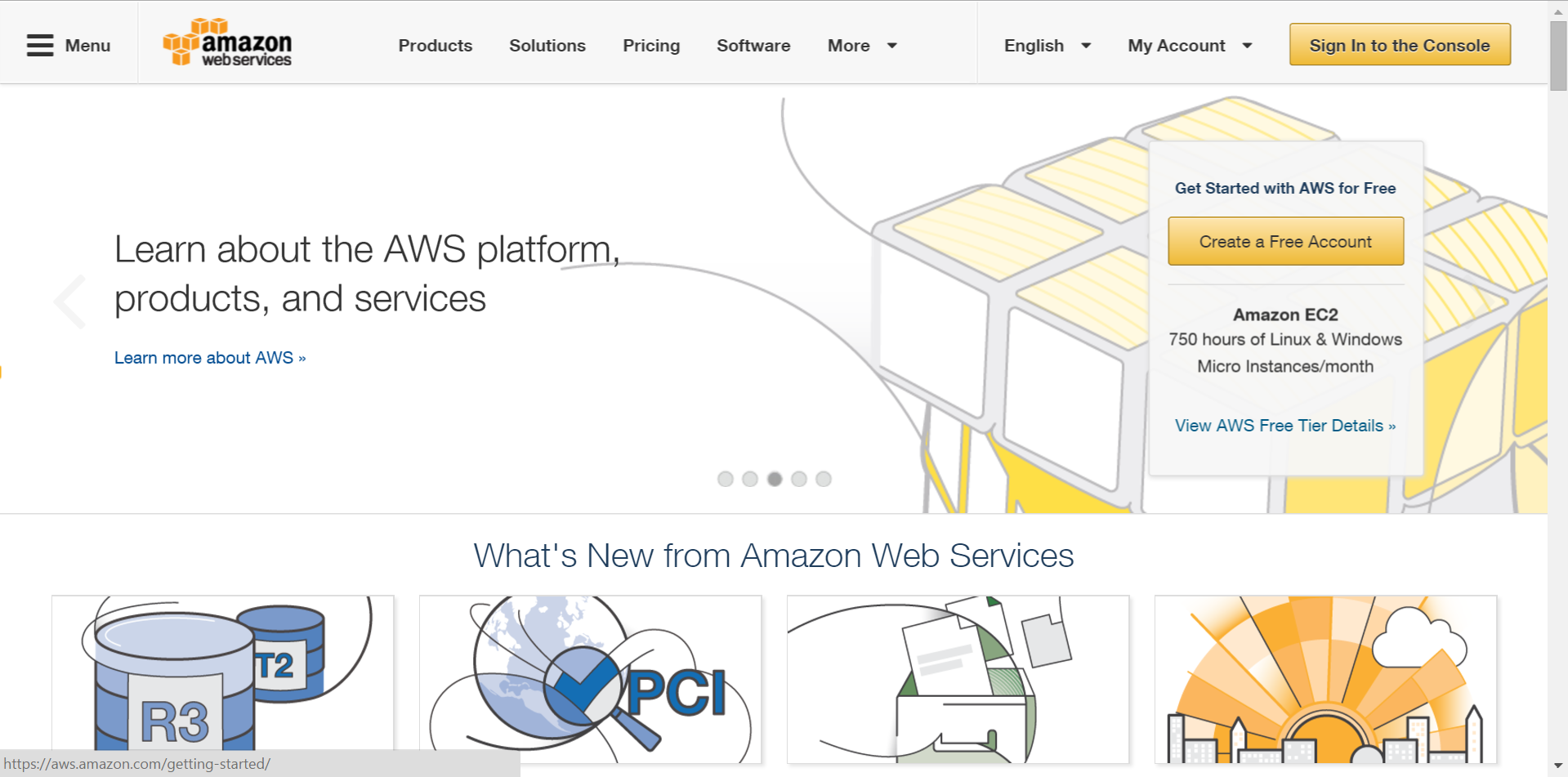
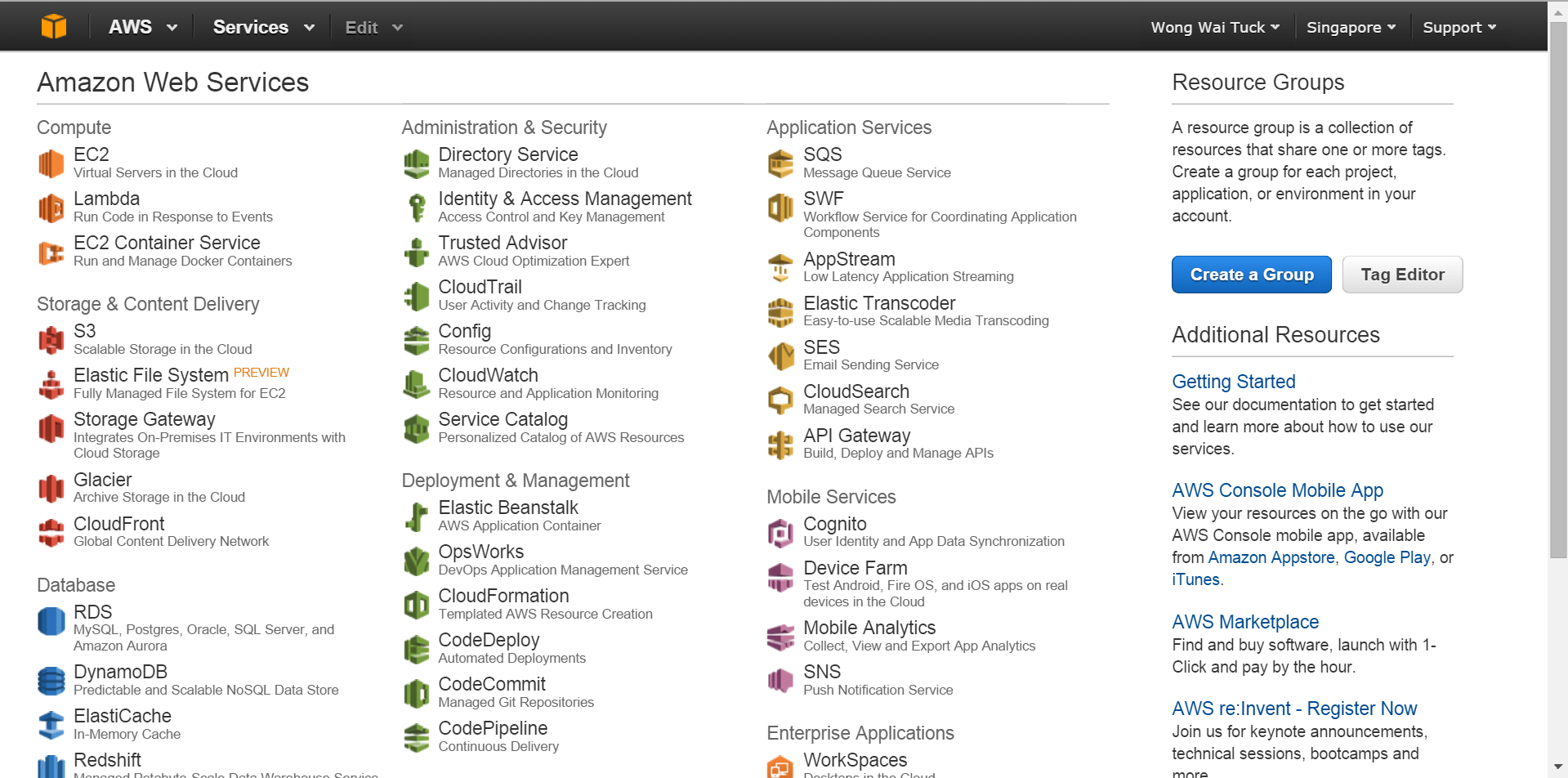
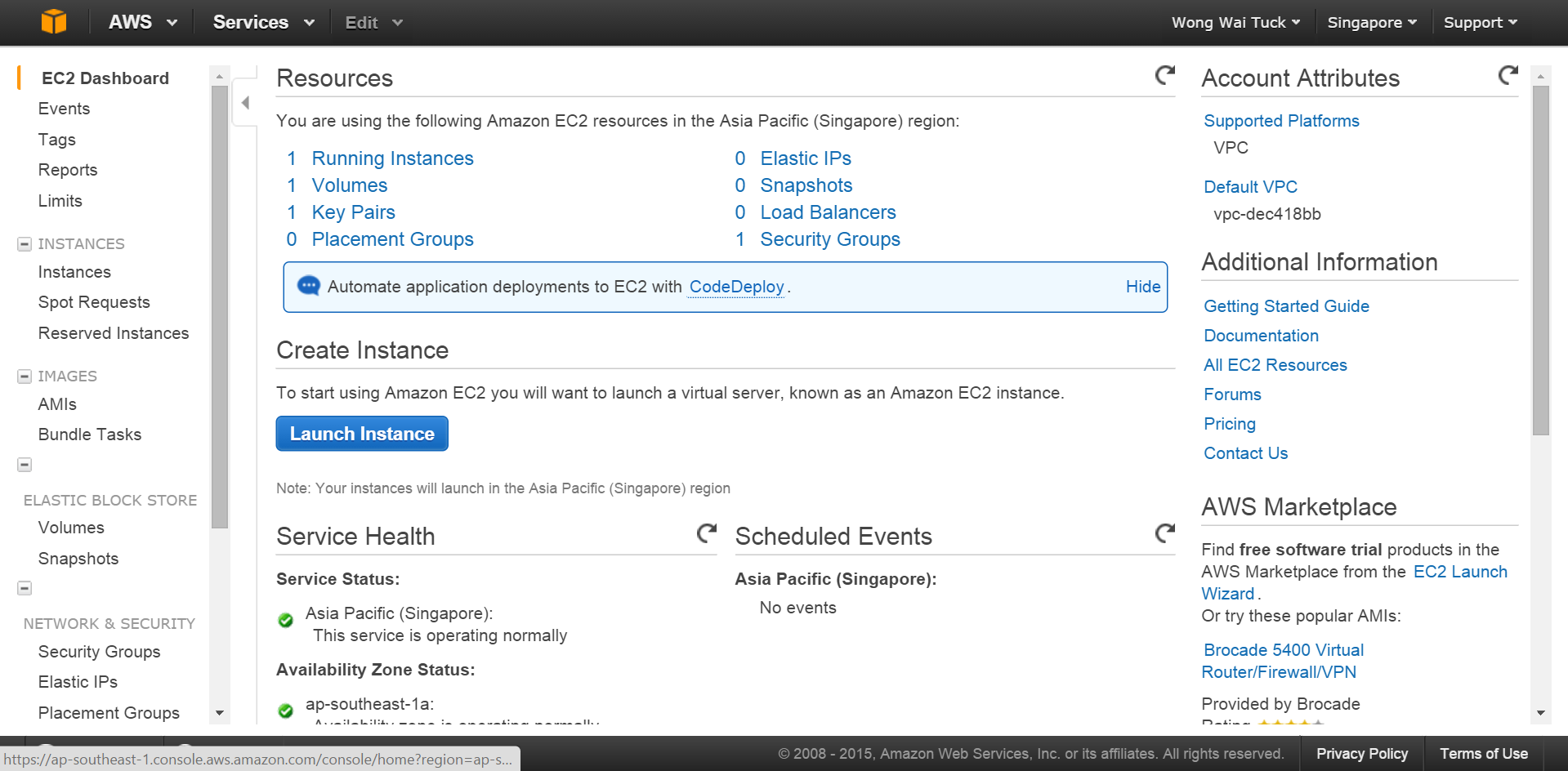
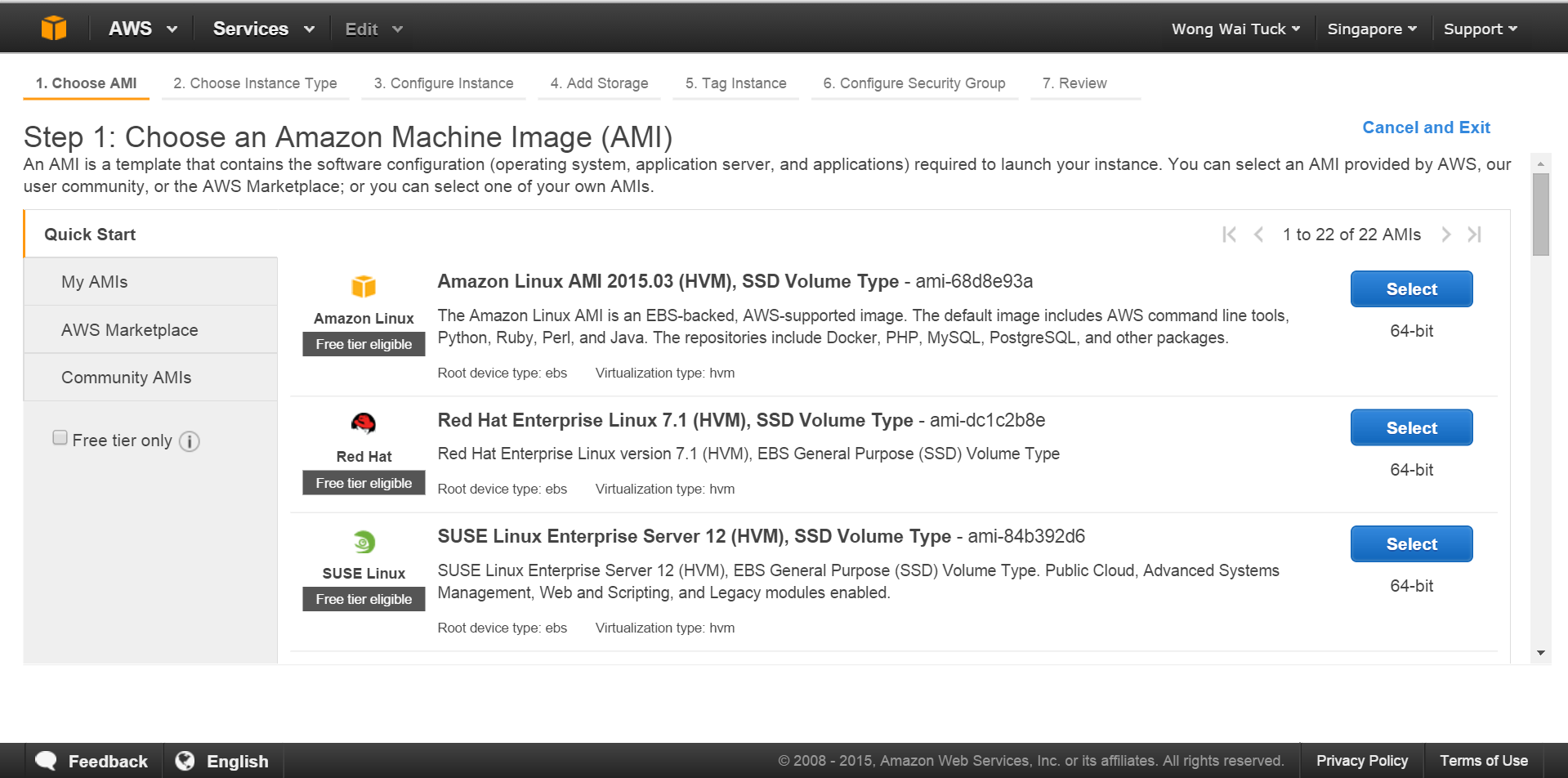
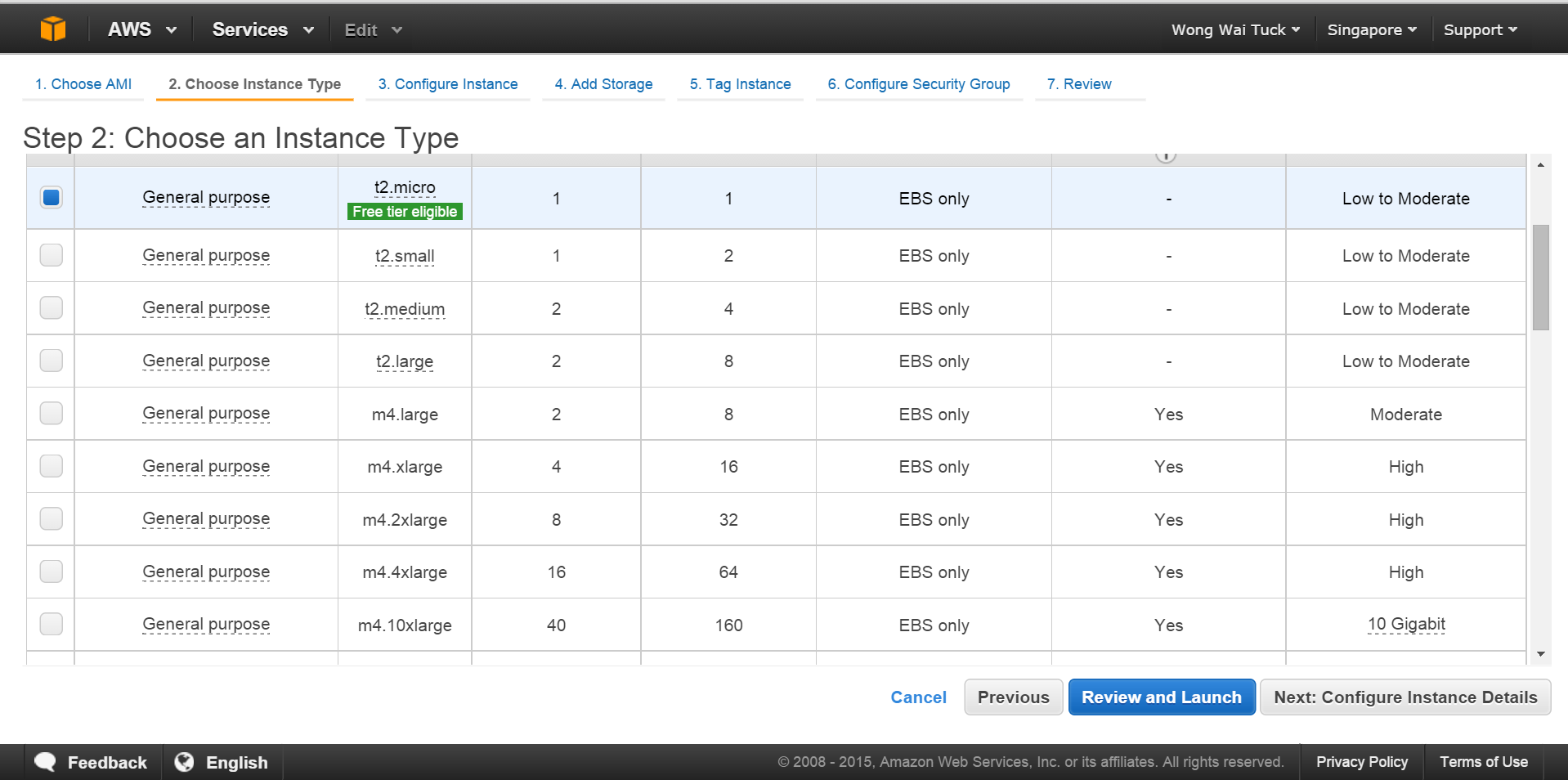
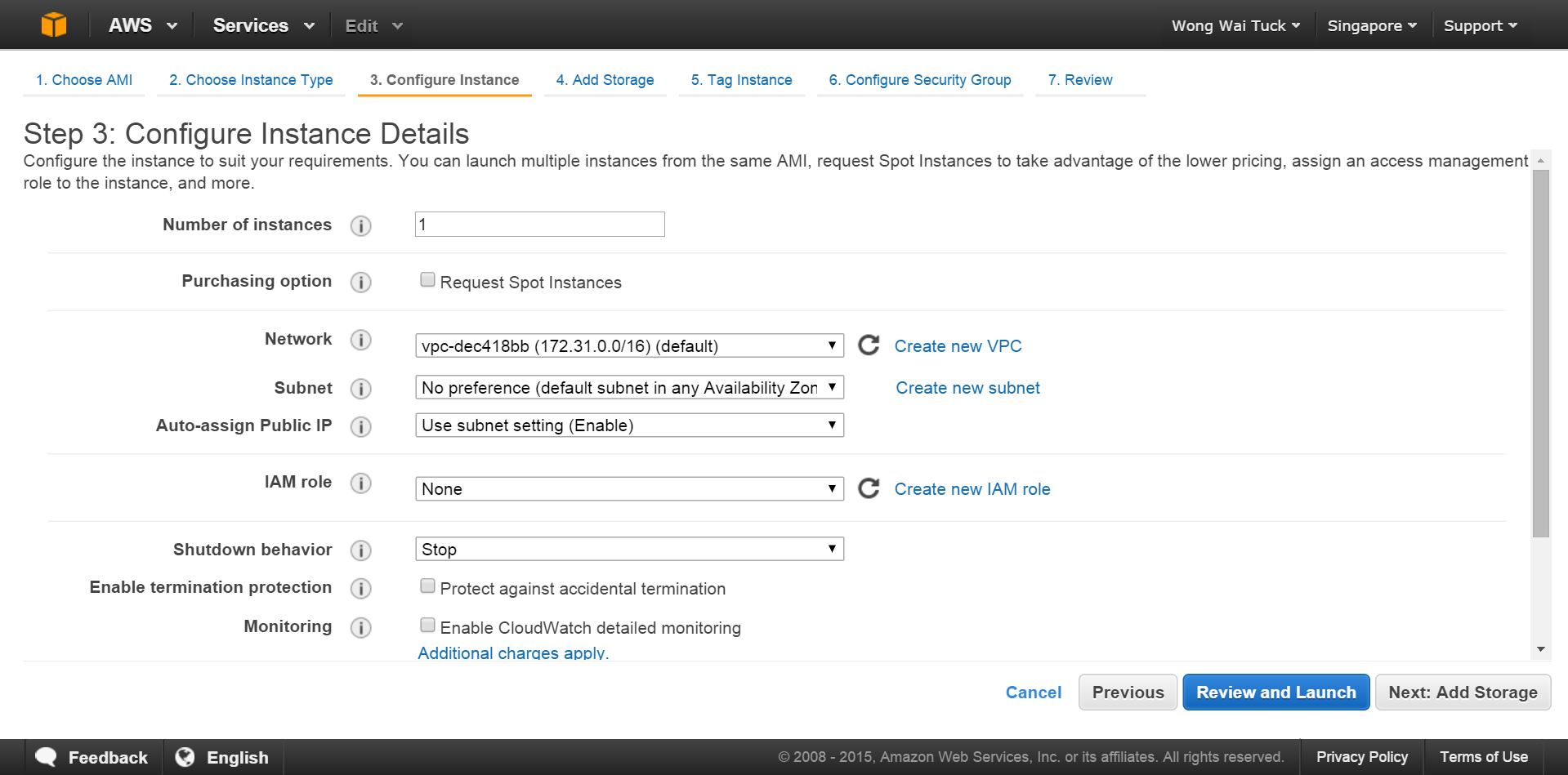
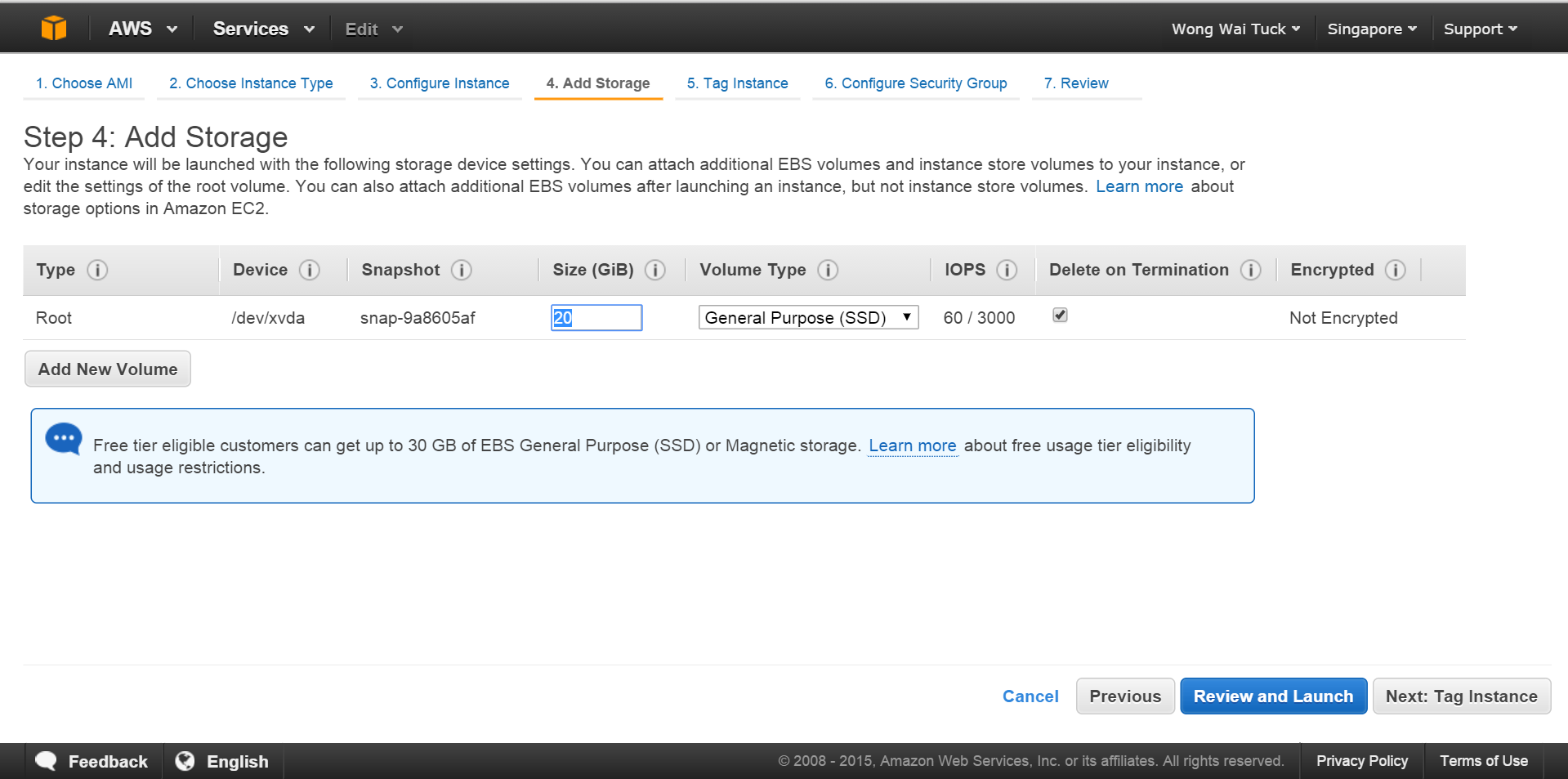
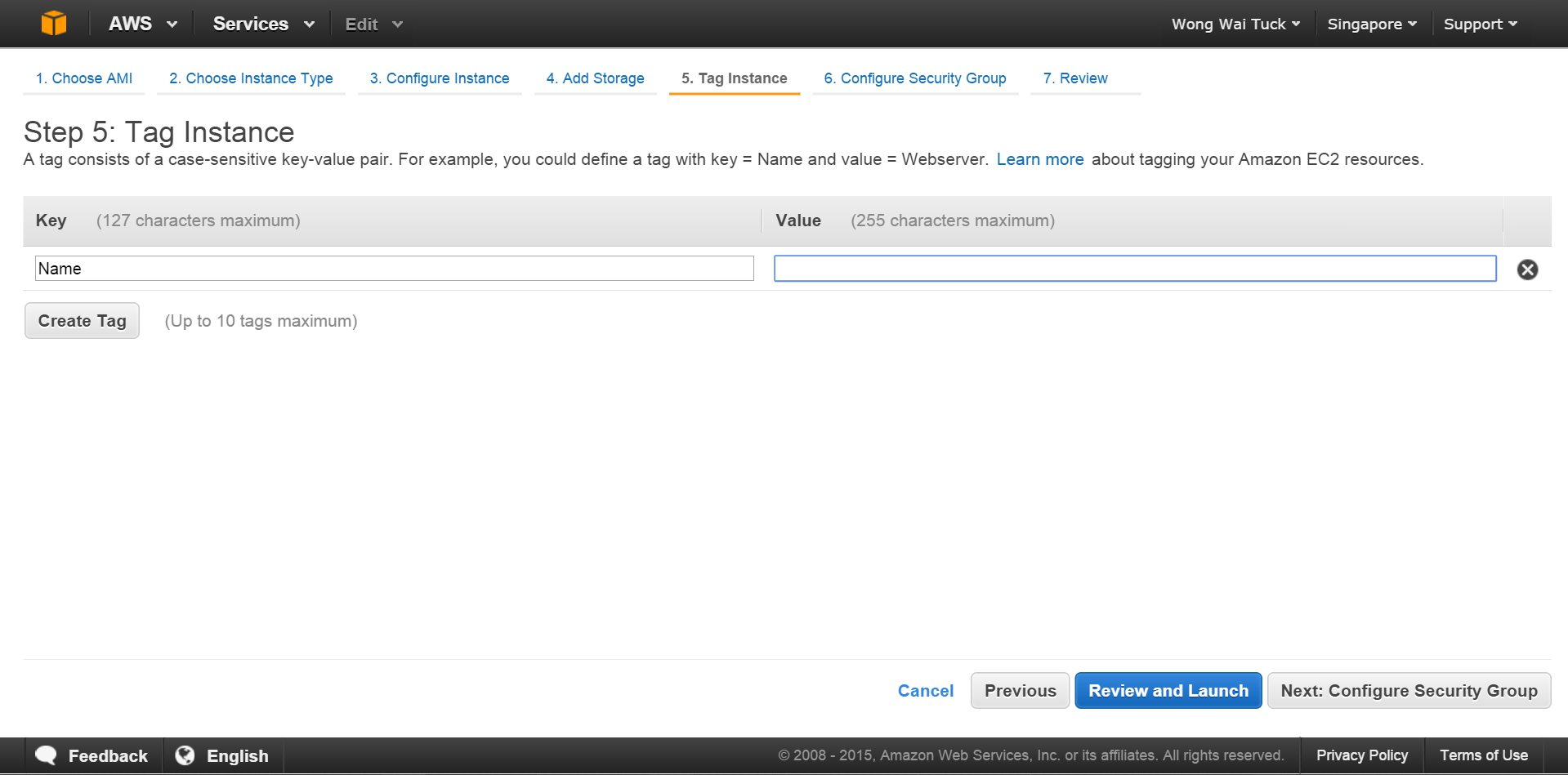
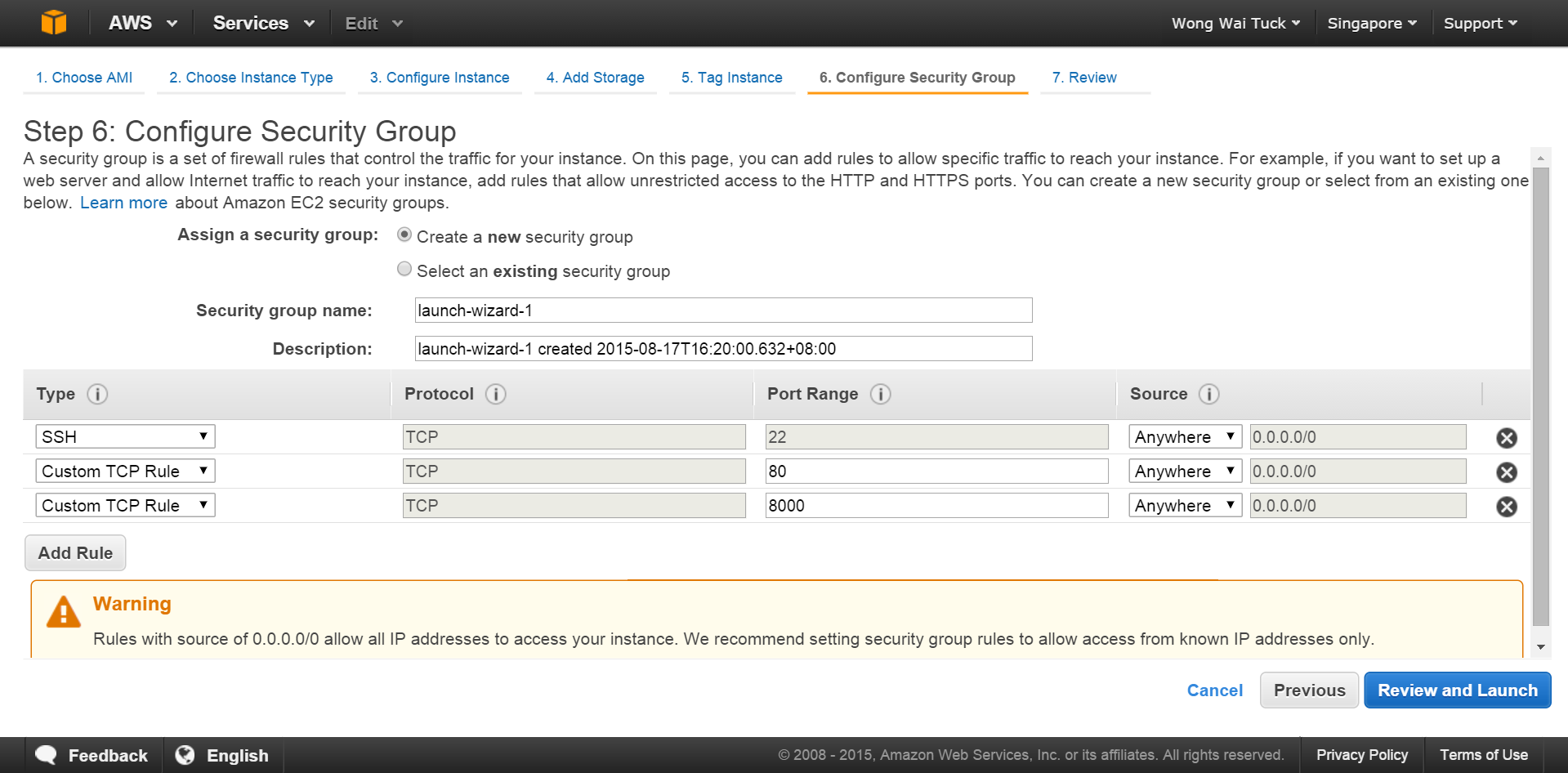
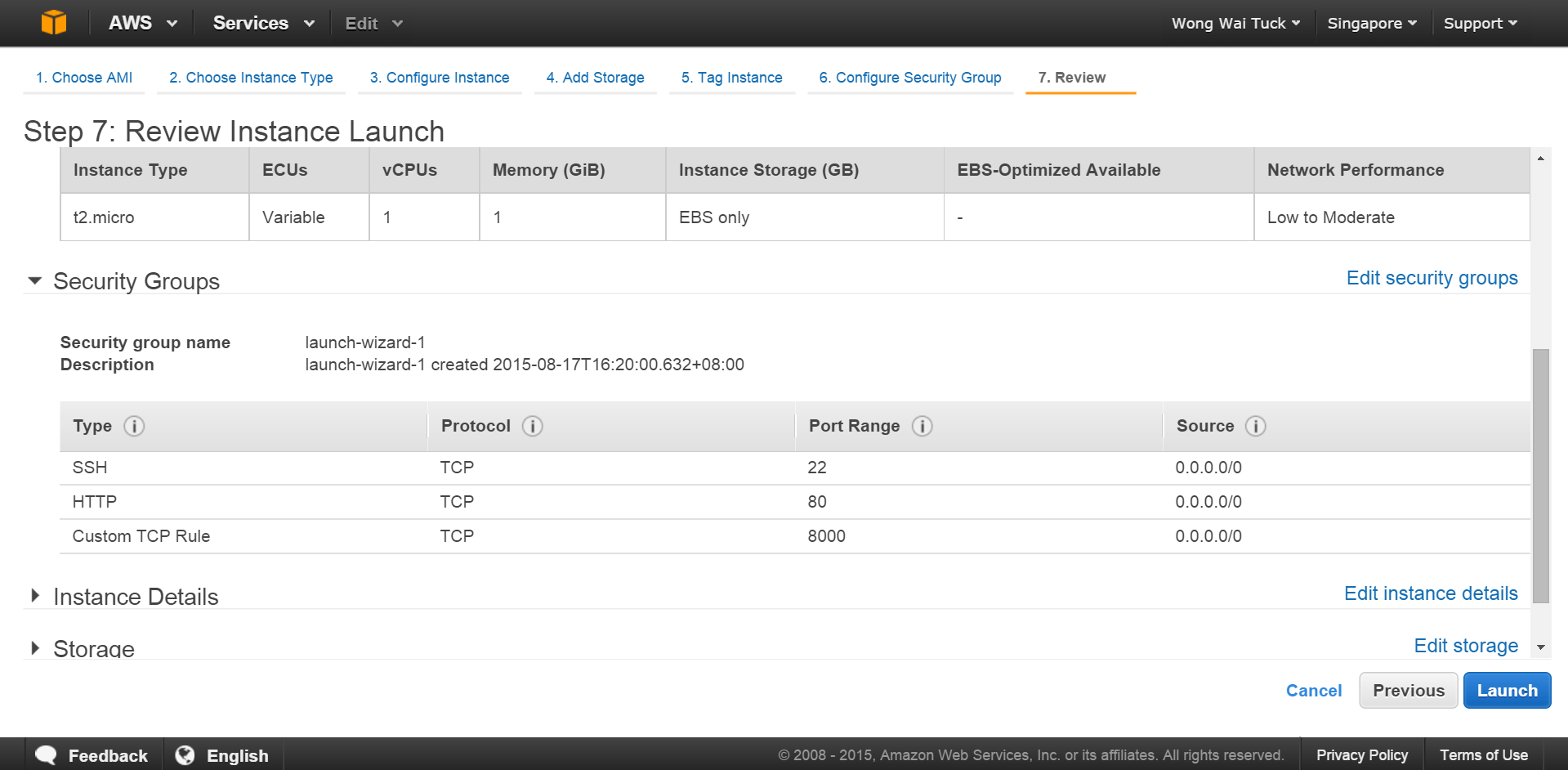
Setting up tmpnb Server

1. Login to AWS on through the following URL.  
   <https://aws.amazon.com/>
2. Click to “Sign In To The Console“ to enter the AWS Portal and create credentials if you have not done so already.  
   
3. Click on EC2. We’ll be using EC2 to host our tmpnb server. Make sure the region you have selected is in Singapore.



1. You will be greeted with the default managing page for EC2. Click the “Launch Instance” button to launch an instance.  
   
2. Select Amazon AMI   
   
3. Select an instance type that is suitable, then click “Next: Configure Instance Details”.  
   
4. Select “Next: Add Storage”  
   
5. Change the size to a larger number (say 20GB), and press “Next: Tag Instance”.   
   
6. Press “Next: Configure Security Group”  
   
7. Make sure that “Create a new security group” is selected add the following Custom TCP Rules as below and press “Review and Launch”.  
   
8. Review and launch your instance!  
   
9. Login to your EC2 instance, following the instructions on the screen.
10. Type the following command without quotes to install docker, and if prompted, allow the installation:

sudo yum install docker

1. Type the following command to start the docker daemon

sudo service docker start

1. Type the following command in the AWS EC2 instance terminal to save a unique random token to your linux environment. This will be used for authentication by your docker containers.

export TOKEN=$( head -c 30 /dev/urandom | xxd -p )

1. Type the following command in the AWS EC2 instance terminal to start the proxy server. This proxy server will be used to route the request from the web to your tmpnb orchestrator running locally.

sudo docker run --net=host -d -e CONFIGPROXY\_AUTH\_TOKEN=$TOKEN --name=proxy jupyter/configurable-http-proxy --default-target <http://127.0.0.1:9999>

1. Type the following command to download the docker image containing the custom ipython notebook that you want to run on the tmpnb docker instance.

sudo docker pull waituck/custom\_nb

1. Finally, type the following command to run your tmpnb server!

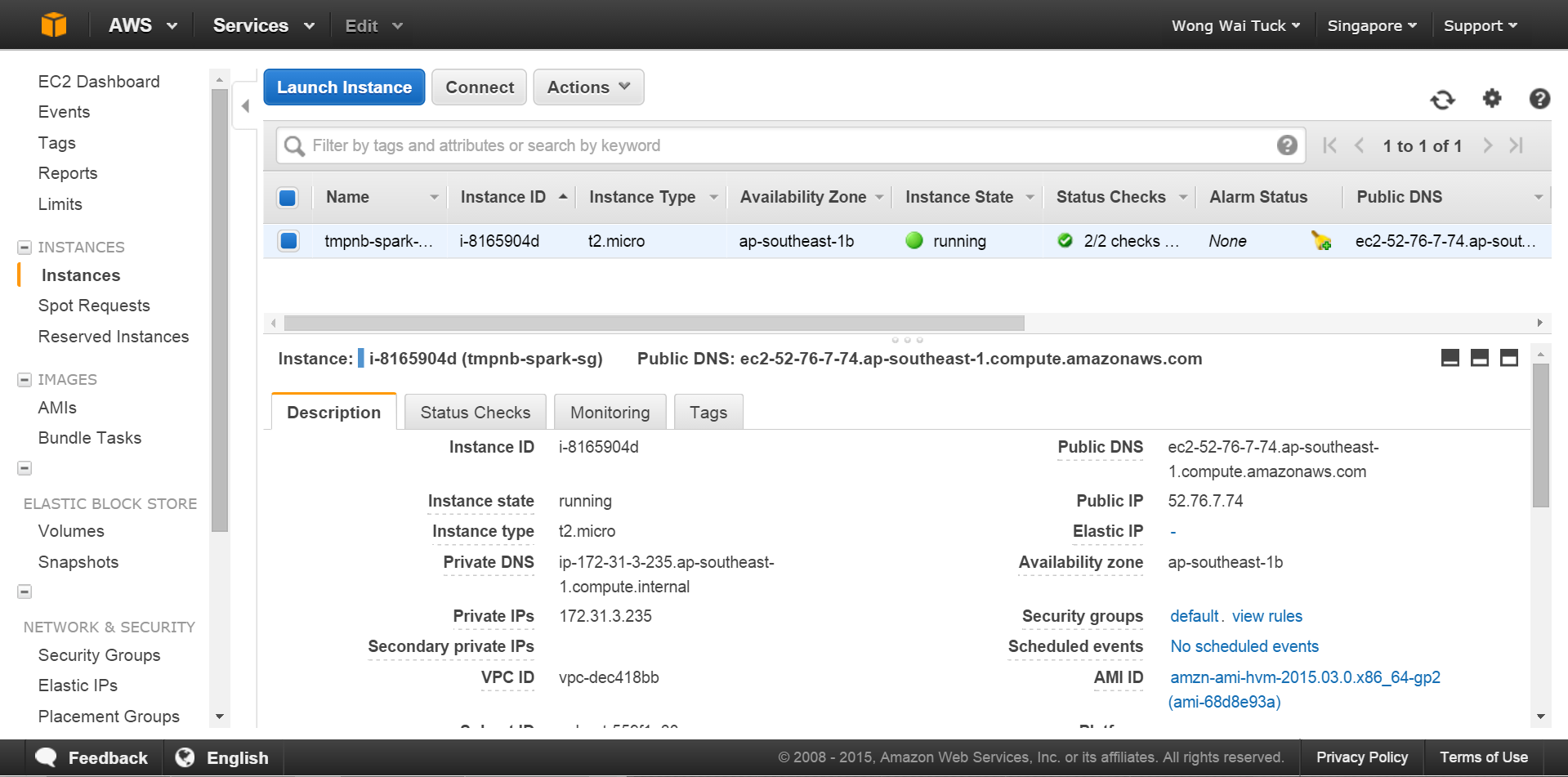
sudo docker run --net=host –name=tmpnb -d -e CONFIGPROXY\_AUTH\_TOKEN=$TOKEN \

-v /var/run/docker.sock:/docker.sock \

jupyter/tmpnb python orchestrate.py --image='waituck/custom\_nb' --pool\_size=10 --command="ipython notebook --NotebookApp.base\_url={base\_path} --ip=0.0.0.0 --port {port}"

1. Access the final tmpnb server with the following url:

**http://<Public IP>:8000**  
**Replace <Public IP> with your public IP that can be found below**



Finally, if you wish to rerun once you shut down your amazon instance, here are the commands you have to enter:

* 1. Restart Docker

sudo service docker restart

* 1. Remove the previously started docker instances  
     sudo docker rm proxy  
     sudo docker rm tmpnb
  2. Set the Environment Variable

export TOKEN=$( head -c 30 /dev/urandom | xxd -p )

* 1. Start the Docker Instances

sudo docker run --net=host -d -e CONFIGPROXY\_AUTH\_TOKEN=$TOKEN --name=proxy jupyter/configurable-http-proxy --default-target <http://127.0.0.1:9999>

sudo docker run --net=host –-name=tmpnb -d -e CONFIGPROXY\_AUTH\_TOKEN=$TOKEN \

-v /var/run/docker.sock:/docker.sock \

jupyter/tmpnb python orchestrate.py --image='waituck/custom\_nb' --pool\_size=10 --command="ipython notebook --NotebookApp.base\_url={base\_path} --ip=0.0.0.0 --port {port}"

1. Access the final tmpnb server with the following url:

**http://<Public IP>:8000**

\*NOTE: the mem\_limit of teach docker container in the tmpnb server may be modified, as wella s the pool\_size.

Refer to <https://github.com/jupyter/tmpnb> for the full list of command line options.