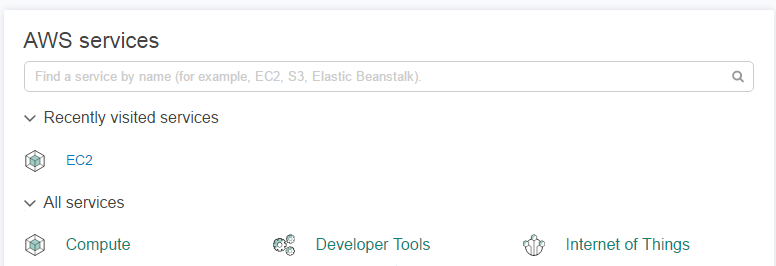
AWS Instance launch

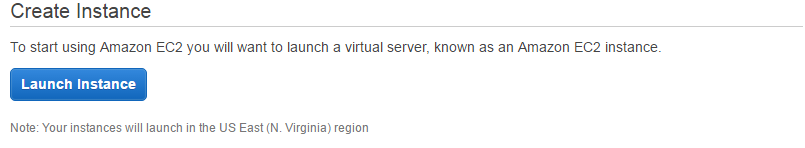
1. Sign-in to our account.



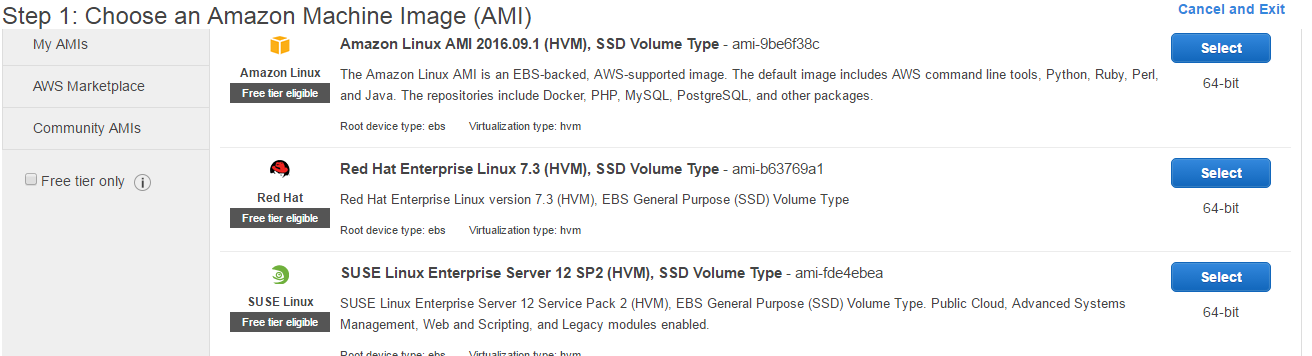
1. Launch an EC2 Instance by clicking EC2.



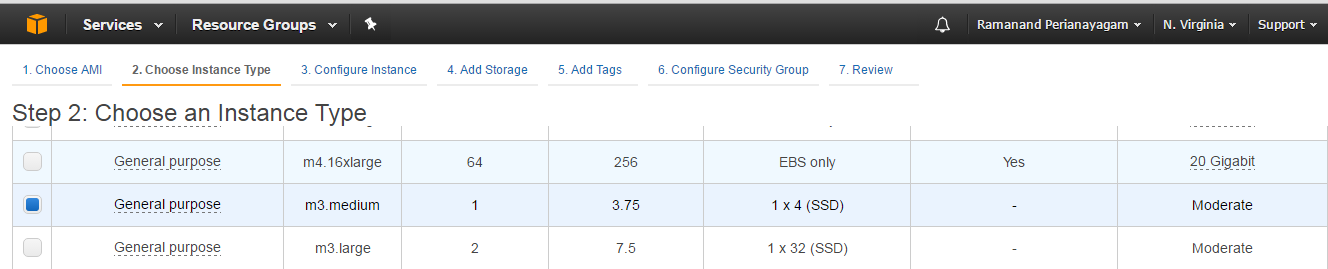
1. Launch instance by selecting “**Launch Instance**”.



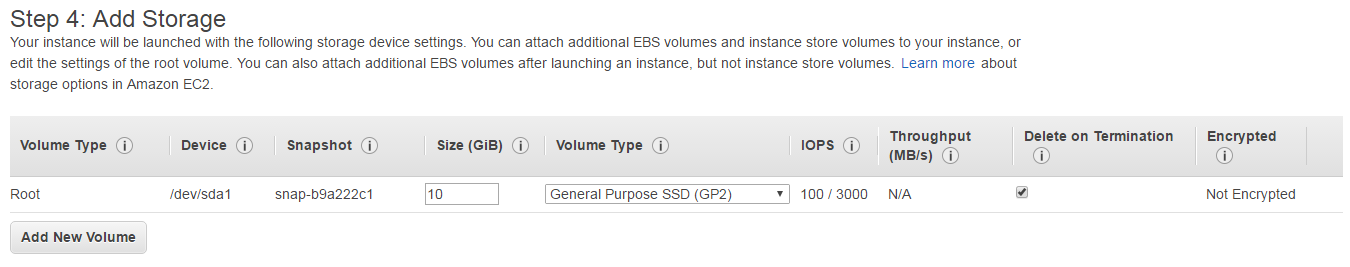
1. Select any instance of our choice and goto Next.



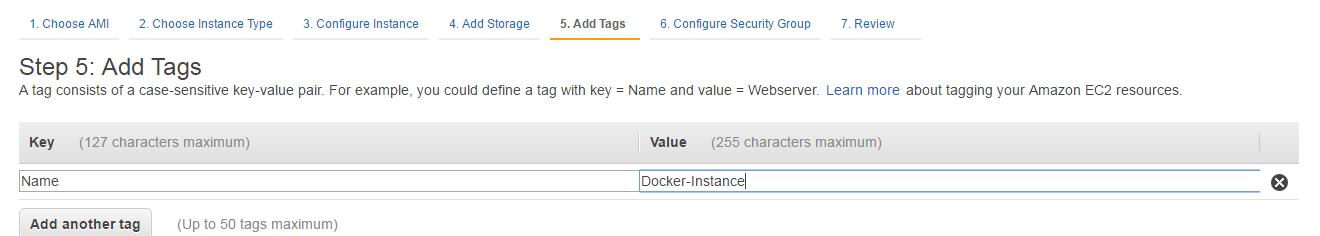
1. Choose the size of the instance and goto Next.



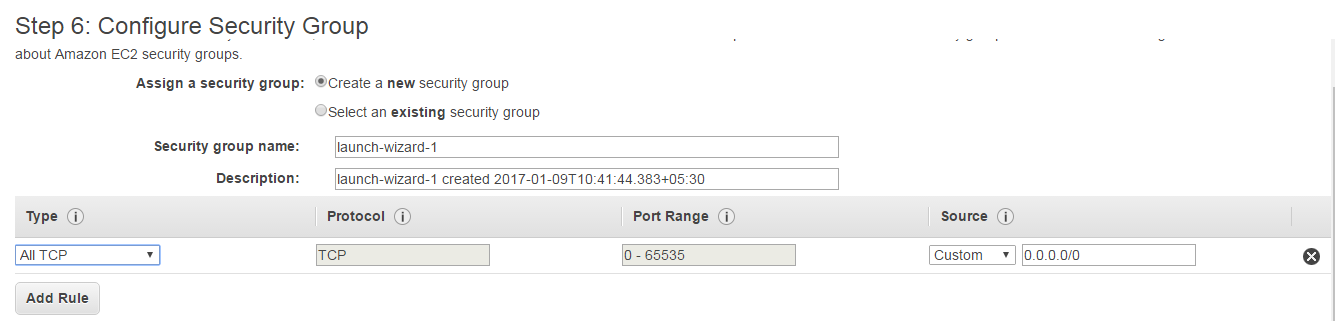
1. Select storage size and goto Next.



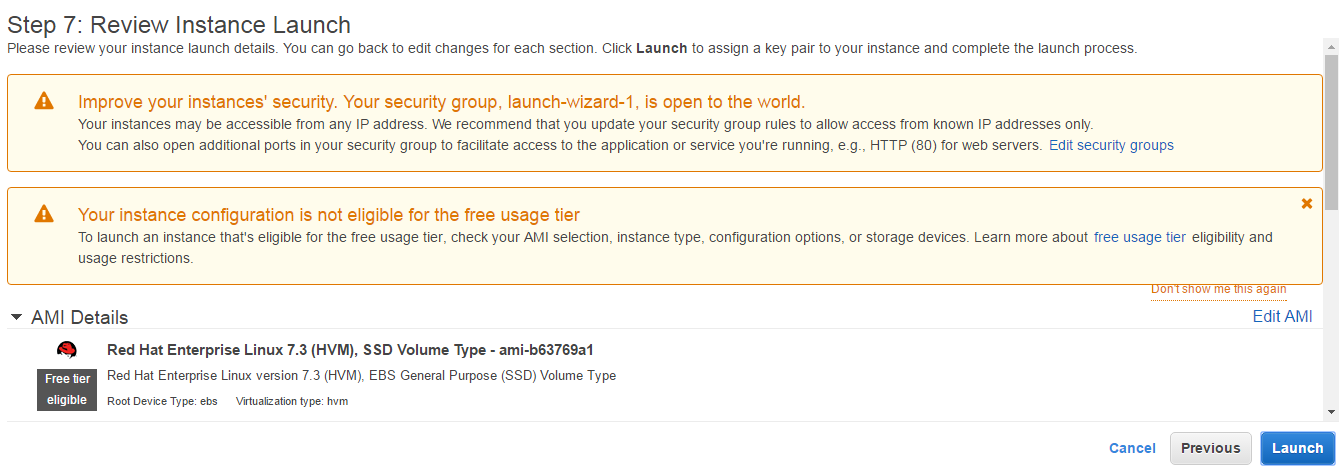
1. Give a name for your Instance and goto Next.



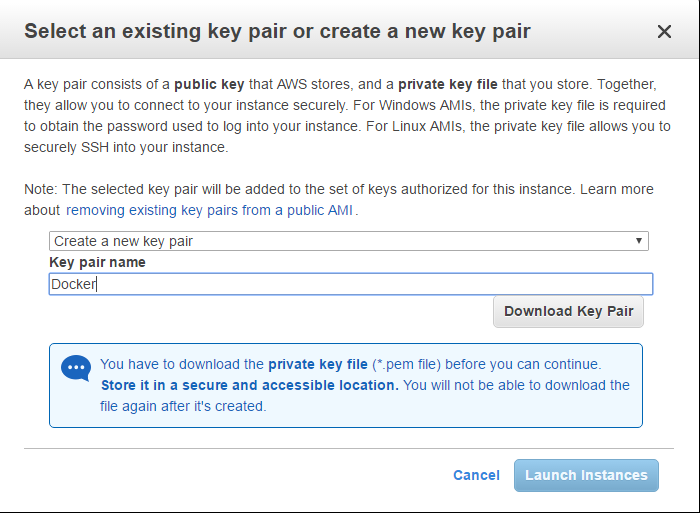
1. Create a new security group or use existing.(here, using a new security group) and goto Next.



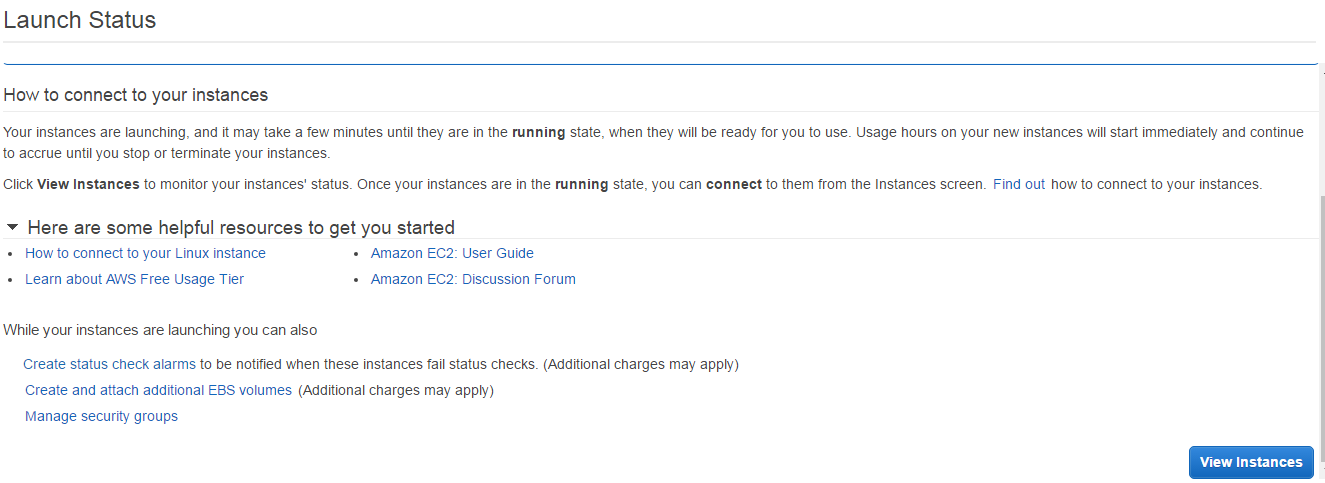
1. Check your configurations and click launch.



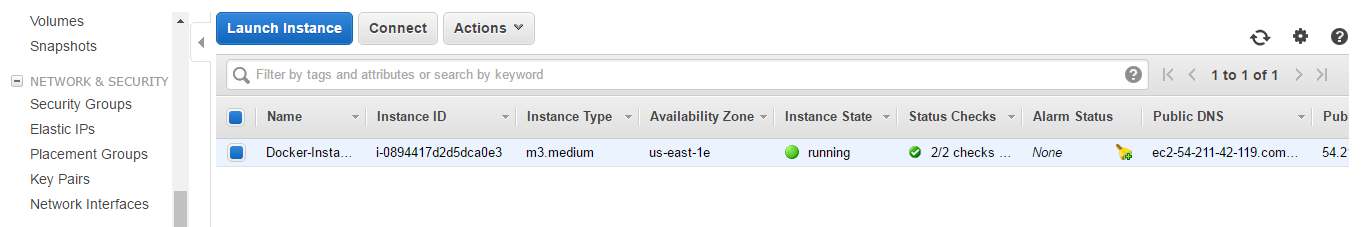
1. Create a new key-pair, download the new key-pair. The key –pair will be downloaded as **.pem** file. Save the file. Then click on launch instance



1. Click on **view instances** to know the status of the instance launched.

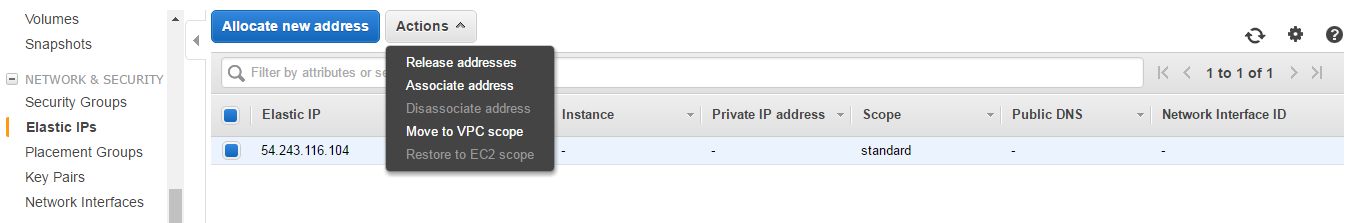


1. We can see the instance launched in the dashboard.

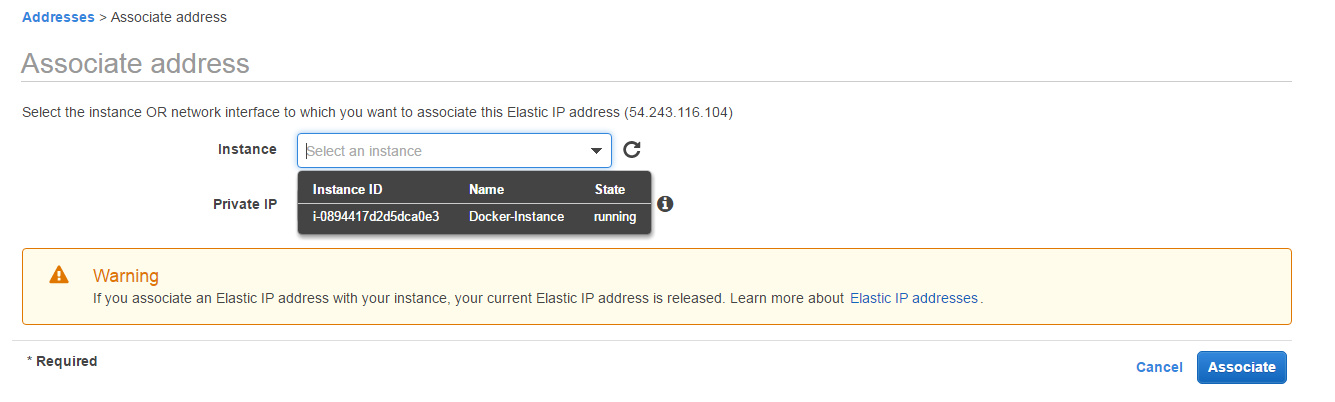


Note: Due to internet access issues we are using an Elastic Ip which will be our machine ip.

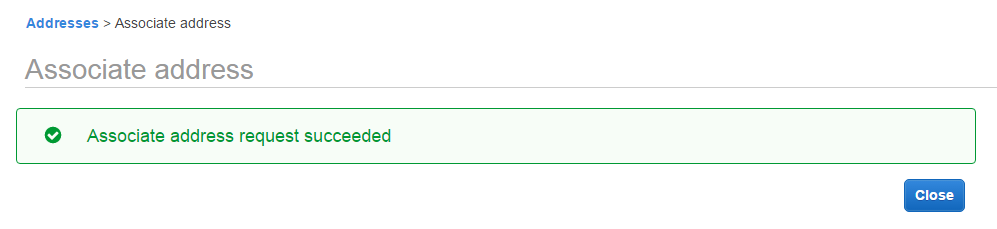
1. Elastic Ip configuration(Optional).
2. Under Network and Security, select Elastic Ips.
3. We are using an existing elastic ip.



1. Click on the elastic ip and under actions select Associate Address.
2. New screen will appear, click on **Instance** text box, it will list the running instances. Select the instance and click **Associate.**

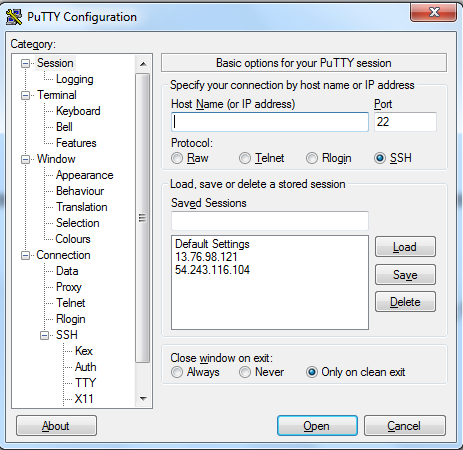


1. Success message shows the elastic ip is configured with the instance ip. We can now access our instance using this elastic ip.

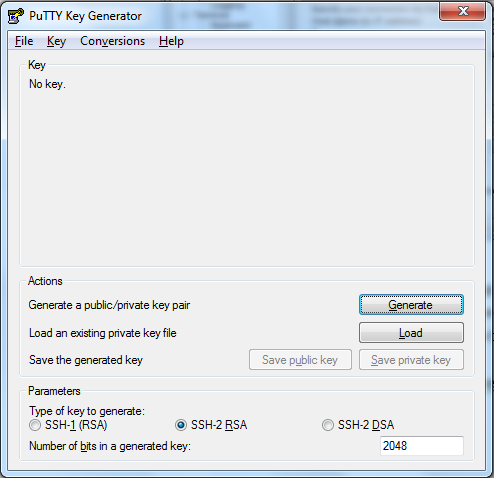


Access the Instance using Putty

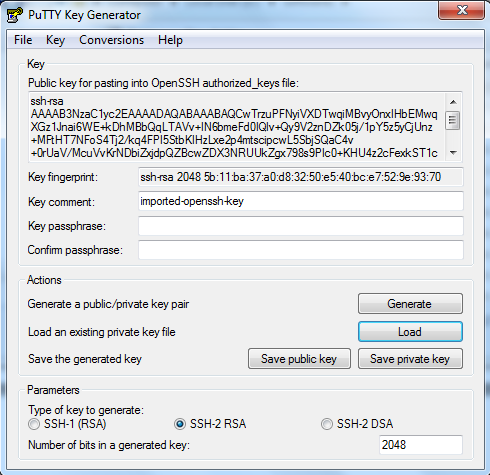
1. Open the Putty tool. In the host name give the elastic ip we configured, here it is 54.243.116.104.



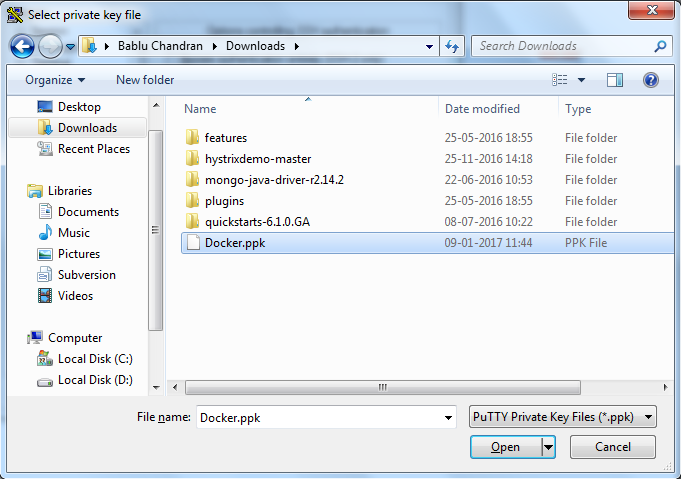
1. Then, under **SSH** click **Auth,** a newscreen will appear, it will ask for **.ppk** file. As of now we don’t have .ppk file, we have to create one using the .pem file we saved before. Follow the steps…
2. Open the **puttygen** tool, if you don’t have one , download it.



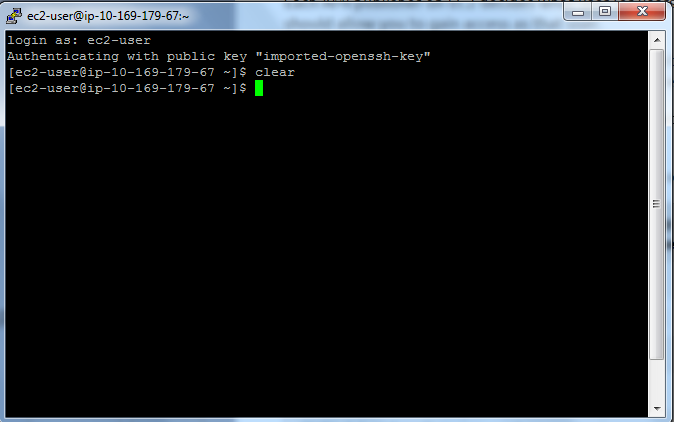
1. Load the .pem file and click **save private key.** A .ppk file will be created.



1. Coming back to putty, choose the .ppk file and click open.



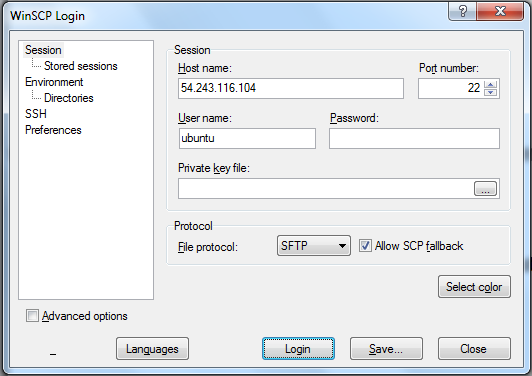
1. Give the credentials for the new linux machine.



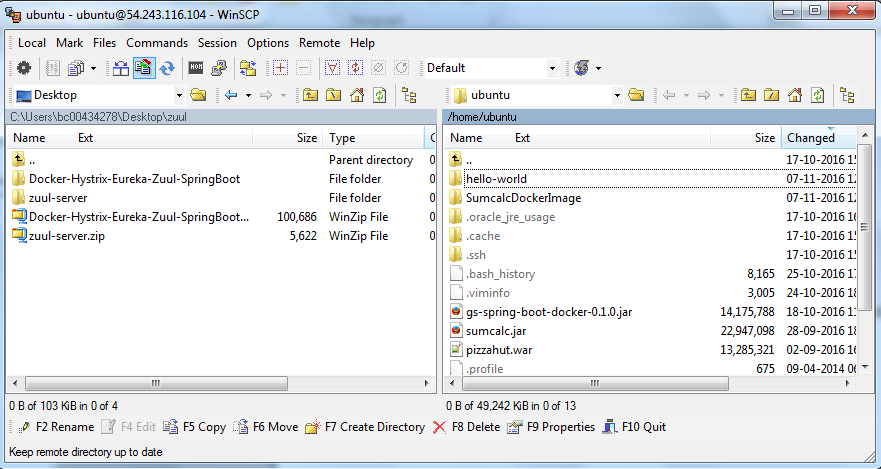
The new linux machine is now ready to use.

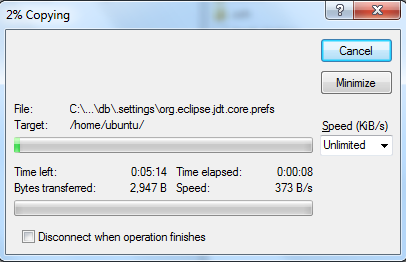
Moving file into the New Instance

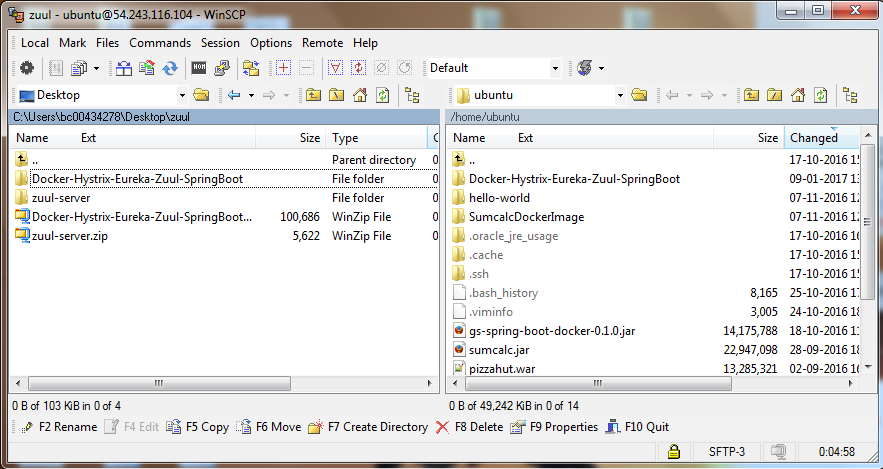
1. Open the **winscp** tool and give the credentials of the new instance .



1. Simply drag the file/folder from our local machine to the newly launched instance.

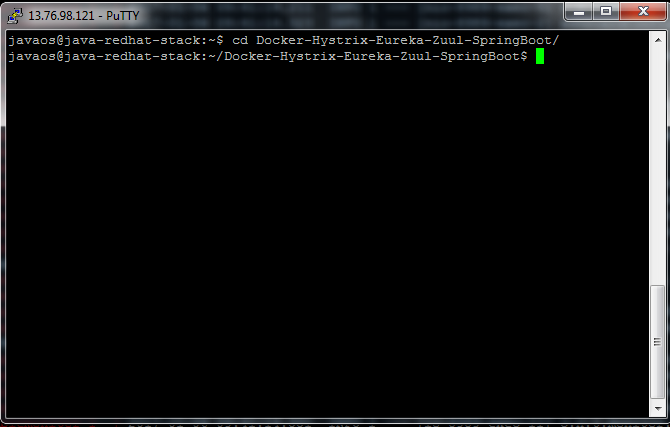






1. Then, using the putty tool move to the specific location where we copied our application. Here it is in **/home/javaos/techmDemo/Docker-Hystrix-Eureka-Zuul-SpringBoot**

|  |
| --- |
| cd /home/javaos/techmDemo/Docker-Hystrix-Eureka-Zuul-SpringBoot |



1. Run the application.

Note: If java\_home and maven\_home not set, set using …

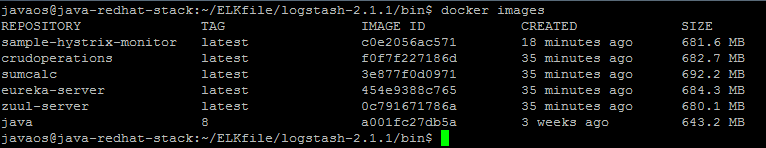
|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | export JAVA\_HOME=/home/javaos/softwares/jdk1.8.0\_111/ | | export M2\_HOME=/home/javaos/softwares/apache-maven-3.3.9 | | export PATH=/home/javaos/softwares/jdk1.8.0\_111/bin:/home/javaos/softwares/apache-maven-3.3.9/bin:$PATH | |

Run the application using command

|  |
| --- |
| mvn clean install |

1. After running the application check whether all the docker images are build.

|  |
| --- |
| docker images |



1. Create a **docker-compose.yml** file in any of the folder, here its in **/home/javaos/techmDemo/Docker-Hystrix-Eureka-Zuul-SpringBoot** folder.

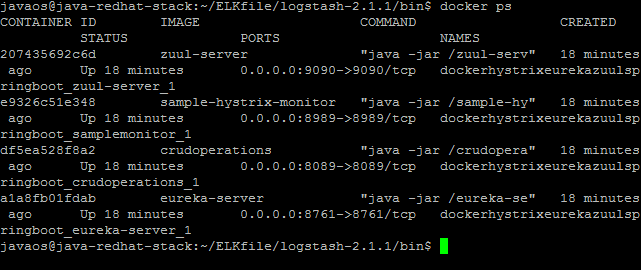
**docker-compose.yml**

|  |
| --- |
| eureka-server:  image: eureka-server  ports:  - "8761:8761"  db-operation-service:  image: db-operation-service  hostname: db-operation-service  links:  - eureka-server  environment:  EUREKA\_HOST: eureka-server  EUREKA\_PORT: 8761  SERVER\_PORT: 9099  SPRING\_PROFILES\_ACTIVE: docker  ports:  - "9099:9099"  db:  image: db  hostname: db  links:  - eureka-server  environment:  EUREKA\_HOST: eureka-server  EUREKA\_PORT: 8761  SERVER\_PORT: 8089  SPRING\_PROFILES\_ACTIVE: docker  ports:  - "8089:8089" |

|  |
| --- |
| samplemonitor:  image: sample-hystrix-monitor  hostname: samplemonitor  links:  - eureka-server  # - sampleservice  - db  - db-operation-service  environment:  EUREKA\_HOST: eureka-server  EUREKA\_PORT: 8761  SERVER\_PORT: 8989  SPRING\_PROFILES\_ACTIVE: docker  ports:  - "8989:8989"  zuul-server:  image: zuul-server  hostname: zuul-server  links:  - eureka-server  # - sampleservice  - db  - db-operation-service  environment:  EUREKA\_HOST: eureka-server  EUREKA\_PORT: 8761  SERVER\_PORT: 9090  SPRING\_PROFILES\_ACTIVE: docker  ports:  - "9090:9090" |

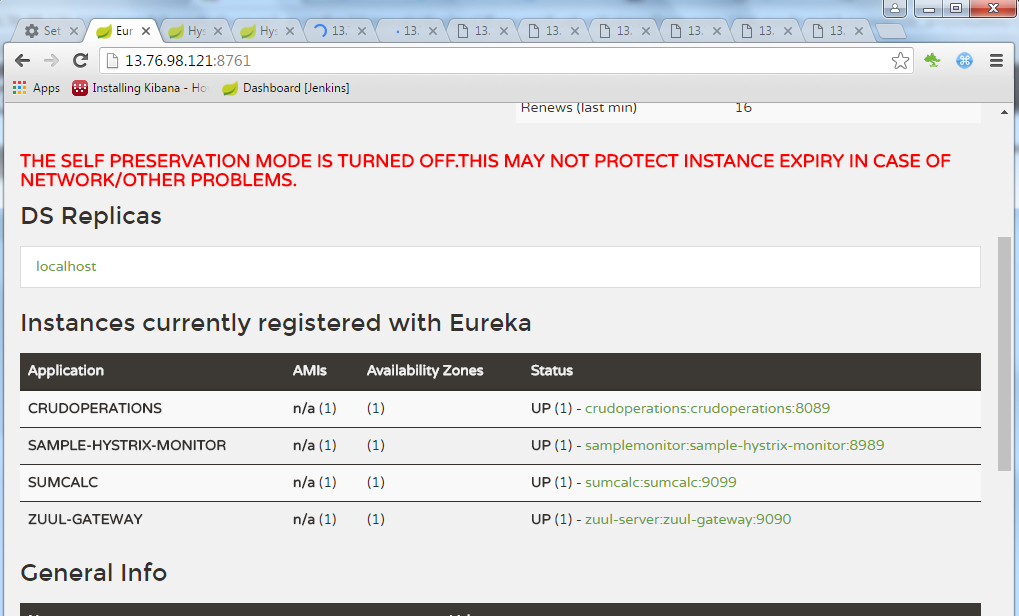
1. Wait for a few minutes for all the applications to start and check using …

|  |
| --- |
| docker ps |



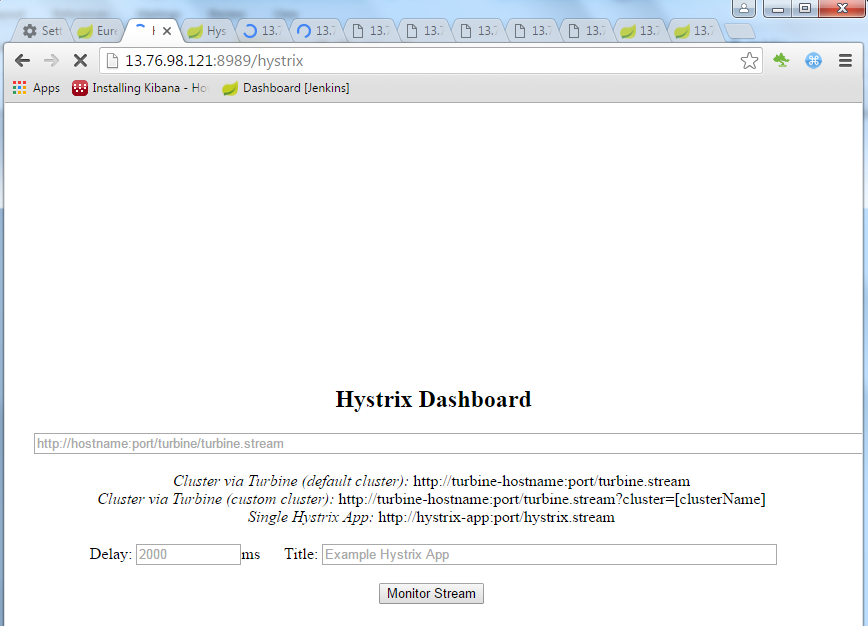
1. After all the application starts , access each application in the bowser
2. **Eureka dashboard**

|  |
| --- |
| [**http://13.76.98.121:8761/**](http://13.76.98.121:8761/) |

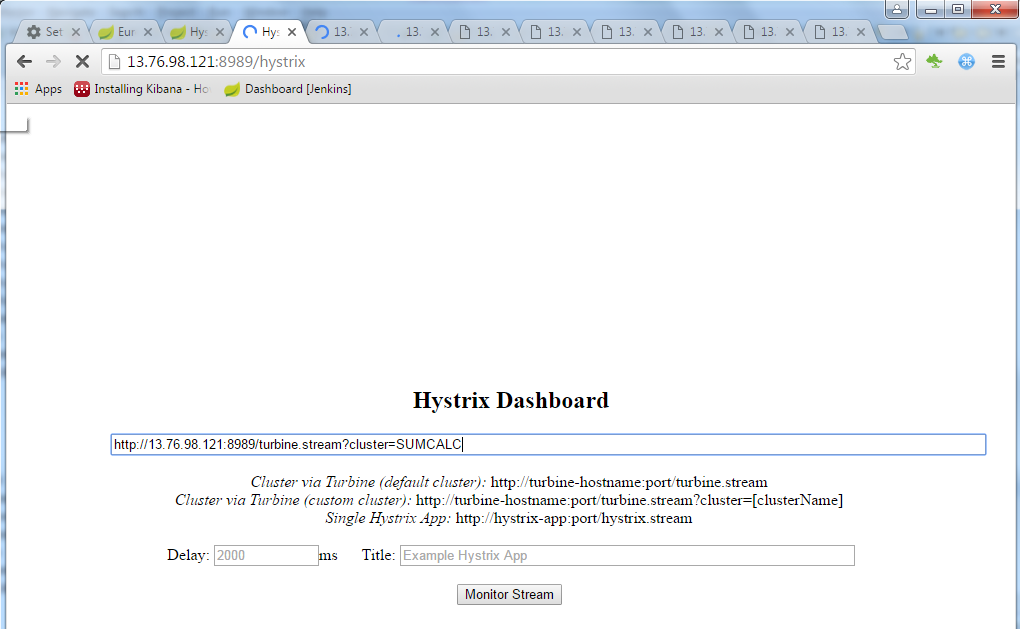


1. **Hystrix dashboard**

|  |
| --- |
| [**http://13.76.98.121:8989/hystrix/**](http://13.76.98.121:8989/hystrix/) |



**For DB-OPERATION-SERVICE**



Hit the services

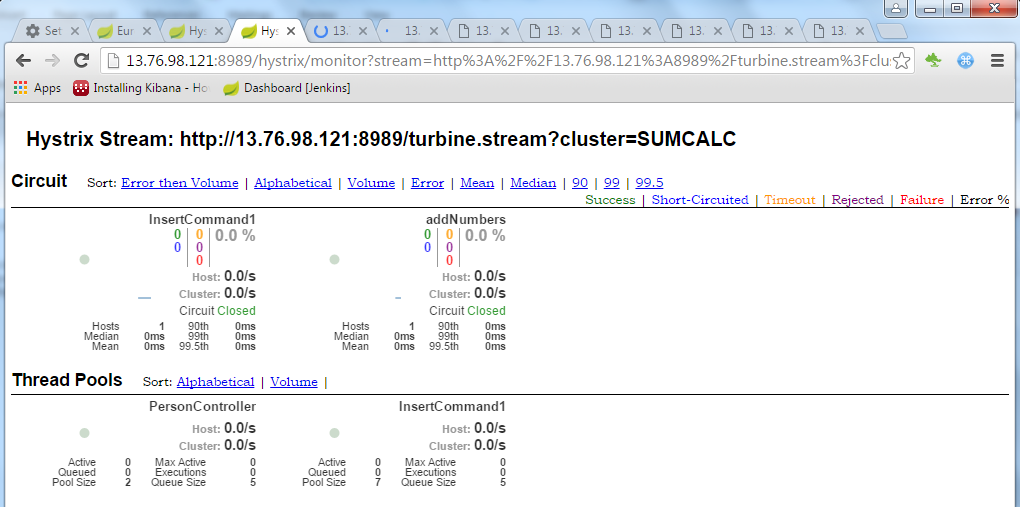
1. Insert command

|  |
| --- |
| [**http://13.76.98.121:9099/person/insert**](http://13.76.98.121:9099/person/insert) |

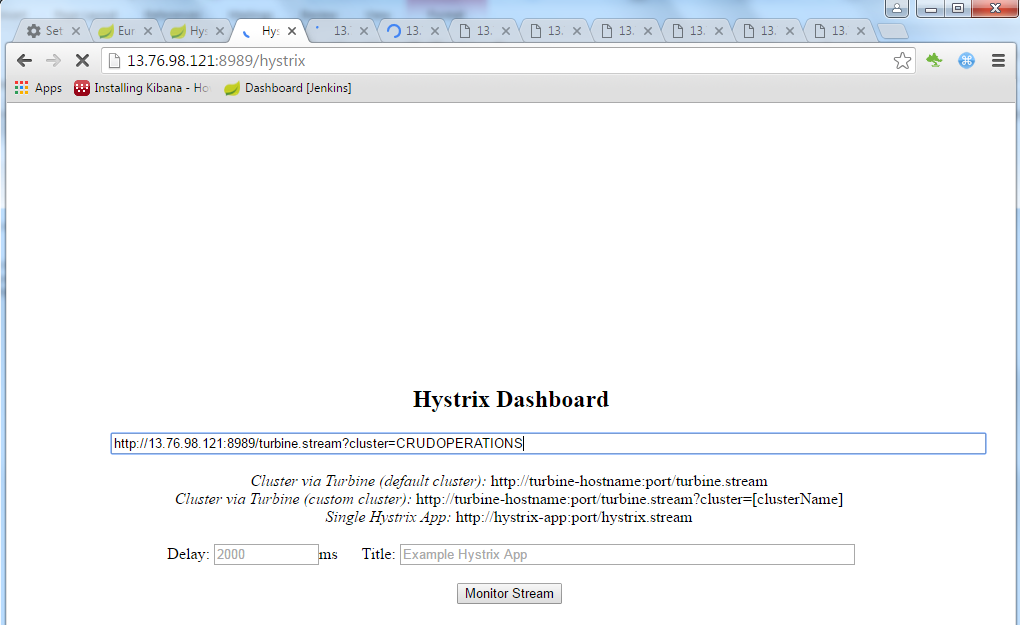
1. Add numbers

|  |
| --- |
| [**http://13.76.98.121:9099/person/sum?i=10&j=20**](http://13.76.98.121:9099/person/sum?i=10&j=20) |

**Output**



**For DB**



Hit the services

1. Insert command

|  |
| --- |
| [**http://13.76.98.121:8089/insert**](http://13.76.98.121:8089/insert) |

1. Update command

|  |
| --- |
| [**http://13.76.98.121:8089/update**](http://13.76.98.121:8089/update) |

1. Delete command

|  |
| --- |
| [**http://13.76.98.121:8089/delete**](http://13.76.98.121:8089/delete) |

Check the hystrix dashboard for the output

**Output:**

