

# Secure Starter Kit Cloud Connect Installation & Setup Guide

Date: December 08, 2020 | Version 1.0  
FINAL

---



The Solutions People



## CONTENTS

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>INTRODUCTION .....</b>   | <b>3</b>  |
| 1.1      | Purpose of the Document .....   | 3         |
| 1.2      | Prerequisites, Background information & AWS Cloud Services Descriptions ..... | 3         |
| <b>2</b> | <b>AWS ACCOUNT CREATION &amp; SETUP EC2 SERVICE.....</b>                      | <b>5</b>  |
| 2.1      | Login or Create your AWS Account.....   | 5         |
| 2.2      | AWS EC2 Instance Service .....  | 6         |
| 2.3      | EC2 Dashboard .....   | 6         |
| 2.4      | Creating an EC2 Instance .....  | 7         |
| 2.5      | Convert key to Putty Format.....  | 11        |
| 2.6      | Configure Putty.....  | 12        |
| <b>3</b> | <b>INSTALLING DOCKER ON EC2 .....</b>   | <b>14</b> |
| 3.1      | Execute below command .....   | 14        |
| <b>4</b> | <b>CONFIGURATION OF EC2 INSTANCE, RDS SERVICE AND SQL DATABASE.....</b>       | <b>16</b> |
| 4.1      | Application access.....   | 16        |
| 4.2      | Application – “Allow” or “Deny” listing.....                                  | 17        |
| 4.3      | AWS RDS Service – Database Setup and Configuration.....                       | 21        |
| 4.4      | Creating a Database .....   | 21        |
| 4.5      | Creating an IAM User .....  | 27        |
| 4.6      | MySQL Setup and Configuration .....   | 29        |
| <b>5</b> | <b>CONFIGURE IMAGE ON DOCKER AND EC2 .....</b>                                | <b>38</b> |
| 5.1      | Execute below commands in PUTTY.....  | 38        |
| 5.2      | Log into SSK Cloud Connect .....  | 38        |
| <b>6</b> | <b>CHECKOUT PROJECT.....</b>  | <b>40</b> |
| <b>7</b> | <b>BUILD PROJECT.....</b>   | <b>41</b> |
| 7.1      | Execute below command .....   | 41        |
| <b>8</b> | <b>REFERENCES .....</b>   | <b>42</b> |

## 1 INTRODUCTION

### 1.1 Purpose of the Document

The Cloud Connect Installation & Setup Guide provides an overview of the AWS services required to run the demo's provided in the Security Starter Quick Start Guides, as well as detailed instructions to setup and configure those required services. Each of these services **MUST** be setup and configured (only once), prior to running the demo's outlined in the Security Starter Quick Start Guides.

### 1.2 Prerequisites, Background information & AWS Cloud Services Descriptions

1. **AWS Account Management Console** – the user will need to create their own AWS Account and is used as the basis for the configuration of the other services required to run the demo's provided in the Security Starter Kits. The creation of an account provides the following access and feature;
  - Discover and experiment with over 150 AWS services, many of which you can try for [free](#).
  - Build your cloud-based applications in [any AWS data center throughout the world](#).
  - Manage and monitor [users](#), [service usage](#), [health](#), and [monthly billing](#).
  - Get [in-console help](#) from AWS Support.
  - **Link to create AWS Account:** <https://portal.aws.amazon.com/billing/signup#/start>
2. **AWS EC2 Instance Service** – <https://aws.amazon.com/ec2>

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment.

3. **AWS Relational Database Service (Amazon RDS)** – <https://aws.amazon.com/rds/>
- *The output of the setup and configuration of the EC2 instance will provide the user with URL and Login credentials required to run their own instance of the Security Starter Kit Cloud Connect Tool.*

Makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups. It frees you to focus on your applications so you can give them the fast performance, high availability, security and compatibility they need.

Amazon RDS is available on several [database instance types](#) - optimized for memory, performance or I/O - and provides you with six familiar database engines to choose from, including [Amazon Aurora](#), [PostgreSQL](#), [MySQL](#), [MariaDB](#), [Oracle Database](#), and [SQL Server](#). You can use the [AWS Database Migration Service](#) to easily migrate or replicate your existing databases to Amazon RDS.

- *MySQL is employed as the database instance type when configuring Amazon RDS service.*

4. **Docker Hub** - Cloud-based application registry and development team collaboration services.  
<https://www.docker.com/>  
<https://hub.docker.com/>

Docker Hub is the world's largest repository of [container images](#) with an array of content sources including container community developers, open source projects and independent software vendors (ISV) building and distributing their code in containers. Users get access to free public repositories for storing and sharing images or can choose subscription plan for private repos.

- *Docker is the repository service used to store source code for our web-based, open-source “Security Starter Kit Cloud Connect Tool”. The user will need to update, configure and build an “Image” from the source code stored on Docker Hub using their specific AWS Account and AWS Service credentials. These instructions are provided below in the document.*

  5. **AWS OTA Role Access** - <https://docs.aws.amazon.com/freertos/latest/userguide/create-ota-user-policy.html>

When you create an OTA update, the [OTA Update Manager service](#) creates an [AWS IoT job](#) to notify your devices that an update is available. The OTA demo application runs on your device and creates a FreeRTOS task that subscribes to notification topics for AWS IoT jobs and listens for update messages. When an update is available, the OTA Agent publishes requests to AWS IoT and receives updates using the HTTP or MQTT protocol, depending on the settings you chose. The OTA Agent checks the digital signature of the downloaded files and, if the files are valid, installs the firmware update. If you don't use the FreeRTOS OTA Update demo application, you must integrate the [OTA Agent library](#) into your own application to get the firmware update capability.

- *OTA setup and configuration is listed in Section 5 of the SSK Cloud Connect Users Guide.*
- *These steps do not need to be completed to run the demo outlined in the Quick Start Guide, but will need to be configured in order to perform OTA firmware updates from within AWS Cloud Services.*

## 2 AWS ACCOUNT CREATION & SETUP EC2 SERVICE

### 2.1 Login or Create your AWS Account

**Note:** If the User does not have an AWS Account, you will need to create one and this is used as the basis for the configuration of the other services required to run the demo's provided in the Security Starter Kits.

Login URL: <https://aws.amazon.com/console/>

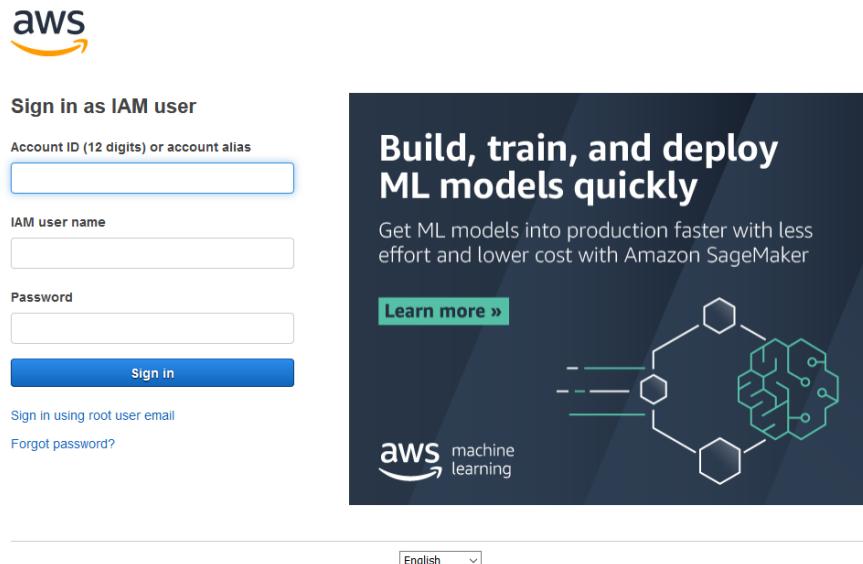


Figure 1: Login page

A screenshot of the 'Create an AWS account' page. The page has a light gray background. On the left, there is a section titled 'AWS Accounts Include 12 Months of Free Tier Access' with a subtext about free tier access for EC2, S3, and DynamoDB. Below this, there is a link to 'aws.amazon.com/free'. On the right, there is a form with fields for 'Email address', 'Password', 'Confirm password', and 'AWS account name' (with a green checkmark icon). A large yellow 'Continue' button is at the bottom of the form. Below the form, there is a link to 'Sign in to an existing AWS account'. At the very bottom, there is small text about copyright and terms of use.

Figure 2: Create New Account page

## 2.2 AWS EC2 Instance Service

- Go to AWS Console >> Services >> Select EC2 (Under Compute section).

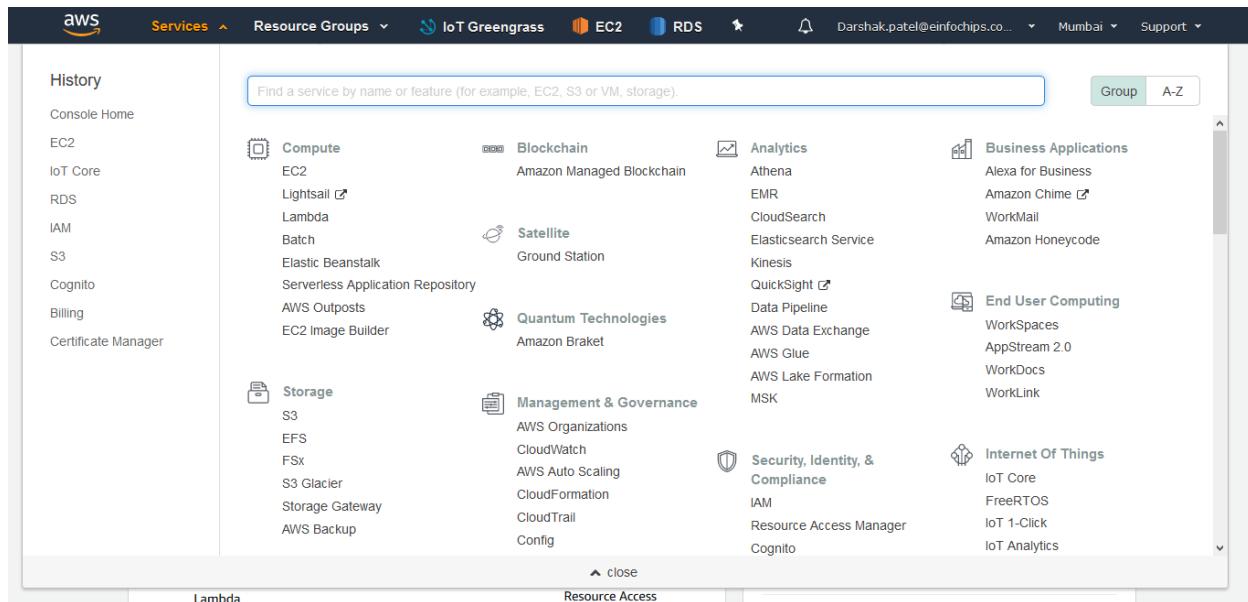


Figure 2: Select EC2 Instance

## 2.3 EC2 Dashboard

- Go to Instances >> Instances Click on it.

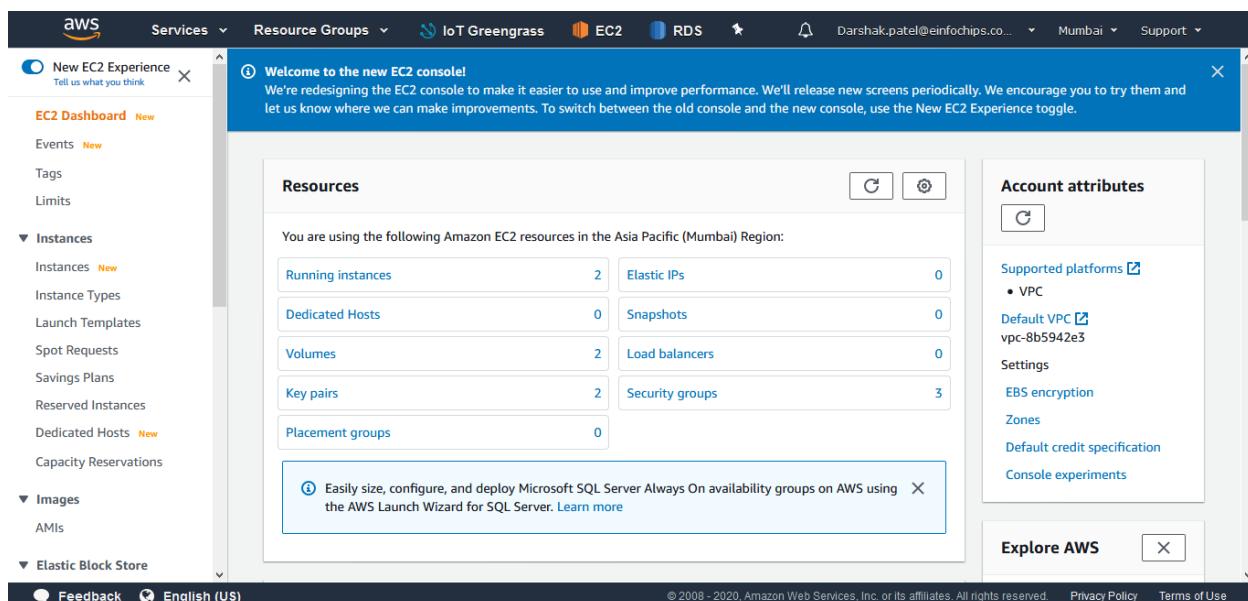


Figure 2: List EC2 Dashboard

## 2.4 Creating an EC2 Instance

### Step 1: Click on Launch instances (Top right corner)

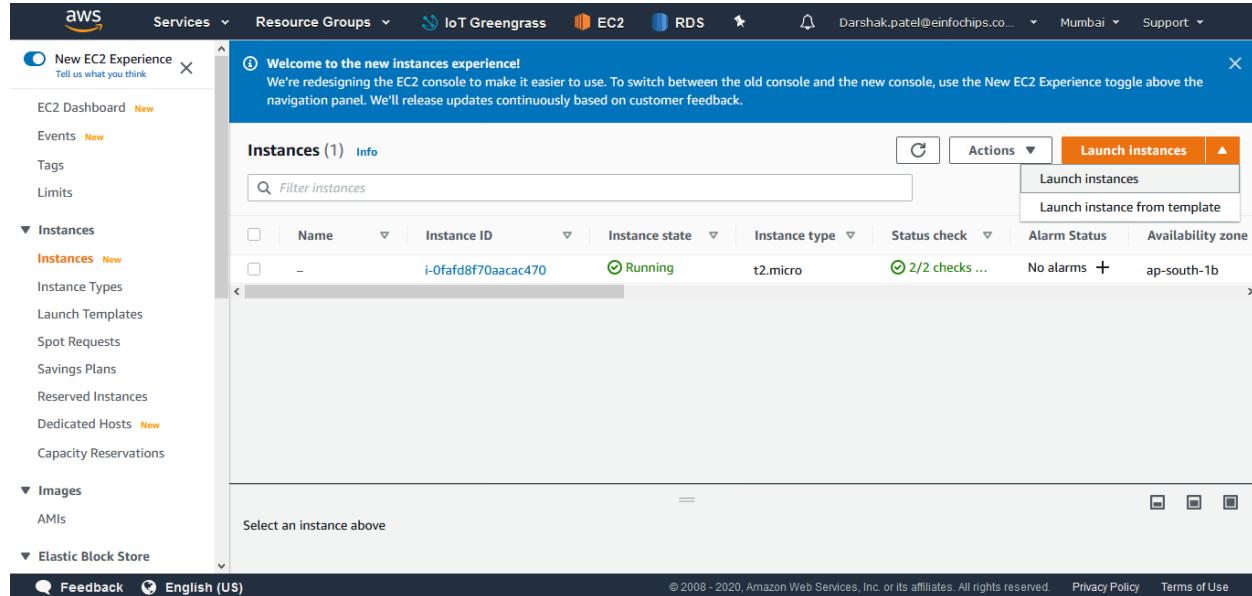


Figure 3: Launch EC2 Instance

- Choose an Amazon Machine Image  
Search “Ubuntu Server 18.04 LTS” in textbox then press select button.

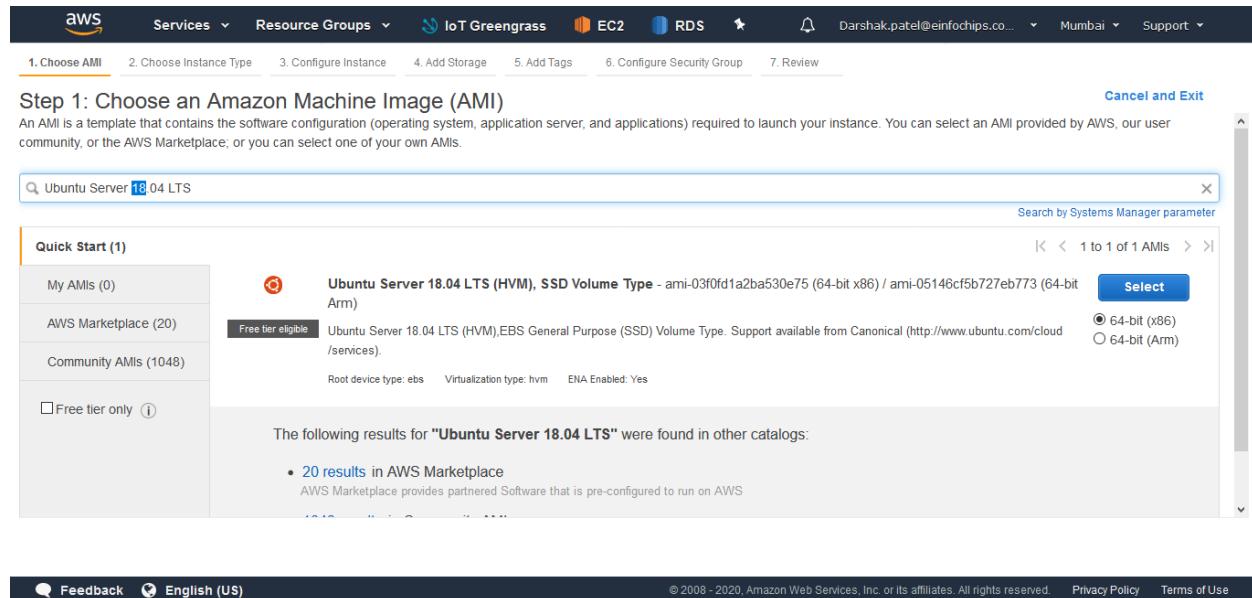


Figure 4: Ubuntu AMI

## Step 2: Choose an Instance Type (Change as per your performance requirement)

- Click on Next: Configure Instance details

| Family          | Type                                  | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance | IPv6 Support |
|-----------------|---------------------------------------|-------|--------------|-----------------------|-------------------------|---------------------|--------------|
| General purpose | t2.nano                               | 1     | 0.5          | EBS only              | -                       | Low to Moderate     | Yes          |
| General purpose | <b>t2.micro</b><br>Free tier eligible | 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          |

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

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

Cancel Previous Review and Launch Next: Configure Instance Details

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Figure 5: Configure EC2 Instance Type

## Step 3: Configure Instance Details (Don't alter anything if don't needed)

- Click on Next: Add Storage

Number of Instances (1) Launch into Auto Scaling Group

Purchasing option Request Spot instances

Network (vpc-8b5942e3 (default)) Create new VPC

Subnet (No preference (default subnet in any Availability Zone)) Create new subnet

Auto-assign Public IP (Use subnet setting (Enable))

Placement group (Add instance to placement group)

Capacity Reservation (Open)

Domain join directory (No directory) Create new directory

IAM role (None) Create new IAM role

Cancel Previous Review and Launch Next: Add Storage

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Figure 6: Configure Instance details

## Step 4: Add Storage

- Change size to 16 GB (Default 8 GB) then press Next: Add Tags

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

| Volume Type | Device    | Snapshot              | Size (GiB) | Volume Type               | IOPS       | Throughput (MB/s) | Delete on Termination               | Encryption    |
|-------------|-----------|-----------------------|------------|---------------------------|------------|-------------------|-------------------------------------|---------------|
| Root        | /dev/sda1 | snap-061cd34c66ebbd58 | 16         | General Purpose SSD (gp2) | 100 / 3000 | N/A               | <input checked="" type="checkbox"/> | Not Encrypted |

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Cancel Previous Review and Launch Next: Add Tags

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Figure 7: Add Storage

## Step 5: Add Tags (Don't do anything)

- Press Next: Configure Security Group

## Step 6: Configure Security Group

- Fill up Security group Name: SSK Security Group (Also add description)
- Then press “Review and Launch”

**Note:** Ensure that both SSH and HTTP are listed as “type” below, otherwise click “Add Rule” to enable those services.

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group:  Create a new security group  Select an existing security group

Security group name: SSK Security Group

Description: SSK Group created 2020-09-22T02:03:07.677+05:30

| Type | Protocol | Port Range | Source                    | Description                |
|------|----------|------------|---------------------------|----------------------------|
| SSH  | TCP      | 22         | My IP<br>157.32.227.98/32 | e.g. SSH for Admin Desktop |
| HTTP | TCP      | 80         | Anywhere<br>0.0.0.0/0     | e.g. SSH for Admin Desktop |

Add Rule

Cancel Previous Review and Launch Next: Review and Launch

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Figure 8: Configure Security group

## Step 7: Review Instance Launch

- Press Launch

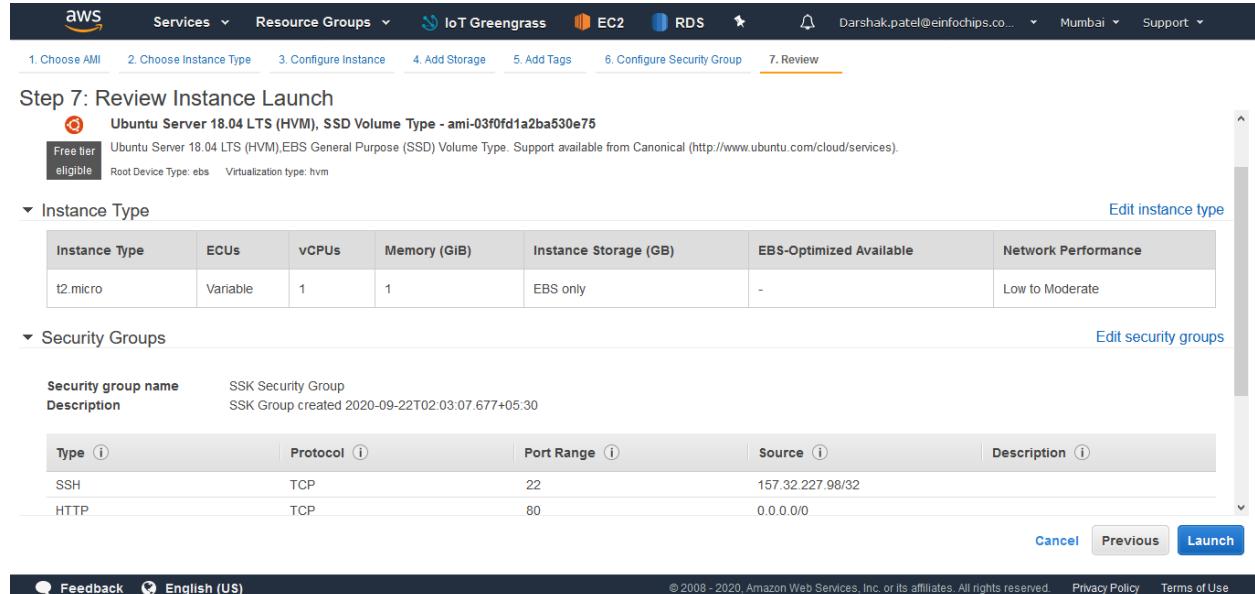


Figure 9: Review Instance

## Step 8: Create New key pair

- Select “Create a new key pair” then name “SSK\_Key”

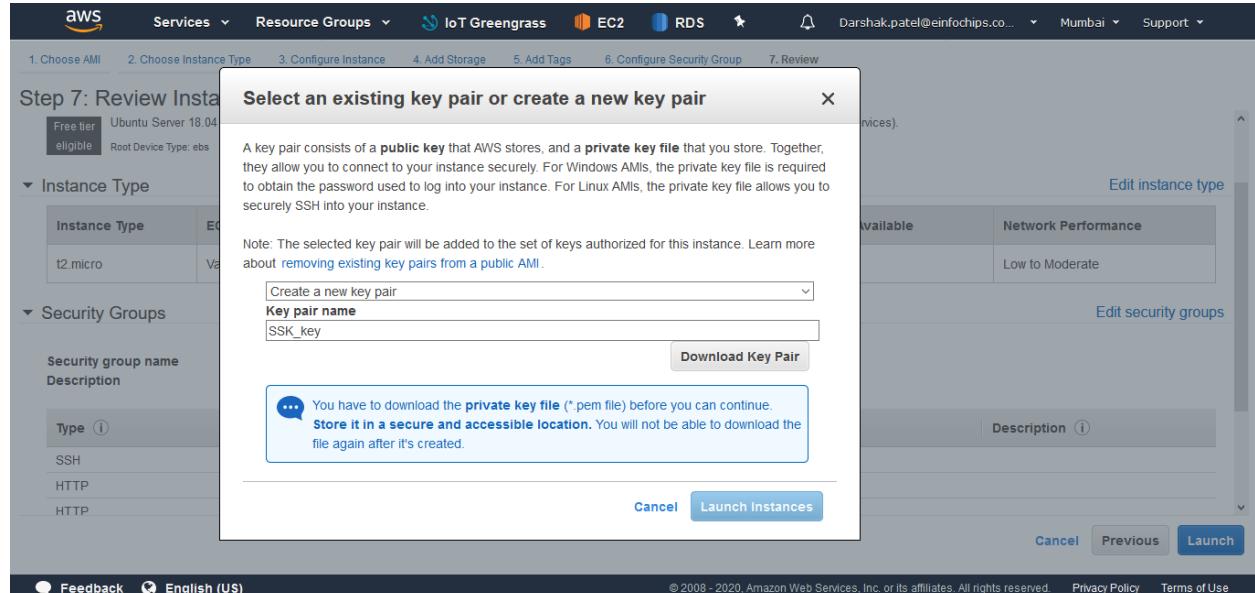


Figure 10: Configure key

### Step 9: Download key pair (To Connect EC2 Instance)

- Keep Certificate key file at secure place which will be used to connect EC2 instance.
- Then press “Launch Instance”.

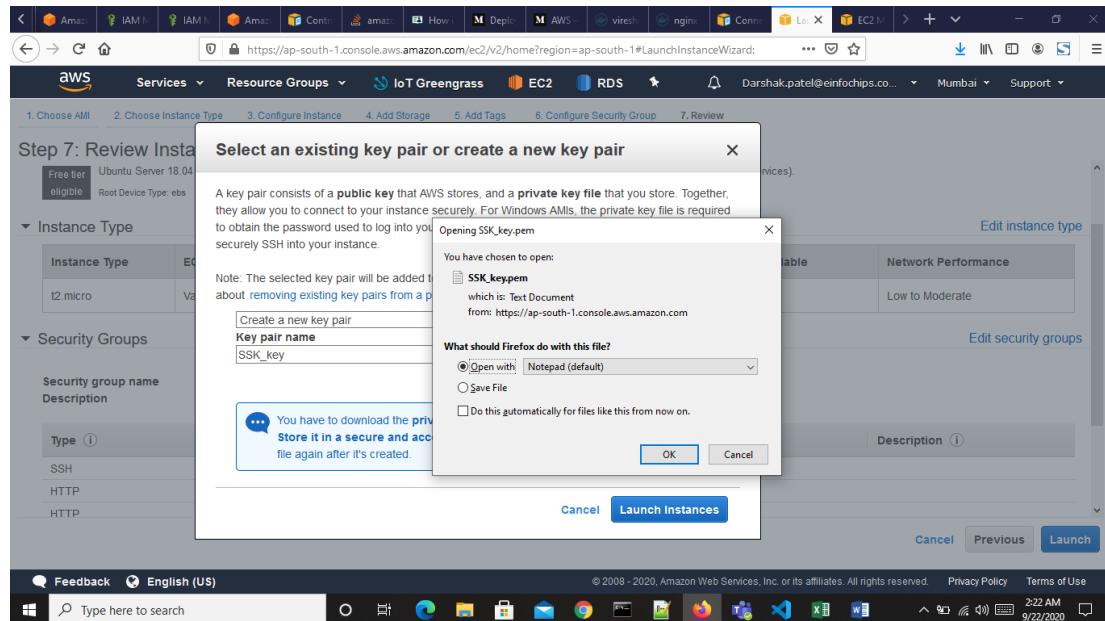


Figure 11: Download key

## 2.5 Convert key to Putty Format

### Step 1: Convert SSK\_key.pem file to SSK\_key.ppk (Using Putty)

Open PuTTYgen (From Windows) press Load button.

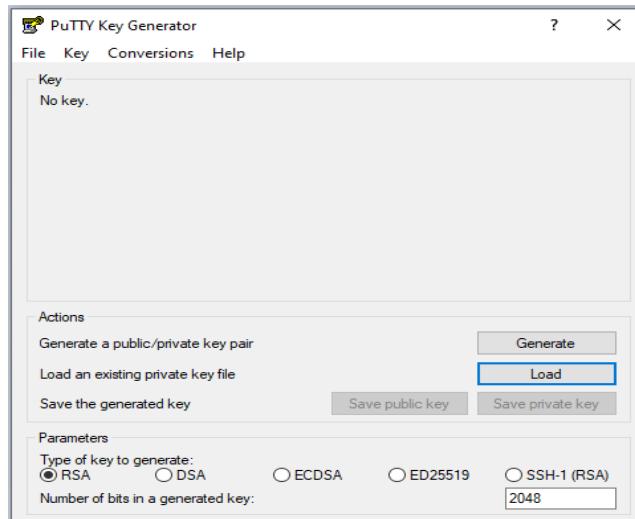


Figure 1: Convert PPK file

- It will ask for file to choose, here you'll need to provide SSK\_key.pem file (select all file format)
- After successfull loading of key it will popup the successfully loaded key
- Press “Save private key”. (Ignore passphrase warning)
- Name the file ” SSK\_key” and Save file along with ppk file

**Ref Link:**

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html#putty-private-key>

## 2.6 Configure Putty

- Open Putty and save the session with following details

Host Name: ubuntu@<host ip address> (Host Ip address can be obtained from EC2 instance)

i.e.

Host Name: [ubuntu@13.235.8.114](#)

Session Name: SSK EC2

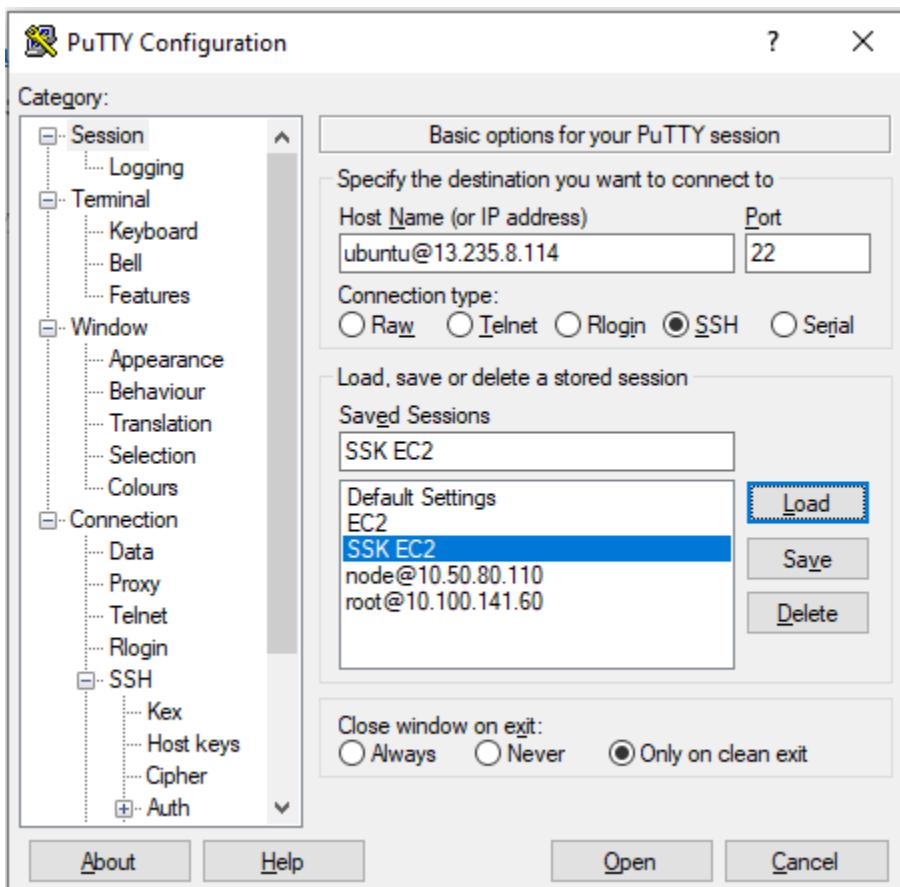


Figure 1: Configure putty

## To configure key

- Go to Connection >> SSH >> Auth >> Select private key
- Then again save it and press open button.

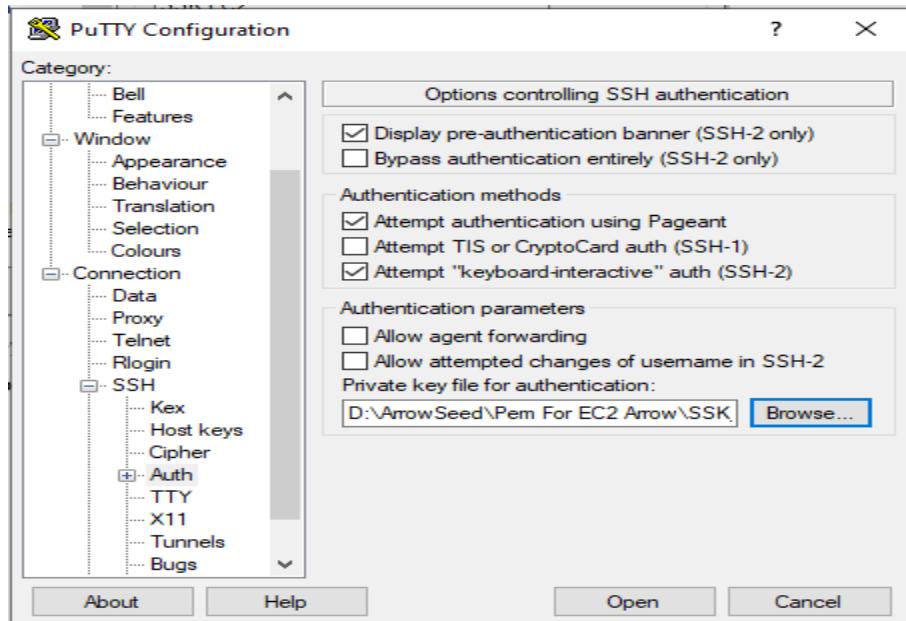


Figure 2: add key to putty

- Here you can now connect to the AWS EC2 Instance.

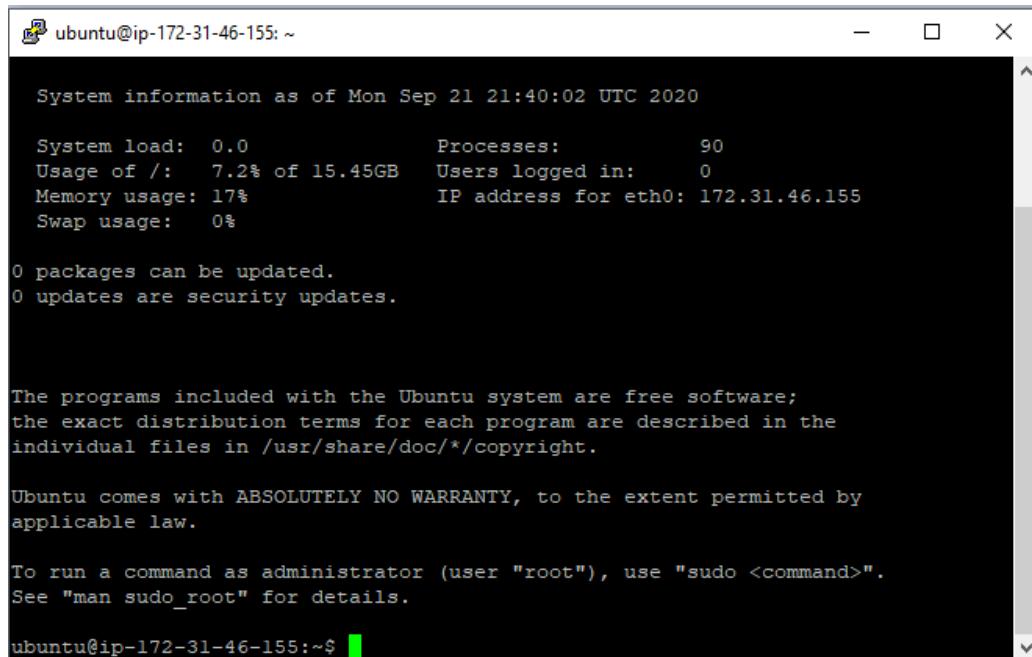


Figure 3: Connected AWS

### 3 INSTALLING DOCKER ON EC2

#### 3.1 Execute below command

Step 1: Update your existing list of packages

```
$ sudo apt-get update
```

Step 2: Next, install a few prerequisite packages which will let apt use packages over HTTPS:

```
$ sudo apt-get install apt-transport-https ca-certificates curl gnupg-agent  
software-properties-common
```

Step 3: Add Docker's official GPG key:

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add
```

-

Step 4: Add the Docker repository to APT sources

```
$ sudo add-apt-repository "deb [arch=amd64]  
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

Step 5: Update the package database with the Docker packages

```
$ sudo apt-get update
```

Step 6: Install Docker

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Step 7: To verify installation

```
$sudo docker --version
```

```
ubuntu@ip-172-31-46-155:~$ sudo docker --version  
Docker version 19.03.13, build 4484c46d9d
```

Figure 1: Docker version



## 4 CONFIGURATION OF EC2 INSTANCE, RDS SERVICE AND SQL DATABASE

### 4.1 Application access

After executing docker run command to access application using following page:

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with 'Instances' selected. The main area displays an 'Instance summary for i-04edc66ab39475528'. Key details shown include:

| Instance ID         | Public IPv4 address  | Private IPv4 addresses                       |
|---------------------|--|--|
| i-04edc66ab39475528 | 13.235.8.114   open address                                      | 172.31.46.155                                |
| Instance state      | Public IPv4 DNS  | Private IPv4 DNS                             |
| Running             | ec2-13-235-8-114.ap-south-1.compute.amazonaws.com   open address | ip-172-31-46-155.ap-south-1.compute.internal |
| Instance type       | VPC ID   | Subnet ID                                    |
| t2.micro            | vpc-8b5942e3   | subnet-9bfcd9f3                              |
| IAM Role            | -  |  |

Figure 1: Application access

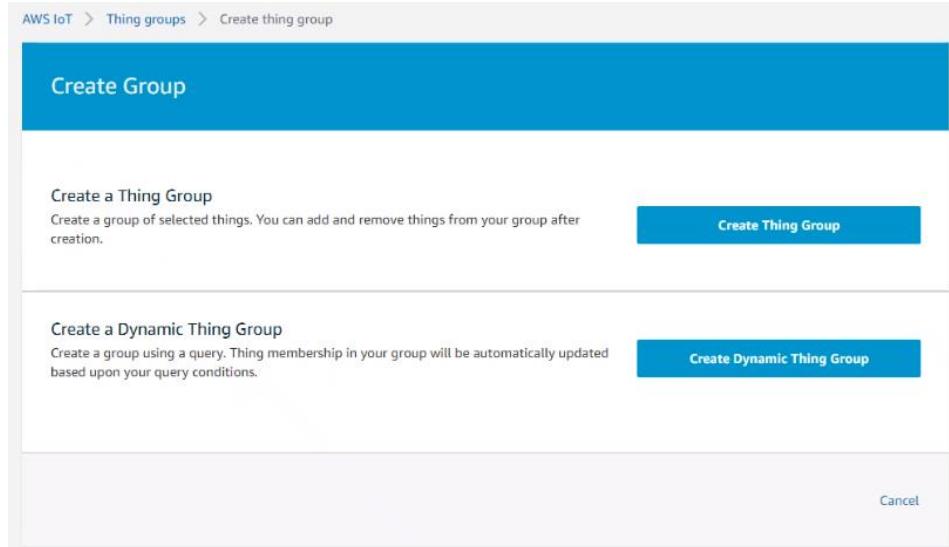
**Make note** of the Public IPv4 DNS address provided and it needs to be in the following format, with a leading HTTP:// as shown below;

Public Access URL: <http://ec2-13-235-8-114.ap-south-1.compute.amazonaws.com/>

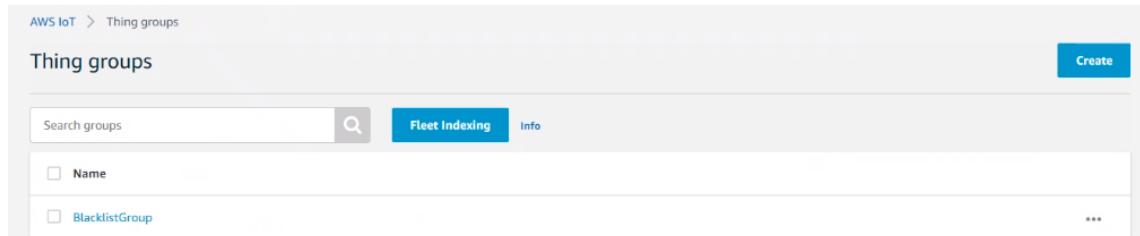
## 4.2 Application – “Allow” or “Deny” listing

1. Create Thing group “BlacklistGroup” and create one default policy “blacklist-policy” then attach policy to thing group.

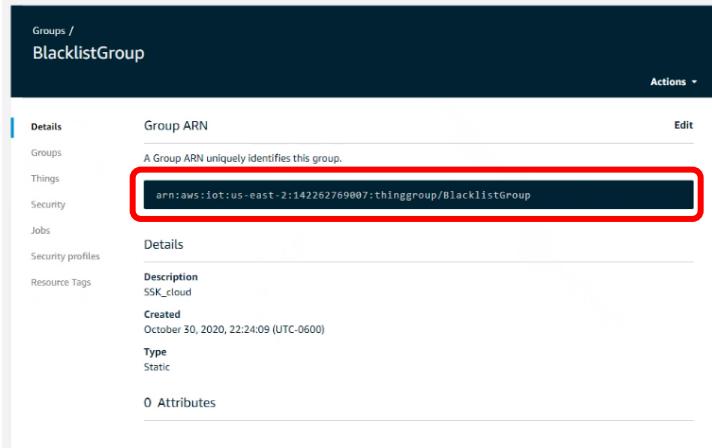
- To create ‘Thing group’ : navigate to IoT Core → Manage → Thing groups → Create



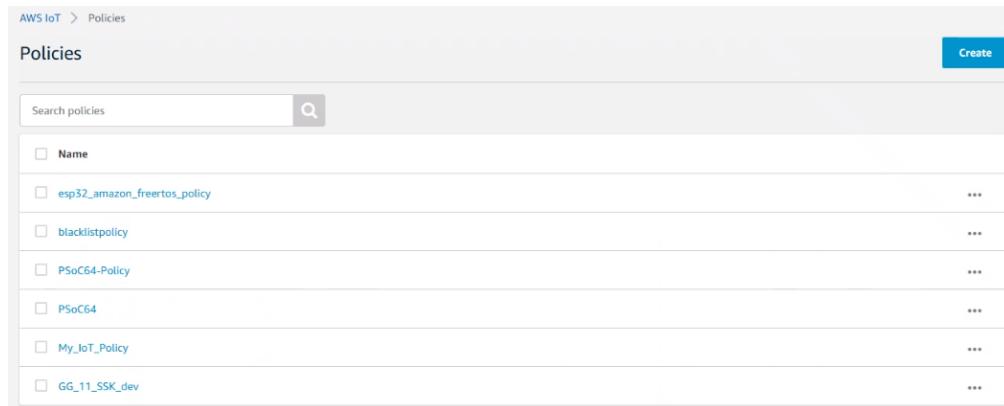
- Provide a name to your Thing Group, like what is shown below.



- Click on “BlacklistGroup” (or whatever name you gave it) and make note of your Group ARN listed below;



2. To create default policy: Navigate to: IoT Core → Secure → Policies



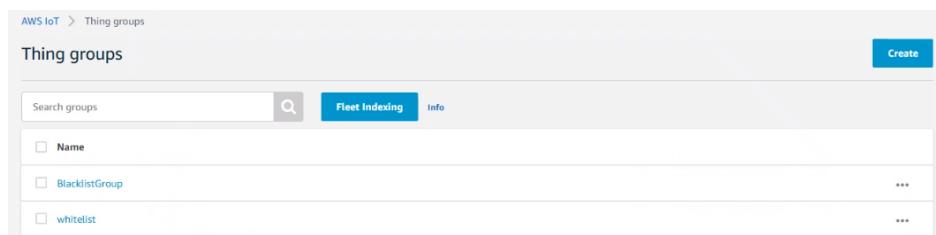
- Click create and enter the name of your policy. Under Add Statements click Advanced mode. JSON statement will be seen and then edit with the information as shown below.

**Note:** You will be entering your specific ARN Group provided in the previous step next to “Resource”;

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "iot:>",
      "Resource": "Group ARN noted from the step above:topic/replaceWithATopic"
    }
  ]
}
```

3. Next, you need to attach the “Policy” to the “Thing Group”;

- Navigate to: IoT Core → Manage → Thing Groups and click on the Group you just created;



The screenshot shows the AWS IoT Groups / whitelist page. The left sidebar has options: Groups, Things, Security, Jobs, Security profiles, and Resource Tags. The main area shows a table with one row:

| Details           | Group ARN   | Actions |
|-------------------|---|---------|
| Groups            | A Group ARN uniquely identifies this group.<br><code>arn:aws:iot:us-east-2:142262769007:thinggroup/whitelist</code> | Edit    |
| Things            |   |         |
| Security          |   |         |
| Jobs              |   |         |
| Security profiles |   |         |
| Resource Tags     |   |         |

Below the table, there are sections for Description (You do not have description for the thing group yet.), Created (November 19, 2020, 09:35:55 (UTC-0700)), Type (Static), and 0 Attributes.

- Click “Security” on the left and then “Edit”, Select the “Policy” you recently created;

The screenshot shows the Policy configuration page for the whitelist group. It has sections for Select a policy to attach to this group and Select another policy to attach to this group. The Select a policy section contains a search bar and a list of policies: esp32\_amazon\_freetos\_policy, blacklistpolicy, PSoC64-Policy, and PSoC6. The Select another policy section shows No policy selected and a Select button. The Directly attached section is empty.

- You should see the policy statements you had edited from the previous steps, then click “Save”

The screenshot shows the AWS IAM Policy editor interface. At the top, it says "Policy". Below that, a message says "Browse and select the policies you want to attach to this group." A section titled "Select a policy to attach to this group" contains a list with "blacklistpolicy" selected. There are "Remove" and "Select" buttons next to the list. Below this, another section titled "Select another policy to attach to this group" shows "No policy selected" with a "Select" button. Under "Directly attached", there is a JSON code block showing a single policy statement:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "iot:*",
      "Resource": "arn:aws:iot:us-east-2:142262769007:topic/replaceWithATopic"
    }
  ]
}
```

At the bottom, there are "Cancel" and "Save" buttons.

The screenshot shows the AWS Groups / whitelist page. The title is "Groups / whitelist". On the left, a sidebar has tabs for Details, Groups, Things, Security (which is selected), Jobs, Security profiles, and Resource Tags. In the main area, under "Policies", it says "blacklistpolicy is attached to this group." Under "Things in this group have the following permissions:", it lists "iot:\*" with an "EXPLICITLY DENIED" note and a link to "arn:aws:iot:us-east-2:142262769007:topic/replaceWithATopic". There is also an "Edit" button.

4. In order to perform OTA updates, the user will need to Create an OTA Role and the link to the instructions within AWS is provided below. You will also need to create an OTA Job, which is part of the SSK Cloud Connect Tool and outlined in Section 5 of the SSK Cloud Connect Users Guide;

To create OTA update role follow below URL:

URL: <https://docs.aws.amazon.com/freertos/latest/userguide/create-service-role.html>

## 4.3 AWS RDS Service – Database Setup and Configuration

- Go to Services >> Database >> RDS(Select)
- Click Left side navigation “Databases” will show following page.

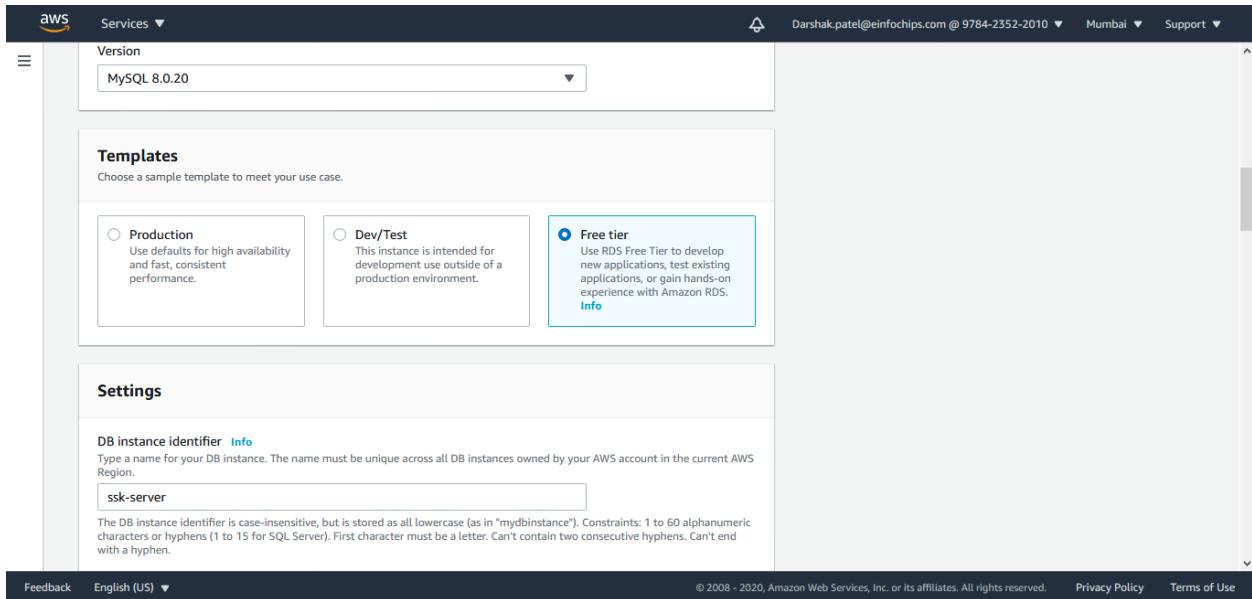
The screenshot shows the AWS RDS service interface. On the left, there's a sidebar with various options like Dashboard, Databases, Query Editor, etc. The main area is titled 'Databases' and shows a table with one row. The row contains the DB identifier 'seed-server', Instance type 'Instance', Engine 'MySQL Community', Region & AZ 'ap-south-1b', Size 'db.t2.micro', and a status indicator 'Normal'. There are buttons for 'Group resources', 'Modify', 'Actions', 'Restore from S3', and a prominent orange 'Create database' button.

## 4.4 Creating a Database

- Select “Standard Create”

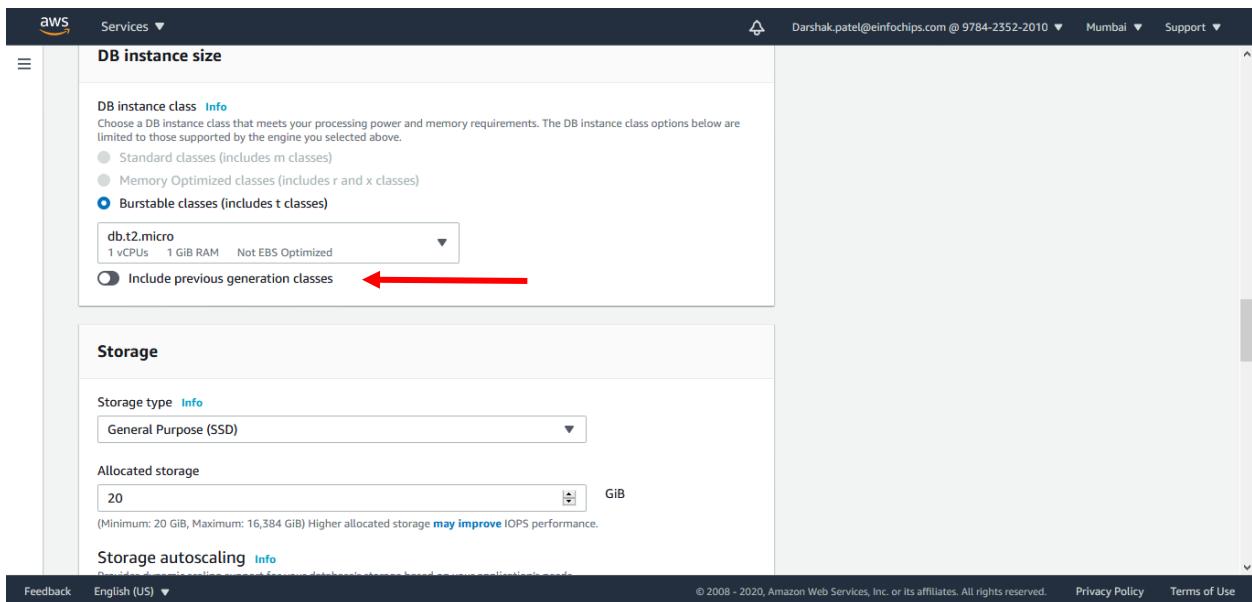
This screenshot shows the 'Choose a database creation method' step. It has two options: 'Standard Create' (selected) and 'Easy Create'. 'Standard Create' allows setting all configuration options, while 'Easy Create' uses recommended best-practice configurations. Below this, the 'Engine options' section is shown, with 'MySQL' selected. Other engine options available are Amazon Aurora, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server. The MySQL option is highlighted with a blue border.

- Select “Free Tier”



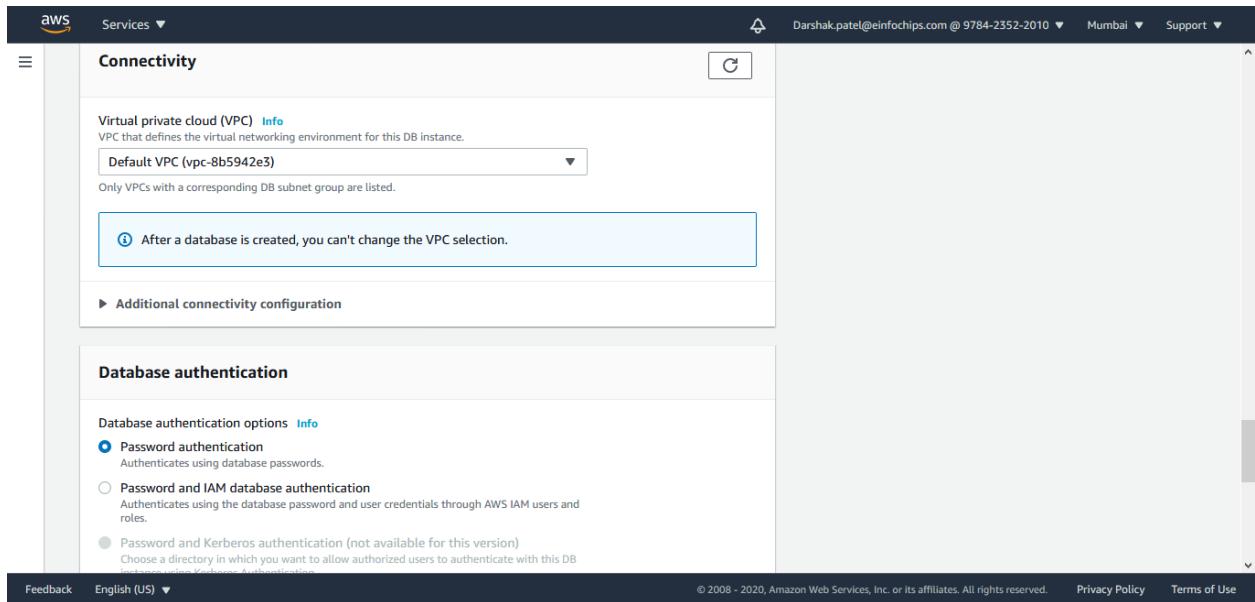
The screenshot shows the AWS RDS MySQL instance creation interface. In the 'Templates' section, the 'Free tier' option is selected, highlighted with a blue border. The 'DB instance identifier' field contains 'ssk-server'. A red arrow points to the 'Free tier' button in the 'Templates' section.

- Enable “Include previous generation classes”

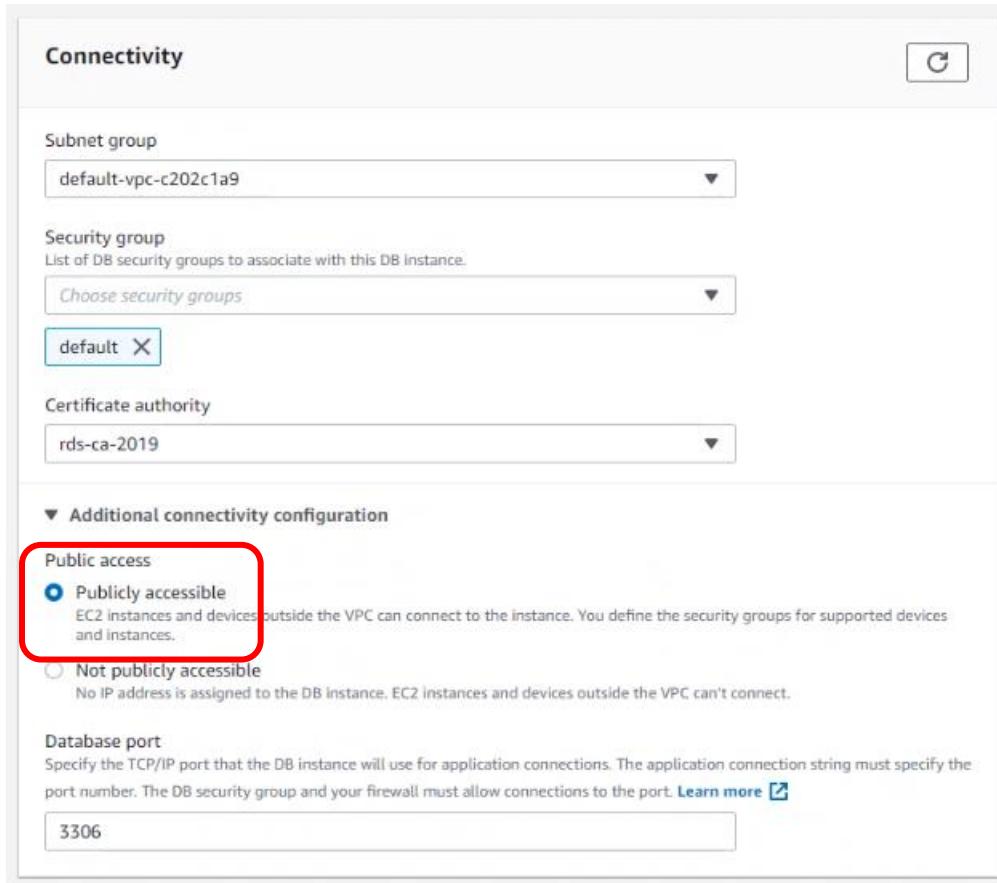


The screenshot shows the 'DB instance size' configuration screen. Under 'DB instance class', the 'Burstable classes (includes t classes)' option is selected. The 'db.t2.micro' class is chosen, which has 1 vCPUs and 1 GiB RAM. A red arrow points to the 'Include previous generation classes' checkbox, which is also selected. Other options like 'Standard classes' and 'Memory Optimized classes' are shown but not selected.

- Select “Default VPC” for the Virtual Private Cloud and “Password Authentication”
- Make note of the database password entered



- Select “Publicly accessible” under Additional Connectivity Configuration



**Connectivity**

Subnet group: default-vpc-c202c1a9

Security group: Choose security groups (default selected)

Certificate authority: rds-ca-2019

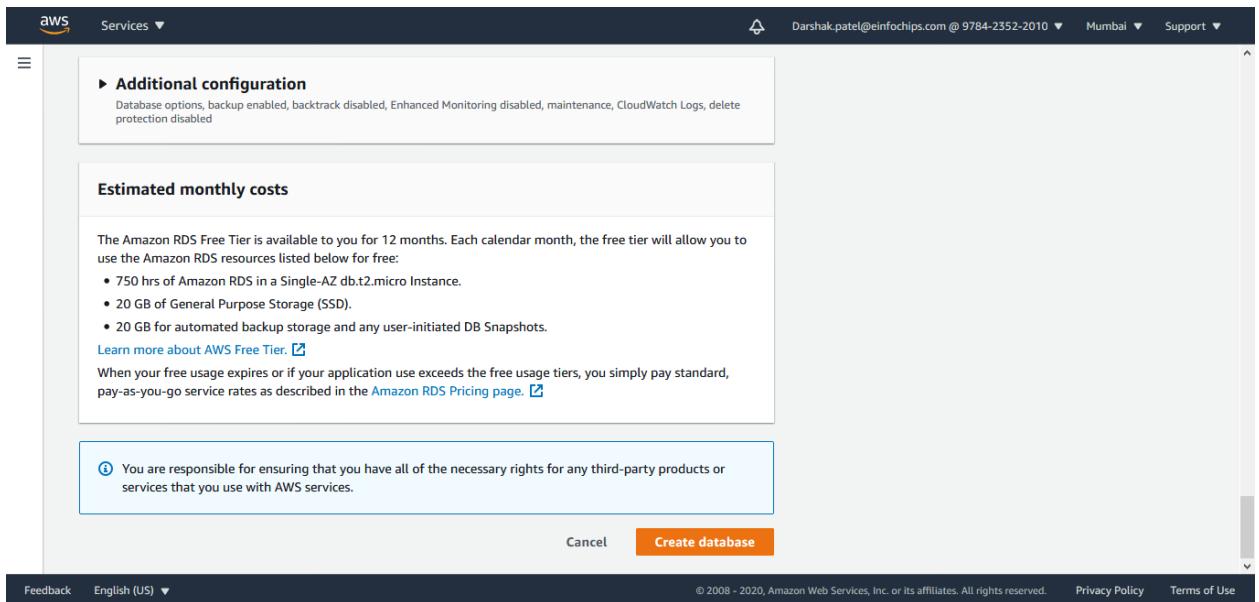
**Additional connectivity configuration**

**Public access**

- Publicly accessible: EC2 instances and devices outside the VPC can connect to the instance. You define the security groups for supported devices and instances.
- Not publicly accessible: No IP address is assigned to the DB instance. EC2 instances and devices outside the VPC can't connect.

Database port: 3306

- Click “Create database”



aws Services Darshak.patel@einfochips.com @ 9784-2352-2010 Mumbai Support

**Additional configuration**  
Database options, backup enabled, backtrack disabled, Enhanced Monitoring disabled, maintenance, CloudWatch Logs, delete protection disabled

**Estimated monthly costs**

The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

Learn more about AWS Free Tier.

When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the [Amazon RDS Pricing page](#).

**Important**: You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

**Create database**

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

## SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

The screenshot shows the AWS RDS console for the 'ssk-server' database. The 'Connectivity & security' tab is selected. A yellow box highlights the database endpoint URL 'ssk-server.c0zdybumpk6e.us-east-2.rds.amazonaws.com'.

| Endpoint  | Networking   | Security   |
|---|--|--|
| ssk-server.c0zdybumpk6e.us-east-2.rds.amazonaws.com<br>3306 | Availability zone: us-east-2c<br>VPC: vpc-c202c1a9<br>Subnet group: default-vpc-c202c1a9<br>Subnets: subnet-8aff77c2, subnet-291ff442, subnet-fbac8d81 | VPC security groups: default (sg-9f61e6f8) (active)<br>Public accessibility: Yes<br>Certificate authority: rds-ca-2019<br>Certificate authority date: Aug 22nd, 2024 |

- Make note of the RDS URL that is created and highlighted above.

The screenshot shows the AWS RDS console for the 'ssk-server' database. The 'Configuration' tab is selected. A yellow box highlights the 'Master username' field, which contains 'admin\_jq'.

| Configuration   | Instance class  | Storage  | Performance Insights             |
|---|---|--|----------------------------------|
| DB instance id: ssk-server<br>Engine version: 8.0.20<br>DB name: -<br>License model: General Public License<br>Option groups: default:mysql-8-0 | Instance class: db.t2.micro<br>vCPU: 1<br>RAM: 1 GB<br>Availability: Master username: admin_jq<br>IAM db authentication | Encryption: Not Enabled<br>Storage type: General Purpose (SSD)<br>IOPS: -<br>Storage: 20 GiB<br>Storage autoscaling: Enabled | Performance Insights enabled: No |

- Make note of the User Name highlighted above, under the configurations tab.

- Once MySQL Database is created, ensure the below Security group rules are set by clicking on the default security group under “Security group rules” in Amazon RDS

The screenshot shows the AWS Amazon RDS instance configuration page for a MySQL Community instance. The left sidebar lists various options like Dashboard, Databases, Query Editor, etc. The main panel has tabs for Connectivity & security, Monitoring, Logs & events, Configuration, Maintenance & backups, and Tags. The Connectivity & security tab is selected. It displays endpoint and port details (Endpoint: ssk-server.czqdvbumpk6e.us-east-2.rds.amazonaws.com, Port: 3306), networking information (Availability zone: us-east-2c, VPC: vpc-c202c1a9, Subnet group: default-vpc-c202c1a9, Subnets: subnet-8eff77c2, subnet-291ff442, subnet-fbac8d81), and security details (VPC security groups: default (sg-9f61e6f8) active, Public accessibility: Yes, Certificate authority: rds-ca-2019, Certificate authority date: Aug 22nd, 2024). Below this, the Security group rules section shows two entries:

| Security group        | Type               | Rule      |
|-----------------------|--------------------|-----------|
| default (sg-9f61e6f8) | CIDR/IP - Inbound  | 0.0.0.0/0 |
| default (sg-9f61e6f8) | CIDR/IP - Outbound | 0.0.0.0/0 |

## Inbound Rules:

The screenshot shows the AWS EC2 Security Groups page. The URL indicates we are editing the inbound rules for the 'sg-9f61e6f8 - default' security group. The page title is 'Edit inbound rules'. It shows two existing inbound rules:

| Rule Number | Type        | Protocol | Port range | Action |
|-------------|-------------|----------|------------|--------|
| 1           | All traffic | All      | All        | Allow  |
| 2           | All traffic | All      | All        | Allow  |

Below the rules, there is a note: "⚠ NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created." At the bottom, there are 'Cancel', 'Preview changes', and a prominent orange 'Save rules' button.

## Outbound Rules:

**Outbound rules**

**Outbound rule 1**

|                                       |  |   |
|---------------------------------------|--|---|
| Type <a href="#">Info</a>             | Protocol <a href="#">Info</a>                            | Port range <a href="#">Info</a>             |
| All traffic                           | All  | All   |
| Destination type <a href="#">Info</a> | Destination <a href="#">Info</a>                         | Description - optional <a href="#">Info</a> |
| Custom                                | <input type="text" value="0.0.0.0/0"/> <a href="#">X</a> |   |

**Outbound rule 2**

|                                       |  |   |
|---------------------------------------|--|---|
| Type <a href="#">Info</a>             | Protocol <a href="#">Info</a>                                | Port range <a href="#">Info</a>             |
| All traffic                           | All  | All   |
| Destination type <a href="#">Info</a> | Destination <a href="#">Info</a>                             | Description - optional <a href="#">Info</a> |
| Custom                                | <input "::="" 0"="" type="text" value=""/> <a href="#">X</a> |   |

[Add rule](#)

**NOTE:** Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

[Cancel](#) [Preview changes](#) [Save rules](#)

## 4.5 Creating an IAM User

- In AWS go to Services >> IAM >> Users and select ‘Add User’

**Identity and Access Management (IAM)**

[Add user](#) [Delete user](#)

|                          | User name | Groups        | Access key age | Password age |
|--------------------------|-----------|---------------|----------------|--------------|
| <input type="checkbox"/> | ADMIN     | Administrator | ⚠ 307 days     | 307 days     |
| <input type="checkbox"/> | root_user | Administrator | ✓ 4 days       | None         |
| <input type="checkbox"/> | User_Name | Administrator | ✓ Today        | None         |

## SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

- Pick a username and give it Programmatic access. Note this will be the username you will use to log into SSK Cloud Connect

The screenshot shows the 'Add user' wizard in the AWS IAM console. Step 1: Set user details. A 'User name' field contains 'NAME'. Below it, under 'Access type', 'Programmatic access' is selected. Other options like 'AWS Management Console access' are available but not selected.

- Choose 'Next: Permissions' and add your IAM user to the Administrator group with the Administrator Access policy. If you don't have an Administrator group then choose "Attach existing policies directly, search for the 'Administrator Access' policy and attach it

The screenshot shows the 'Add user' wizard in the AWS IAM console. Step 2: Set permissions. The 'Set permissions' section is open, showing three options: 'Add user to group', 'Copy permissions from existing user', and 'Attach existing policies directly'. The 'Add user to group' option is selected. Below it, the 'Add user to group' section shows the 'Administrator' group selected, with the 'AdministratorAccess' policy attached. There is also a 'Create group' button.

- Click Next: Tags >> Next: Review >> Create user

Add user

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

|                      |  |
|----------------------|--|
| User name            | NAME                                     |
| AWS access type      | Programmatic access - with an access key |
| Permissions boundary | Permissions boundary is not set          |

Permissions summary

The user shown above will be added to the following groups.

| Type  | Name          |
|-------|---------------|
| Group | Administrator |

Tags

No tags were added.

[Cancel](#) [Previous](#) [Create user](#)

- Download the '**new\_user\_credentials.csv**' and save it in a safe location

## 4.6 MySQL Setup and Configuration

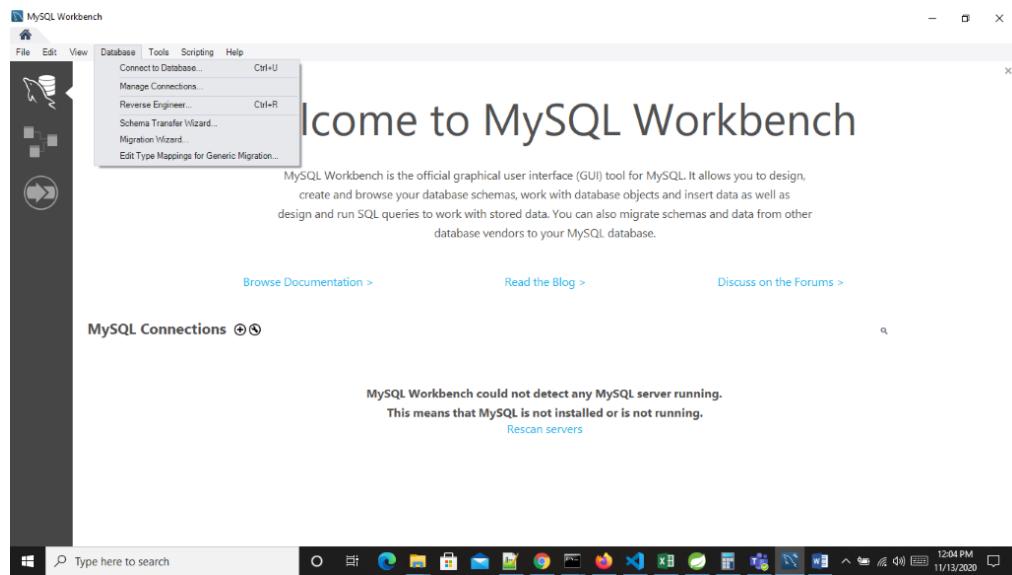
1. Install MySQL Workbench (link provided below), then Open MySQL workbench.

<https://dev.mysql.com/downloads/workbench/>

2. Install Postman

<https://www.postman.com/>

3. On Tab Database & select Manage connections.(Database -> Manage Connections)



4. It will open pop up model. Press New Button then fill up details as per below:

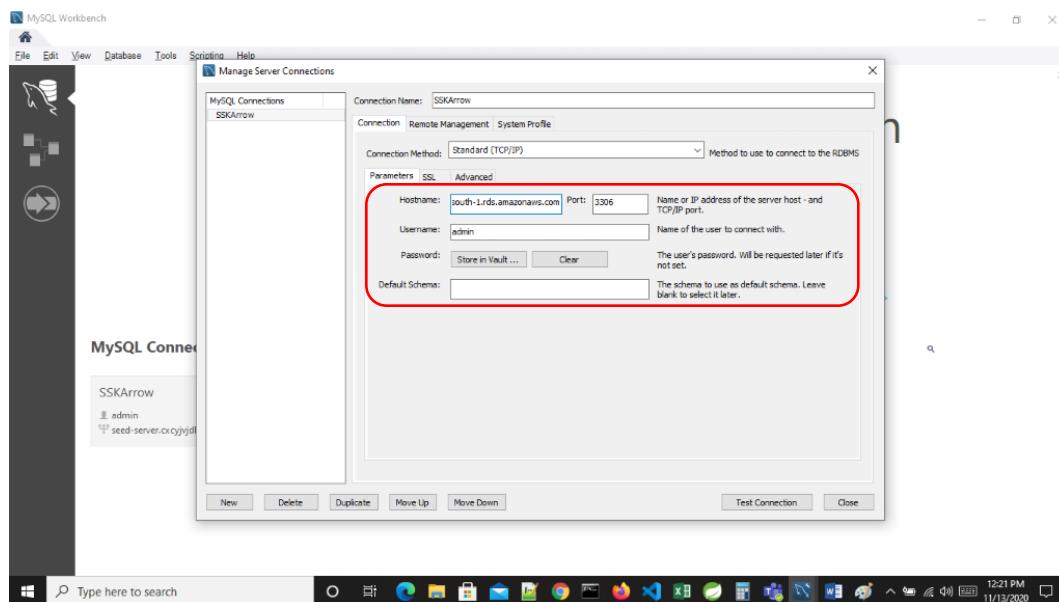
**Connection Name:** <Name of connection>

**Host Name:** <AWS RDS HOST URL>

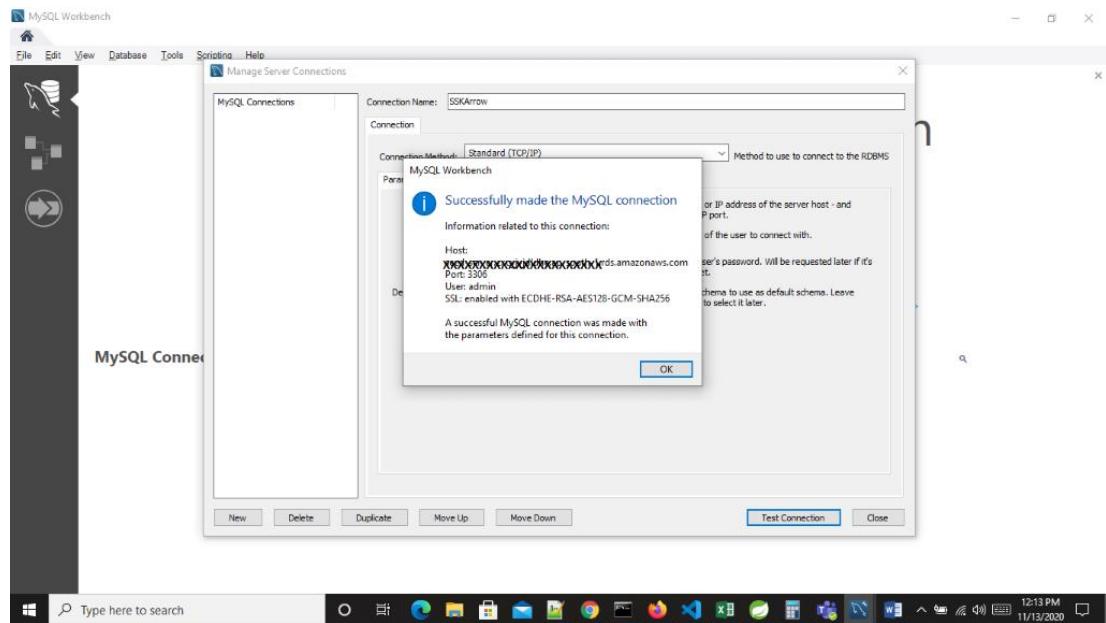
**Port:** 3306 (Default value)

**User Name:** < AWS RDS User name>

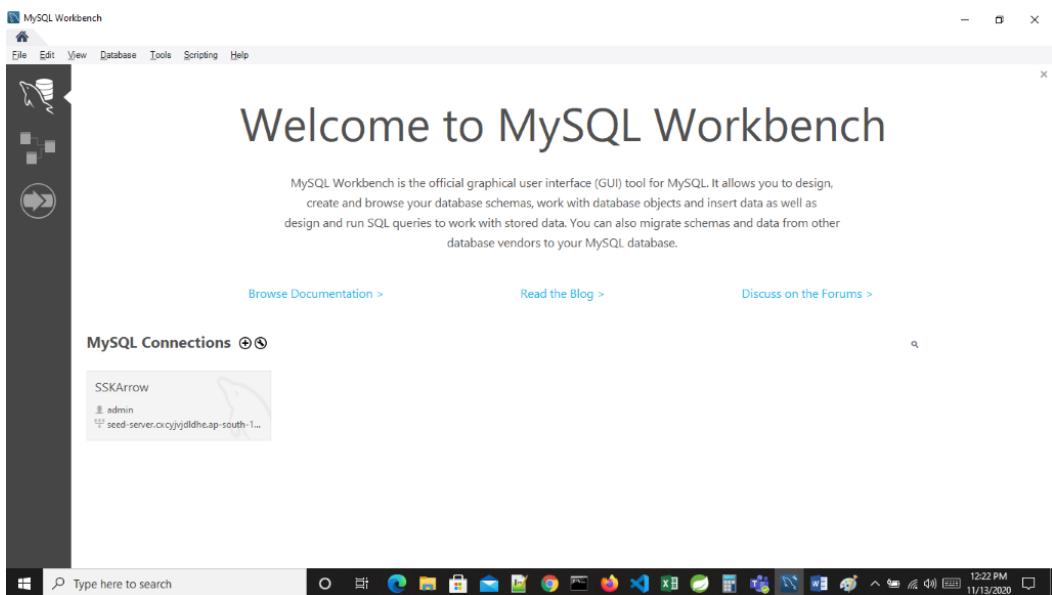
**Password:** <AWS RDS user password> (Store in Vault if needed)



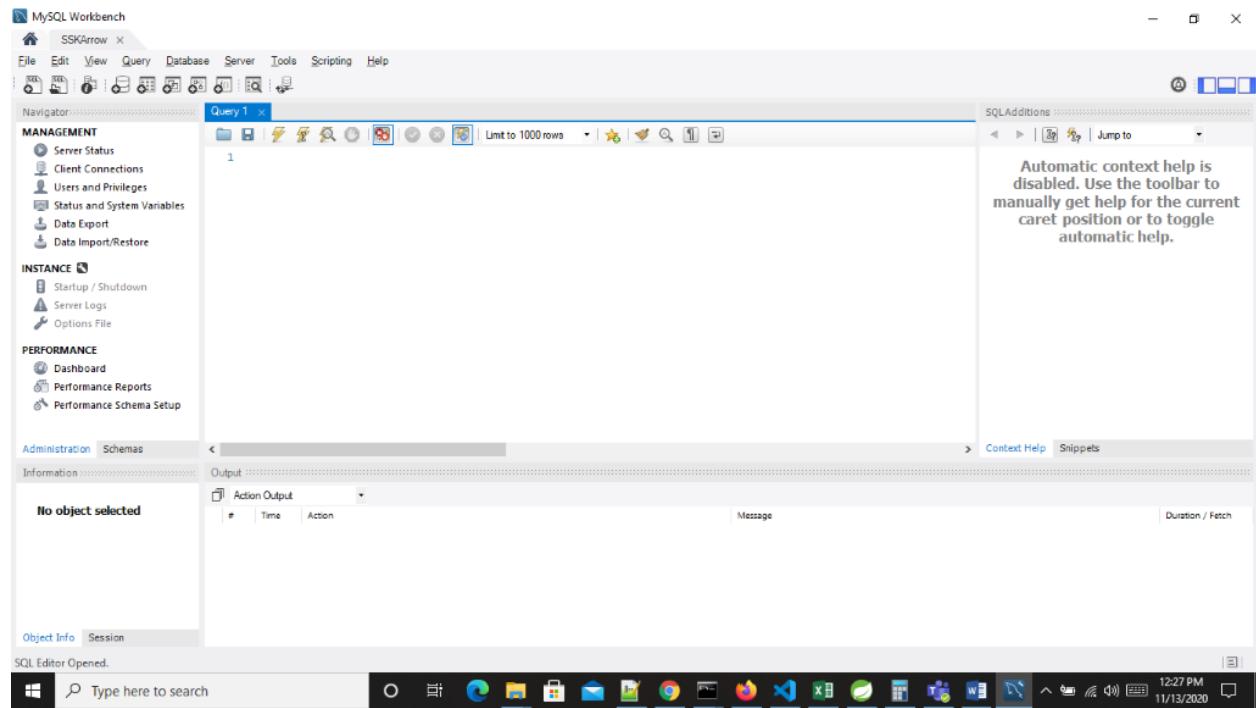
5. Press on Test Connection. It will pop up successful connection message. Then click on close button



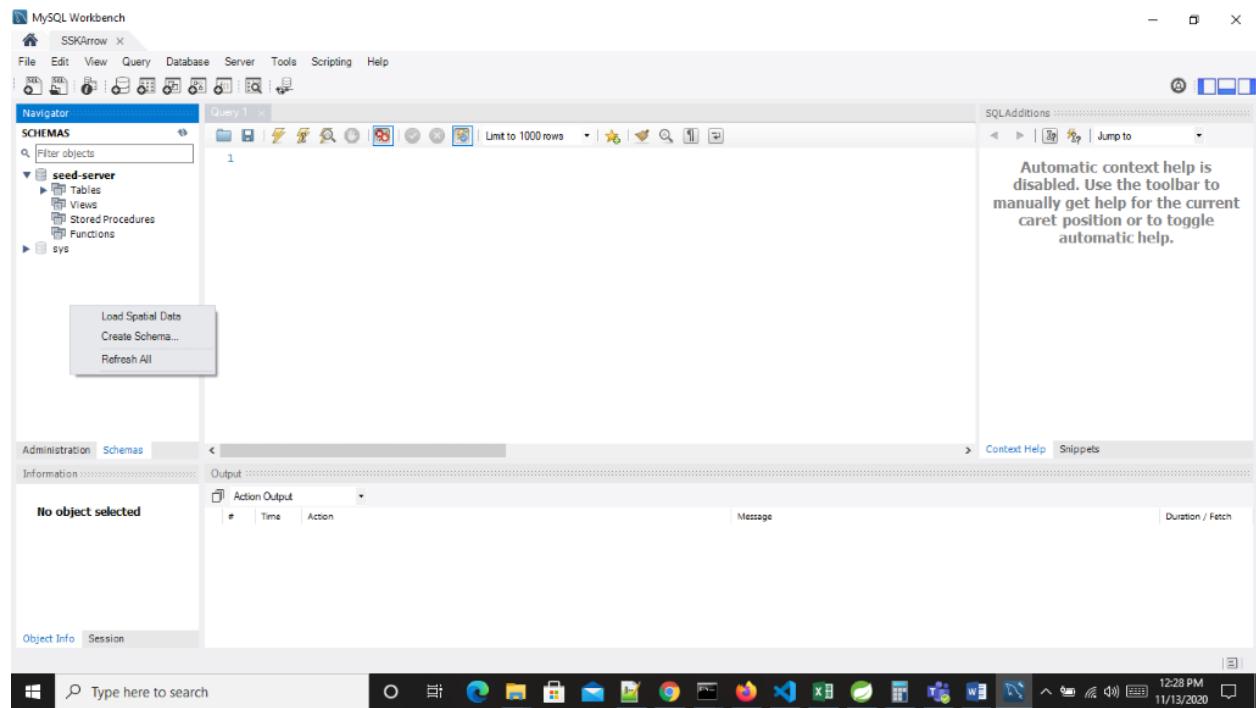
6. Further it will show following details. Click on created connection button.



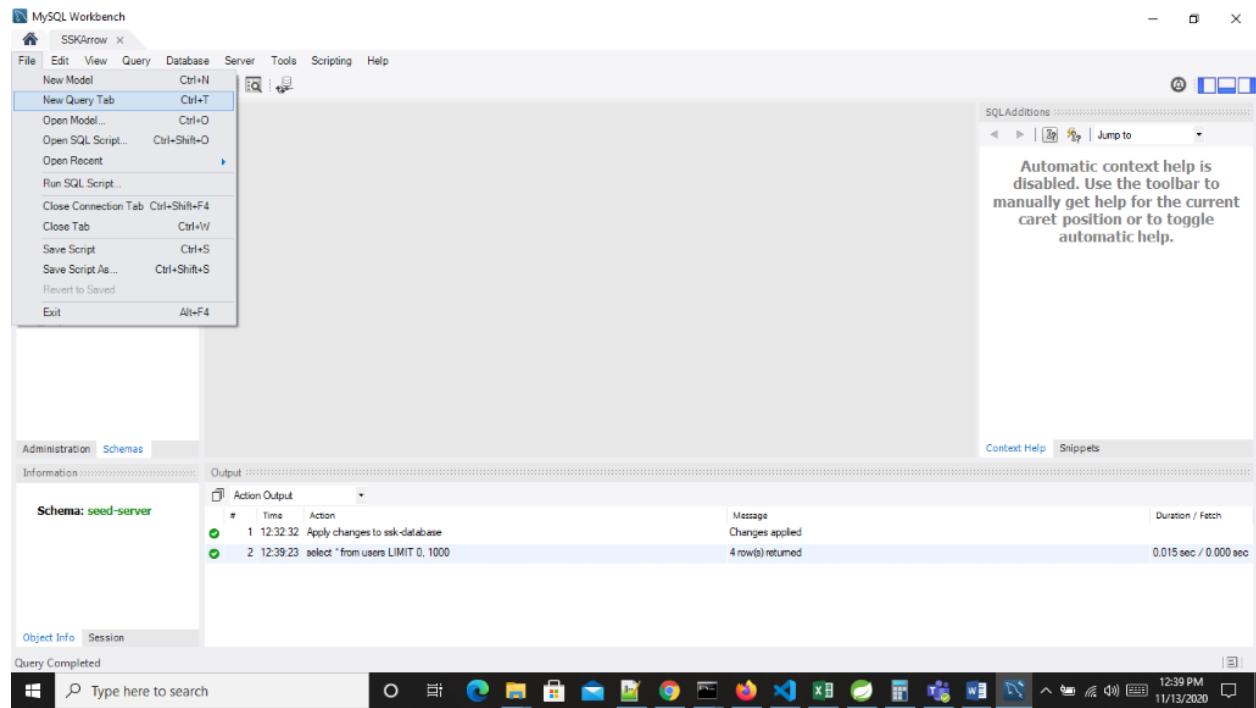
7. It will open Administration tab.



8. Select Schemas Tab & right click on mouse.



9. Click on File Tab then select “New Query Tab”



10. Edit and Execute schema.sql

- Go to File ->Open SQL Script..., navigate to the location of 'schema.sql' on your PC, and select it
- Modify lines 6 and 7 and choose a unique name for the schema database, for example `ssk-database`
- **Make note** of this name
- Highlight lines 6 and 7 as shown below and click the yellow bolt one time to execute the



selected lines in schema.sql

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigator schema data
SCHEMAS
Filter objects
ssk-database
Tables
awsregion
awsuser_token
roles
user_roles
userpassword_history
users
Views
Stored Procedures
Functions
Limit to 1000 rows
1
2  /*revoke all on db_example.* from 'springuser'@'localhost';
3  grant select, insert, delete, update on db_example.* to 'springuser'@'localhost';
4  */
5  /*
6  • CREATE DATABASE IF NOT EXISTS `ssk-database`;
7  • USE `ssk-database`;
8  */
9  • CREATE TABLE `roles` (
10    `id` int NOT NULL AUTO_INCREMENT,
11    `name` varchar(255) DEFAULT NULL,
12    `role_type` varchar(64) NOT NULL,
13    PRIMARY KEY (`id`),
14    UNIQUE KEY `uk_roles_role_type` (`role_type`)
15  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
16  /*
17  • CREATE TABLE `users` (
18    `userid` varchar(255) NOT NULL,
19    `user_name` varchar(64) NOT NULL,
20    `password` varchar(68) DEFAULT NULL,
21    `reset_token` varchar(255) DEFAULT NULL,
22    `token_expiry_date` datetime(6) DEFAULT NULL,
23    `created_time` datetime(6) DEFAULT NULL,
24    `modified_time` datetime(6) DEFAULT NULL,
25    PRIMARY KEY (`userid`),
26    UNIQUE KEY `uk_users_username` (`user_name`)
27  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
28  /*
29  • CREATE TABLE `user_roles` (
30    `user_id` varchar(255) NOT NULL,
31    `role_id` int NOT NULL,
32    PRIMARY KEY (`user_id`,`role_id`),
33    KEY `FK_user_roles_roles` (`role_id`),

```

## 11. Edit and Execute data.sql

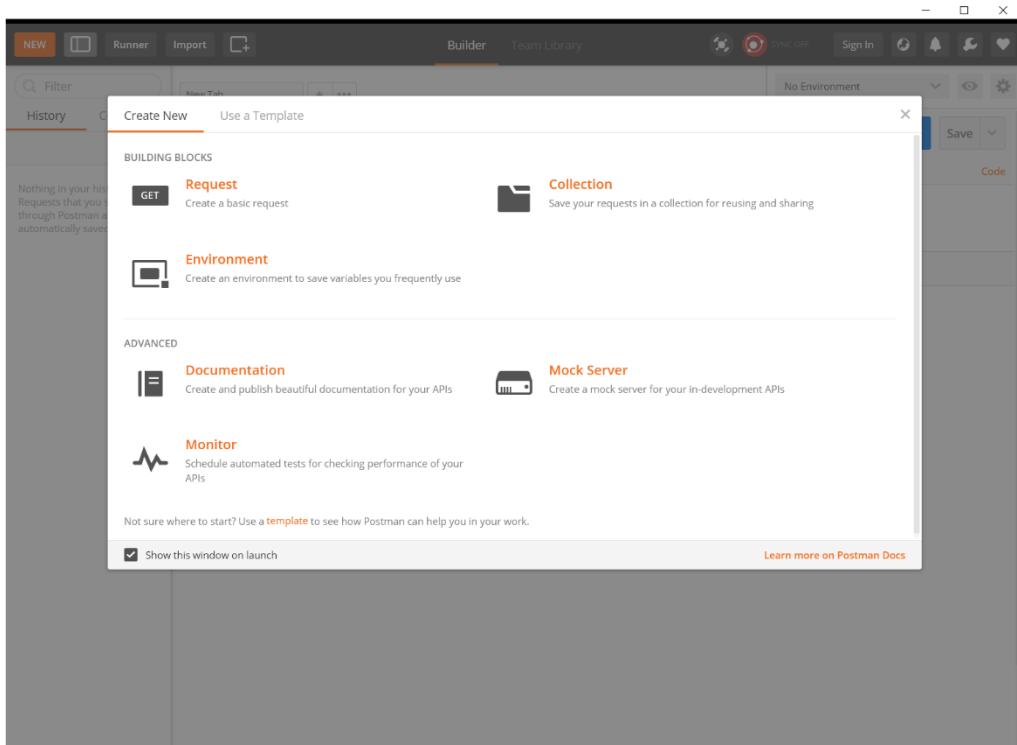
- Go to File->Open SQL Script..., navigate to the location of data.sql on your PC, and select it
- Locate the User name, Access key ID, and Secret access key of your IAM user. Note these credentials can be found in the ‘new\_user\_credentials.csv’ file that was downloaded after creating an IAM user
- Modify line 18 of data.sql by entering your IAM credentials. It should have a similar structure shown below

```

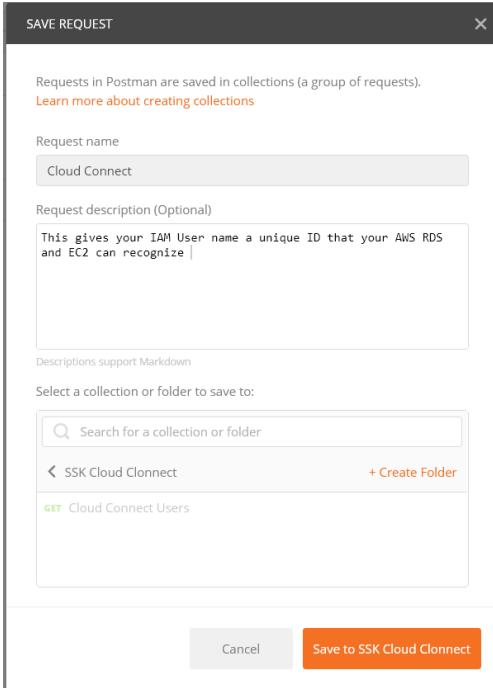
INSERT IGNORE INTO
awsuser_token(user_name,access_key_id,secret_access_key,is_root_account,create_date)
values('<your_iam_username>','<your_access_key>','<your_secret_key>',1,now());

```

- Launch the Postman app
- Under the ‘Create New’ tab select ‘Request’



- Enter a request name, create a new collection to save the request to, and save it



- Under the 'Authorization' tab select 'AWS Signature' next to Type and enter your IAM credentials into the 'AccessKey' and 'SecretKey' text boxes
- Next to 'Get' <Enter request URL> copy and paste the below URL and change the highlighted text with the username of your IAM

<https://iam.amazonaws.com/?Action= GetUser&UserName=IAM-username&Version=2010-05-08>

- Leave ‘AWS Region’ empty and enter ‘iam’ (all lowercase) next to Service Name as shown below

The screenshot shows the Postman Builder interface. A GET request is defined with the URL <https://iam.amazonaws.com/?Action= GetUser&UserName=IAM-username&Version=2010-05-08>. The 'Authorization' tab is active, set to 'AWS Signature'. Other tabs include Headers (4), Body, Pre-request Script, and Tests. The 'Service Name' field is set to 'iam'. The left sidebar shows a history of requests, including one for 'Cloud Connect Users'.

- Now choose ‘Send’ and copy the ID that was generated below next to ‘<UserId>’

The screenshot shows the Postman Test Results tab displaying the XML response of the 'GetUserResponse' request. The XML structure includes 'GetUserResult' and 'User' elements. The 'UserId' element at line 7 contains the value 'AIDASCH4FSVXRM7SB0Z5T'. The status is 200 OK and the time taken is 437 ms.

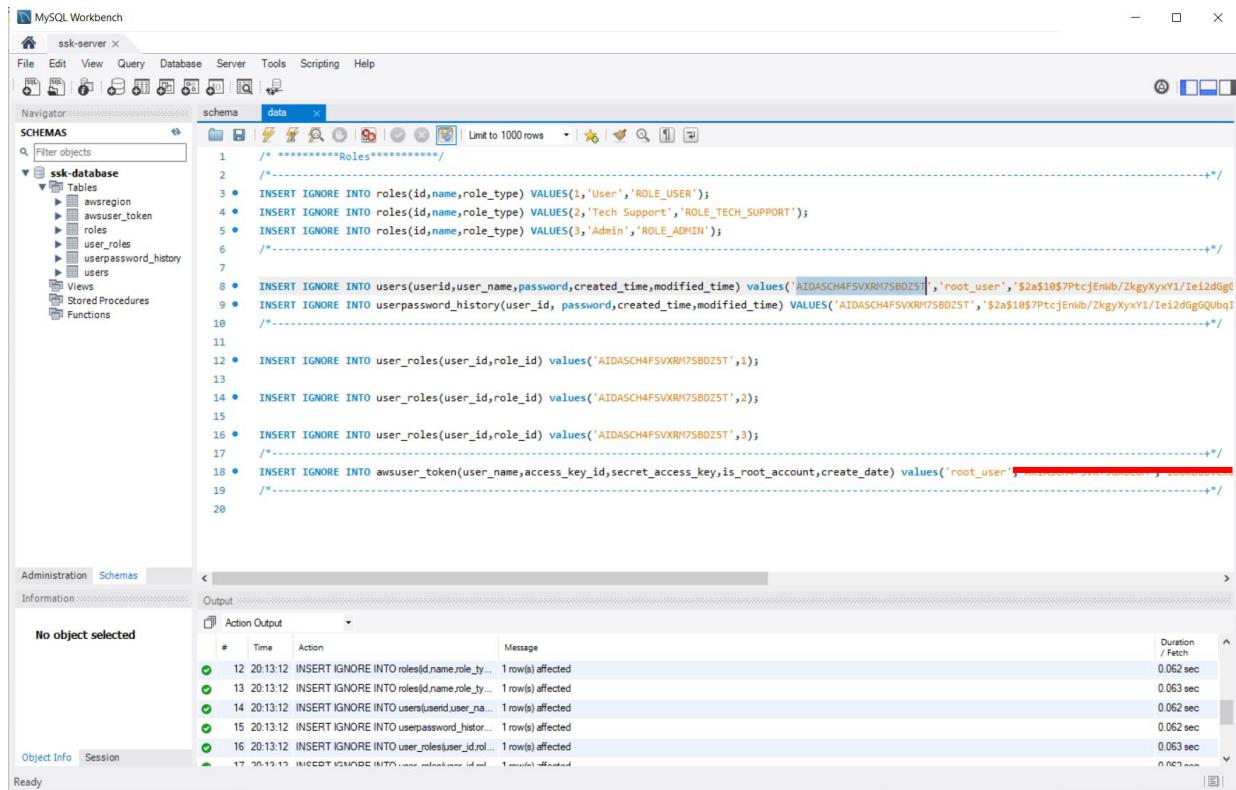
```

1 <GetUserResponse xmlns="https://iam.amazonaws.com/doc/2010-05-08">
2   <GetUserResult>
3     <User>
4       <Path>/</Path>
5       <Arn>arn:aws:iam::142262769007:user/root_user</Arn>
6       <UserName>root_user</UserName>
7       <UserId>AIDASCH4FSVXRM7SB0Z5T</UserId>
8       <CreateDate>2020-11-20T00:43:07Z</CreateDate>
9     </User>
10    </GetUserResult>
11    <ResponseMetadata>
12      <RequestId>0ee95db8-5084-4705-a8fd-5da81fd6522f</RequestId>
13    </ResponseMetadata>
14  </GetUserResponse>

```

- In MySQL Workbench modify data.sql by replacing the generic User ID in lines 8, 9, 12, 14, and 16 with the User ID that was created above
- In line 8 also replace ‘seed.dev@einfochips.com’ with the username of your IAM user

## SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE



```

1  /* *****Roles***** */
2  /*
3   * INSERT IGNORE INTO roles(id,name,role_type) VALUES(1,'User','ROLE_USER');
4   * INSERT IGNORE INTO roles(id,name,role_type) VALUES(2,'Tech Support','ROLE_TECH_SUPPORT');
5   * INSERT IGNORE INTO roles(id,name,role_type) VALUES(3,'Admin','ROLE_ADMIN');
6  */
7
8   * INSERT IGNORE INTO users(userid,user_name,password,created_time,modified_time) values('AIDASCH4FSVXR75BDZ5T','root_user','$2a$10$7PtcjEnlb/ZkgvXyxY1/Iei2dgg0');
9   * INSERT IGNORE INTO userpassword_history(user_id, password,created_time,modified_time) VALUES('AIDASCH4FSVXR75BDZ5T','$2a$10$7PtcjEnlb/ZkgvXyxY1/Iei2dgg6QUbqI
10 */
11
12   * INSERT IGNORE INTO user_roles(user_id,role_id) values('AIDASCH4FSVXR75BDZ5T',1);
13
14   * INSERT IGNORE INTO user_roles(user_id,role_id) values('AIDASCH4FSVXR75BDZ5T',2);
15
16   * INSERT IGNORE INTO user_roles(user_id,role_id) values('AIDASCH4FSVXR75BDZ5T',3);
17 */
18   * INSERT IGNORE INTO awouser_token(user_name,access_key_id,secret_access_key,is_root_account,create_date) values('root_user','AIDASCH4FSVXR75BDZ5T','$2a$10$7PtcjEnlb/ZkgvXyxY1/Iei2dgg6QUbqI
19 */
20

```

No object selected

| #  | Time     | Action                                      | Message           | Duration  |
|----|----------|---|-------------------|-----------|
| 12 | 20:13:12 | INSERT IGNORE INTO roles(id,name,role_ty... | 1 row(s) affected | 0.062 sec |
| 13 | 20:13:12 | INSERT IGNORE INTO roles(id,name,role_ty... | 1 row(s) affected | 0.063 sec |
| 14 | 20:13:12 | INSERT IGNORE INTO users(userid,user_na...  | 1 row(s) affected | 0.062 sec |
| 15 | 20:13:12 | INSERT IGNORE INTO userpassword_hist...     | 1 row(s) affected | 0.062 sec |
| 16 | 20:13:12 | INSERT IGNORE INTO user_roles(user_id,ro... | 1 row(s) affected | 0.063 sec |
| 17 | 20:13:12 | INSERT IGNORE INTO user_roles(user_id,ro... | 1 row(s) affected | 0.062 sec |

- Execute 'data.sql' one time by clicking the yellow bolt

## 5 CONFIGURE IMAGE ON DOCKER AND EC2

### 5.1 Execute below commands in PuTTY

Step 1: Check “seed-server” container exists and running

```
$sudo docker ps -a
```

Step 2: Stop server & delete container (Only execute if exists & running otherwise skip it)

```
$sudo docker stop ssk-server
$sudo docker rm ssk-server
```

Step 3: Check “seed-server” image exists

```
$sudo docker images
```

Step 4: Delete image (Only execute if exists otherwise skip it)

```
$sudo docker rmi arrowelectronics/ssk
```

Step 5: Pull latest image

```
$sudo docker login -u <username> -p <password>
$sudo docker pull arrowelectronics/ssk:latest
```

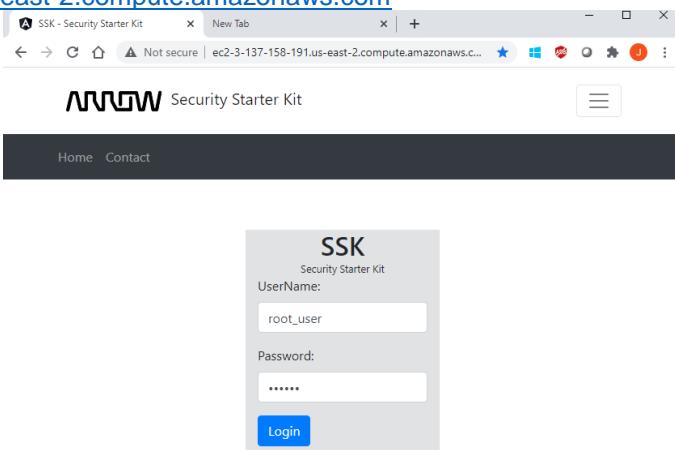
Step 6: Run image (Change highlighted text with your own values)

```
$sudo docker run --name ssk-server -e
MYSQL_URL="jdbc:mysql://<RDS_ENDPOINT>:3306/<SCHEMA_NAME>?useSSL=false&serverTimezone=UTC&useLegacyDatetimeCode=false&allowPublicKeyRetrieval=true" -e MYSQL_UNAME=<RDS_UserName>" -e
MYSQL_PASSWORD=<RDS_Password>" -d -p 80:8080 -v
/home/ubuntu/seedserver:/var/lib/ arrowelectronics/ssk
```

where <SCHEMA\_NAME> is the name configured in schema.sql and <RDS\_ENDPOINT>, <RDS\_UserName>, and <RDS\_Password> are the same values that were used to initially connect to MySQL Workbench

### 5.2 Log into SSK Cloud Connect

- Open a web browser and enter the URL that was noted from section 4.1
  - Note this is the Public IPv4 DNS address of your EC2 Instance i.e. <http://ec2-3-137-158-191.us-east-2.compute.amazonaws.com>



- Enter the UserName which is the name of the IAM username configured in ‘data.sql’ i.e. ‘root\_user’
- Enter the default password: ArrowSSKportal@2020
- You can now login to your freshly installed SSK Cloud Connect Dashboard!
- Please refer to the [SSK\\_Cloud\\_Connect\\_Users\\_Guide.pdf](#) for configuring the different AWS services within the Arrow SSK Cloud Connect web-based tool.

## 6 CHECKOUT PROJECT

1. Execute following command

```
git clone ssh://<name>@git.einfochips.com:29418/secure-end-to-end-device  
git checkout seed-server
```

2. Manage permission regarding project

URL: <https://git.einfochips.com:8080/q/status:open>

## 7 BUILD PROJECT

### 7.1 Execute below command

1. Build Angular project (Go to /secure-end-to-end-device/seed-client)  
ng build --prod
2. Copy build file under static folder  
From path /secure-end-to-end-device/seed-client/dist/seed-client/  
To  
/secure-end-to-end-device/seed-cloud/seed-server/src/main/resources/static
3. To Build Spring boot application execute following command  
.gradlew clean build
4. To Build image (Go to /secure-end-to-end-device/seed-cloud/seed-server)  
docker build -t arrowelectronics/ssk:latest .
5. Push image to Docker hub  
docker push arrowelectronics/ssk:latest

## 8 REFERENCES

- [1] [https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2\\_GetStarted.html](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html)
- [2] <https://docs.docker.com/engine/install/ubuntu/>
- [3] <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-18-04>
- [4] <https://aws.amazon.com/console/>