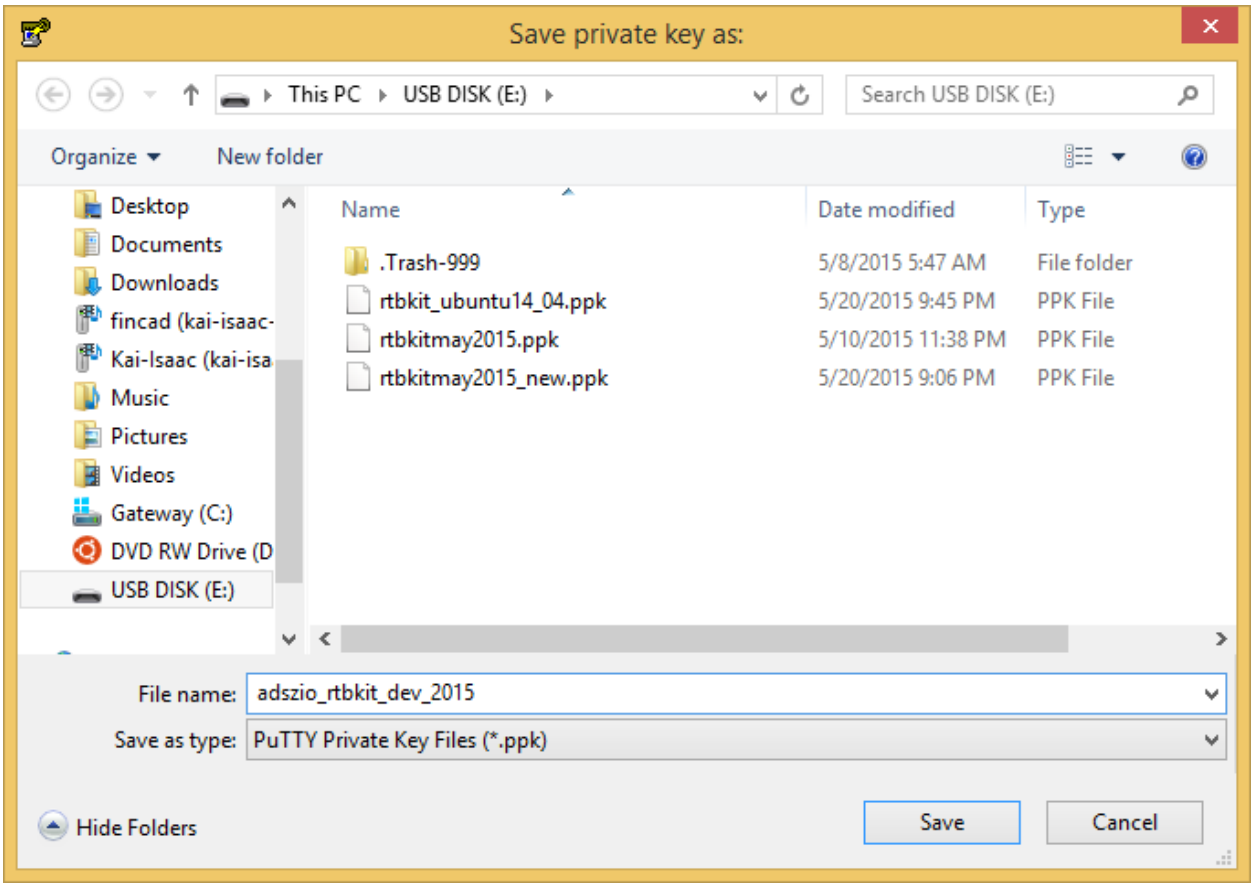
Conversions->Import Key



Click Load and select your .pem file.

Click Save private key

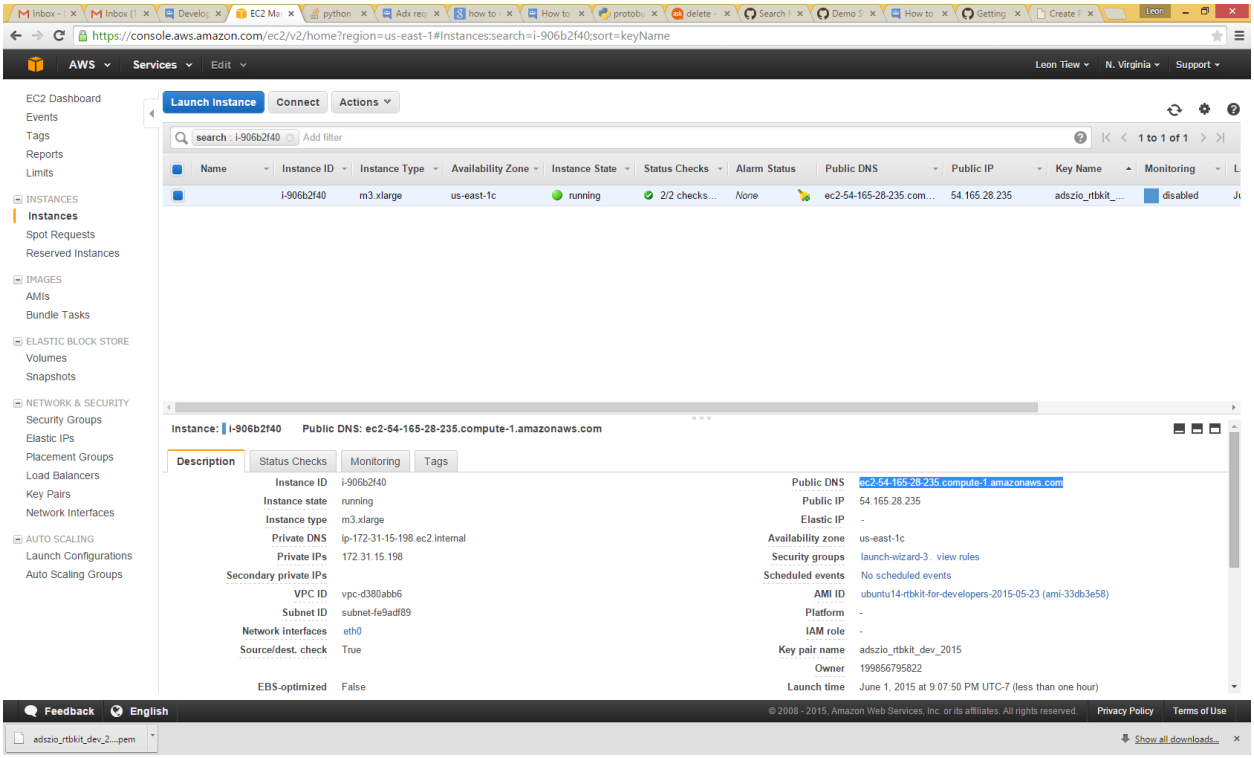




Please save your .ppk file in a safe place.

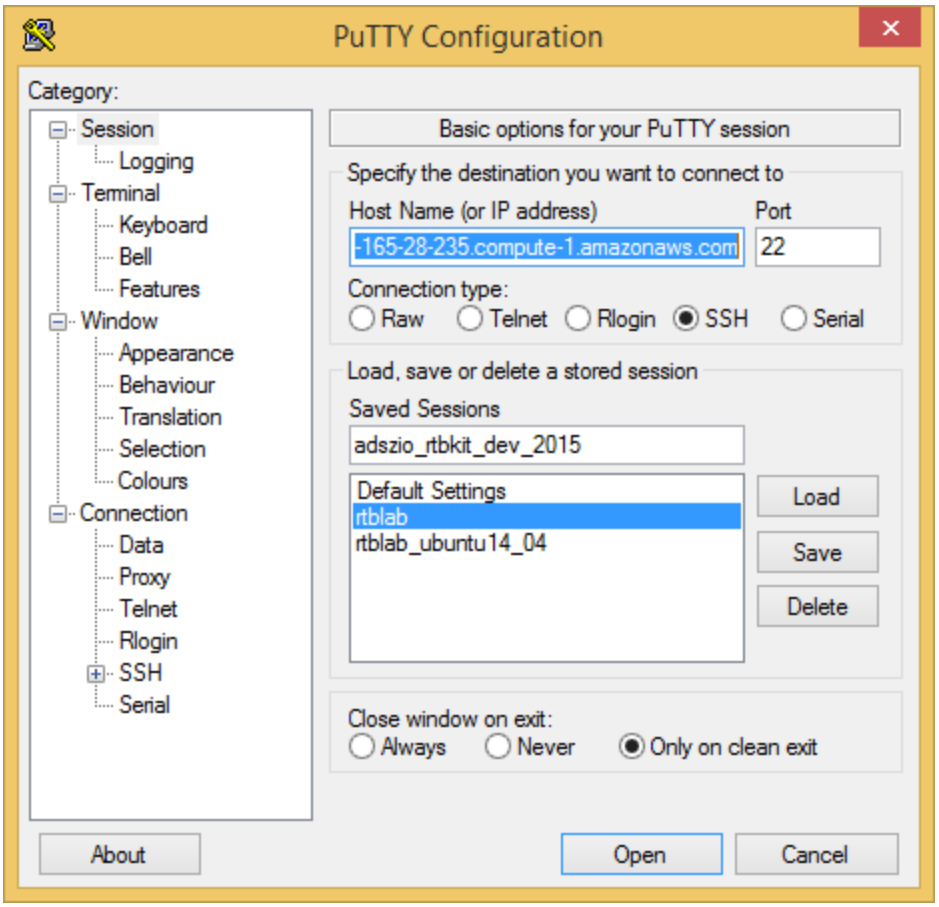
Done.

Prepare to login your AWS EC2 using your .PPK file

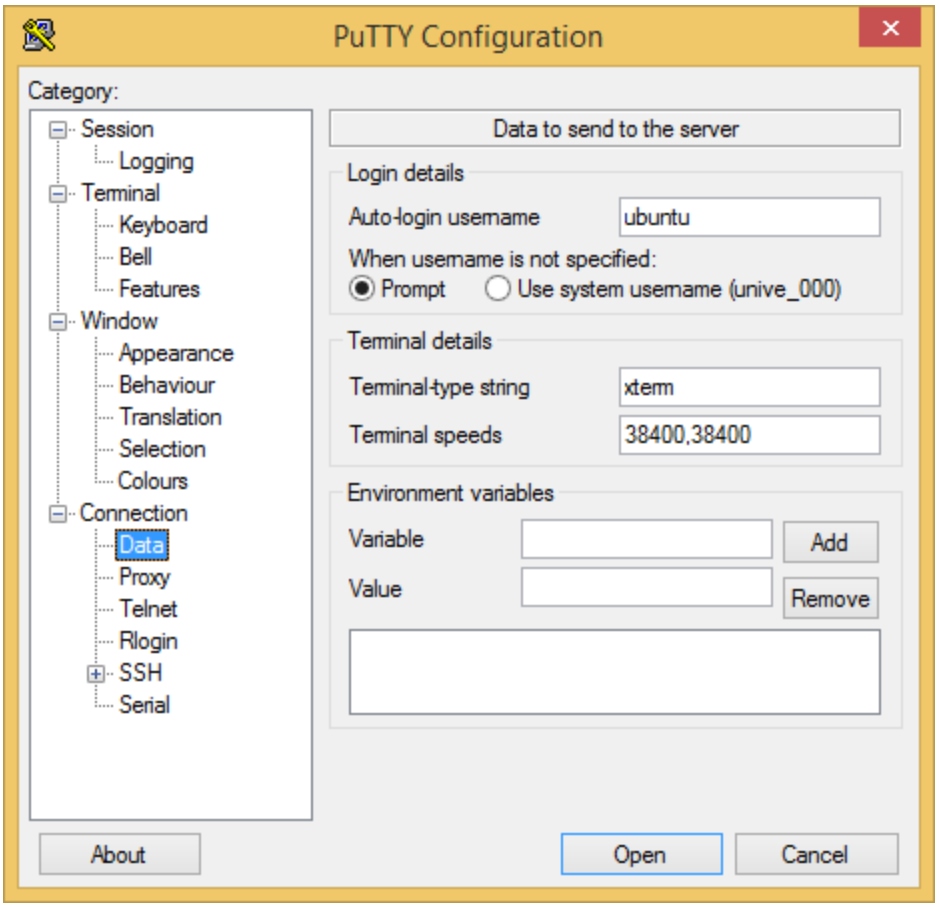


Copy Public DNS

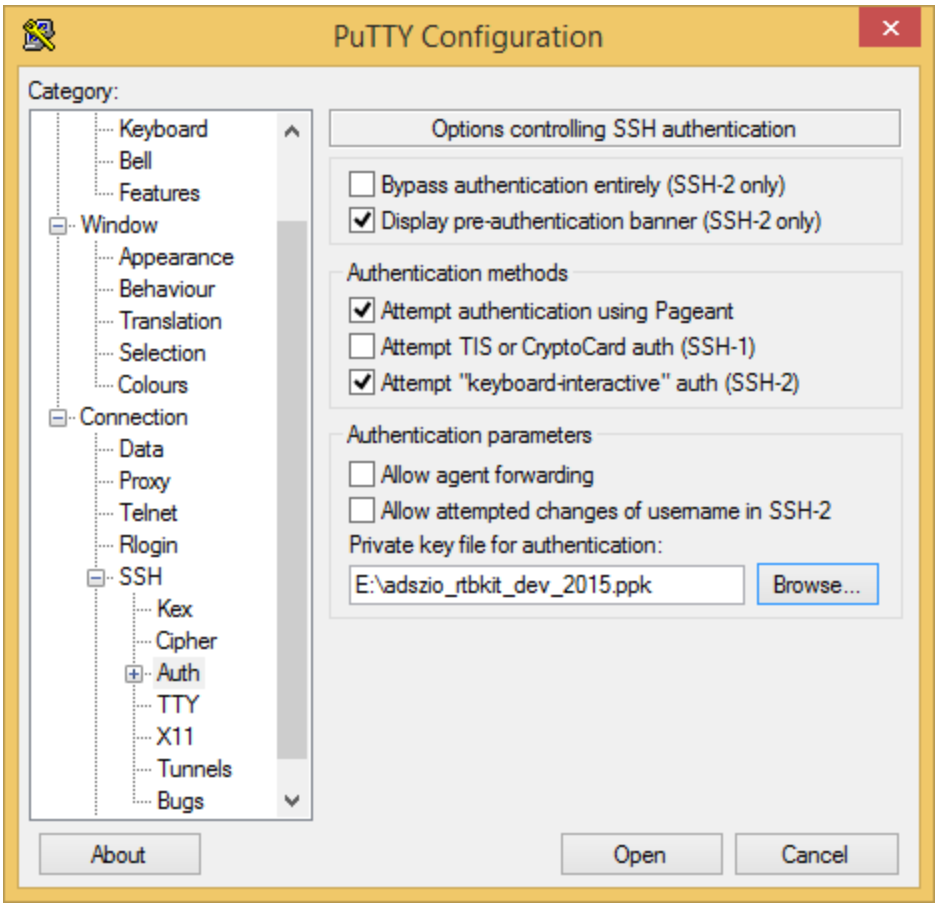
Create a session using Putty (please feel free to use other SSH tool that you are familiar)



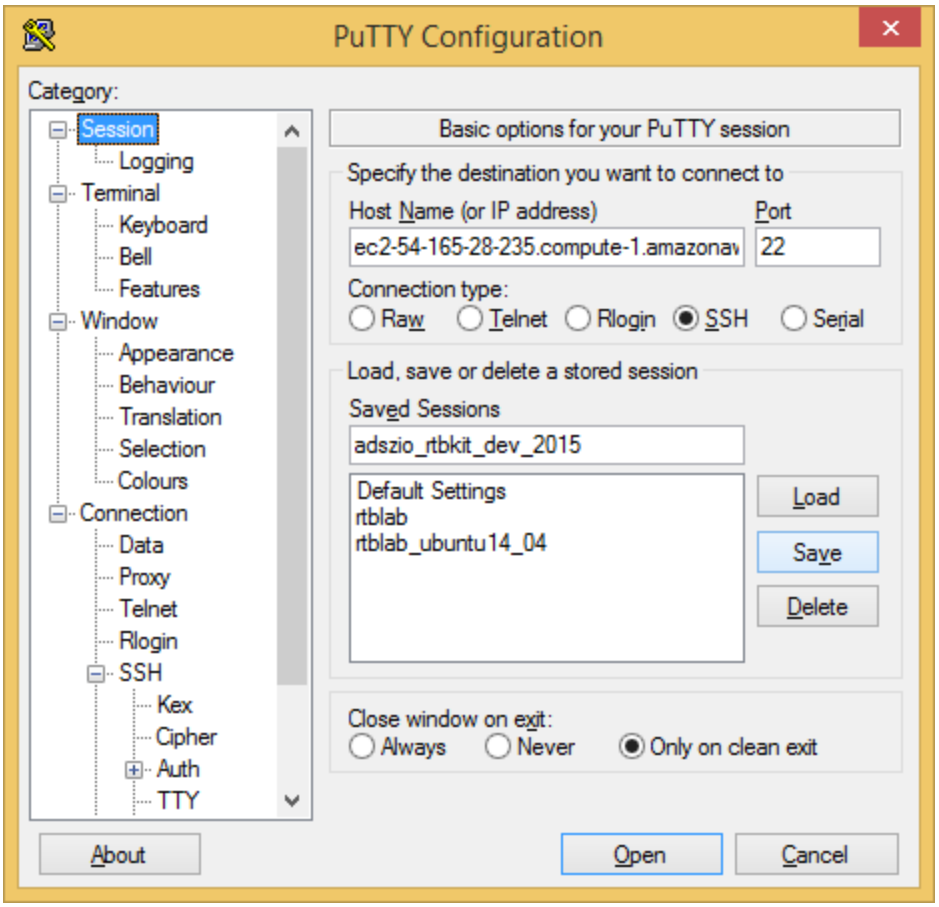
Create adszio_rtbkit_dev-2015 session.



Enter Auto-login username = ubuntu

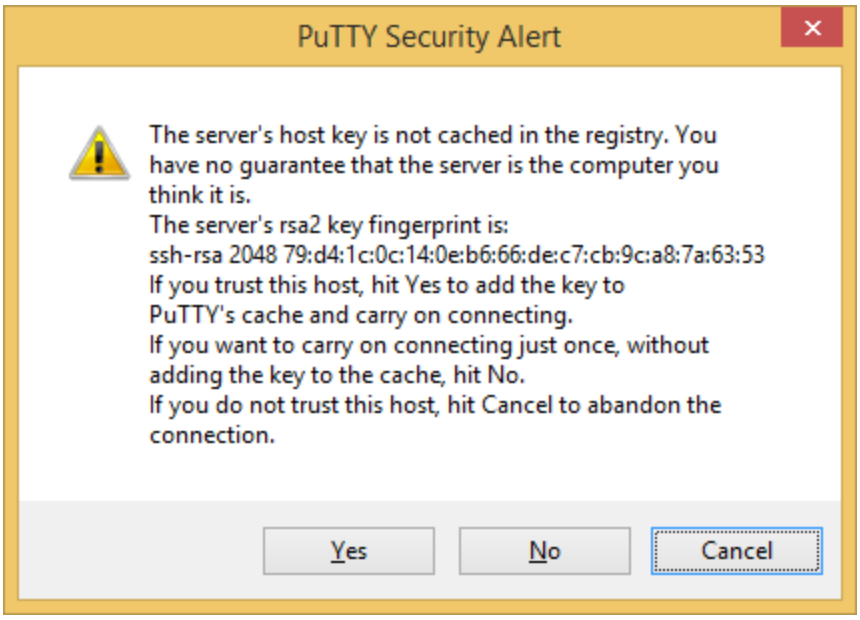


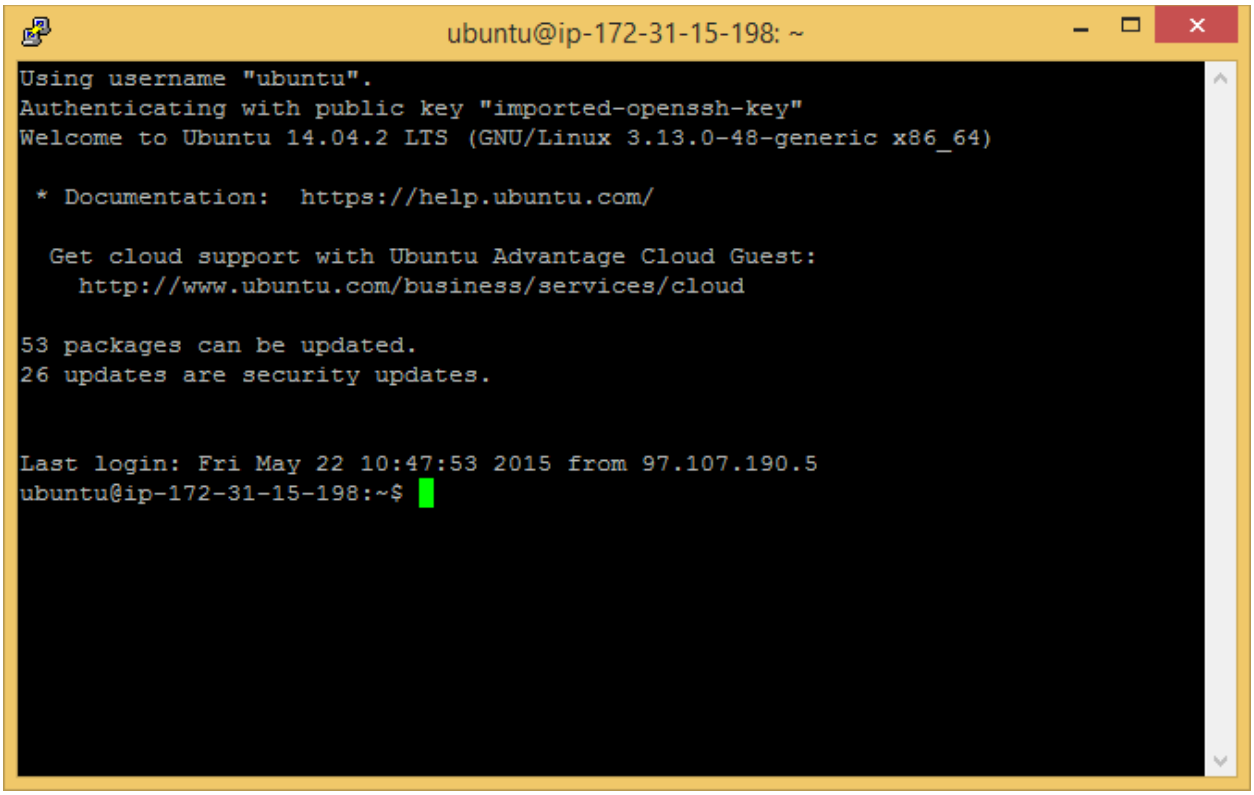
Select your .PPK file



Remember to save your session.

Click Open





sudo su - rtbkit

Test your RTBKit Stack making sure every core module is working

I will provide you the scripts to launch the services below.

For now do it manually so that you know how to start each RTBKit module.

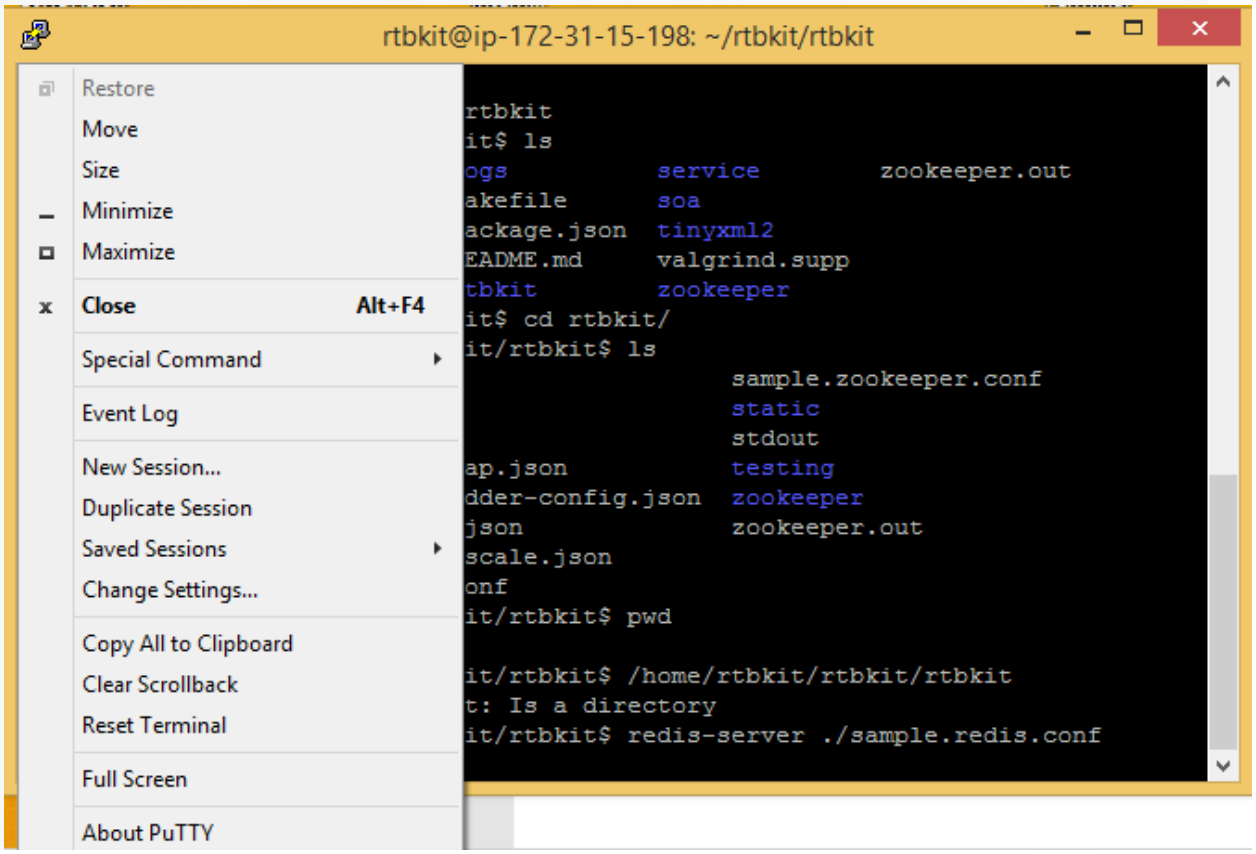
1) Start zookeeper

```
~/local/bin/zookeeper/bin/zkServer.sh start
```

2) Start Redis

```
cd /home/rtbkit/rtbkit/rtbkit
redis-server ./sample.redis.conf
```

Duplicate putty session



sudo su - rtbkit to login again

3) Start Carbon

```
sudo -i /opt/graphite/bin/carbon-cache.py start
```

4) Start Mock Exchange
cd rtbkit
./build/x86_64/bin/mock_exchange_runner

5) Finally start rtbkit stack
Duplicate a putty session (instruction given above)

sudo su – rtbkit
cd rtbkit
./launch.sh

This will start the tmux session

6) Look at a list of demo account

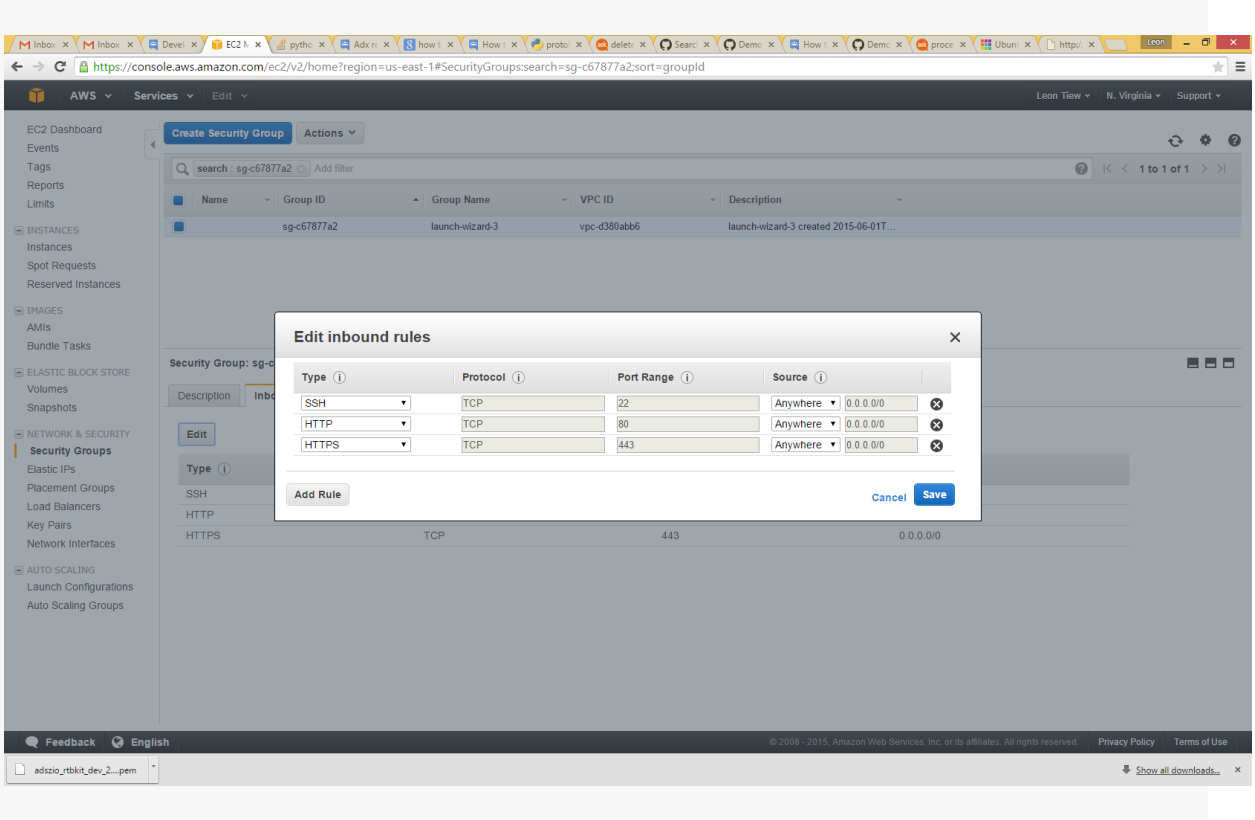
```
curl http://localhost:9985/v1/accounts
```

7) Add budget to the account “hello”

```
curl http://localhost:9985/v1/accounts/hello/budget -d '{ "USD/1M": 123456789 }'
```

Setup Graphite WebApp for Visualization

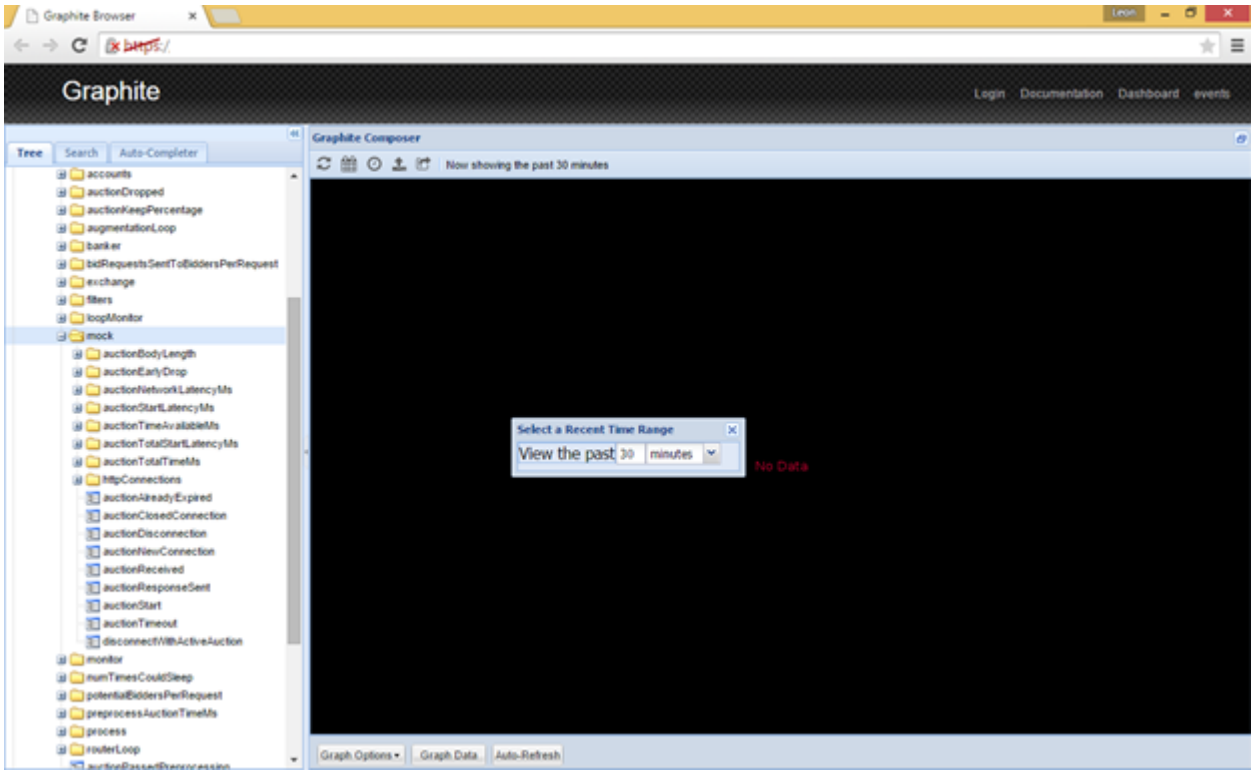
Add new rules for the Security Group in you AWS EC2 for Graphite



Launch Graphite

```
Open your browser and enter you EC2 Public or Elastic IP (handle to create one)
```

Let's use Graphite to visualize bidding information

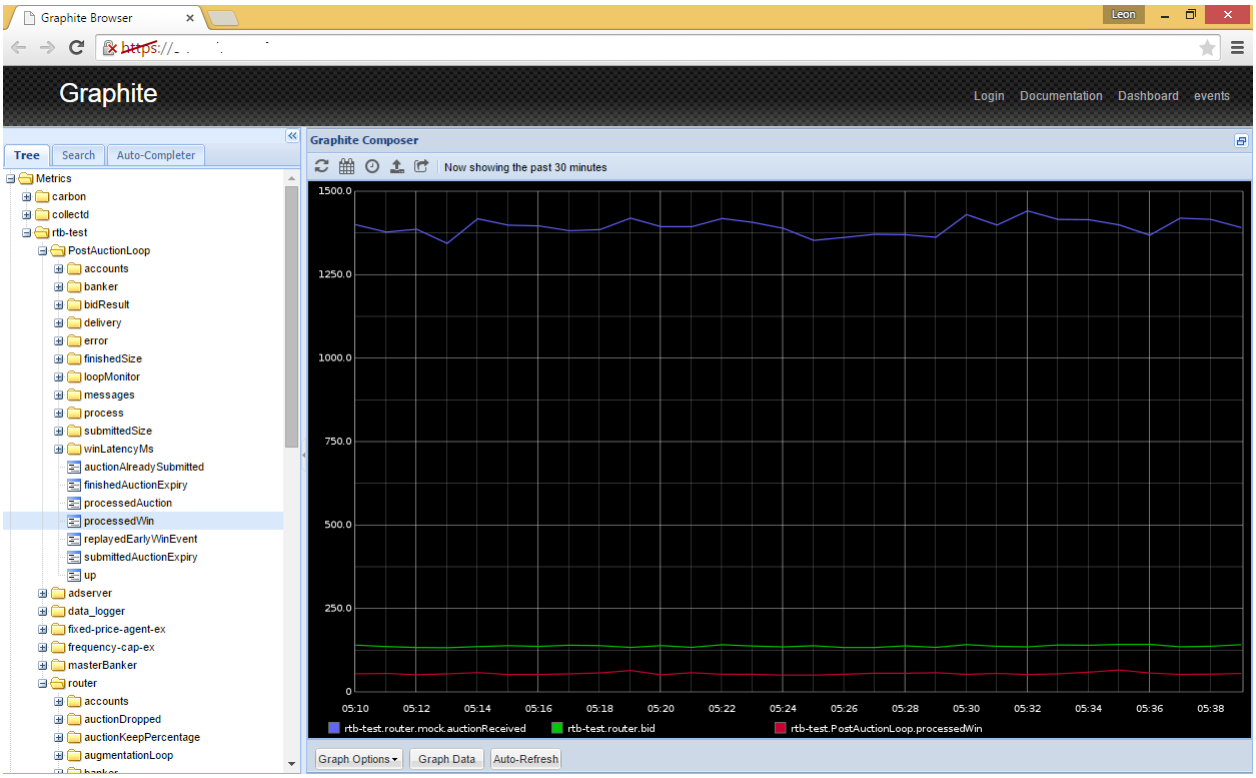


Set view the pass 30 min

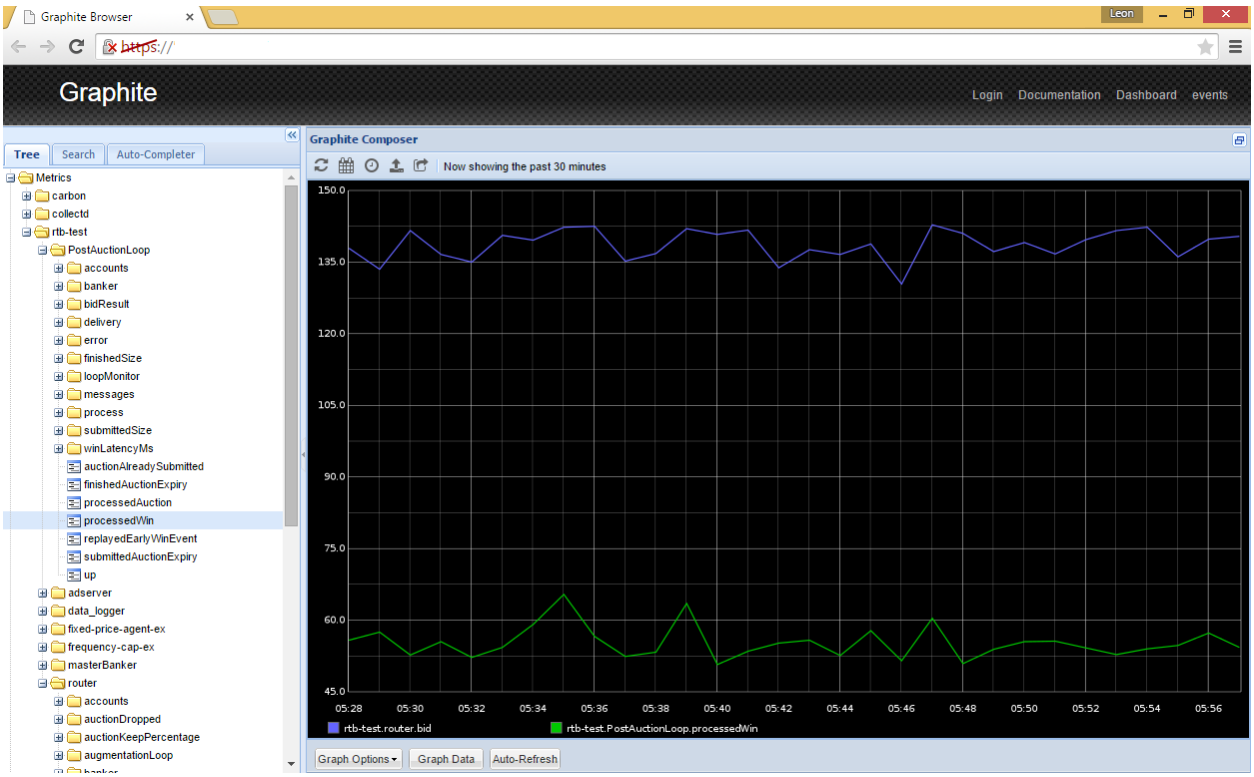
Create a custom graph

From the navigation tree, select

1) rtb-test.router.mock.auctionReceived (blue)
2) rtb-test.router.bid (green)
3) rtb-test.postAuction.processedWin (red)



A better graph

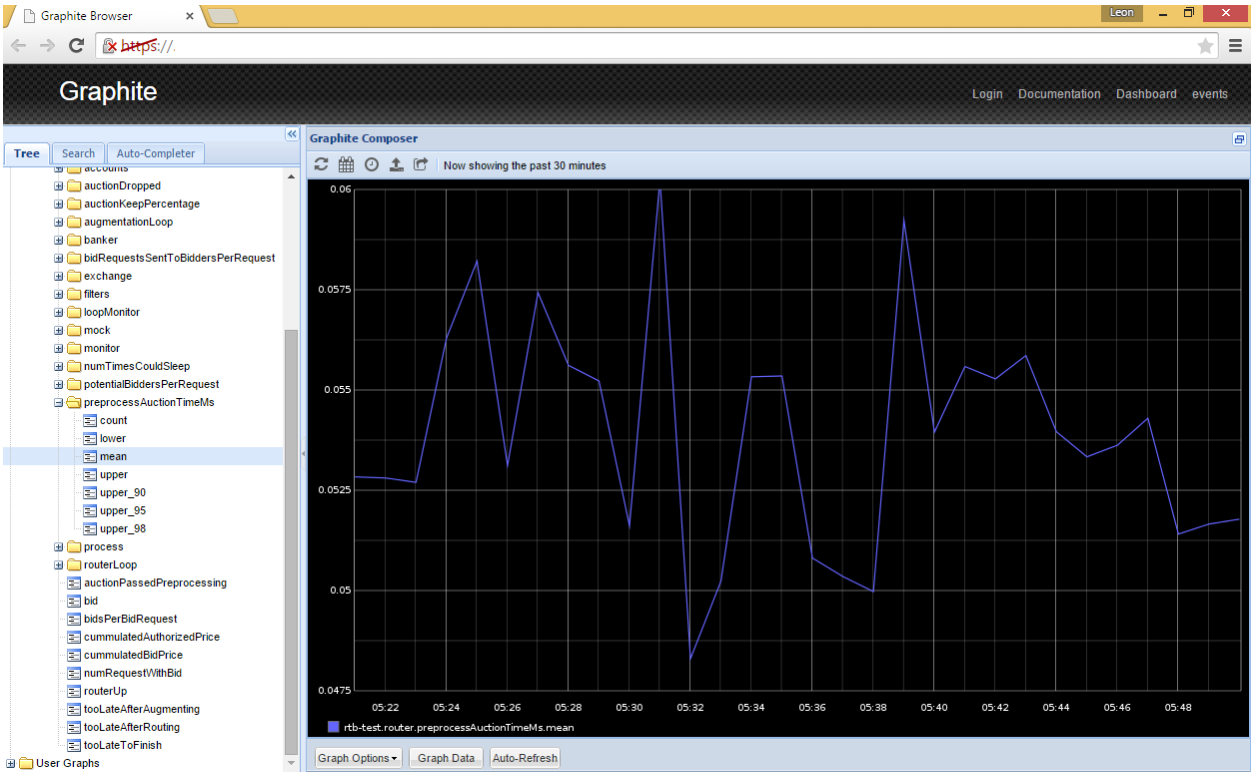


Here you can see that out of the 1350 plus bid requests received, the bidding agent placed over 135 plus bids and out of that it received over 50 – 60 winning bids (almost 37% - 44% winning bid in average if my math is correct)

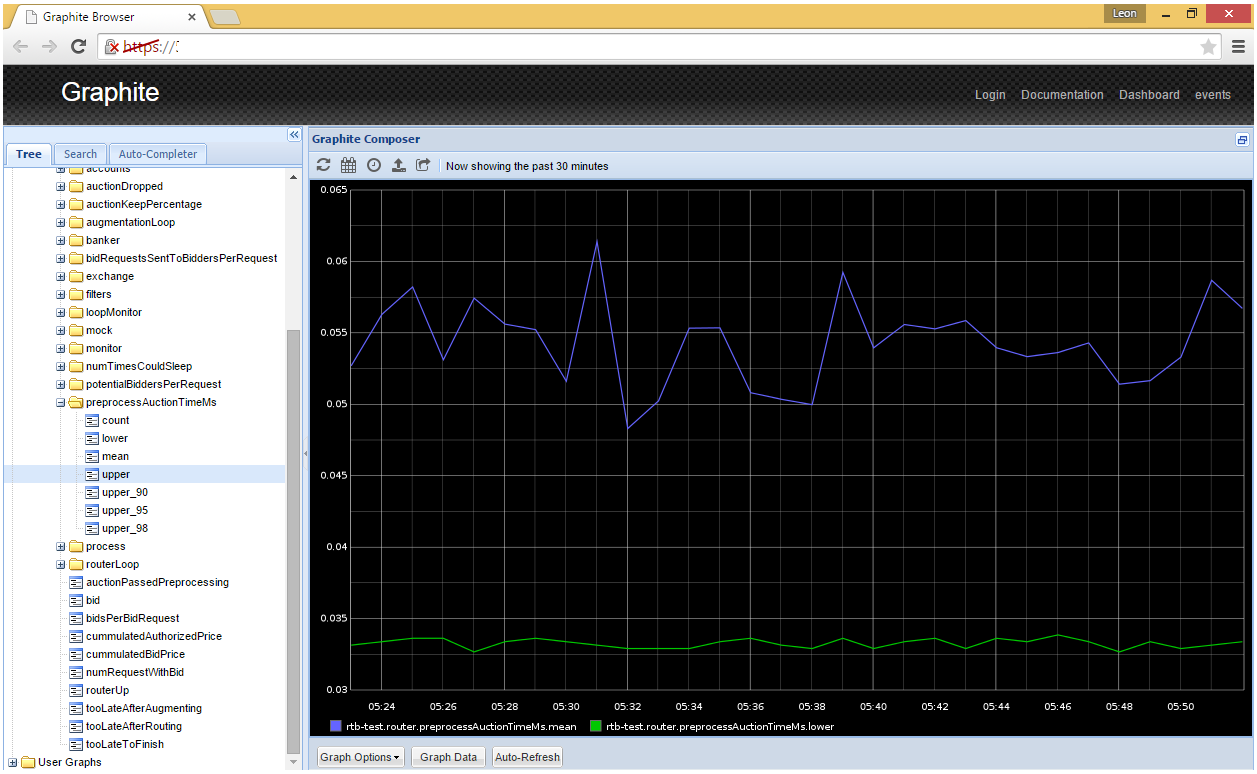
Monitor the Pre-process Auction Time

Remember most Add Exchange request a response time in 100 ms

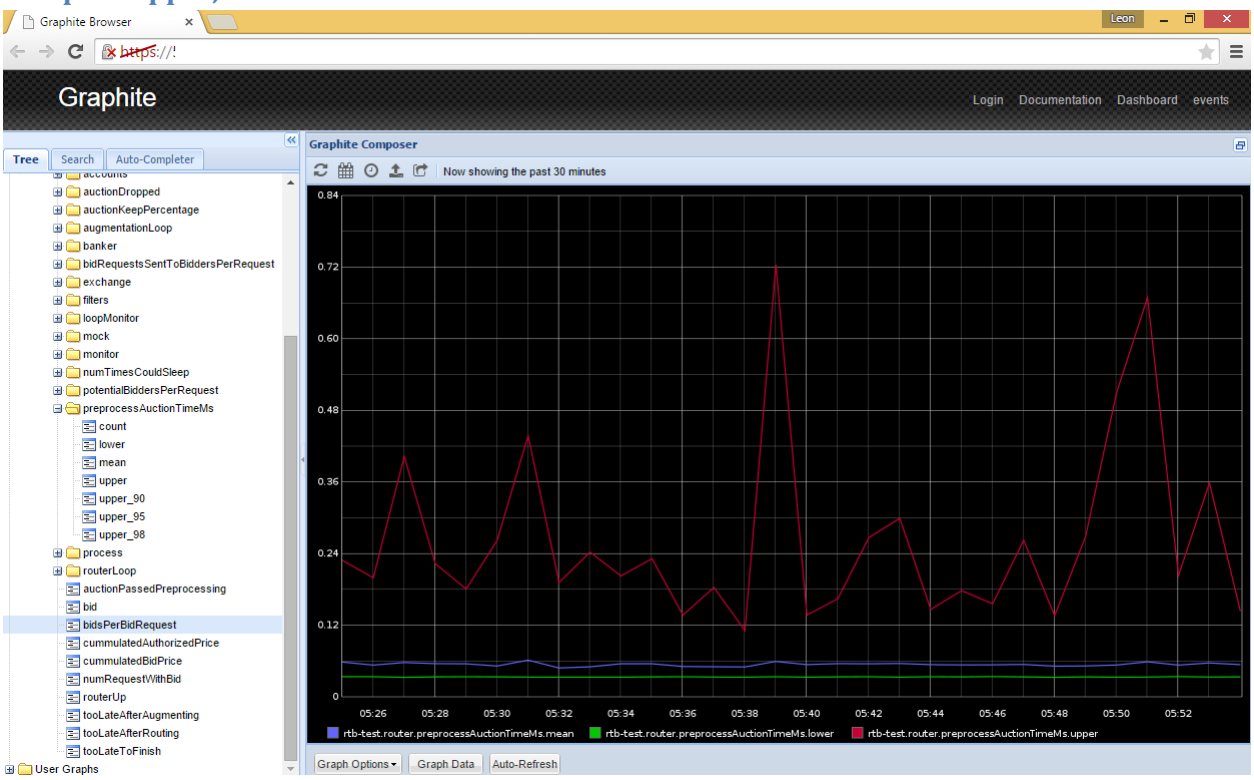
Preprocess Auction Time in Milliseconds (mean)



Compare mean and lower



Compare upper, mean and lower



The beauty of Graphite tool is, RTBkit no longer a black box, you can build custom graph and visualize all bidding transactions that interested you. Not the mention, it is also a good tool for debugging.

I hope this instruction is useful and get you up to speed with RTBKit faster; hence, save your precious time.