AWARE DASHBOARD SETUP INSTRUCTIONS ON AWS

Version 1.1

TABLE OF CONTENTS

Revision History

Date	Version	Description	Author
9/10/2018	1.0	Initial Version of the Document	Abhijit
4/21/2021	1.1	Updates for Ubuntu 18	D. Bellew

1 INTRODUCTION

The purpose of this document is to outline the install steps followed to setup the aware dashboard so that UPenn can test out the aware framework. Due to network Issues faced in trying to setup this install in-house, a decision was taken to try and install the dashboard on AWS. The online guide located at http://www.awareframework.com/hosting-your-own-aware-dashboard/ was used as a basis for the install and this document

1.1 REFERENCES

The following resources were leveraged to complete the install steps and test out the deployment

- Sharath Chandra Guntuku
- Salvatore Giorgi
- http://www.awareframework.com/hosting-your-own-aware-dashboard/

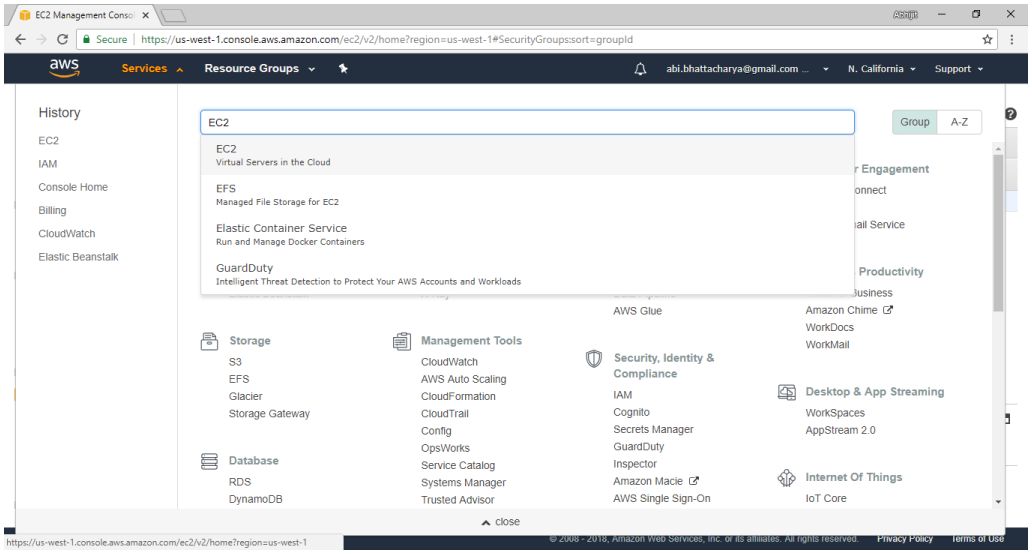
2 SETTING UP THE SERVER ON AWS

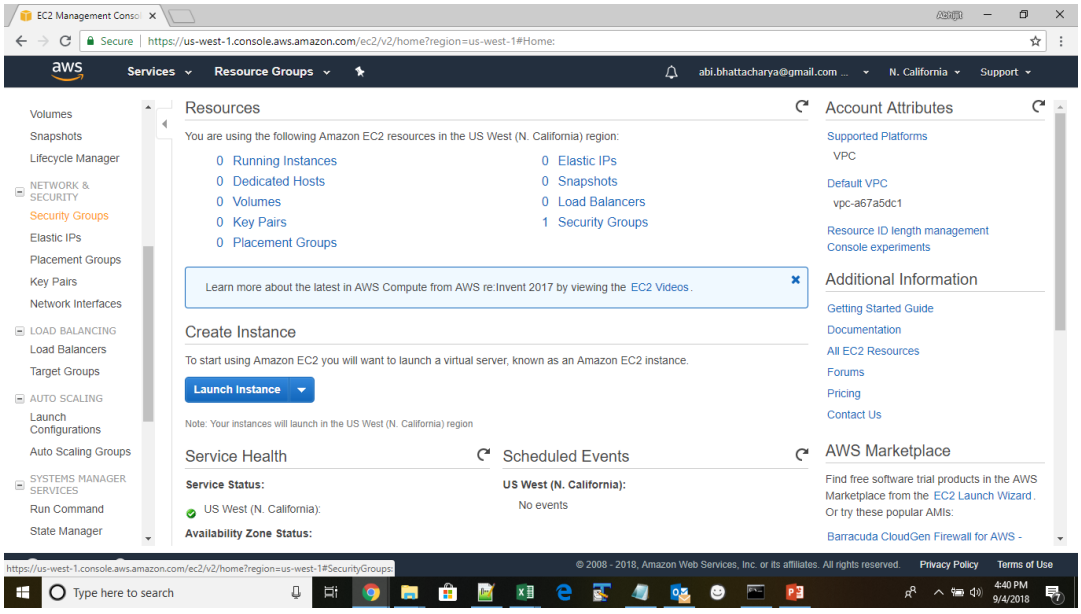
An Ubuntu 14.04 server was setup on AWS using the steps below.

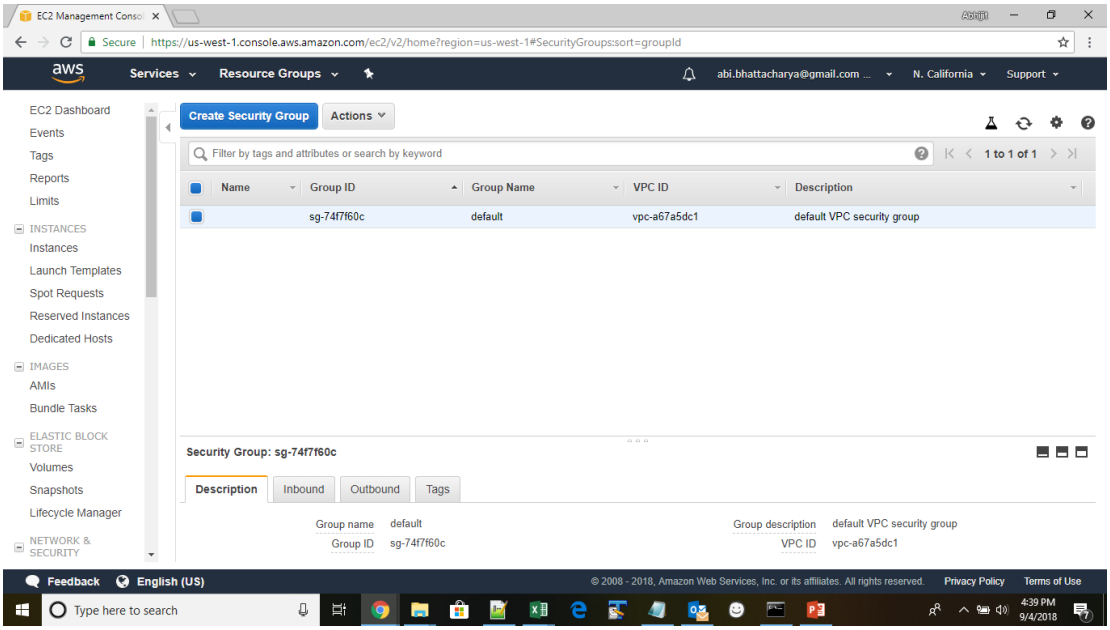
- DB: Notes - Changes for this document are for updates fo Ubuntu 18.4
- (“lsb_release -a” to show Ubuntu version)

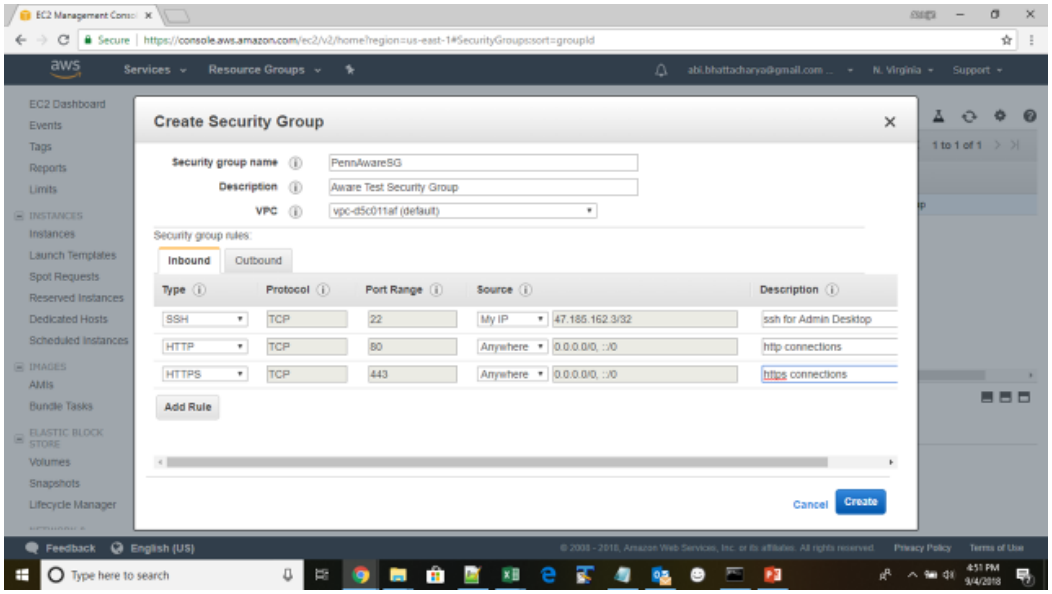
2.1 CREATE SECURITY GROUP

A security group was created by going to the EC2 dashboard in the N. Virginia Region of AWS


EC2 Management Consol
Secure | https://us-west-1.console.aws.amazon.com/ec2/v2/home?region=us-west-1#SecurityGroups:sort=groupId
aws
Services
Resource Groups
abi.bhattacharya@gmail.com ...
N. California
Support
History
EC2
IAM
Console Home
Billing
CloudWatch
Elastic Beanstalk
EC2
Virtual Servers in the Cloud
EFS
Managed File Storage for EC2
Elastic Container Service
Run and Manage Docker Containers
GuardDuty
Intelligent Threat Detection to Protect Your AWS Accounts and Workloads
Group
A-Z
Storage
S3
EFS
Glacier
Storage Gateway
Management Tools
CloudWatch
AWS Auto Scaling
CloudFormation
CloudTrail
Config
OpsWorks
Service Catalog
Systems Manager
Trusted Advisor
Security, Identity & Compliance
IAM
Cognito
Secrets Manager
GuardDuty
Inspector
Amazon Macie
AWS Single Sign-On
AWS Glue
Desktop & App Streaming
WorkSpaces
AppStream 2.0
Amazon Chime
WorkDocs
WorkMail
Database
RDS
DynamoDB
Internet Of Things
IoT Core
close
https://us-west-1.console.aws.amazon.com/ec2/v2/home?region=us-west-1
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


EC2 Management Consol
Secure | https://us-west-1.console.aws.amazon.com/ec2/v2/home?region=us-west-1#Home:
aws
Services
Resource Groups
abi.bhattacharya@gmail.com ...
N. California
Support
Volumes
Snapshots
Lifecycle Manager
NETWORK & SECURITY
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
LOAD BALANCING
Load Balancers
Target Groups
AUTO SCALING
Launch Configurations
Auto Scaling Groups
SYSTEMS MANAGER SERVICES
Run Command
State Manager
Resources
You are using the following Amazon EC2 resources in the US West (N. California) region:
0 Running Instances
0 Dedicated Hosts
0 Volumes
0 Key Pairs
0 Placement Groups
0 Elastic IPs
0 Snapshots
0 Load Balancers
1 Security Groups
Learn more about the latest in AWS Compute from AWS re:Invent 2017 by viewing the EC2 Videos.
Create Instance
To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.
Launch Instance
Note: Your instances will launch in the US West (N. California) region
Service Health
Service Status:
US West (N. California):
Availability Zone Status:
Scheduled Events
US West (N. California):
No events
Account Attributes
Supported Platforms
VPC
Default VPC
vpc-a67a5dc1
Resource ID length management
Console experiments
Additional Information
Getting Started Guide
Documentation
All EC2 Resources
Forums
Pricing
Contact Us
AWS Marketplace
Find free software trial products in the AWS Marketplace from the EC2 Launch Wizard. Or try these popular AMIs:
Barracuda CloudGen Firewall for AWS -
https://us-west-1.console.aws.amazon.com/ec2/v2/home?region=us-west-1#SecurityGroups:
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Type here to search
4:40 PM
9/4/2018

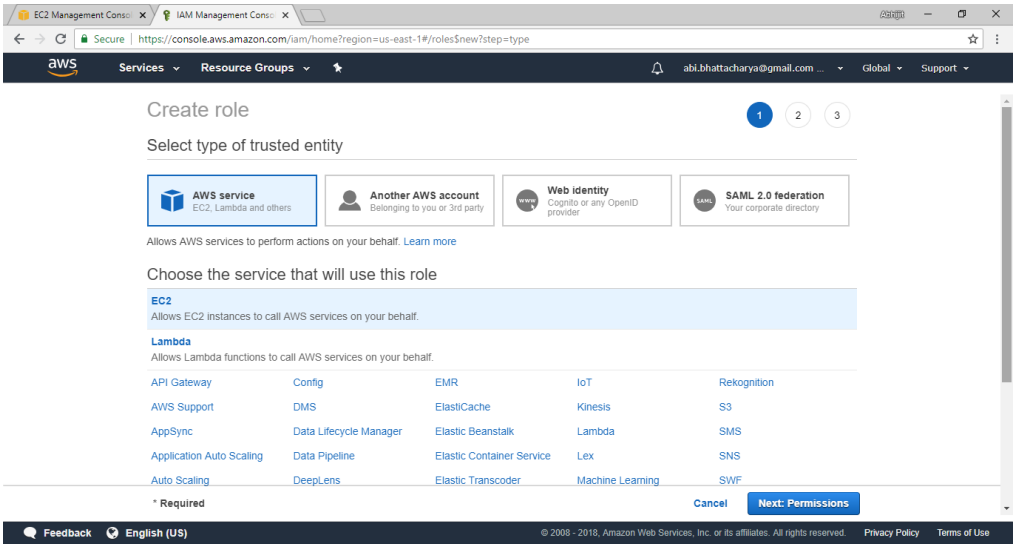


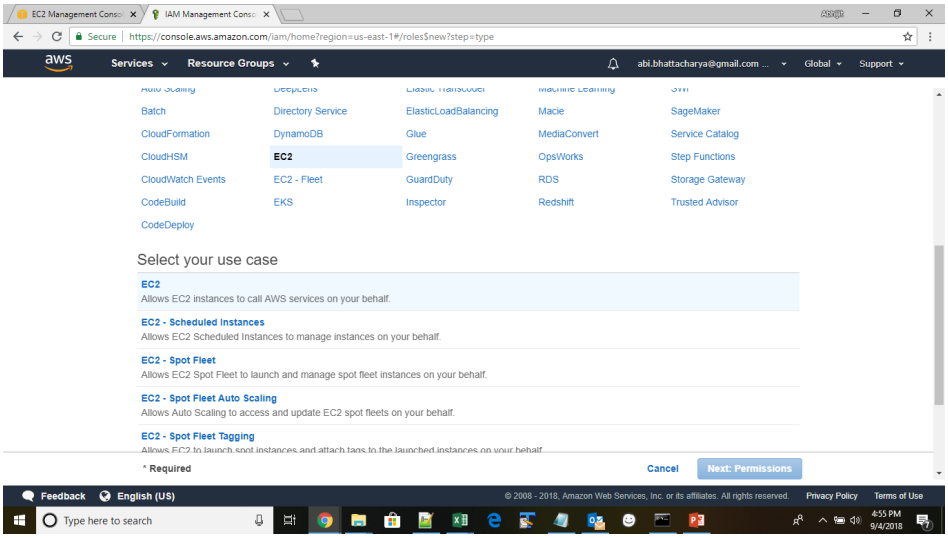


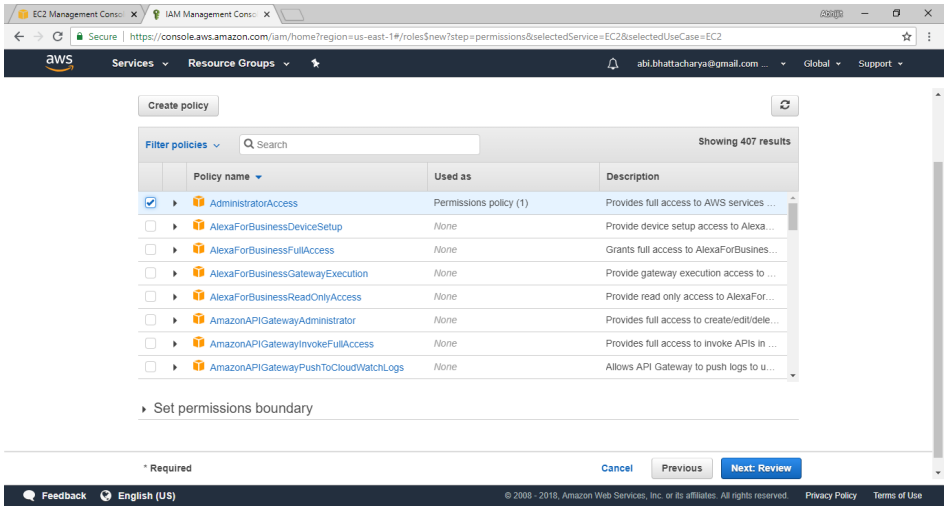
Select “Anywhere” in the source for SSH to allow anyone to SSH into the server and hit create to create the security group.

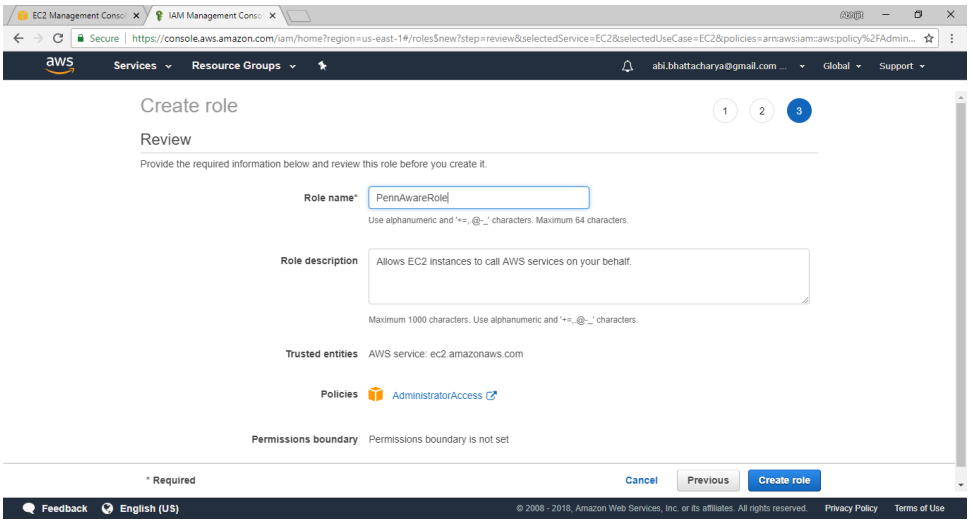
2.2 CREATE AN AWS ROLE

Create a new role with admin access policy to be able to launch a new instance. Roles can be created under IAM service in AWS


Secure | https://console.aws.amazon.com/iam/home?region=us-east-1#/roles$new?step=type
Services
Resource Groups
abi.bhattacharya@gmail.com ...
Global
Support
Create role
1
2
3
Select type of trusted entity
AWS service
EC2, Lambda and others
Another AWS account
Belonging to you or 3rd party
Web identity
Cognito or any OpenID provider
SAML 2.0 federation
Your corporate directory
Allows AWS services to perform actions on your behalf. Learn more
Choose the service that will use this role
EC2
Allows EC2 instances to call AWS services on your behalf.
Lambda
Allows Lambda functions to call AWS services on your behalf.
API Gateway
Config
EMR
IoT
Rekognition
AWS Support
DMS
ElastiCache
Kinesis
S3
AppSync
Data Lifecycle Manager
Elastic Beanstalk
Lambda
SMS
Application Auto Scaling
Data Pipeline
Elastic Container Service
Lex
SNS
Auto Scaling
DeepLens
Elastic Transcoder
Machine Learning
SWF
* Required
Cancel
Next: Permissions
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

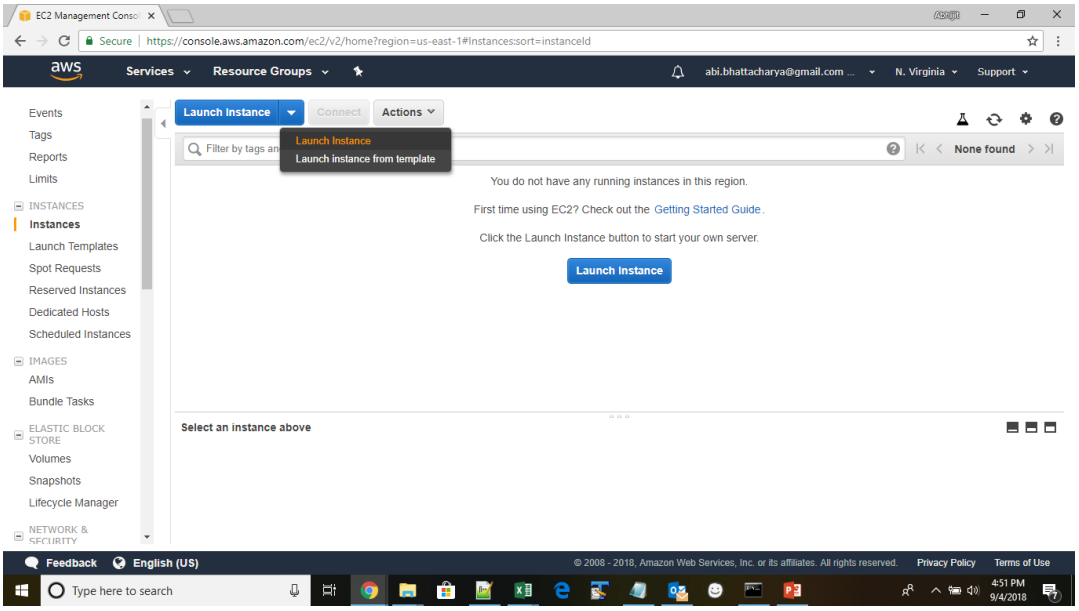

Secure | https://console.aws.amazon.com/iam/home?region=us-east-1#/roles$new?step=type
Services
Resource Groups
abi.bhattacharya@gmail.com ...
Global
Support
Batch
Directory Service
ElasticLoadBalancing
Macie
SageMaker
CloudFormation
DynamoDB
Glue
MediaConvert
Service Catalog
CloudHSM
EC2
Greengrass
OpsWorks
Step Functions
CloudWatch Events
EC2 - Fleet
GuardDuty
RDS
Storage Gateway
CodeBuild
EKS
Inspector
Redshift
Trusted Advisor
CodeDeploy
Select your use case
EC2
Allows EC2 instances to call AWS services on your behalf.
EC2 - Scheduled Instances
Allows EC2 Scheduled Instances to manage instances on your behalf.
EC2 - Spot Fleet
Allows EC2 Spot Fleet to launch and manage spot fleet instances on your behalf.
EC2 - Spot Fleet Auto Scaling
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.
EC2 - Spot Fleet Tagging
* Required
Cancel
Next: Permissions
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Type here to search
4:55 PM
9/4/2018

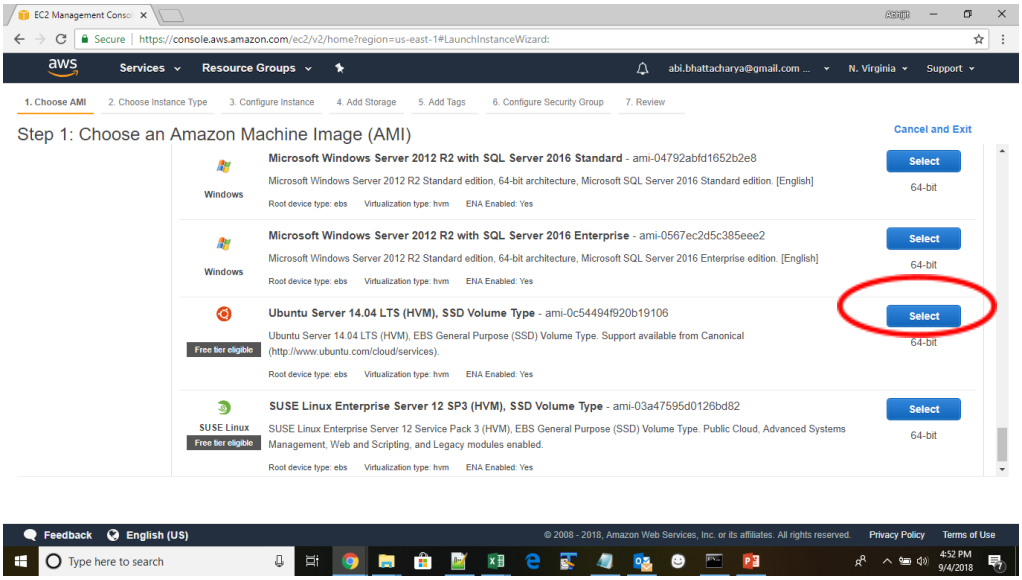




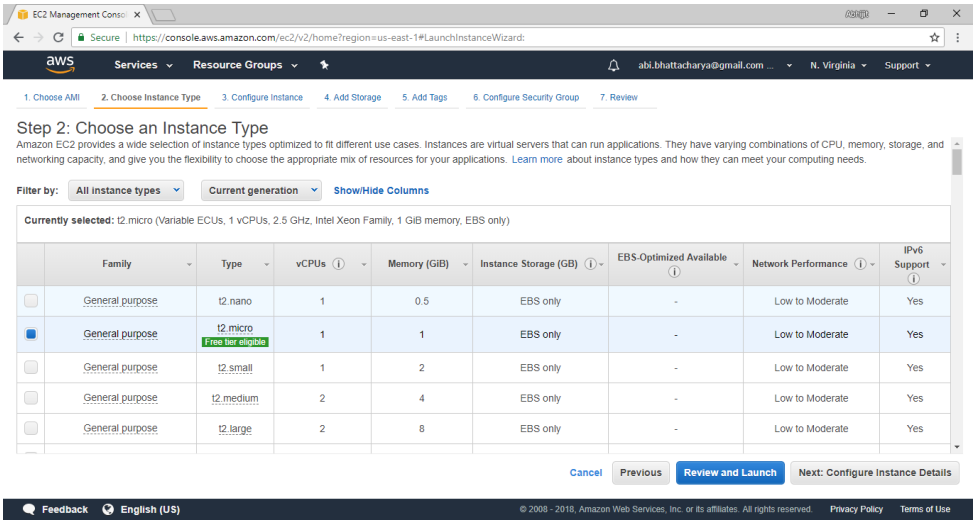
2.3 LAUNCH INSTANCE

Launch a new Ubuntu 14.04 instance using the role and security group created

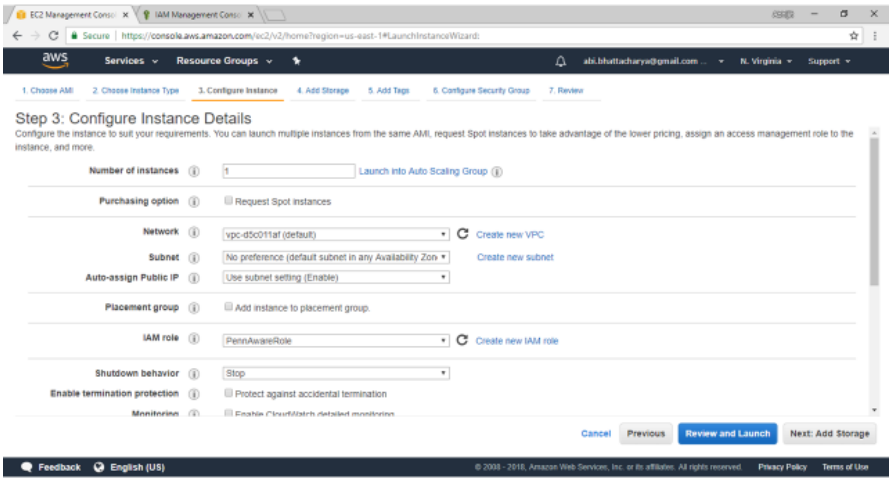




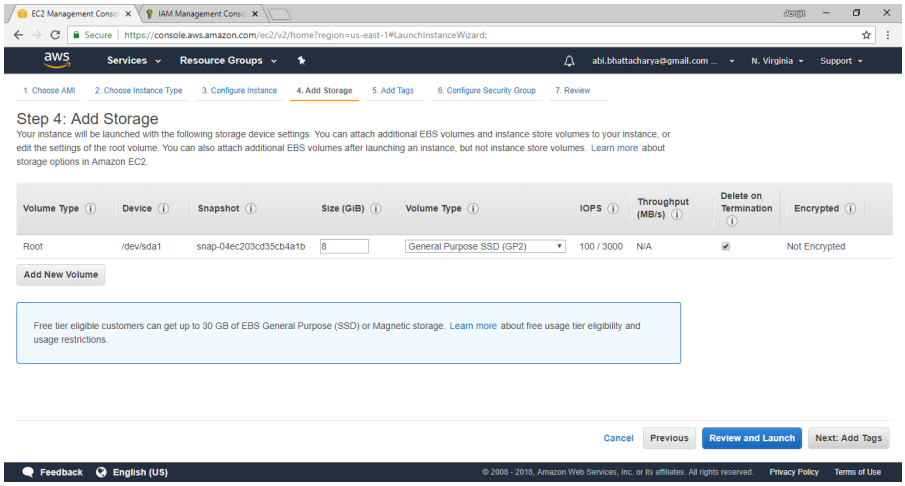
Choose a t2 micro instance



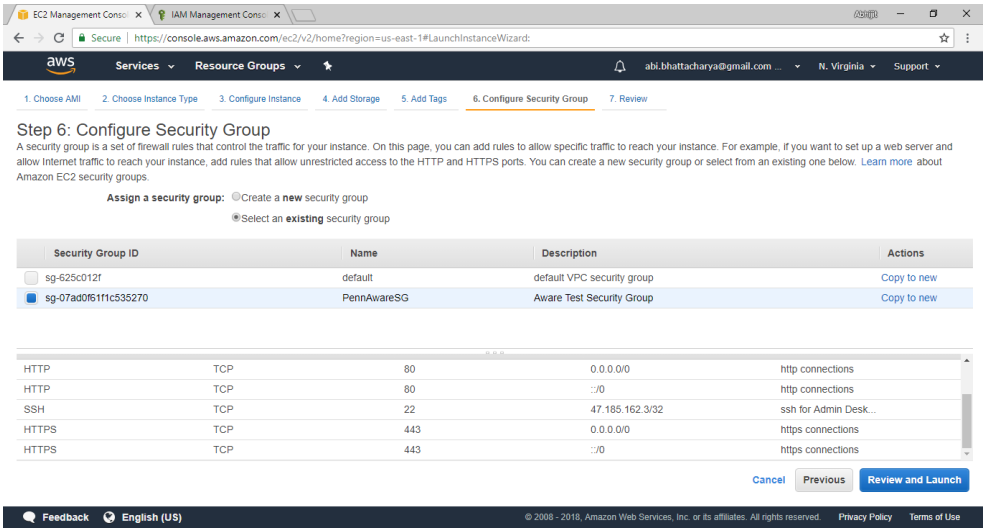
Use role created in instance details



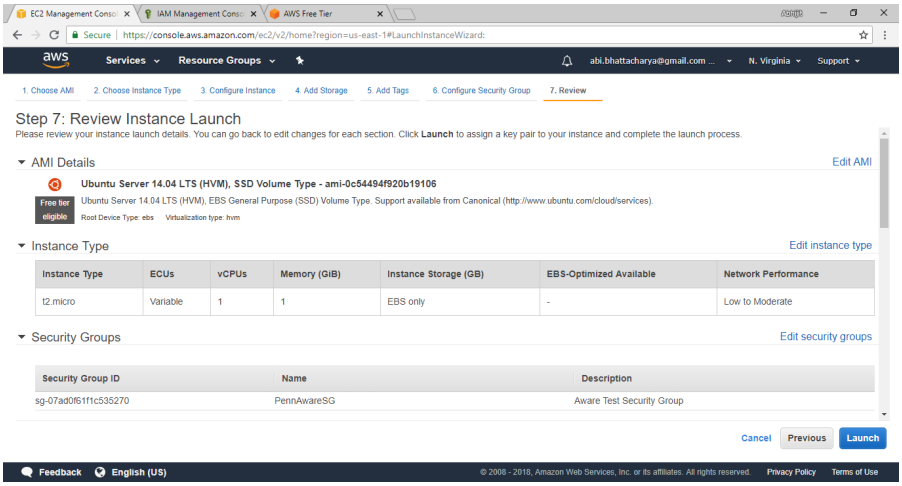
Change disk size to 16 GB

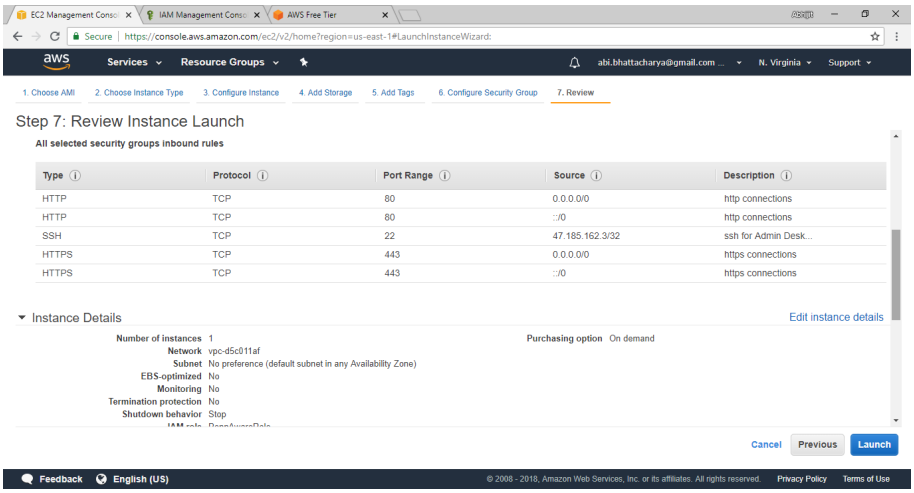


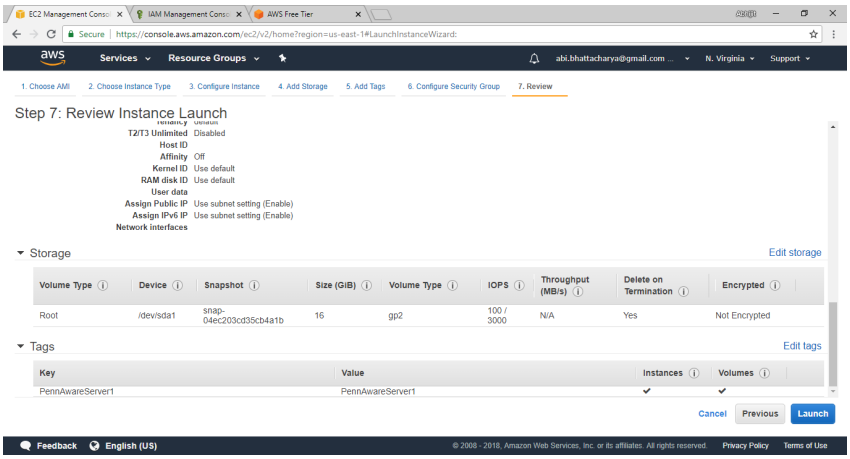
Pick the security group created previously



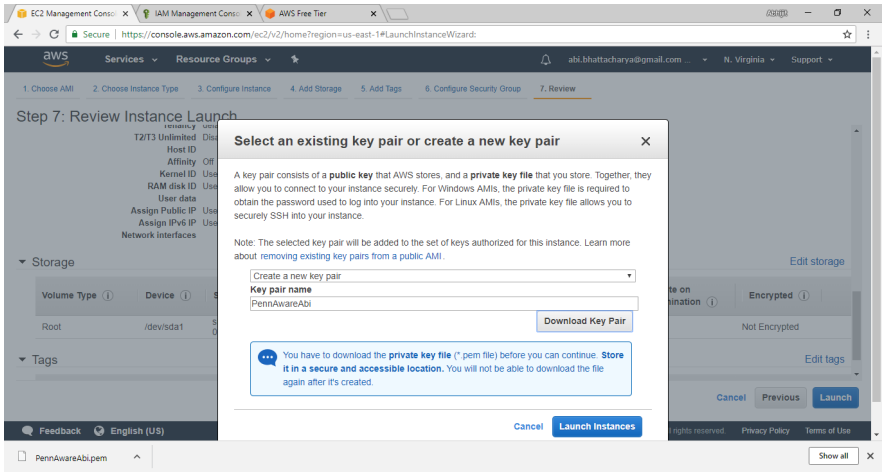
Launch the instance after review of settings

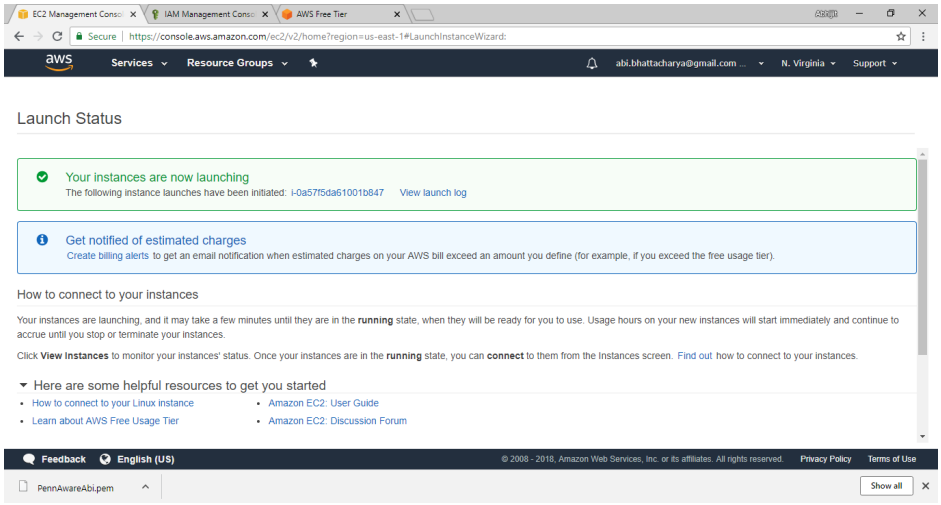






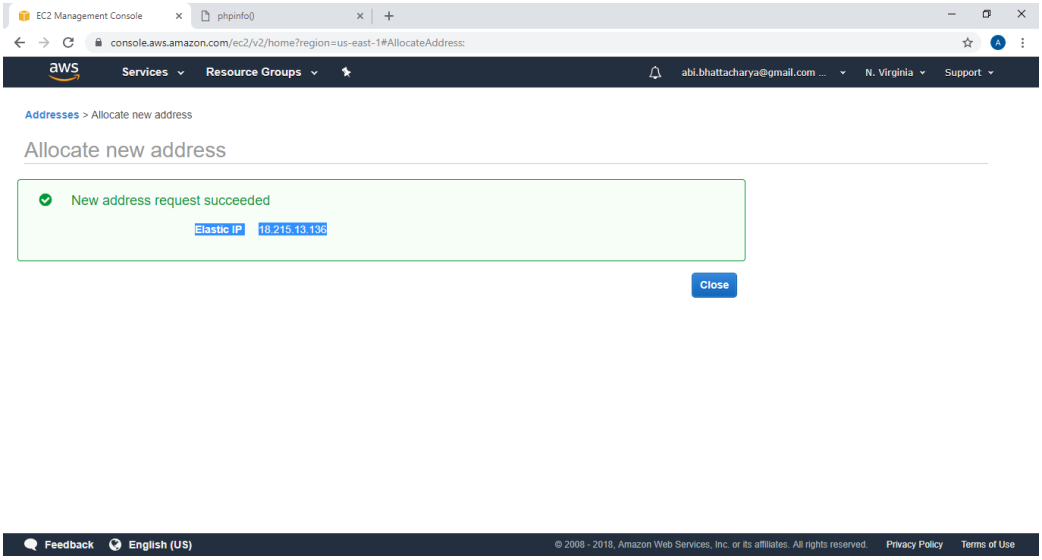
Name and download the key-pair



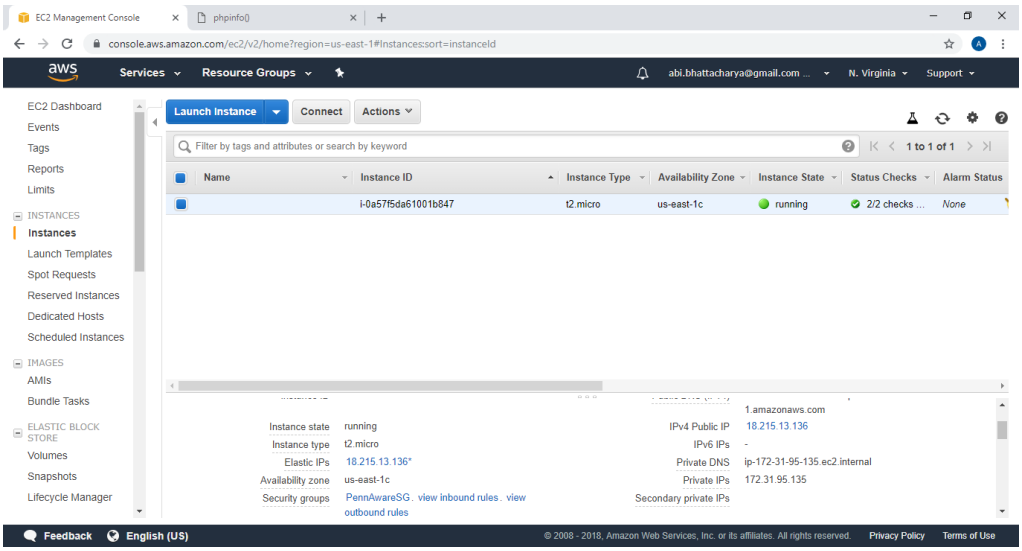


2.4 CREATE AND ASSIGN AN ELASTIC IP ADDRESS

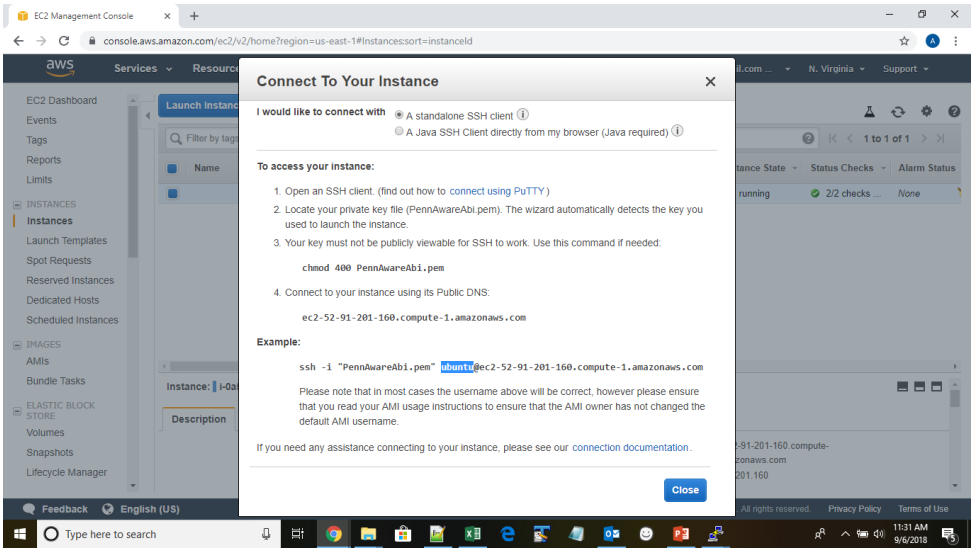
Create elastic IP address from the EC2 console and assign the IP address to the running instance

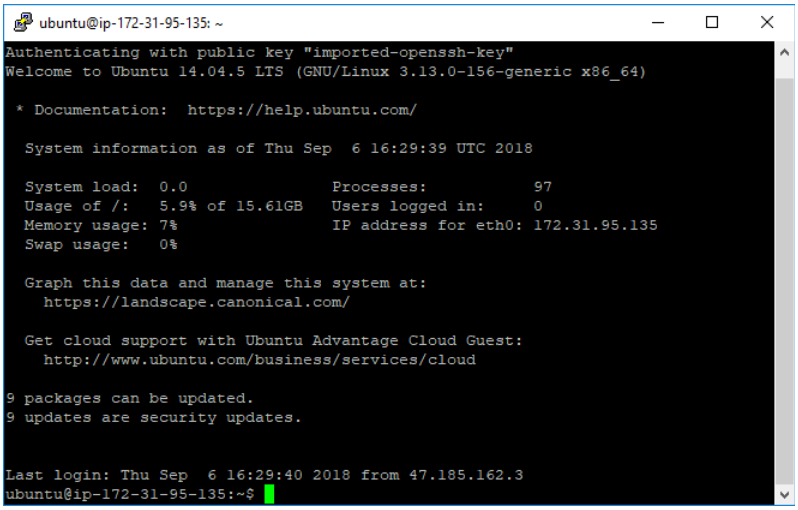


After assigning the IP address, get the public IP address from the instance details in the EC2 dashboard



Putty/connect via SSH to the server at the public IP address using the key pair to access the server via ssh. Hit the connect button next to the launch instance button to get the connection details to the instance.





The public IP address set up in this case was 18.215.13.136.

2.5 GET A NAME ASSIGNED TO THE IP ON THE DNS SERVER

An A name of aware-cloud was added to the wwbp.org domain and was pointed to the static elastic IP address of the server - 18.215.13.136

3 INSTALL REMAINING ELEMENT OF THE LAMP STACK

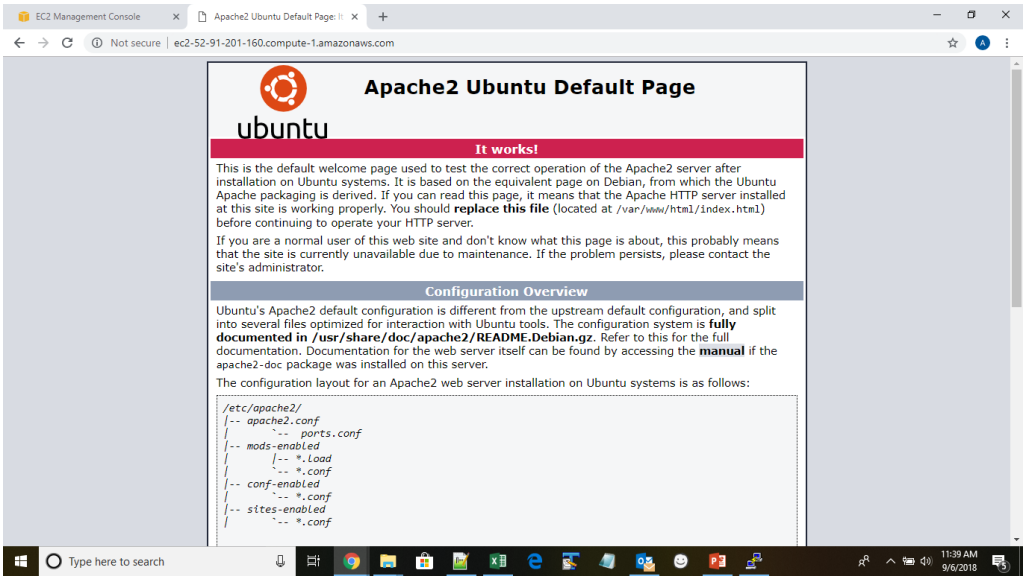
In addition to the Linux server, MySQL, apache and PHP modules were also installed to prepare for the aware dashboard server install. The following site was referenced for this section:

https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-14-04

3.1 INSTALL APACHE

Once, connected to the server via ssh, execute the following commands on the shell prompt to install apache:

- sudo apt-get update
- sudo apt-get install apache2
- Check for proper local installation by:
    - "curl http://localhost" (you should get back html text)
- If you don't know the external DNS ip (from connect screen of the EC2 dashboard), you can use:
    - "curl ipecho.net/plain ; echo"
- Check external access from a different machine browser and use
    - "http://<external DNS ip address>" (you should get back the below screen)



3.2 INSTALL MYSQL AND SETUP THE DATABASE

Run the following commands on the ssh terminal to install MySQL.

- “sudo apt-get update”
- “sudo apt-get dist-upgrade”
- “sudo add-apt-repository ppa:ondrej/php”
- “sudo apt-get update”
- “sudo apt-get install mysql-server php5.6-mysql”
- “sudo mysql_secure_installation -u root”

The following answers were given to questions on the secure Installation script:

Validate Password Plugin? N

New password: <enter new Password>

Remove anonymous users? [Y/n] Y

Disallow root login remotely? [Y/n] Y

Remove test database and access to it? [Y/n] Y

Reload privilege tables now? [Y/n] Y

All done!

- For some reason the above “New Password” doesn’t work, so we need to change the root password for real:
    - “sudo service mysql stop”
    - “sudo mkdir /var/run/mysqld”
    - “sudo chown mysql:mysql /var/run/mysqld”
    - “sudo mysqld_safe --skip-grant-tables --skip-networking &”
    - “mysql -u root mysql”
        - mysql> UPDATE mysql.user SET authentication_string=CONCAT('*', UPPER(SHA1(UNHEX(SHA1('<put password here>'))))), plugin='mysql_native_password' WHERE User='root' AND Host='localhost';
        - mysql> \q;
    - “sudo mysqladmin -S /var/run/mysqld/mysqld.sock shutdown”
    - “sudo service mysql start”

After installation test the installation using the following command at the ssh terminal

- “mysql -uroot -p”

Enter Password: <root password from above>

- mysql> exit

You can check the status of MySQL using the command

- “sudo service mysql status”

You can stop the service using

- “sudo service mysql stop”

You can start the service using

- “sudo service mysql start”

3.3 INSTALL PHP

Install PHP by running the following commands at the ssh terminal

- “sudo apt-get install php5.6 libapache2-mod-php5.6 php5.6-mcrypt”

4 SETTING UP THE AWARE DASHBOARD

4.1 PULLING DOWN THE LATEST VERSION OF THE AWARE SERVER

The following commands were executed at the ssh terminal to pull down the latest code of the aware server

- cd /var/www/html
- sudo git clone https://github.com/denzilferreira/aware-server.git
- cd /var/www/html/aware-server
- sudo git pull

4.2 CERTBOT AND SSL CERT INSTALL

The following commands were run at the ssh terminal:

- sudo apt-get update
- sudo apt-get install software-properties-common
- sudo add-apt-repository ppa:certbot/certbot
    - The PPA had been DEPRECIATED.
    - Press [ENTER] to continue [...] <ENTER>
- sudo apt-get update
- sudo apt-get install python-certbot-apache
- sudo certbot --apache

The last command automatically configures Apache and assigns the certificates to your host. When executed, the script asks a few questions. These were answered as indicated below:

- Enter email address (used for urgent renewal and security notices) (Enter 'c' to cancel): sal.giorgi@gmail.com

- Please read the Terms of Service at https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must agree in order to register with the ACME server at https://acme-v02.api.letsencrypt.org/directory

(A)gree/(C)ancel: A

- Would you be willing to share your email address with the Electronic Frontier Foundation, a founding partner of the Let's Encrypt project and the non-profit organization that develops Certbot? We'd like to send you email about our work encrypting the web, EFF news, campaigns, and ways to support digital freedom.

(Y)es/(N)o: N

- No names were found in your configuration files. Please enter in your domain name(s) (comma and/or space separated) (Enter 'c' to cancel): aware.wwbp.org

The script says that it did the following:

Created an SSL vhost at /etc/apache2/sites-available/000-default-le-ssl.conf
Enabled Apache socache_shmcb module
Enabled Apache ssl module
Deploying Certificate to VirtualHost /etc/apache2/sites-available/000-default-le-ssl.conf
Enabling available site: /etc/apache2/sites-available/000-default-le-ssl.conf

- Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP ac cess.

1: No redirect - Make no further changes to the webserver configuration.
2: Redirect - Make all requests redirect to secure HTTPS access. Choose this for new sites, or if you're confident your site works on HTTPS. You can undo this change by editing your web server's configuration.
Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 1

Summary result of running the script:

IMPORTANT NOTES:
- Congratulations! Your certificate and chain have been saved at:
  /etc/letsencrypt/live/aware-cloud.wwbp.org/fullchain.pem
  Your key file has been saved at:
  /etc/letsencrypt/live/aware-cloud.wwbp.org/privkey.pem
  Your cert will expire on 2018-12-05. To obtain a new or tweaked
  version of this certificate in the future, simply run certbot again
  with the "certonly" option. To non-interactively renew *all* of
  your certificates, run "certbot renew"
- Your account credentials have been saved in your Certbot
  configuration directory at /etc/letsencrypt. You should make a
  secure backup of this folder now. This configuration directory will
  also contain certificates and private keys obtained by Certbot so
  making regular backups of this folder is ideal.
- If you like Certbot, please consider supporting our work by:
  Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
  Donating to EFF: https://eff.org/donate-le

4.3 SECURING THE AWARE DASHBOARD

The apache configuration file for the dashboard virtual host must be edited to enable the aware dashboard. Per the lets encrypt setup the conf file updated for ssl is /etc/apache2/sites-available/000-default-le-ssl.conf

Edit that file and make the following changes:

- Change the document root from

DocumentRoot /var/www/html

To

DocumentRoot /var/www/html/aware-server

- Change log files from

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

To

ErrorLog /var/log/apache2/aware_error.log
CustomLog /var/log/apache2/aware_access.log combined

- locate the line

ServerName aware-cloud.wwbp.org

And add the following below it:

<Directory "/var/www/html/aware-server">
  Allow from all
  Options +Indexes
 </Directory>

- Locate the line

SSLCertificateChainFile /etc/letsencrypt/live/aware-cloud.wwbp.org/chain.pem

And add the following below it

ErrorLog /var/log/apache2/ssl_error_log
 LogLevel debug
 TransferLog /var/log/apache2/ssl_access_log

Please note that these instructions deviate from the aware server install instructions. The cypher suite provided in the aware server instructions is significantly shorter than the one included by Letsencrypt in its include file that actually has a lot of the settings

NOTE: These instructions are in case you are creating a new set of .pem files. If you are being given a set of ssl key files, replace the following lines:

SSLCertificateFile /etc/letsencrypt/live/aware.wwbp.org/fullchain.pem
SSLCertificateKeyFile /etc/letsencrypt/live/aware.wwbp.org/privkey.pem
SSLCertificateChainFile /etc/letsencrypt/live/aware.wwbp.org/chain.pem

With whatever 3 files you have been given. (ex.)
SSLCertificateFile /etc/apache2/sslkey/wwbp_org_cert_2020.cer
SSLCertificateKeyFile /etc/apache2/sslkey/wwbp_org_cert_2020.key

SSLCACertificateFile /etc/apache2/sslkey/_.wwbp.org_ca.crt

Please also note that these changes are done for the *:443 virtual host (https)

Edit file /etc/apache2/sites-available/000-default.conf and make the following changes for the *:80 virtual host that hosts the http server needed to lookup public certs

- Change the document root from

DocumentRoot /var/www/html

To

DocumentRoot /var/www/html/public

- Add the following lines before </VirtualHost> tag

ServerPath "/public/"
ServerName aware-cloud.wwbp.org

4.4 SET UP PUBLIC CERTIFICATES IN THE PUBLIC FOLDER FOR THE HTTP VIRTUAL HOST

- "sudo mkdir /var/www/html/public"
- "sudo chmod 777 /var/www/html/public"
- openssl x509 -outform der -in /etc/letsencrypt/live/aware-cloud.wwbp.org/cert.pem -out /var/www/html/public/server.crt
- cp /var/www/html/public/server.crt /var/www/html/public/ca.crt
- chmod -R 777 /var/www/html/public

4.5 MAKE ADJUSTMENTS TO PHP.INI

In the ssh terminal run the following commands:

- "sudo nano /etc/php/5.6/apache2/php.ini"

Make the following changes

- Find the line

upload_max_filesize = 2M

and replace it with

upload_max_filesize = 200M

- look for the word extension and in the sections where extensions are described add the following

extension=mcrypt.so

5 MYSQL CONFIGURATION

This section works very similar to the documentation provided by aware.

5.1 COPY LETSENCRYPT CERTIFICATES AND ALLOW ACCESS TO THEM BY MYSQL

Run the following commands on the ssh terminal:

- "cd /etc/mysql"

If you haven't been given ssl files to use:

- "sudo cp /etc/letsencrypt/live/aware.wwbp.org/cert1.pem /etc/mysql"
- "sudo cp /etc/letsencrypt/live/aware.wwbp.org/chain1.pem /etc/mysql"
- "sudo cp /etc/letsencrypt/live/aware.wwbp.org/fullchaincert1.pem /etc/mysql"
- "sudo cp /etc/letsencrypt/live/aware.wwbp.org/privkey1.pem /etc/mysql"
- sudo chown mysql:mysql /etc/mysql/*.pem

If you have been given files to use: (modfiy for your file locations:

- "sudo cp /etc/apache/sslkey/_.wwbp.org_ca.crt /etc/mysql"
- "sudo cp /etc/apache/sslkey/wwbp_org_cert_2020.cer /etc/mysql"
- "sudo cp /etc/apache/sslkey/wwpb_org_cert_2020.key /etc/mysql"

Add the following lines at the end of the [mysqld] section (probably end of file):

- "sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf"

```
ssl-ca=/etc/mysql/chain1.pem        (or whatever your ca file is)
ssl-cert=/etc/mysql/cert1.pem       (or whatever your chain file is)
ssl-key=/etc/mysql/privkey1.pem     (or whatever your key file is)
```

Restart the MySQL server by running:

- sudo service mysql restart

Check the status by running

- sudo service mysql status

5.2 CREATE A MYSQL DATABASE AND DB USER FOR THE AWARE DASHBOARD

Connect to the database as root and create the database by running the following commands at the ssh terminal:

- mysql -u root -p

  Enter password: <root password>

At the sql prompt that comes up execute the following commands

- CREATE DATABASE aware_dashboard;
- CREATE USER 'dbuser'@'localhost' IDENTIFIED BY 'dbuser';
- GRANT ALL PRIVILEGES ON *.* TO 'dbuser'@'localhost' WITH GRANT OPTION;
- FLUSH PRIVILEGES;
- exit;

5.3 LOAD AWARE DASHBOARD CORE DATABASE

Connect to the database as the new "dbuser" and load the core database by running the following commands at the ssh terminal:

- mysql -u dbuser --password=dbuser aware_dashboard < /var/www/html/aware-server/aware_dashboard.sql

5.4 SET MYSQL CONFIGURATION ON YOUR AWARE DASHBOARD

Edit the database.php file in /var/www/html/aware-server/application/config/database.php ensure the values of the variables shown below are set as they are below:

$db['aware_dashboard']['hostname'] = 'localhost';

$db['aware_dashboard']['port'] = '3306';

$db['aware_dashboard']['username'] = 'dbuser';

$db['aware_dashboard']['password'] = 'dbuser';

$db['aware_dashboard']['database'] = 'aware_dashboard';

6 SET UP MOSQUITTO MQTT SERVER

This section removed as unneeded.

7 INSTALL THE ANDROID SDK

Install the android command line tools to parse uploaded plugin information by running the following in the ssh terminal:

- “cd /usr/local/src”
- “sudo wget http://dl.google.com/android/android-sdk_r24.4.1-linux.tgz”
- “tar zxvf android-sdk_r24.4.1-linux.tgz”
- “sudo apt-get install openjdk-8-jdk openjdk-11-jdk”

Add Android SDK to your user's bash profile by running the following:

- “sudo nano ~/.bashrc”

Add the following content to .bashrc

export ANDROID_HOME = ~/android-sdk-linux
export PATH=$PATH:$ANDROID_HOME/tools:$ANDROID_HOME/platform-tools

Save the file and run the following to complete install:

- “source ./bashrc”
- “cd android-sdk-linux/tools”
- “sudo ./android update sdk --no-ui -t platform-tools”
    - Do you accept the license ‘android-sdk-license-##########’ [y/n]: y

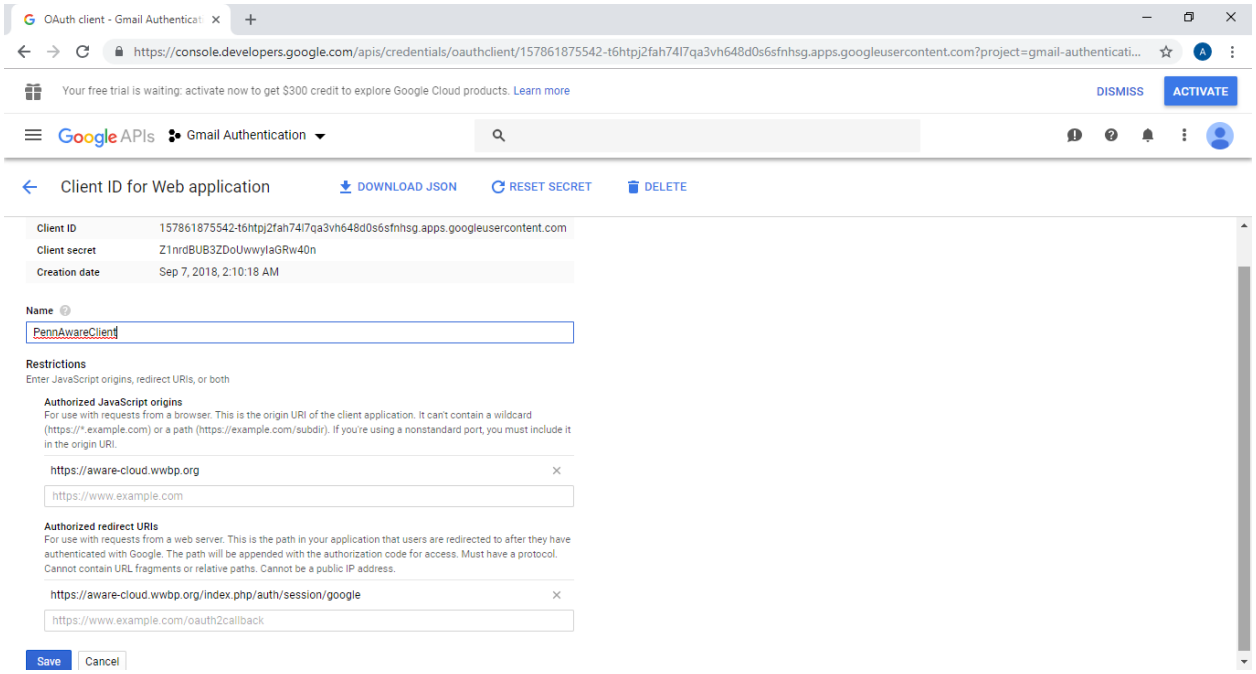
8 CONFIGURING AWARE DASHBOARD

This section follows instructions laid out in the aware dashboard setup guide.

8.1 ADD YOUR SERVER TO GOOGLE OAUTH CREDENTIALS

Go to the Google's Developer Console, at https://console.developers.google.com. Create a new project and then create a new Google OAUTH credentials client with settings shown below.

-DB Notes: Sal Created the keys due to needing to know about privacy policies and other information. He associated the project with the address: sal.giorgi@gmail.com



The Client ID (e.g., 157861875542-t6htpj2fah74l7qa3vh648d0s6sfnhsg.apps.googleusercontent.com) and Client secret (e.g., Z1nrdBUB3ZDoUwwyIaGRw40n) for the final step

8.2 FINAL AWARE DASHBOARD CONFIGURATION

Edit /var/www/html/aware-server/application/config/config.php (sudo vi or nano) and make sure the file has the following settings set (note that for a different environment some of these settings may need to change e.g. each machine must have unique domain name)

$config['encryption_key'] = 'GX$#th@)?FGHty';

$config['cookie_secure'] = TRUE;

$config['android_sdk'] = '/home/ubuntu/android-sdk-linux/';

$config['public_keys'] = '/var/www/html/public/';

```
$config['mqtt_hostname'] = 'aware.wwbp.org';
$config['mqtt_port'] = '8883';

$config['oauth_id'] =
'157861875542-t6htpj2fah74l7qa3vh648d0s6sfnhsg.apps.googleusercontent.com';
$config['oauth_secret'] = 'Z1nrdBUB3ZDoUwwyIaGRw40n';
```

Save the file and restart apache by running

- sudo service apache2 restart