

Secure Starter Kit Cloud Connect Installation & Setup Guide

Date: March 03, 2021 | Version 1.1



The Solutions People



CONTENTS

1	INTRODUCTION.....	3
1.1	Purpose of the Document	3
1.2	Prerequisites, Background information & AWS Cloud Services Descriptions	3
2	AWS ACCOUNT CREATION & SETUP EC2 SERVICE	5
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
3	INSTALLING DOCKER ON EC2	14
3.1	Execute below command	14
4	CONFIGURATION OF EC2 INSTANCE, RDS SERVICE AND SQL DATABASE.....	15
4.1	Application access.....	15
4.2	Application – “Allow” or “Deny” listing.....	16
4.3	AWS RDS Service – Database Setup and Configuration	20
4.4	Creating a Database	20
4.5	Creating an IAM User	26
4.6	MySQL Setup and Configuration	28
5	CONFIGURE IMAGE ON DOCKER AND EC2	37
5.1	Execute below commands in PuTTY.....	37
5.2	Log into SSK Cloud Connect	37
6	CHECKOUT PROJECT	39
7	BUILD PROJECT	40
7.1	Execute below command	40
8	REFERENCES	41

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.

- *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.*

3. **AWS Relational Database Service (Amazon RDS)** – <https://aws.amazon.com/rds/>

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.

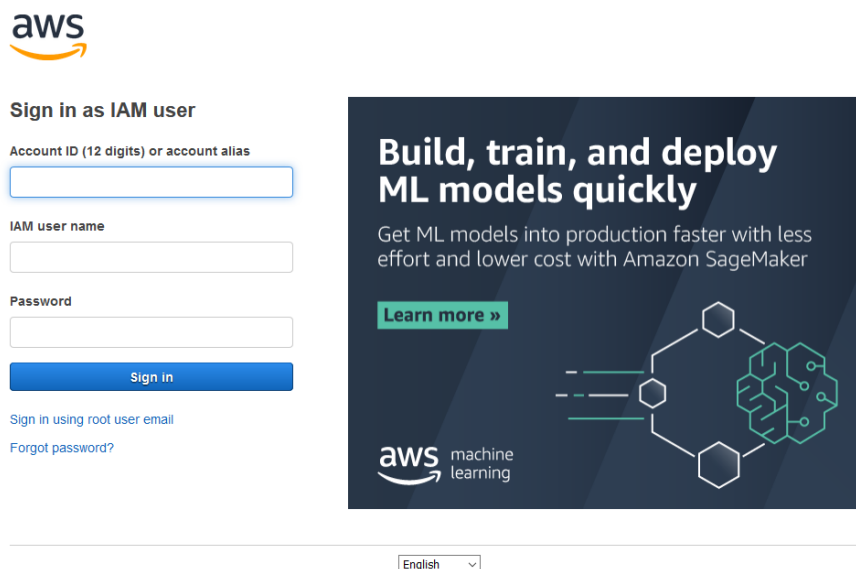
- *OTA setup and configuration is listed in Section 5 of the SSK Cloud Connect Users Guide.*
- *These steps do not need to be completed to run the demo outlined in the Quick Start Guide, but will need to be configured in order to perform OTA firmware updates from within AWS Cloud Services.*

2 AWS ACCOUNT CREATION & SETUP EC2 SERVICE

2.1 Login or Create your AWS Account

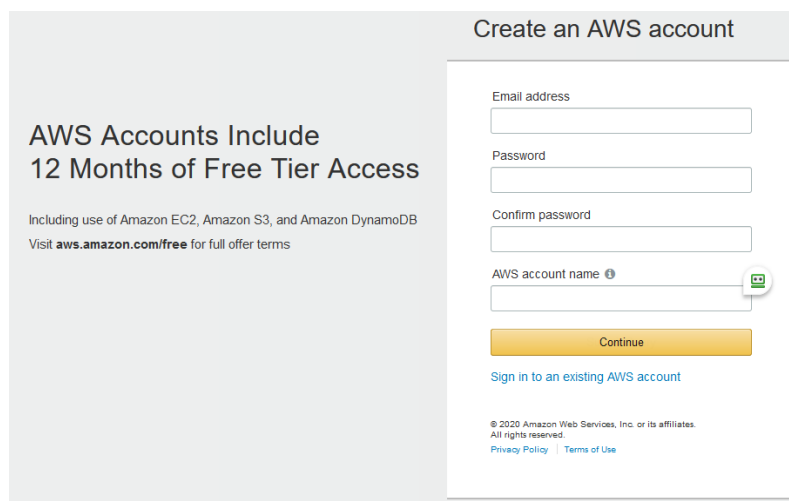
Note: If the User does not have an AWS Account, you will need to create one and this is used as the basis for the configuration of the other services required to run the demo's provided in the Security Starter Kits.

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



The screenshot shows the AWS login page. On the left, under the AWS logo, is the 'Sign in as IAM user' section. It contains three input fields: 'Account ID (12 digits) or account alias', 'IAM user name', and 'Password'. Below these is a blue 'Sign in' button. Links for 'Sign in using root user email' and 'Forgot password?' are at the bottom of this section. On the right is a large promotional banner for 'Build, train, and deploy ML models quickly' featuring the Amazon SageMaker logo and a 'Learn more »' button. At the bottom center, there is a language dropdown menu set to 'English'.

Figure 1: Login page



The screenshot shows the 'Create an AWS account' page. On the left, a grey box contains the text 'AWS Accounts Include 12 Months of Free Tier Access' and mentions 'Including use of Amazon EC2, Amazon S3, and Amazon DynamoDB'. The main form on the right has four input fields: 'Email address', 'Password', 'Confirm password', and 'AWS account name'. Below the 'AWS account name' field is a green icon of a person. A yellow 'Continue' button is positioned below the form fields. At the bottom of the form, there is a link 'Sign in to an existing AWS account'. The footer of the page includes copyright information: '© 2020 Amazon Web Services, Inc. or its affiliates. All rights reserved.' and links to 'Privacy Policy' 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).

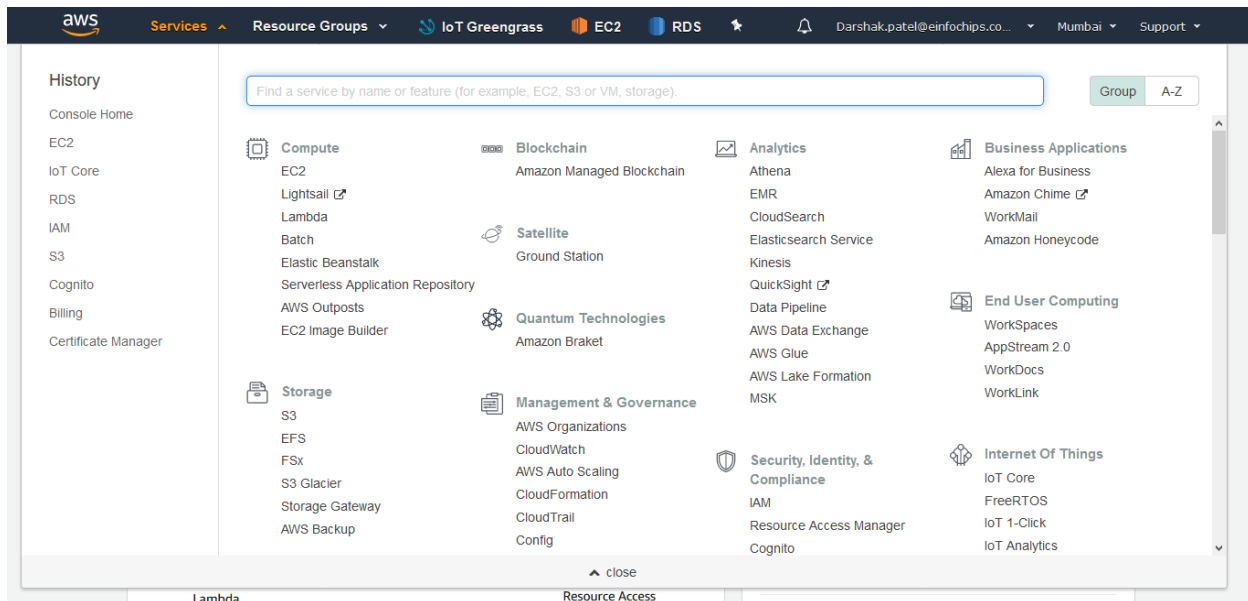


Figure 2: Select EC2 Instance

2.3 EC2 Dashboard

- Go to Instances >> Instances Click on it.

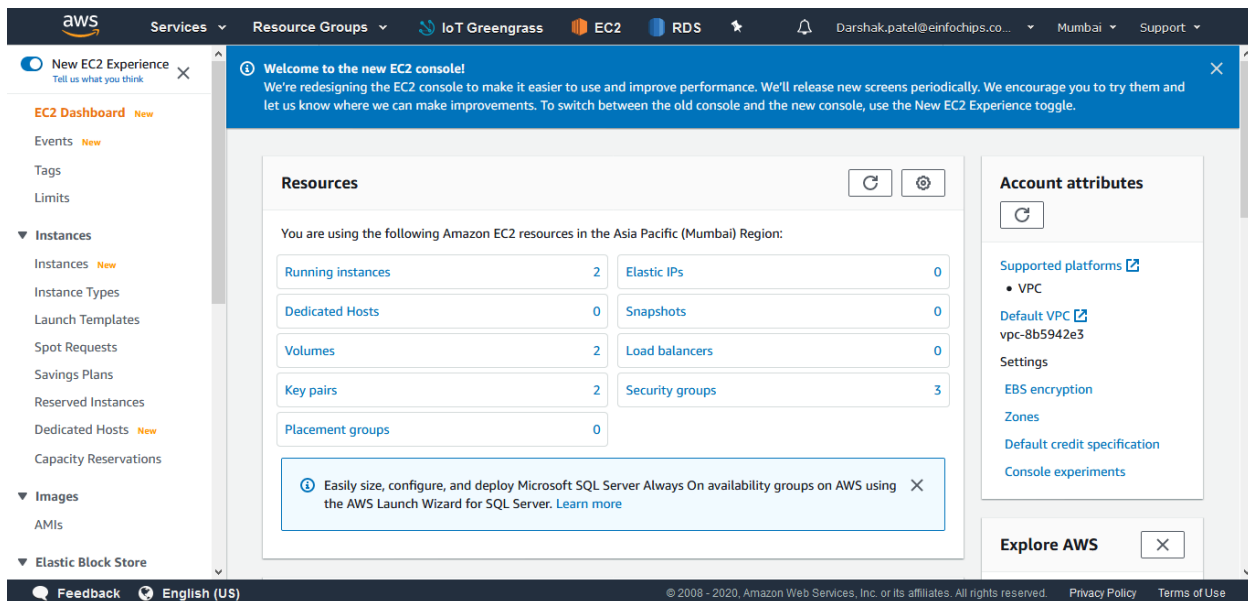


Figure 2: List EC2 Dashboard

2.4 Creating an EC2 Instance

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

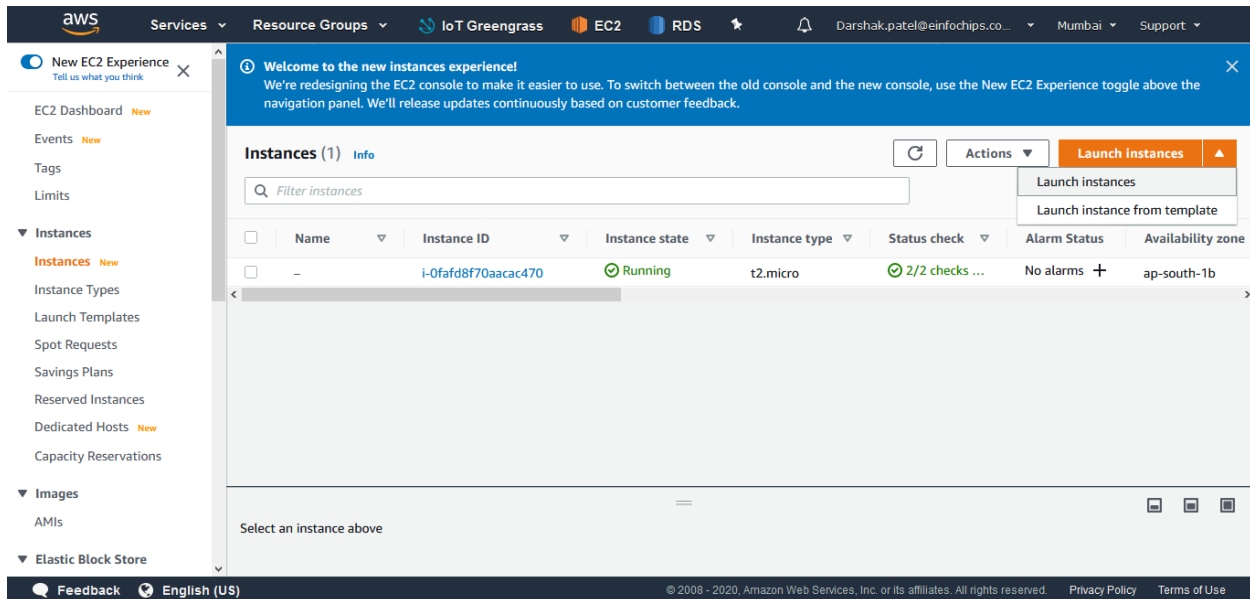


Figure 3: Launch EC2 Instance

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

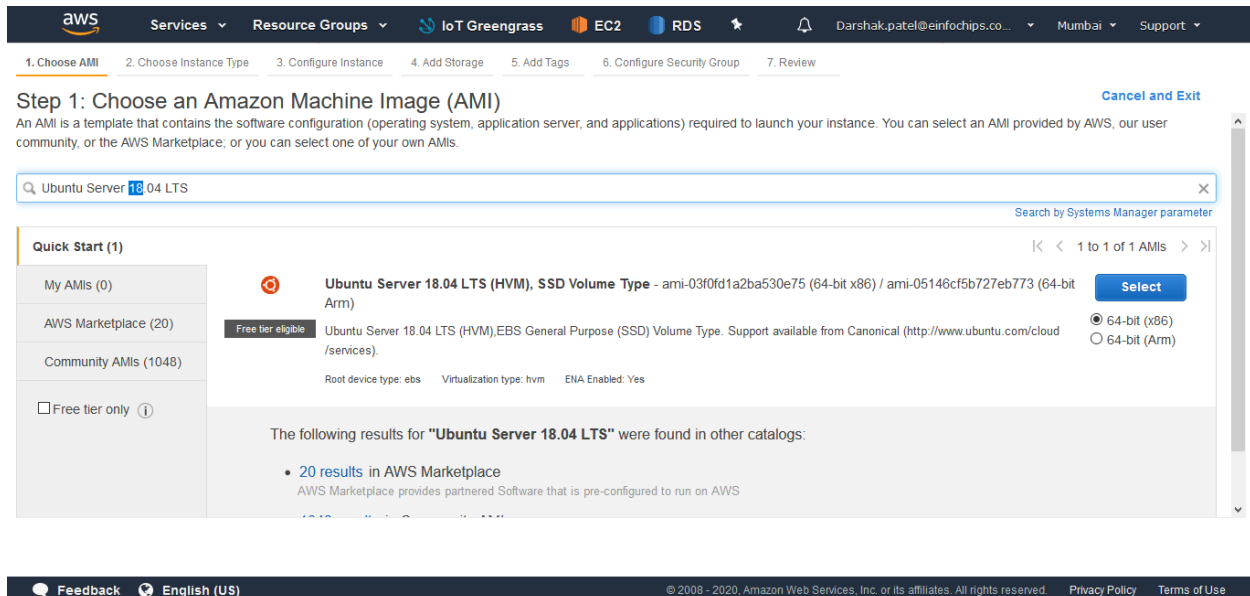


Figure 4: Ubuntu AMI

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

- Click on Next: Configure Instance details

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
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes

[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

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances [Launch into Auto Scaling Group](#)

Purchasing option ☐ Request Spot instances

Network [Create new VPC](#)

Subnet [Create new subnet](#)

Auto-assign Public IP

Placement group ☐ Add instance to placement group

Capacity Reservation

Domain join directory [Create new directory](#)

IAM role [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 Encrypt

[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](#)

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:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	My IP 157.32.227.96/32	e.g. SSH for Admin Desktop
HTTP	TCP	80	Anywhere 0.0.0.0/0	e.g. SSH for Admin Desktop

[Add Rule](#)

[Cancel](#) [Previous](#) [Review and Launch](#)

Figure 8: Configure Security group

Step 7: Review Instance Launch

- Press Launch

The screenshot shows the AWS Management Console interface for the 'Review Instance Launch' step. The instance is an Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-03f0fd1a2ba530e75. The instance type is t2.micro. The security group is SSK Security Group, which has rules for SSH (TCP, port 22) and HTTP (TCP, port 80). The 'Launch' button is visible at the bottom right.

Step 7: Review Instance Launch

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-03f0fd1a2ba530e75

Free tier eligible

Root Device Type: ebs Virtualization type: hvm

▼ Instance Type [Edit 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 [Edit security groups](#)

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	157.32.227.98/32	
HTTP	TCP	80	0.0.0.0/0	

Cancel Previous **Launch**

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

Figure 9: Review Instance

Step 8: Create New key pair

- Select “Create a new key pair” then name “SSK_Key”

The screenshot shows the AWS Management Console interface with a modal dialog box titled 'Select an existing key pair or create a new key pair'. The dialog box contains instructions on how to use a key pair and a form to create a new key pair. The 'Key pair name' field is filled with 'SSK_key'. The 'Download Key Pair' button is visible. The background shows the 'Review Instance Launch' step.

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

SSK_key

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**

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

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”.

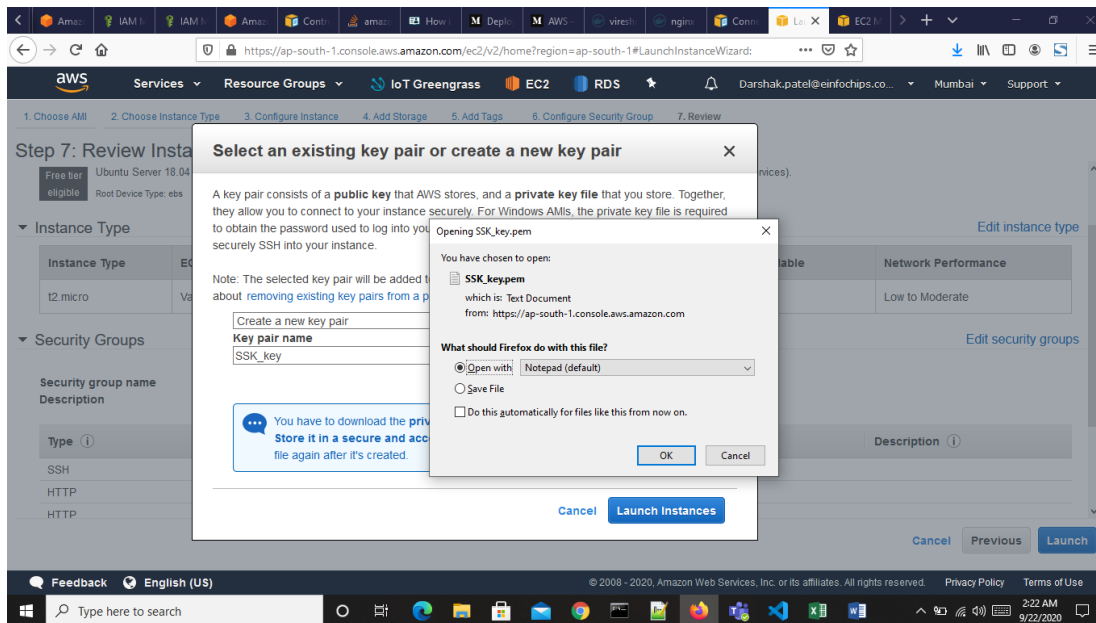


Figure 11: Download key

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.

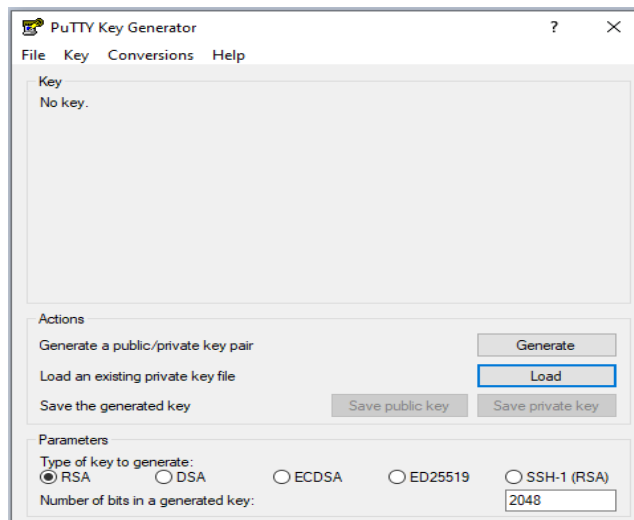


Figure 1: Convert PPK file

- It will ask for file to choose, here you'll need to provide SSK_key.pem file (select all file format)
- After successful 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

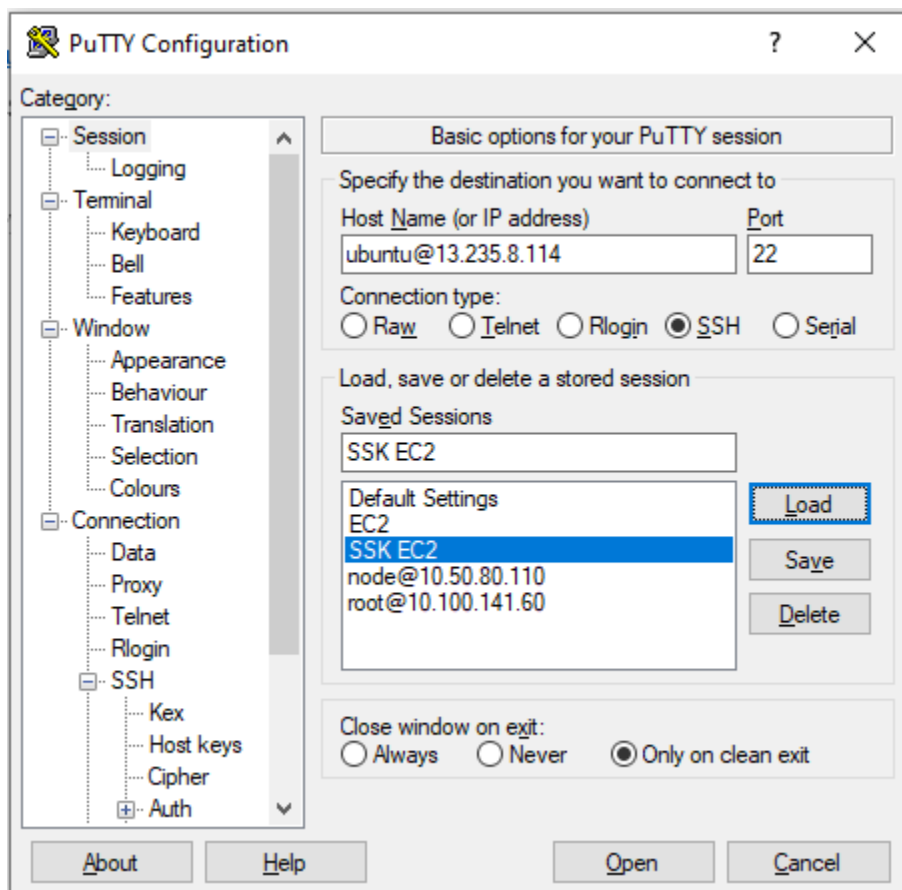


Figure 1: Configure putty

To configure key

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

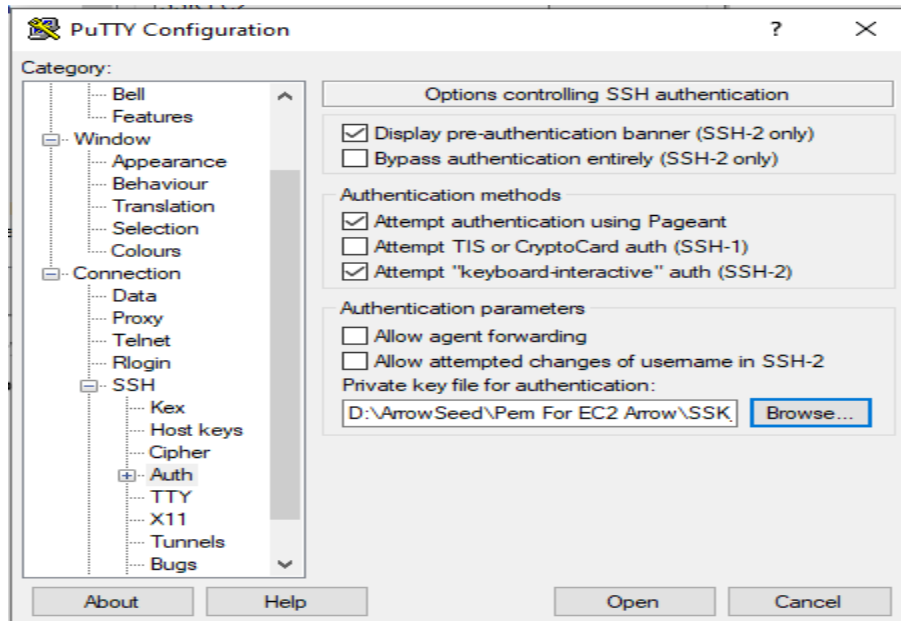


Figure 2: add key to putty

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

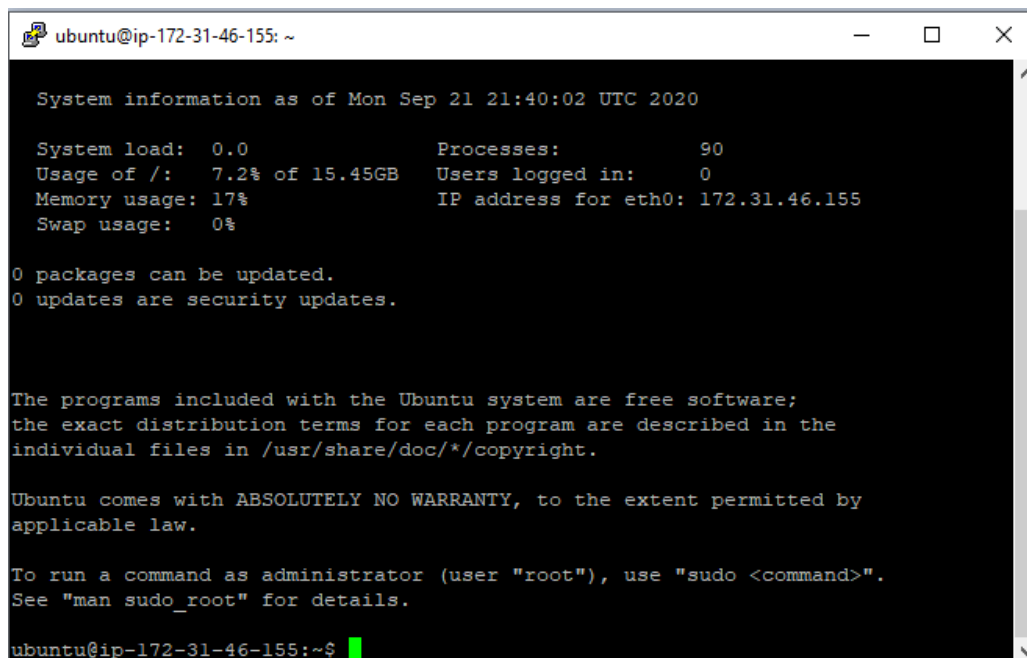


Figure 3: Connected AWS

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

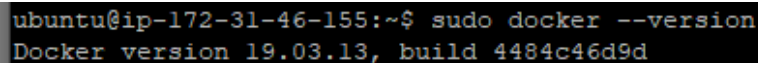
```
$ sudo apt-get update
```

Step 6: Install Docker

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

Step 7: To verify installation

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

A terminal window screenshot with a black background and white text. The prompt is 'ubuntu@ip-172-31-46-155:~\$'. The command entered is 'sudo docker --version'. The output is 'Docker version 19.03.13, build 4484c46d9d'.

```
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:

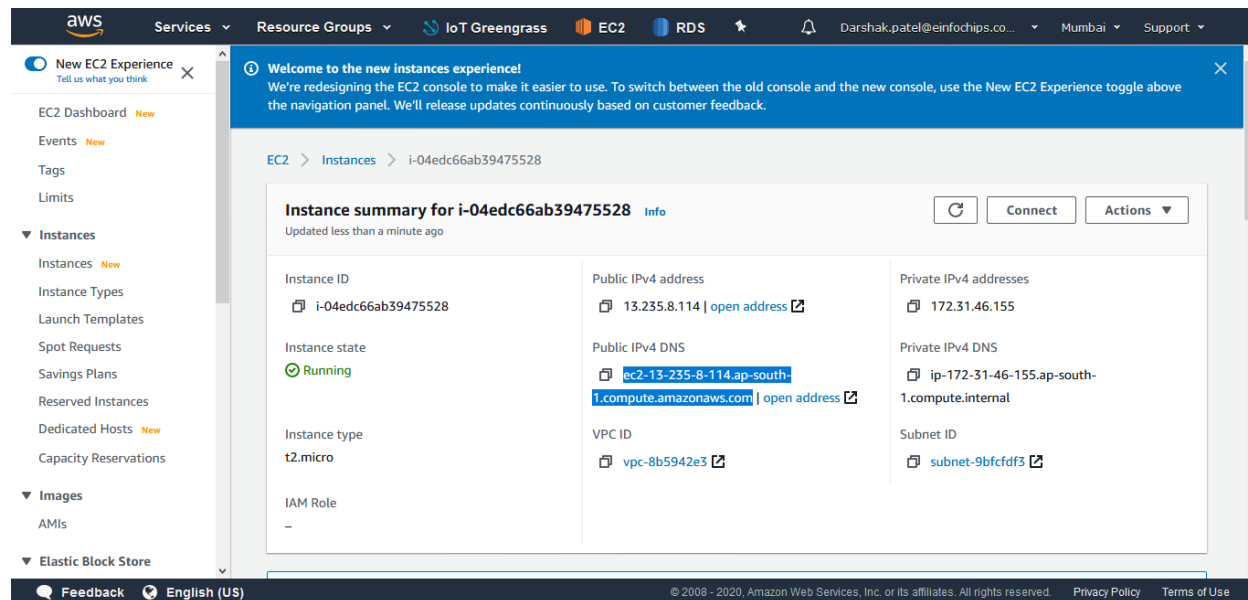


Figure 1: Application access

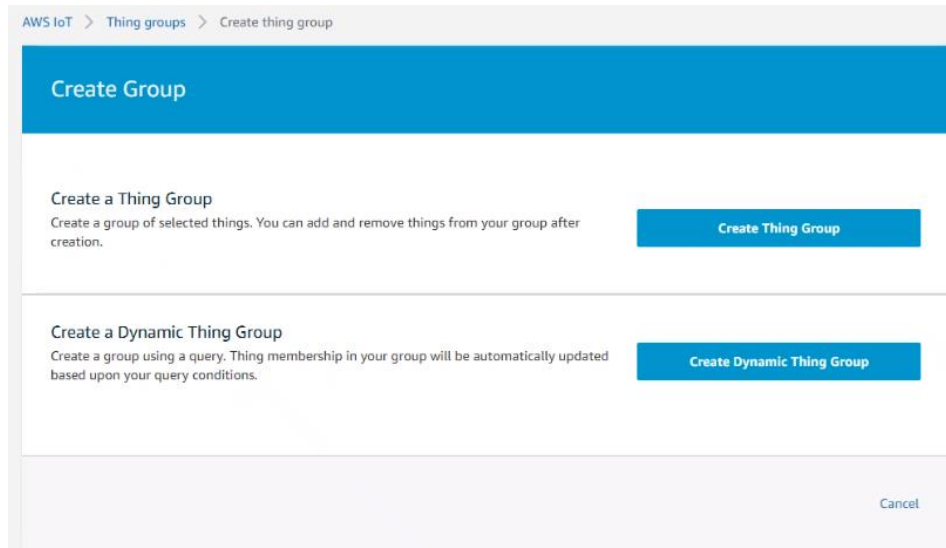
Make note of the Public IPv4 DNS address provided and it needs to be in the following format, with a leading HTTP:// as shown below;

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

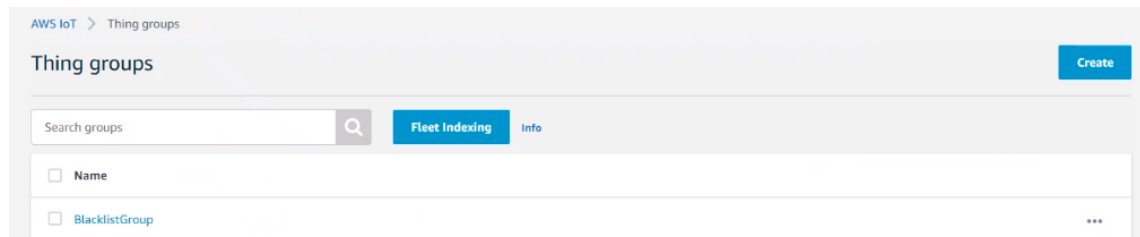
4.2 Application – “Allow” or “Deny” listing

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

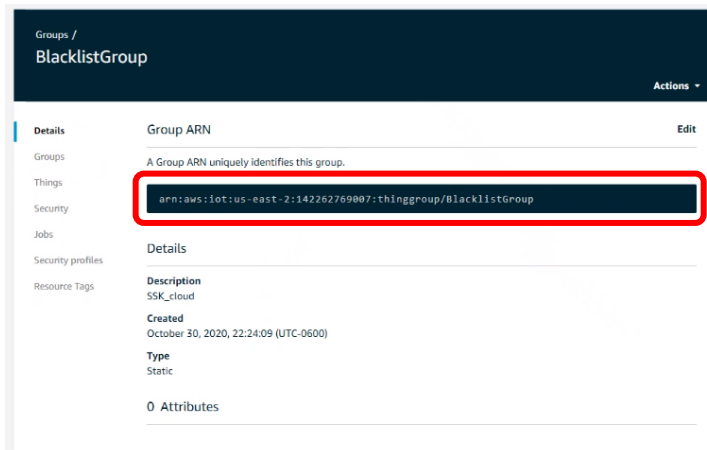
- To create ‘Thing group’ : navigate to IoT Core → Manage → Thing groups → Create



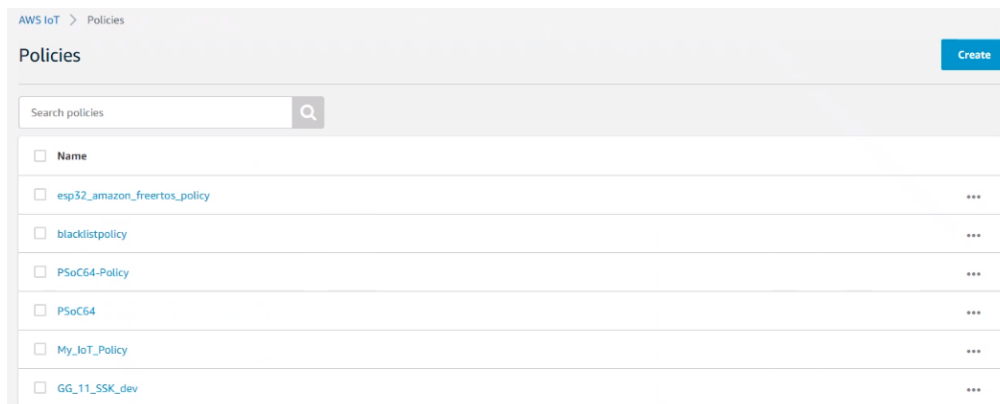
- Provide a name to your Thing Group, like what is shown below.



- 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



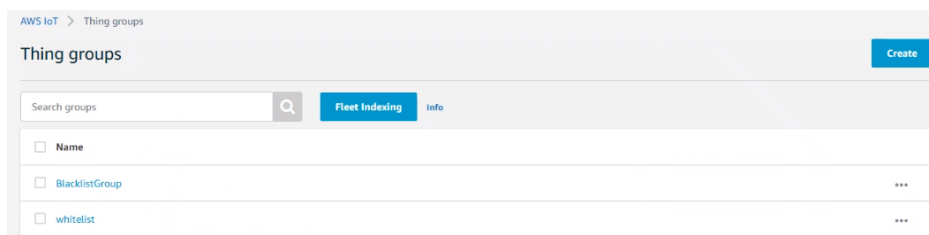
- Click create and enter the name of your policy. Under Add Statements click Advanced mode. JSON statement will be seen and then edit with the information as shown below.

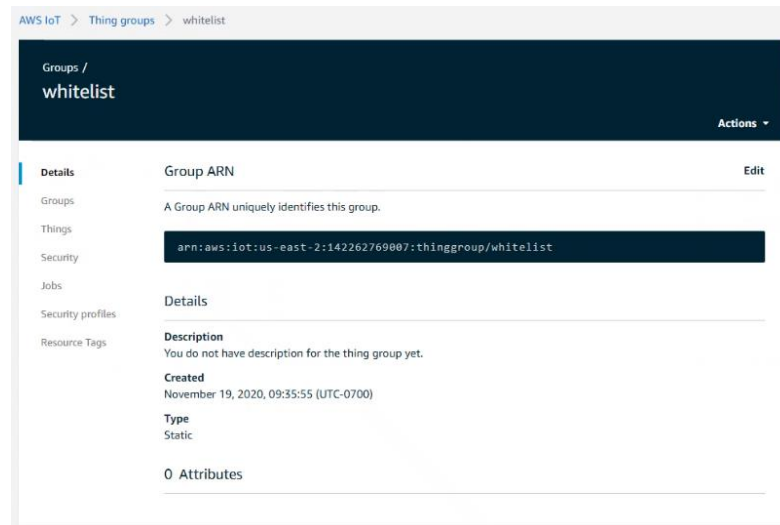
Note: You will be entering your specific ARN Group provided in the previous step next to “Resource”;

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

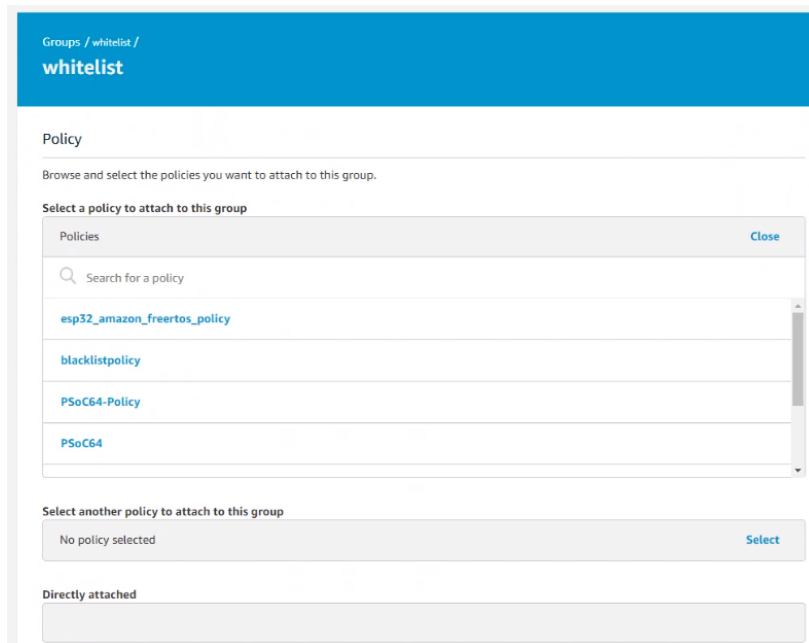
3. Next, you need to attach the “Policy” to the “Thing Group”;

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





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



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

Policy

Browse and select the policies you want to attach to this group.

Select a policy to attach to this group

blacklistpolicy Remove Select

Select another policy to attach to this group

No policy selected Select

Directly attached

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

Cancel Save

Groups / **whitelist** Actions ▾

Details Policies Edit

Groups blacklistpolicy is attached to this group.

Things Things in this group have the following permissions:

Security **iot:***

! EXPLICITLY DENIED

arn:aws:iot:us-east-2:142262769007:topic/replaceWithATopic

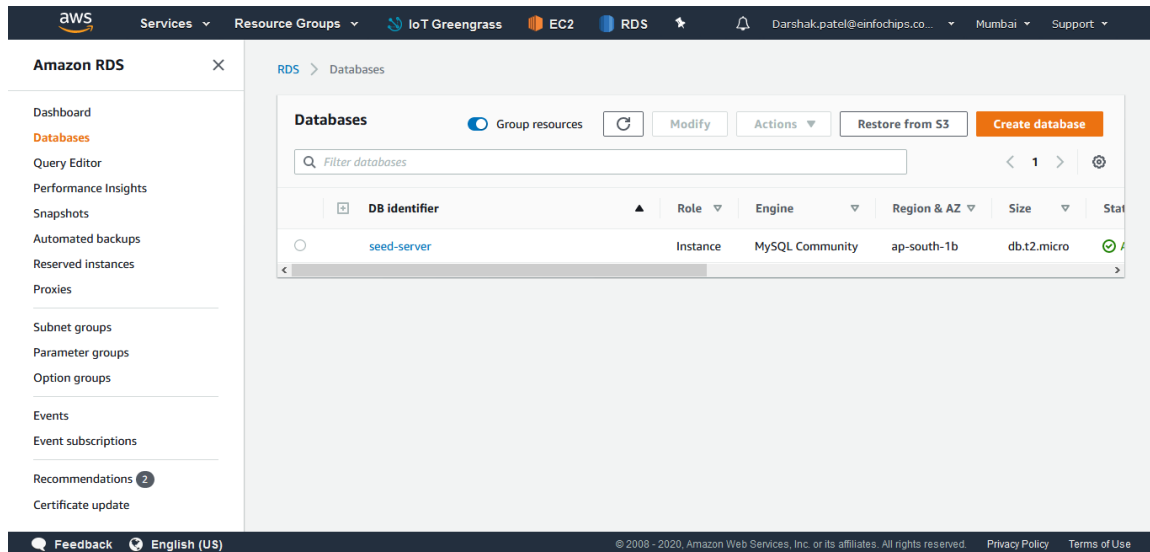
- In order to perform OTA updates, the user will need to Create an OTA Role and the link to the instructions within AWS is provided below. You will also need to create an OTA Job, which is part of the SSK Cloud Connect Tool and outlined in Section 5 of the SSK Cloud Connect Users Guide;

To create OTA update role follow below URL:

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

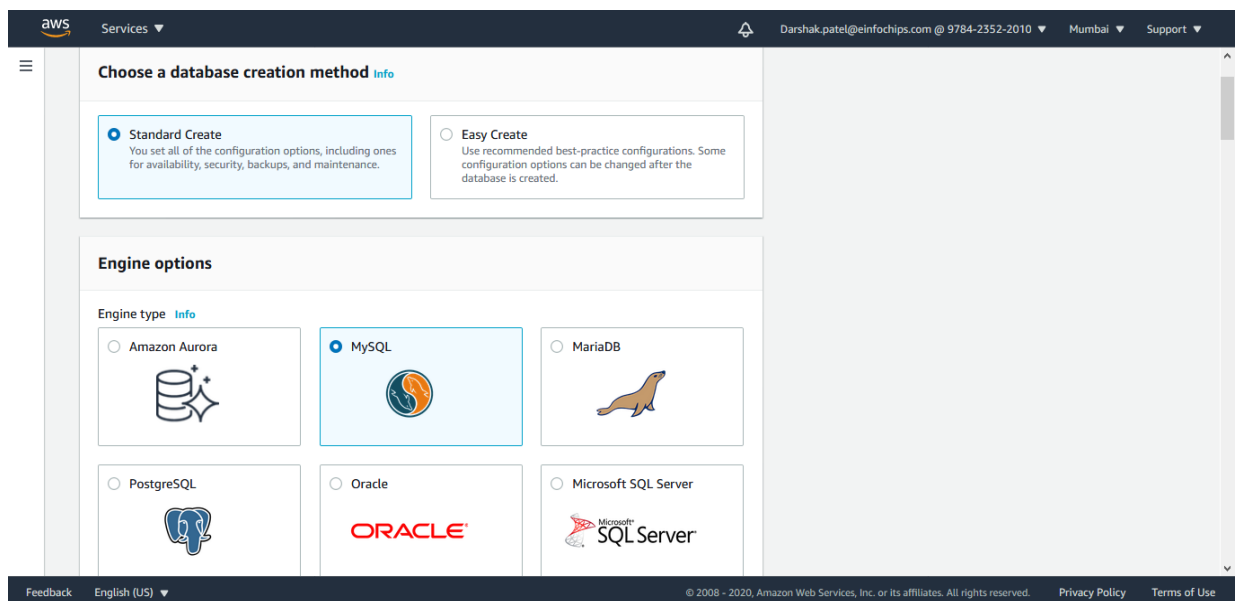
4.3 AWS RDS Service – Database Setup and Configuration

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

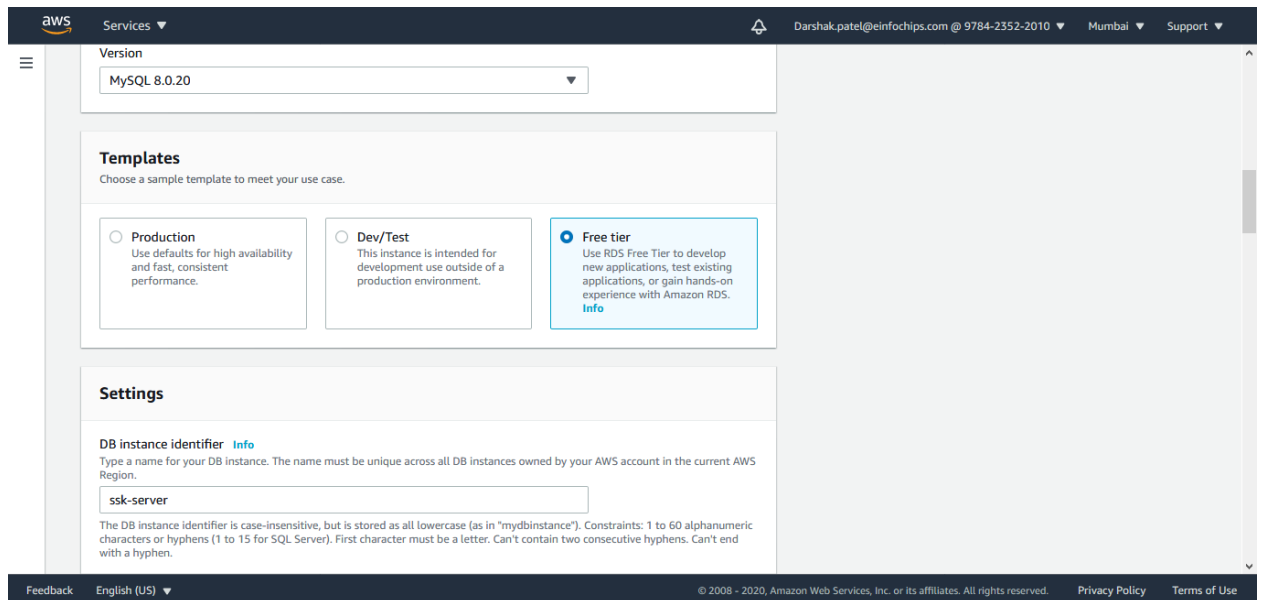


4.4 Creating a Database

- Select “Standard Create”

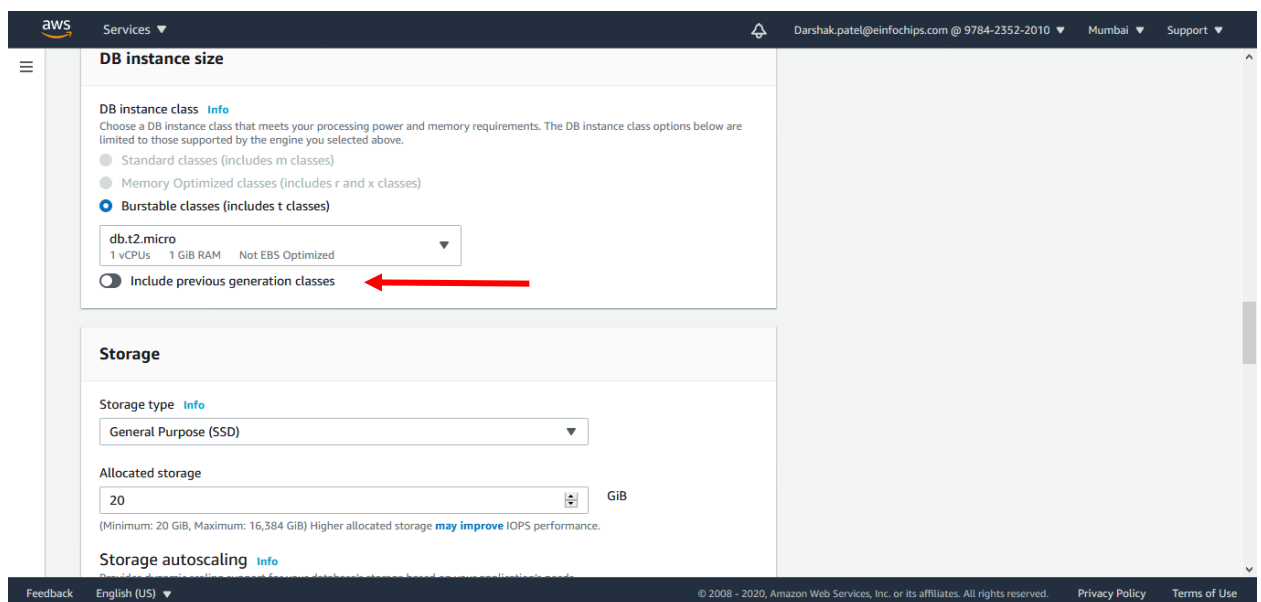


- Select “Free Tier”



The screenshot shows the AWS RDS console interface. At the top, the 'Services' dropdown is set to 'RDS'. The 'Version' dropdown is set to 'MySQL 8.0.20'. Under the 'Templates' section, three options are visible: 'Production', 'Dev/Test', and 'Free tier'. The 'Free tier' option is selected, indicated by a blue circle and a blue border. Below the templates, the 'Settings' section shows the 'DB instance identifier' set to 'ssk-server'. The footer of the console displays '© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.' and links to 'Privacy Policy' and 'Terms of Use'.

- Enable “Include previous generation classes”



The screenshot shows the AWS RDS console interface for the 'DB instance size' section. The 'DB instance class' dropdown is set to 'db.t2.micro'. Below the dropdown, the 'Include previous generation classes' checkbox is checked, and a red arrow points to it. The 'Storage' section shows the 'Storage type' set to 'General Purpose (SSD)' and 'Allocated storage' set to '20' GiB. The footer of the console displays '© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.' and links to 'Privacy Policy' and 'Terms of Use'.

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

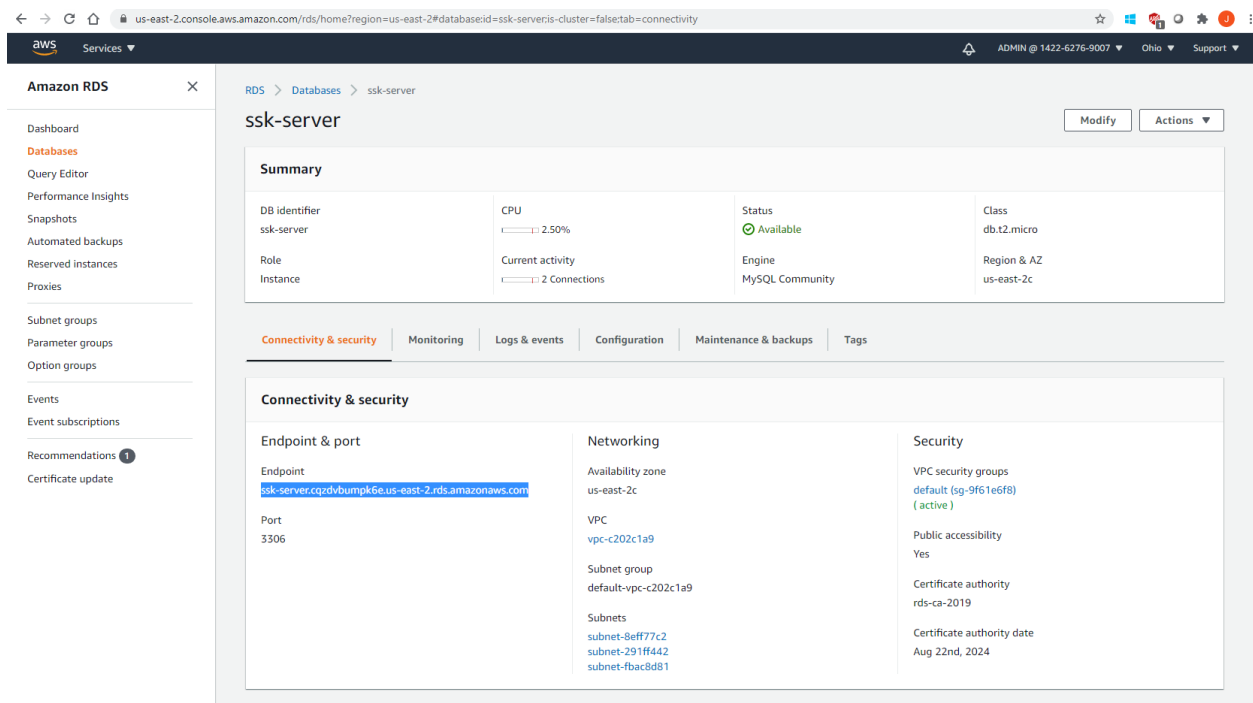
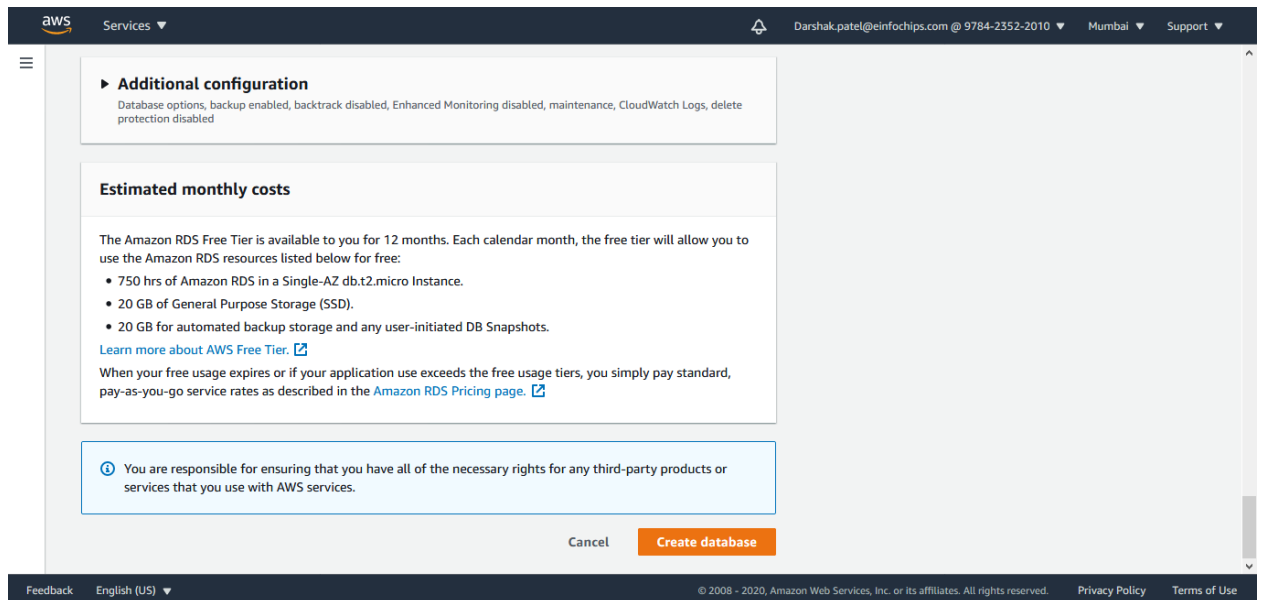
The screenshot shows the AWS Management Console interface. At the top, the 'Services' dropdown is visible. The main content area is titled 'Connectivity'. Under 'Virtual private cloud (VPC)', the 'Default VPC (vpc-8b5942e3)' is selected. A blue information box states: 'After a database is created, you can't change the VPC selection.' Below this is a section for 'Additional connectivity configuration'. The 'Database authentication' section shows three options: 'Password authentication' (selected), 'Password and IAM database authentication', and 'Password and Kerberos authentication (not available for this version)'.

- Select “Publicly accessible” under Additional Connectivity Configuration

This screenshot focuses on the 'Additional connectivity configuration' section. It shows the 'Subnet group' set to 'default-vpc-c202c1a9', the 'Security group' set to 'default', and the 'Certificate authority' set to 'rds-ca-2019'. Under the 'Public access' heading, the 'Publicly accessible' radio button is selected and highlighted with a red rectangle. The description for 'Publicly accessible' states: 'EC2 instances and devices outside the VPC can connect to the instance. You define the security groups for supported devices and instances.' The 'Database port' is set to '3306'.

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

- Click “Create database”



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

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

The screenshot shows the Amazon RDS console with the 'ssk-server' instance selected. The 'Configuration' tab is active, displaying various instance details:

Summary			
DB identifier: ssk-server	CPU: 2.50%	Status: Available	Class: db.t2.micro
Role: Instance	Current activity: 2 Connections	Engine: MySQL Community	Region & AZ: us-east-2c

Instance			
Configuration DB instance id: ssk-server Engine version: 8.0.20 DB name: - License model: General Public License Option groups: default:mysql-8-0	Instance class Instance class: db.t2.micro vCPU: 1 RAM: 1 GB Availability Master username: <u>admin_jg</u> IAM db authentication	Storage Encryption: Not Enabled Storage type: General Purpose (SSD) IOPS: - Storage: 20 GiB Storage autoscaling: Enabled	Performance Insights 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 'Connectivity & security' tab for the 'ssk-server' instance. It displays the following information:

Connectivity & security		
Endpoint & port Endpoint: ssk-server.cqzdvbumpk6e.us-east-2.rds.amazonaws.com Port: 3306	Networking Availability zone: us-east-2c VPC: vpc-c202c1a9 Subnet group: default-vpc-c202c1a9 Subnets: subnet-8eff77c2, subnet-291ff442, subnet-fbac8d81	Security VPC security groups: default (sg-9f61e6f8) (active) Public accessibility: Yes Certificate authority: rds-ca-2019 Certificate authority date: Aug 22nd, 2024

Security group rules (2)

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:

aws

Services

ADMIN @ 1422-6276-9007

Ohio

Support

EC2 > Security Groups > sg-9f61e6f8 - default > Edit inbound rules

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules [Info](#)

Inbound rule 1 [Info](#)

Delete

Type [Info](#)
All traffic

Source type [Info](#)
Custom

Protocol [Info](#)
All

Source [Info](#)
0.0.0.0/0

Port range [Info](#)
All

Description - optional [Info](#)

Inbound rule 2 [Info](#)

Delete

Type [Info](#)
All traffic

Source type [Info](#)
Custom

Protocol [Info](#)
All

Source [Info](#)
:::/0

Port range [Info](#)
All

Description - optional [Info](#)

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

Outbound Rules:

Edit outbound rules [Info](#)

Outbound rules control the outgoing traffic that's allowed to leave the instance.

Outbound rules [Info](#)

Outbound rule 1 [Delete](#)

Type [Info](#): All traffic

Protocol [Info](#): All

Port range [Info](#): All

Destination type [Info](#): Custom

Destination [Info](#): 0.0.0.0/0

Description - optional [Info](#)

Outbound rule 2 [Delete](#)

Type [Info](#): All traffic

Protocol [Info](#): All

Port range [Info](#): All

Destination type [Info](#): Custom

Destination [Info](#): :::/0

Description - optional [Info](#)

[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](#)

Find users by username or access key

<input type="checkbox"/>	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

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

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

The screenshot shows the 'Add user' page in the AWS IAM console. The top navigation bar includes the AWS logo, 'Services' dropdown, a user profile 'ADMIN @ 1422-6276-9007', 'Global' region, and 'Support' link. The page has a progress indicator with four steps: 1 (selected), 2, 3, and 4. The main section is titled 'Set user details' and includes a sub-header 'You can add multiple users at once with the same access type and permissions. [Learn more](#)'. There is a text input field for 'User name*' with the placeholder 'NAME' and a blue link 'Add another user'. Below this is the 'Select AWS access type' section, which says 'Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)'. It lists two options: 'Access type*' with 'Programmatic access' selected (checked), which enables an 'access key ID' and 'secret access key' for the AWS API, CLI, SDK, and other development tools; and 'AWS Management Console access' which is unselected, enabling a 'password' for sign-in to the AWS Management Console. At the bottom, there is a '* Required' label, a 'Cancel' button, and a 'Next: Permissions' button. The footer contains 'Feedback', 'English (US)' dropdown, '© 2018, 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.', 'Privacy Policy', and 'Terms of Use'.

- 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' page in the AWS IAM console, specifically the 'Set permissions' step. The progress indicator shows step 2 is selected. The 'Set permissions' section has three options: 'Add user to group' (selected), 'Copy permissions from existing user', and 'Attach existing policies directly'. Below this is a sub-section 'Add user to group' with a 'Create group' button and a 'Refresh' button. A search bar is present with the text 'Showing 1'. Below the search bar is a table with two columns: 'Group' and 'Attached policies'. The table contains one row: 'Administrator' (checked) and 'AdministratorAccess'. At the bottom, there is a 'Cancel' button, a 'Previous' button, and a 'Next: Tags' button. The footer is the same as the previous screenshot.

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

Add user

1 2 3 4

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

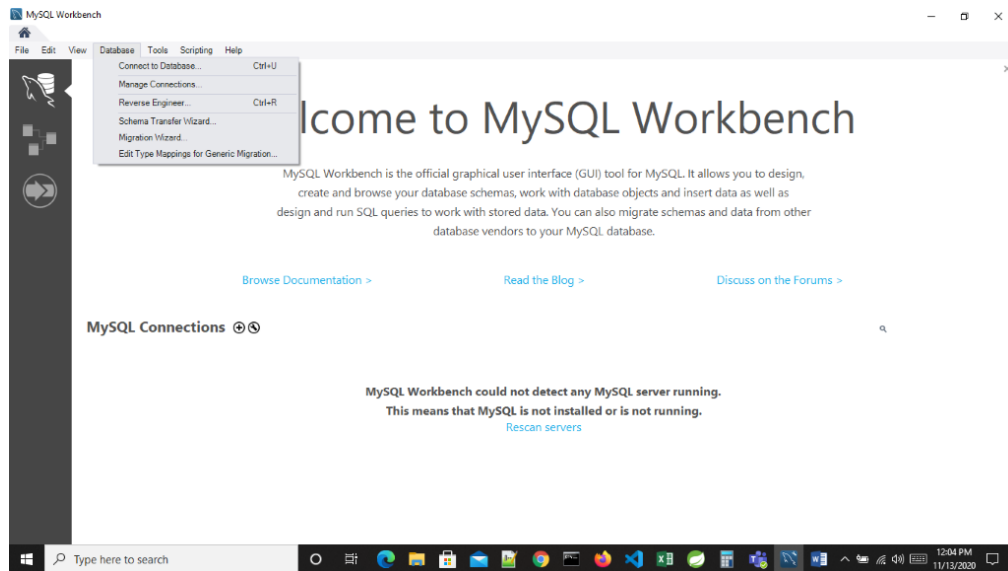
1. Install MySQL Workbench (link provided below), then Open MySQL workbench.

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

2. Install Postman

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

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



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

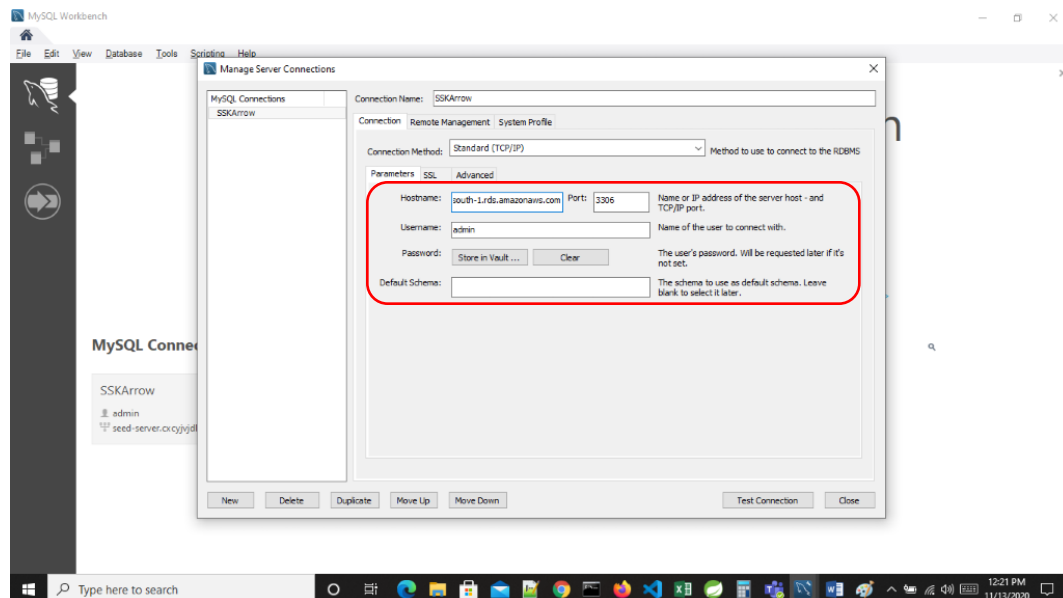
Connection Name: <Name of connection>

Host Name: <AWS RDS HOST URL>

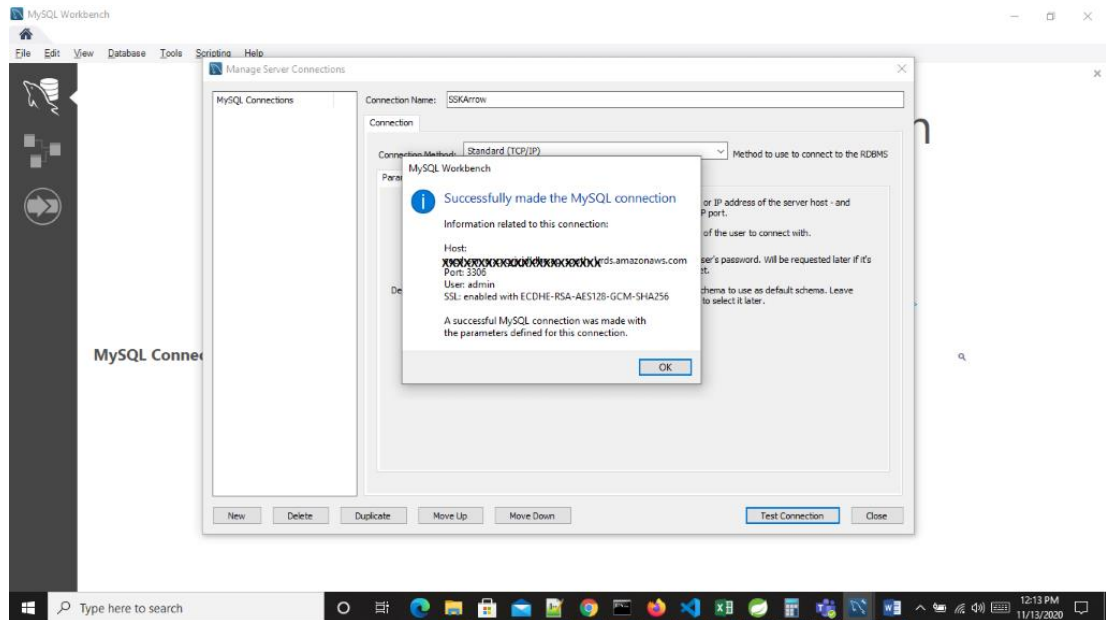
Port: 3306 (Default value)

User Name: < AWS RDS User name>

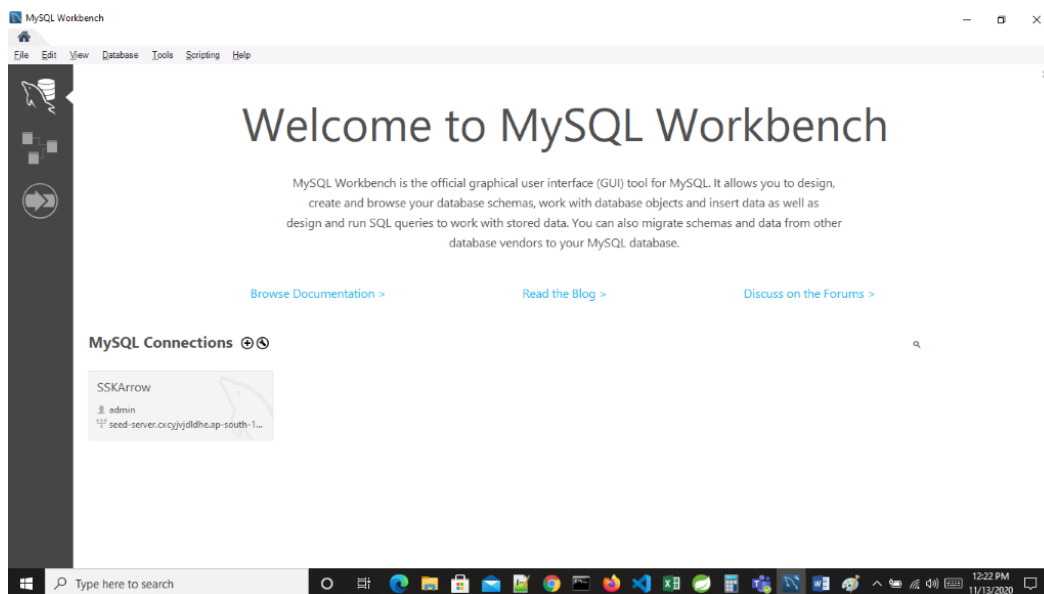
Password: <AWS RDS user password> (Store in Vault if needed)



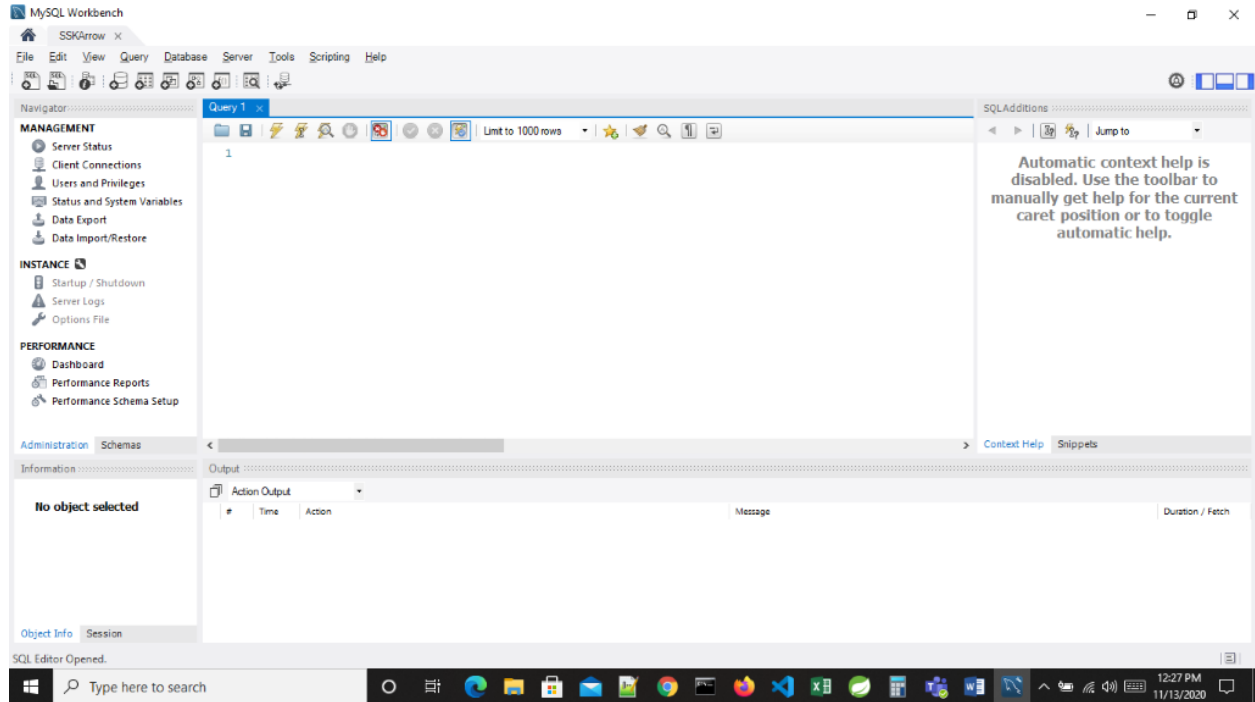
- Press on Test Connection. It will pop up successful connection message. Then click on close button



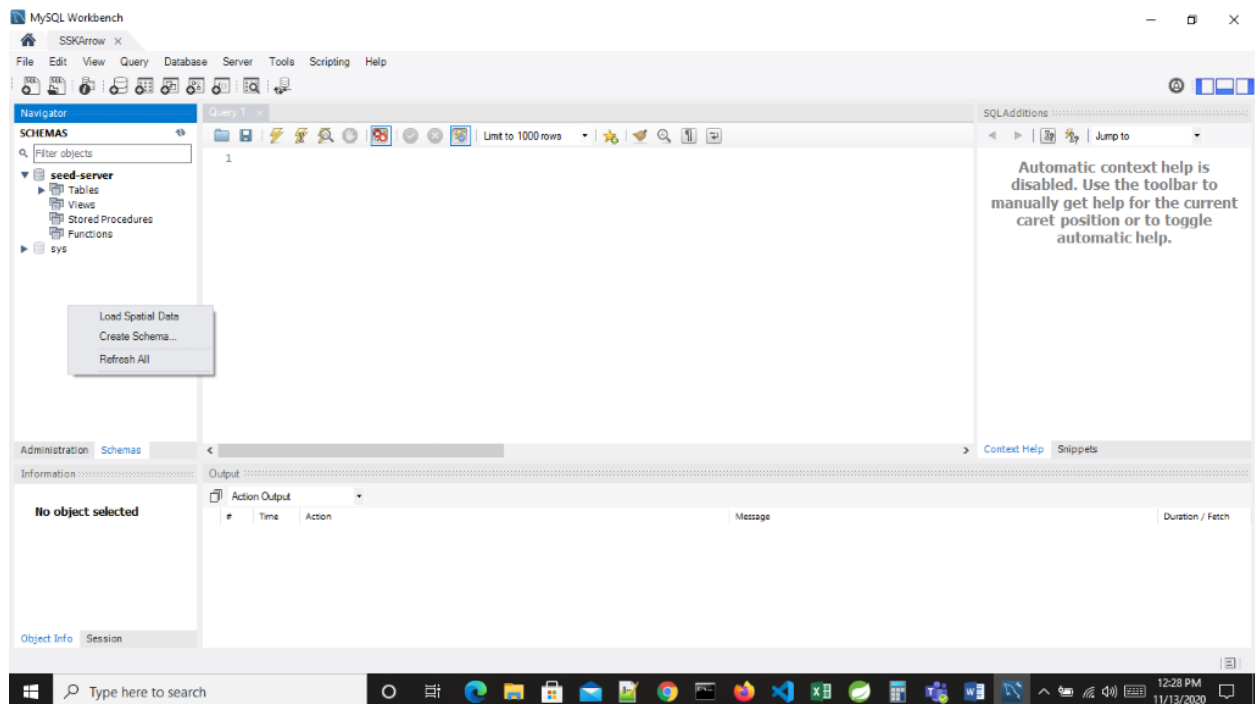
- Further it will show following details. Click on created connection button.



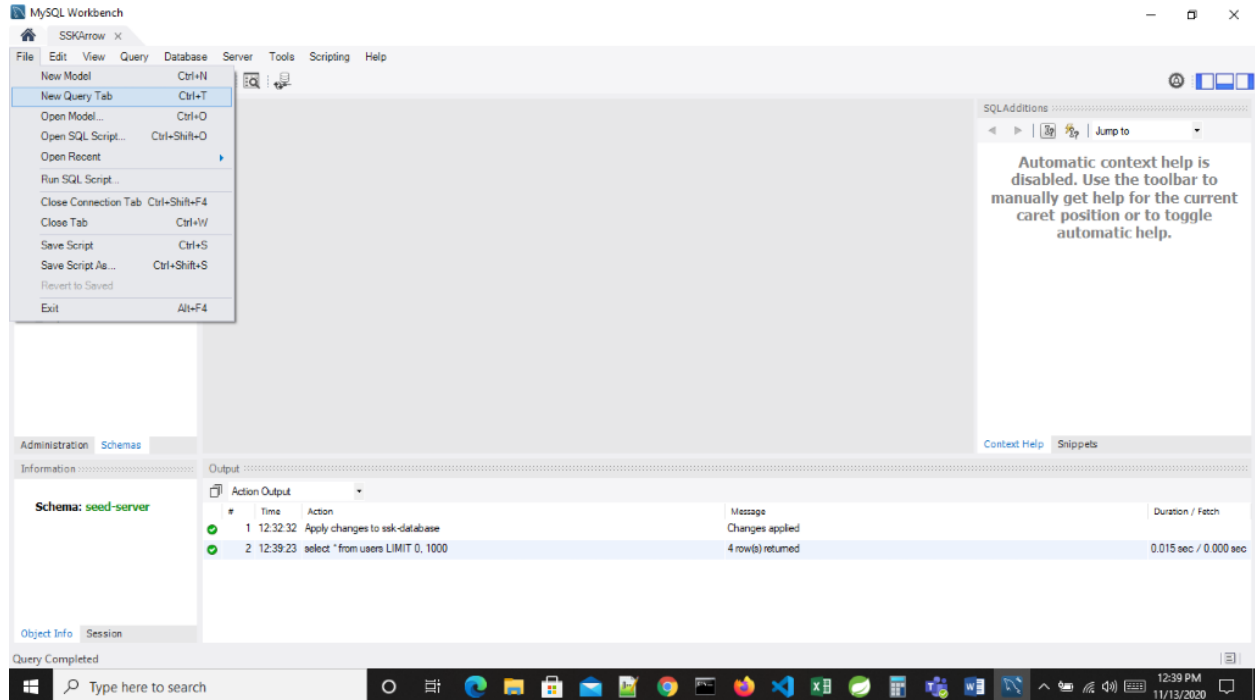
7. It will open Administration tab.



8. Select Schemas Tab & right click on mouse.



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

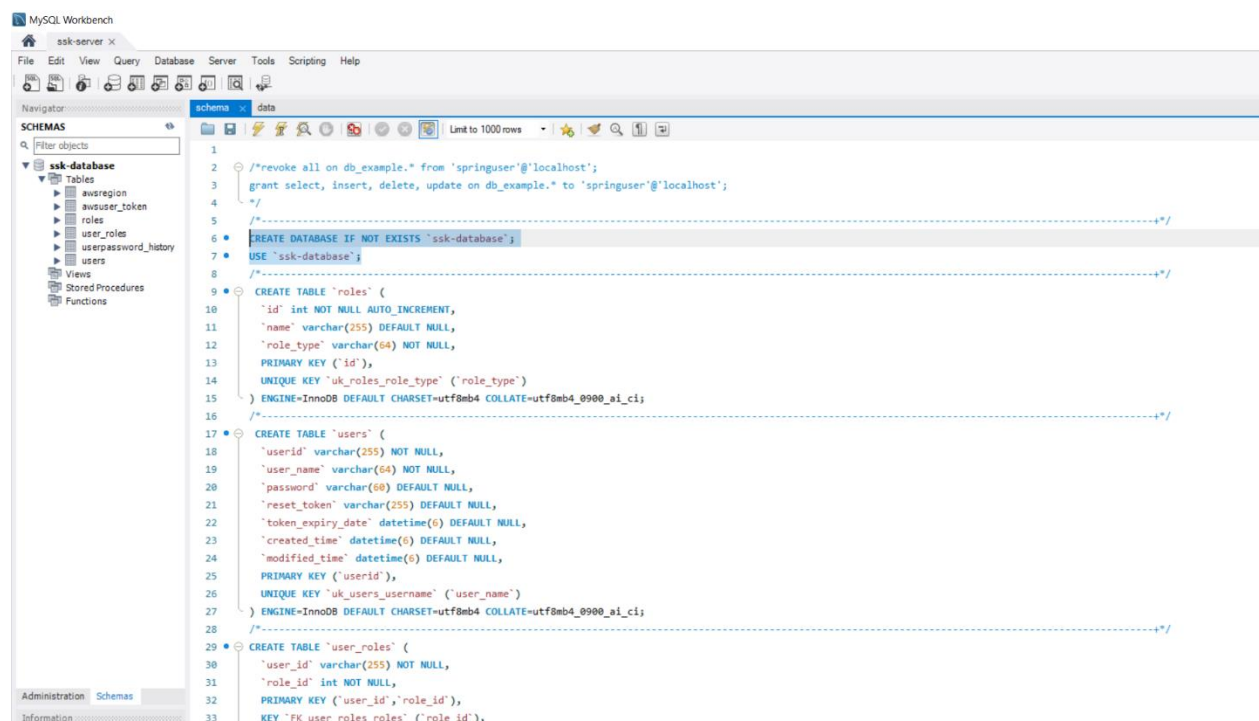


10. Edit and Execute schema.sql

- 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





11. Edit and Execute data.sql

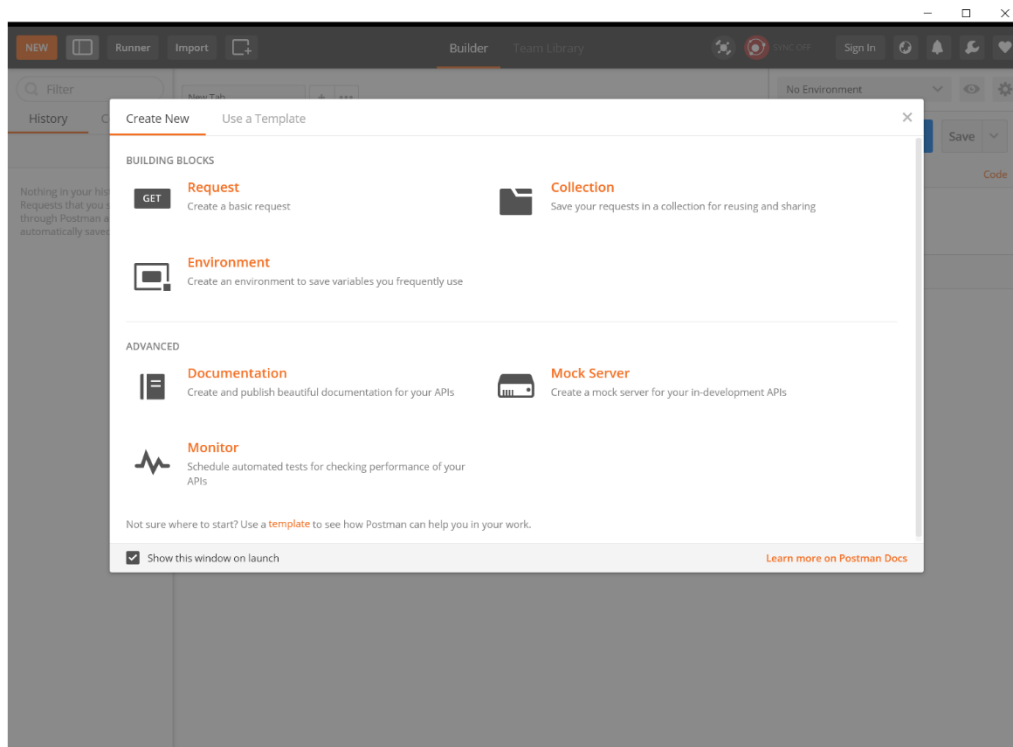
- Go to File->Open SQL Script..., navigate to the location of data.sql on your PC, and select it
- Locate the User name, Access key ID, and Secret access key of your IAM user. Note these credentials can be found in the 'new_user_credentials.csv' file that was downloaded after creating an IAM user
- Modify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

```

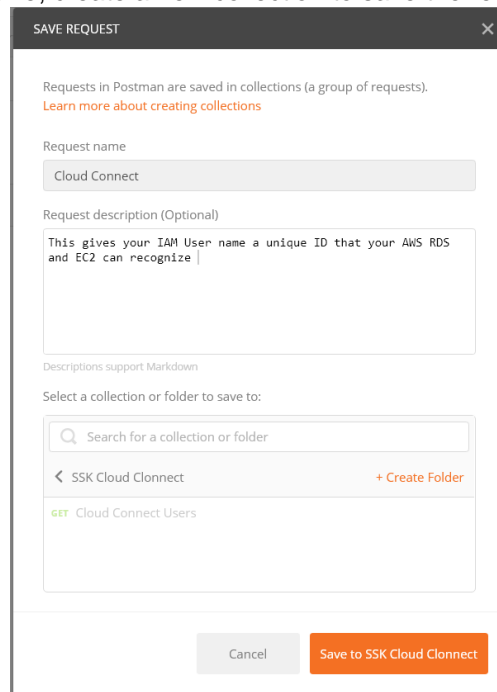
INSERT IGNORE INTO
awsuser_token(user_name,access_key_id,secret_access_key,is_root_account,create_date)
values('<your_iam_username>','<your_access_key>','<your_secret_key>',1,now());

```

- 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 'Cloud Connect Users' configuration in a testing tool. The 'Authorization' tab is selected, displaying the following fields:

- Type:** AWS Signature
- AccessKey:** [Redacted]
- SecretKey:** [Redacted]
- AWS Region:** [Empty]
- Service Name:** iam
- Save helper data to request:** ☐

The URL bar shows: `https://iam.amazonaws.com/?Action=GetUser&UserName=root_user&Version=2010-05-08`

- Now choose 'Send' and copy the ID that was generated below next to '<UserID>'

The screenshot shows the 'Body' tab of the testing tool displaying the XML response from the AWS IAM GetUser API. The response is as follows:

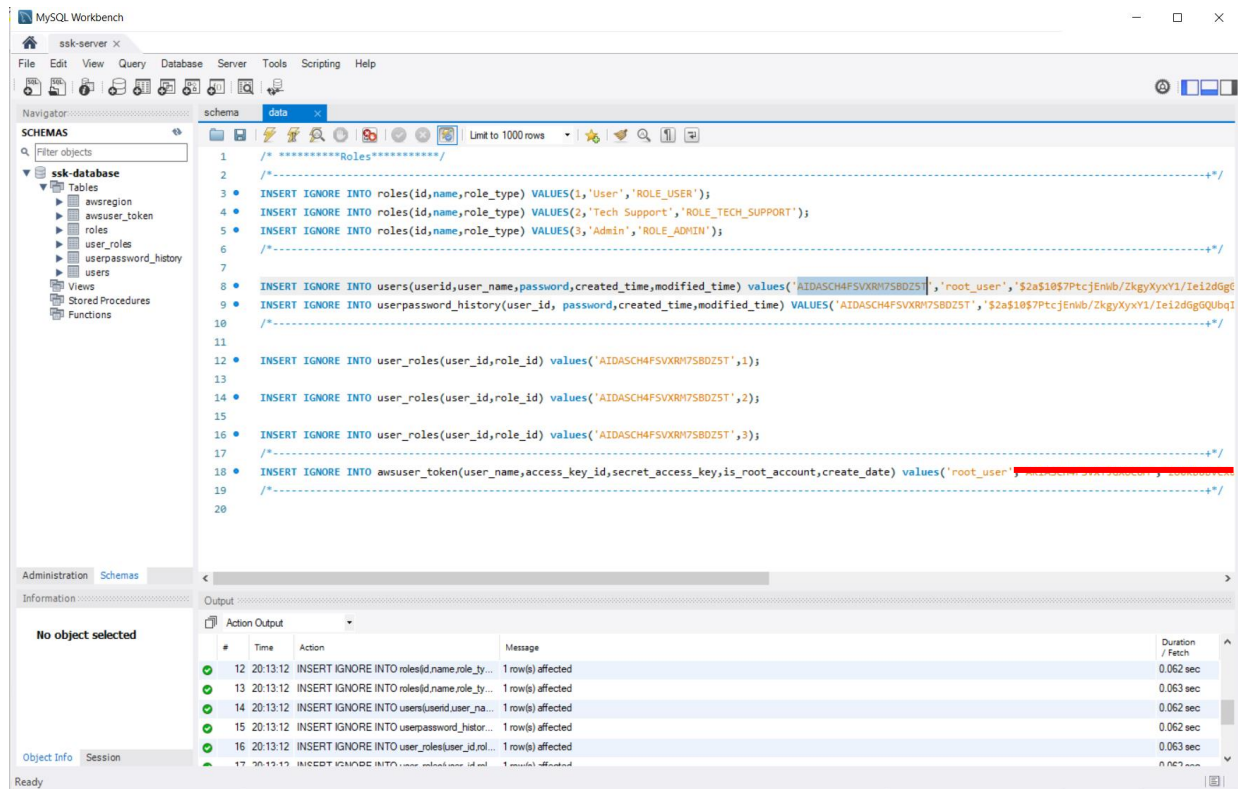
```

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>AIDASCH4FSVXRM7SBDZ5T</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@infochips.com' with the username of your IAM user

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE



- 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
```

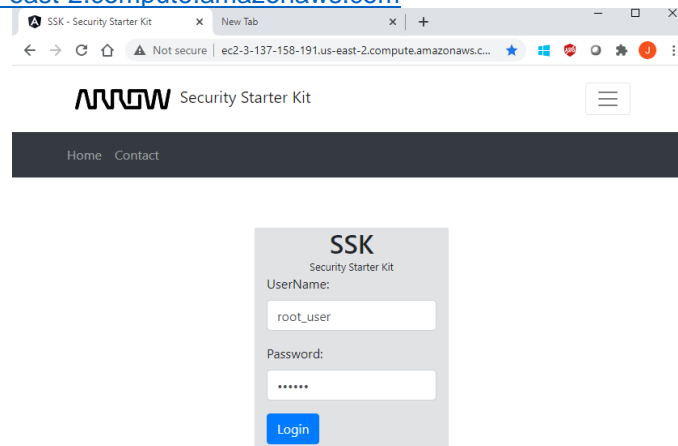
Step 6: Run image (Change highlighted text with your own values)

```
$sudo docker run --name ssk-server -e  
MYSQL_URL="jdbc:mysql://<RDS_ENDPOINT>:3306/<SCHEMA_NAME>?useSSL=  
false&serverTimezone=UTC&useLegacyDatetimeCode=false&allowPublicKeyRet  
rieval=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
- [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/>