

Mainly 2 steps

1. Bring up EC2 Linux System
2. Install mosquito on the above system

Bring up EC2 Linux System

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with options like Instances, Launch Templates, and Capacity Reservations. The main area lists two instances: a Windows instance (i-03587f2ebbcabb21) and an Ubuntu instance (i-06330ba4d5b31193), both currently stopped. At the top, a red box highlights the 'Launch Instance' button. Below it, there's a detailed view of the selected Windows instance, including its instance ID, state, type, IP address, and various network and security details.

The screenshot shows the 'Launch Instance Wizard' step 1: Choose an Amazon Machine Image (AMI). It lists several options, with the 'Ubuntu Server 16.04 LTS (HVM), SSD Volume Type' option highlighted by a red box. This option is described as being available from Canonical. Below this, other options like Microsoft Windows Server 2019 Base and Deep Learning AMI (Ubuntu 16.04) Version 25.0 are listed. On the right side of the wizard, there are 'Select' buttons and checkboxes for choosing the architecture (64-bit x86 or 64-bit Arm).



MQTT Broker SSL Implementation Guideline

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
General purpose	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
General purpose	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
General purpose	t3a.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

Feedback English (US) © 2008-2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use 15:01 01-11-2019

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-0994c095691a46fb5
Free tier eligible

Ubuntu Server 16.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups

Security group name: launch-wizard-6
Description: launch-wizard-6 created 2019-11-01T15:01:49.372+05:30

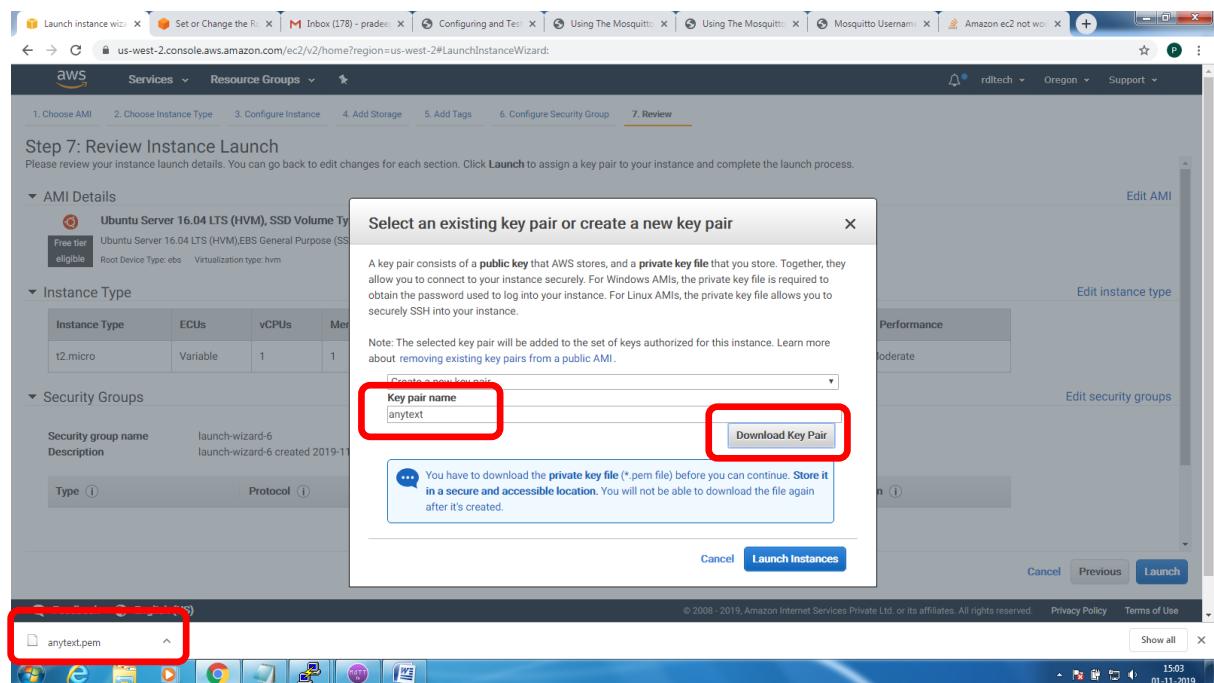
Type	Protocol	Port Range	Source	Description
<i>This security group has no rules</i>				

Instance Details

Cancel Previous Launch

Feedback English (US) © 2008-2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use 15:01 01-11-2019

MQTT Broker SSL Implementation Guideline



Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

Ubuntu Server 16.04 LTS (HVM), SSD Volume Type

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)
t2.micro	Variable	1	1

Security Groups

Security group name: launch-wizard-6
Description: launch-wizard-6 created 2019-11-01

Type: (i) Protocol: (i)

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Create a new key pair

Key pair name: anytext

Download Key Pair

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location**. You will not be able to download the file again after it's created.

Cancel Launch Instances

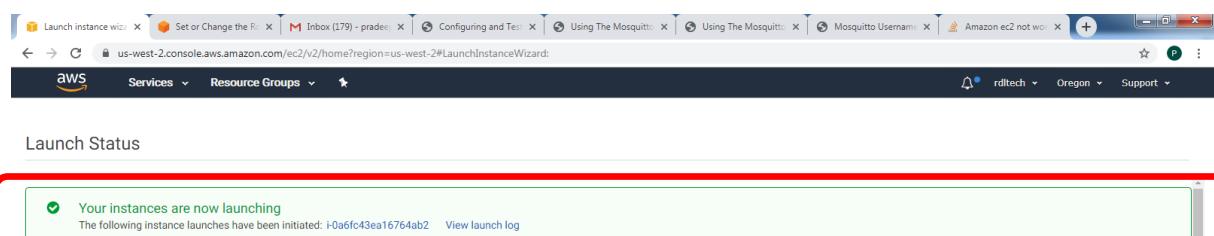
Performance: moderate

Edit instance type

Edit security groups

Cancel Previous Launch

© 2008 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Show all



Your instances are now launching

The following instance launches have been initiated: i-0a6fc43ea16764ab2 View launch log

Get notified of estimated charges

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click **View Instances** to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the Instances screen. Find out how to connect to your instances.

Here are some helpful resources to get you started

- How to connect to your Linux instance
- Learn about AWS Free Usage Tier
- Amazon EC2: User Guide
- Amazon EC2: Discussion Forum

While your instances are launching you can also

Create status check alarms to be notified when these instances fail status checks. (Additional charges may apply)

Create and attach additional EBS volumes (Additional charges may apply)

Manage security groups

Feedback English (US)

© 2008 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Show all

anytext.pem

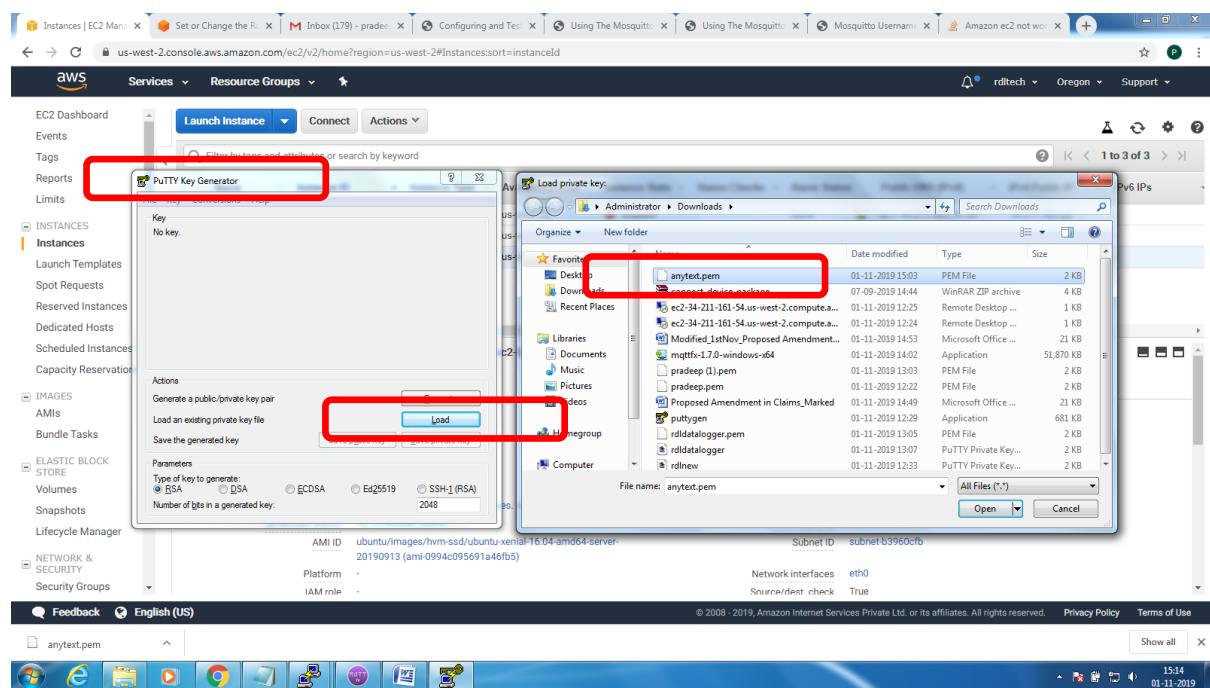
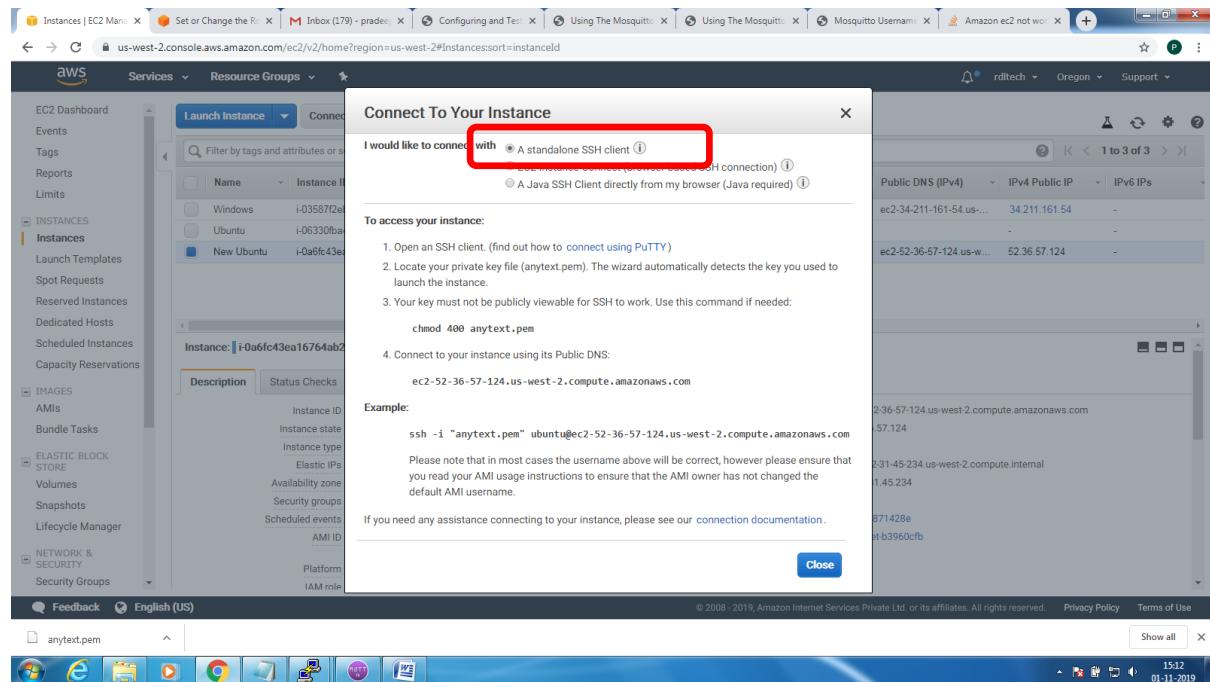
15:30 01-11-2019

MQTT Broker SSL Implementation Guideline

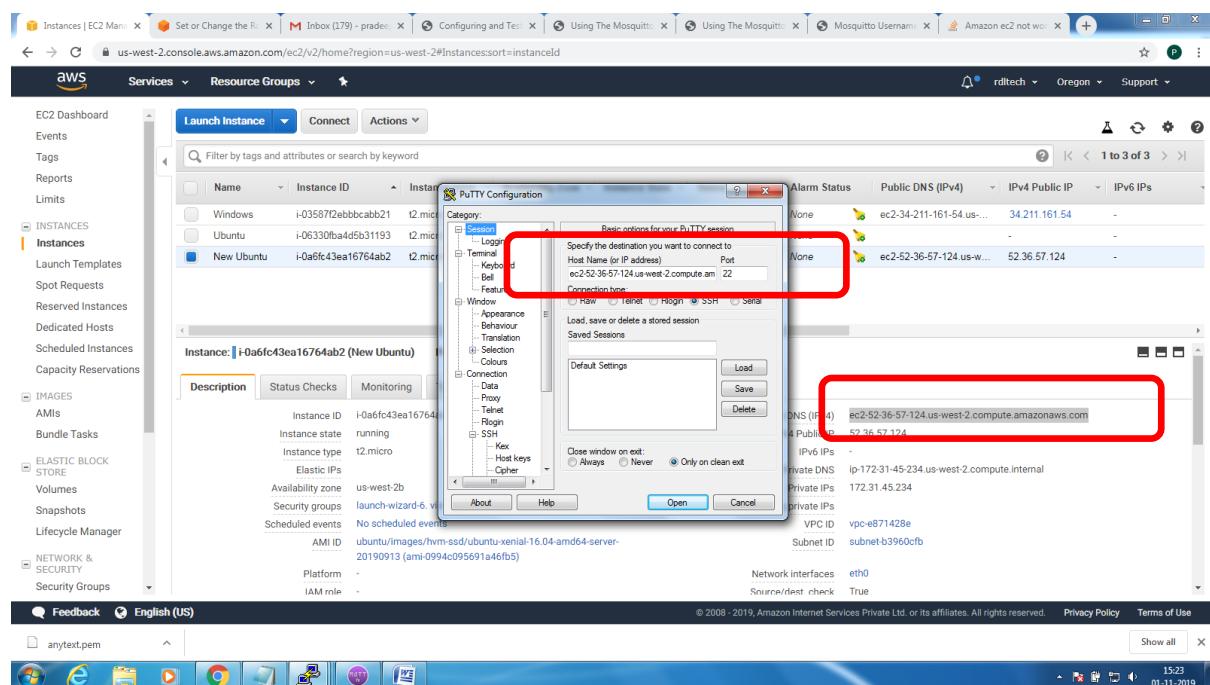
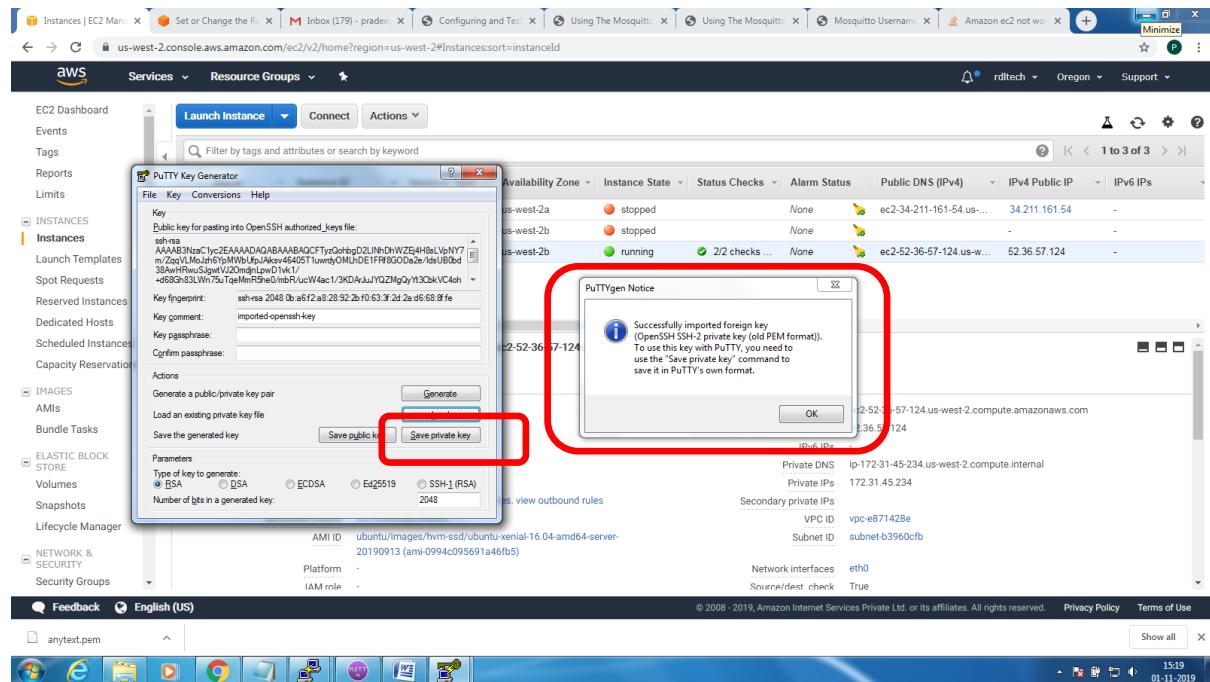
The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Launch Templates, Spot Requests, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, and Security Groups. The main area displays a table of instances. One instance, 'New Ubuntu' (i-0a6fc43ea16764ab2), is selected and highlighted with a red box. To the right of the table, a detailed view of this instance is provided, including its description, status checks, monitoring, and tags. The instance details include: Instance ID: i-0a6fc43ea16764ab2, Instance state: running, Instance type: t2.micro, Availability zone: us-west-2b, Security groups: launch-wizard-6, view inbound rules, view outbound rules, Scheduled events: No scheduled events, AMI ID: ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20190913 (ami-0994c095691a46fb5), Platform: -, IAM role: -. The detailed view also lists Public DNS (IPv4): ec2-52-36-57-124.us-west-2.compute.amazonaws.com, IPv4 Public IP: 52.36.57.124, IPv6 IPs: -, Private DNS: ip-172-31-45-234.us-west-2.compute.internal, Private IPs: 172.31.45.234, Secondary private IPs: -, VPC ID: vpc-e871428e, Subnet ID: subnet-b3960cfb, Network interfaces: eth0, and Source/dest check: True.

This screenshot is similar to the one above, showing the AWS EC2 Instances page. The 'Actions' dropdown menu is open over the 'New Ubuntu' instance, and the 'Create Template From Instance' option is highlighted with a red box. The rest of the interface and instance details are identical to the first screenshot.

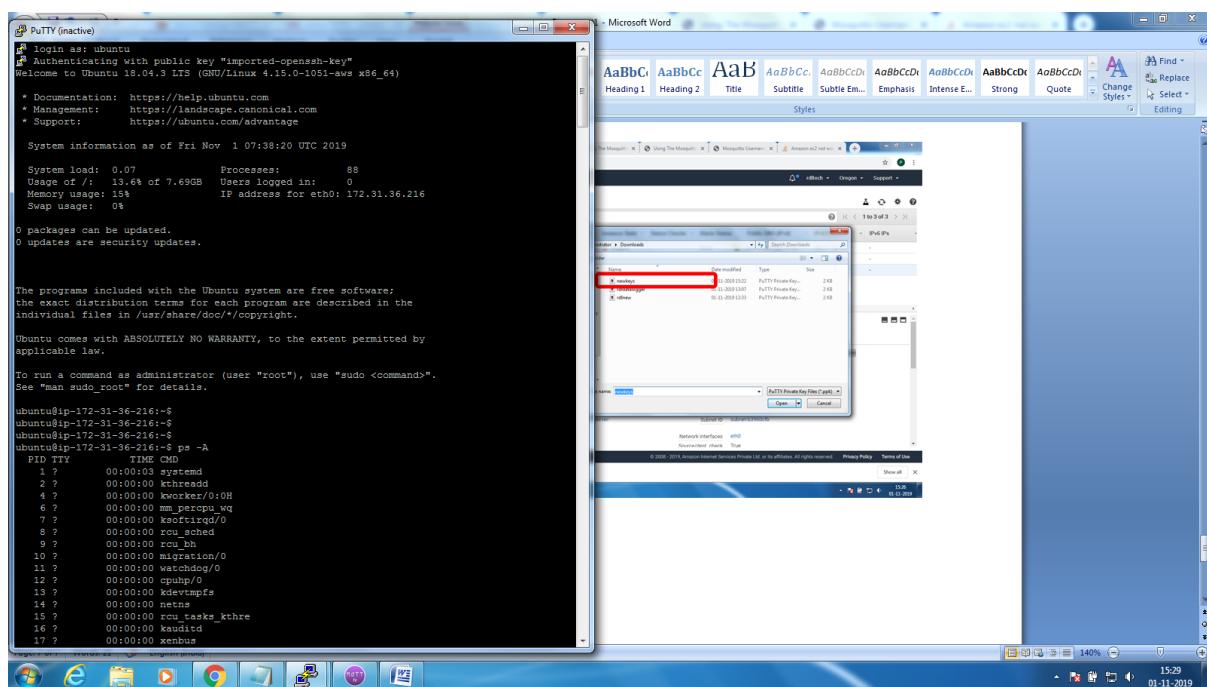
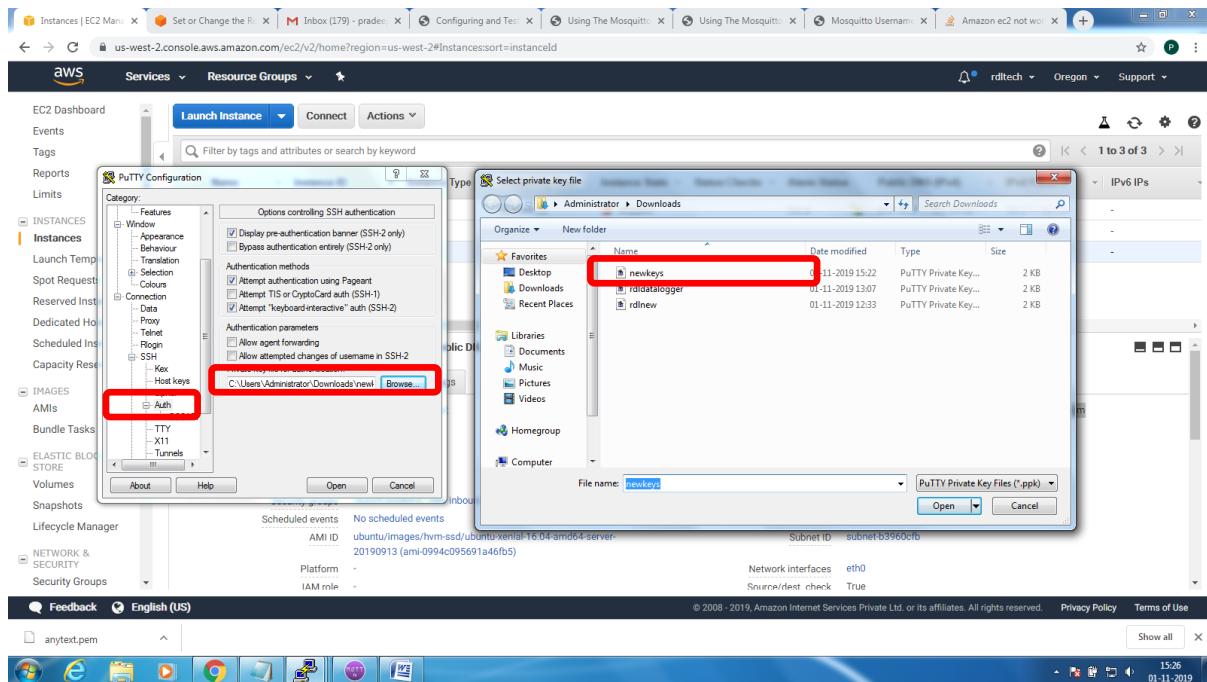
MQTT Broker SSL Implementation Guideline



MQTT Broker SSL Implementation Guideline



MQTT Broker SSL Implementation Guideline



Installation of **mosquitto** on linux system

Follow the below steps to install mosquitto

1. sudo apt-add-repository ppa:mosquitto-dev/mosquitto-ppa
2. sudo apt-get update
3. sudo apt-get install mosquitto
4. sudo apt-get install mosquitto-clients

Follow the below steps to Enable user authentication

- Create a txt file in the following format
- Username:Password
- Issue the following commands to add certificate to this file
- mosquitto_passwd -U passwordfile (text file name)
- Copy this file to /etc/mosquitto

Open **mosquitto.conf** and add these 2 lines to enable user authentication

```
allow_anonymous false  
password_file /etc/mosquitto/passwords.txt
```

Restart the broker to absorb the changes

```
ubuntu@ip-172-31-36-216:~$ mosquitto -v  
1572603369: mosquitto version 1.6.7 starting  
1572603369: Using default config.  
1572603369: Opening ipv4 listen socket on port 1883.  
1572603369: Error: Address already in use
```

To resolve this

```
ps -ef | grep mosquitto  
kill -9 pid  
mosquitto_sub -t '$SYS/#' -v  
or  
root@ip-172-31-36-216:/home/ubuntu# mosquitto
```

```
1572603616: mosquitto version 1.6.7 starting  
1572603616: Using default config.  
1572603616: Opening ipv4 listen socket on port 1883.  
1572603616: Opening ipv6 listen socket on port 1883.
```

MQTT Broker SSL Implementation Guideline

To enable Message flow on to AWS ES2 system, follow the below steps

Create Security Group to allow TCP/1883 traffic in the inbound direction

The screenshot shows the AWS EC2 Dashboard with the 'Security Groups' section selected. A new security group named 'mqtt' is being created. The 'Inbound' tab is selected, showing two rules allowing TCP port 1883 from 0.0.0.0/0.

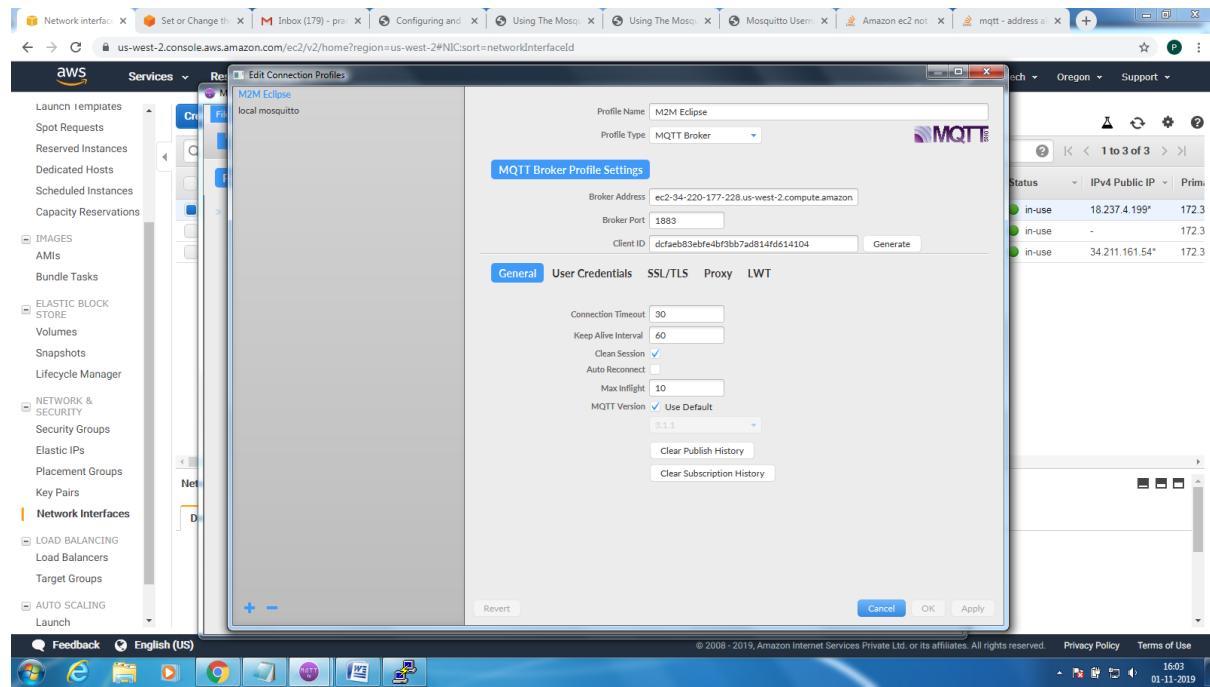
Type	Protocol	Port Range	Source	Description
Custom TCP Rule	TCP	1883	0.0.0.0/0	
Custom TCP Rule	TCP	1883	-/0	

Add the Security Group to the Next Interface associated to the EC2 resource

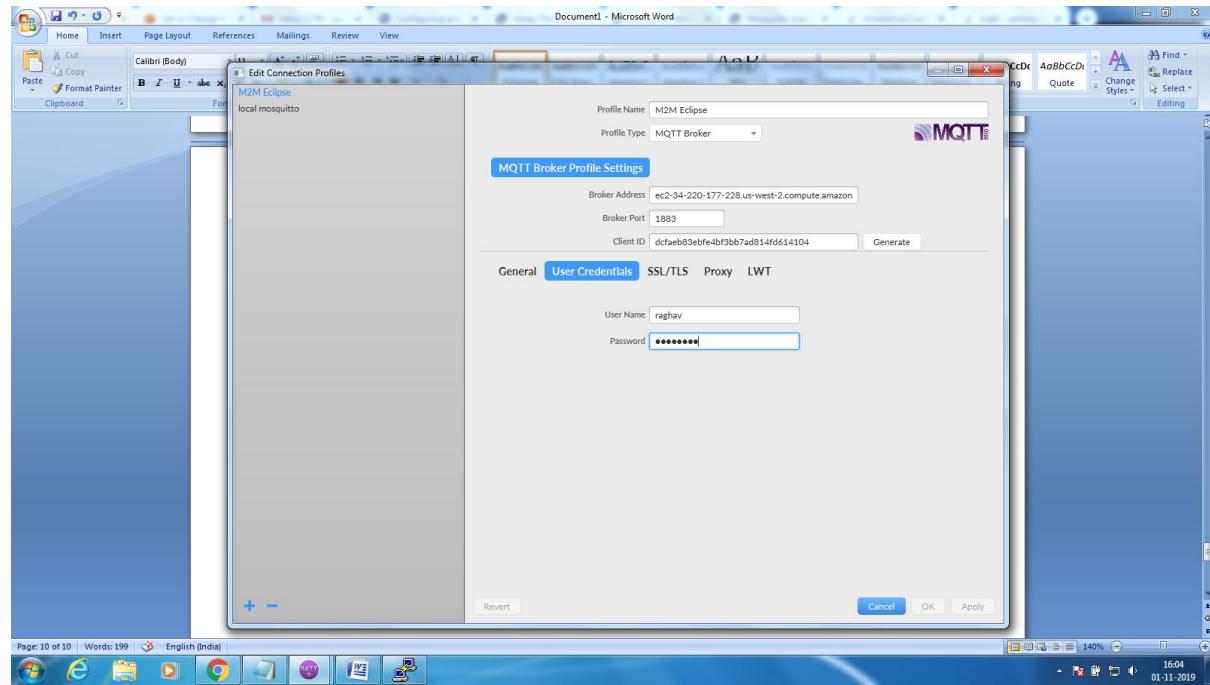
The screenshot shows the AWS EC2 Dashboard with the 'Network Interfaces' section selected. The 'Details' tab for a specific interface is shown, with a context menu open over the 'Security groups' entry. The 'Change Security Groups' option is highlighted.

Name	Network Interface	Subnet ID	VPC ID	Zone	Security groups	Description	Instance ID	Status	IPv4 Public IP	Primary
eni-0162c7e29...	subnet-b3960cfb				launch-wizard-5, mqtt		i-06330fba4d5b31193	in-use	18.237.4.199*	172.3
eni-07c8723...	subnet-b3960cfb				launch-wizard-6		i-0a6fc43ea16764ab2	in-use	-	172.3
eni-0fa61badd...	subnet-5ea5df38				launch-wizard-1		i-03587f2ebbbcab21	in-use	34.211.161.54*	172.3

On MQTTFx client, include Public domain IP4 address as Broker Address



Provide user credentials configured in the password.txt on the MQTT broker



The bring up is now complete, topic could be subscribed and publish on the MQTTfx client

Mosquitto SSL Configuration -MQTT TLS Security

Server Side:

Openssl tool is used to generate the required keys and certificates for both the server and client

Issue the following commands in sequence and make changes to the mosquito.conf on the server for the changes to take effect

1. **openssl genrsa -des3 -out ca.key 2048**
2. **openssl req -new -x509 -days 1826 -key ca.key -out ca.crt**
3. **openssl genrsa -out server.key 2048**
4. **openssl req -new -out server.csr -key server.key**
5. **openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt -days 360**

The directory file listing would look similar to the below

```
drwxr-xr-x  5 root root 4096 Nov  8 05:02 .
drwxr-xr-x 90 root root 4096 Nov  8 06:22 ..
-rw-r--r--  1 root root 1314 Nov  8 04:28 ca.crt
-rw-----  1 root root 1751 Nov  8 04:27 ca.key
-rw-r--r--  1 root root   41 Nov  8 04:29 ca.srl
drwxr-xr-x  2 root root 4096 Nov  8 04:09 ca_certificates/
drwxr-xr-x  2 root root 4096 Nov  8 04:30 certs/
drwxr-xr-x  2 root root 4096 Nov  8 04:09 conf.d/
-rw-r--r--  1 root root  530 Nov  8 04:55 mosquitto.conf
-rw-r--r--  1 root root  116 Nov  8 04:12 password.txt
-rw-r--r--  1 root root 1192 Nov  8 04:29 server.crt
-rw-r--r--  1 root root  993 Nov  8 04:29 server.csr
-rw-----  1 root root 1675 Nov  8 04:28 server.key
```

Copy the ca.crt, server.crt and server.key file to /etc/mosquitto/cert folder

Make the following changes to the mosquito.conf file

Add the following lines

Port 8883

Cafile /etc/mosquitto/certs/ca.crt

Keyfile /etc/mosquitto/certs/server.key

Certfile /etc/mosquitto/certs/server.crt

Most important step is to copy the ca.crt on to the client system.

Incase its get challenging to transfer this file, as it would from AWS EC2 environment. Please use notepad+ (not plain notepad) to copy paste the certificate contents

Client Side: Configure the MQTT Client with the broker address, enable SSL and point to the ca.crt file and connect for application