

Secure Starter Kit Cloud Connect Installation & Setup Guide

Date: December 08, 2020 | Version 1.0
FINAL



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.....	16
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
5	CONFIGURE IMAGE ON DOCKER AND EC2	38
5.1	Execute below commands in PuTTY.....	38
5.2	Log into SSK Cloud Connect	38
6	CHECKOUT PROJECT.....	40
7	BUILD PROJECT.....	41
7.1	Execute below command	41
8	REFERENCES	42

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.*

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

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/>

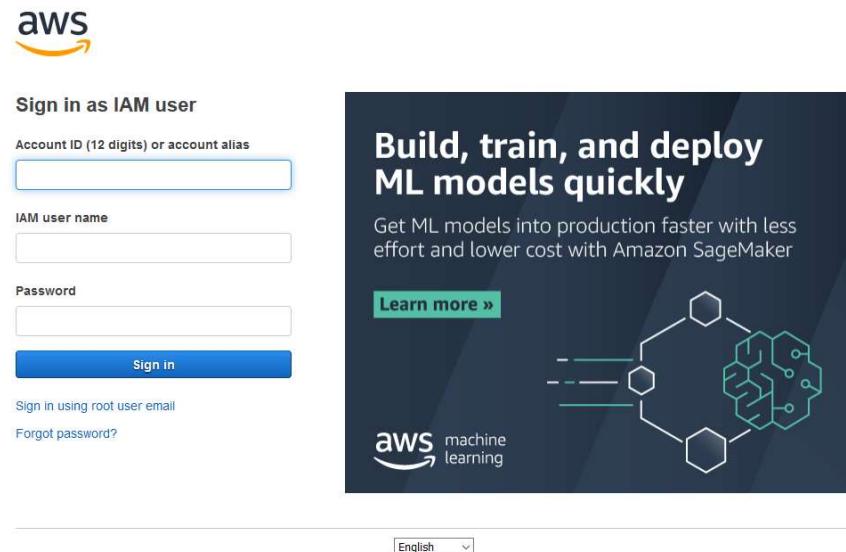


Figure 1: Login page

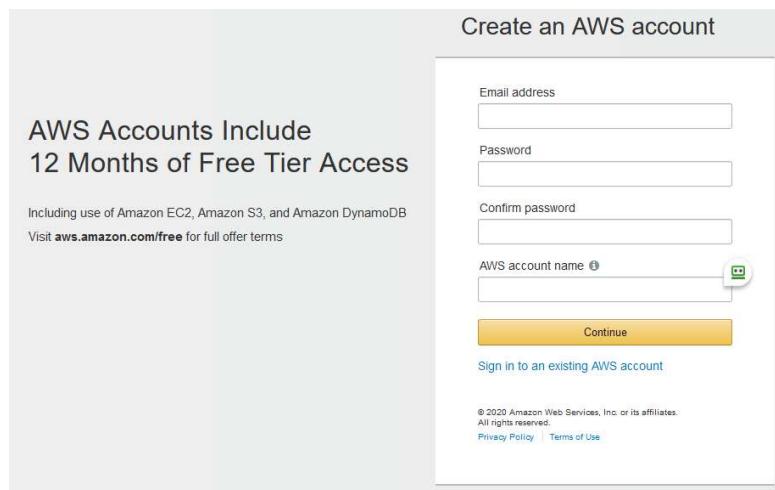


Figure 2: Create New Account page

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

2.2 AWS EC2 Instance Service

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

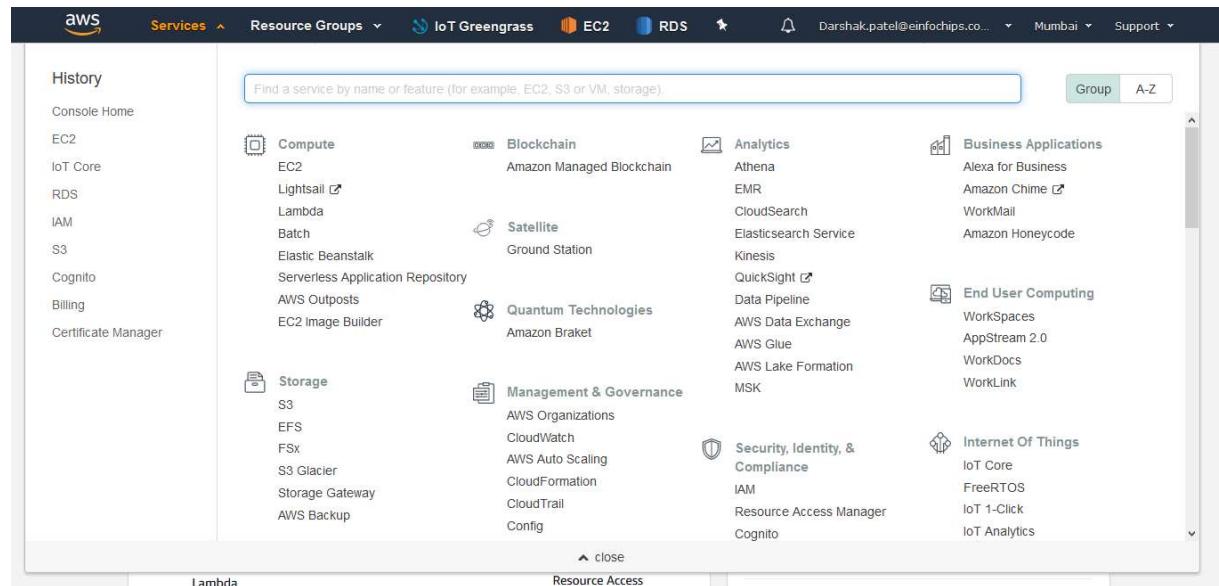


Figure 2: Select EC2 Instance

2.3 EC2 Dashboard

- Go to Instances >> Instances Click on it.

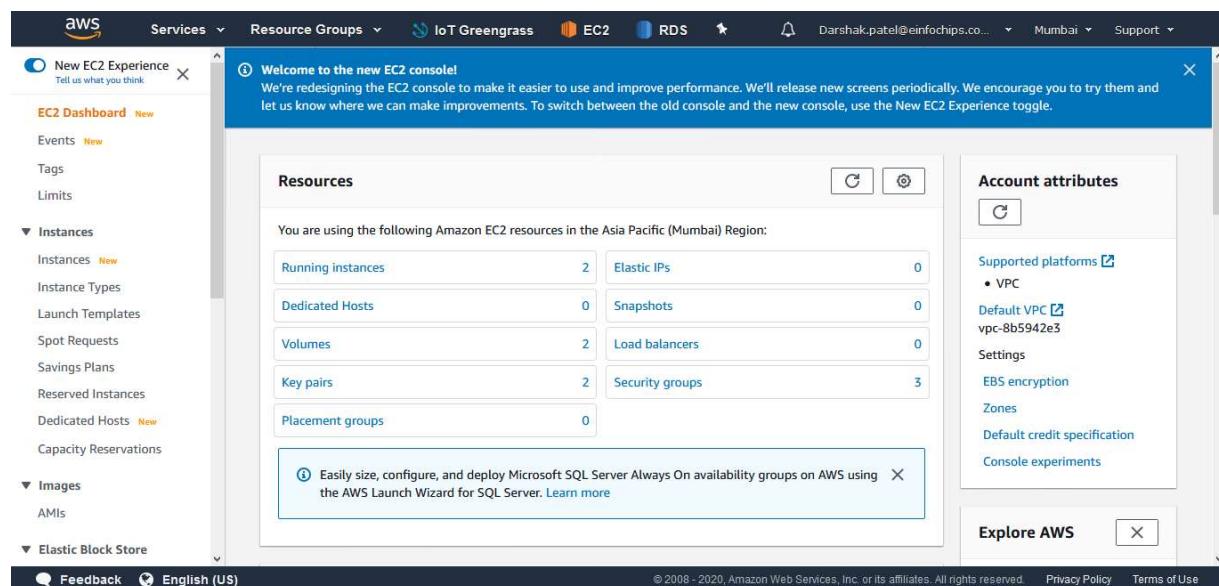


Figure 2: List EC2 Dashboard

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

2.4 Creating an EC2 Instance

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

The screenshot shows the AWS EC2 Instances page. At the top, there is a banner about the new instances experience. Below it, the 'Instances (1) Info' section displays a table with one row. The table columns are: Name, Instance ID, Instance state, Instance type, Status check, Alarm Status, and Availability zone. The instance listed is 'i-0fafd8f70aacac470' with a status of 'Running', type 't2.micro', and availability zone 'ap-south-1b'. There are buttons for 'Launch instances' and 'Launch instance from template' at the top right of the table. On the left sidebar, under 'Instances', there are links for 'Instances New', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts New', and 'Capacity Reservations'. Other sections like 'Images' and 'Elastic Block Store' are also visible.

Figure 3: Launch EC2 Instance

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

The screenshot shows the 'Step 1: Choose an Amazon Machine Image (AMI)' step. The top navigation bar includes steps 1. Choose AMI through 7. Review. The main area shows a search bar with 'Ubuntu Server 18.04 LTS' and a list of results. One result is highlighted: 'Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-03fdf1a2ba530e75 (64-bit x86) / ami-05146cf5b72eb773 (64-bit Arm)'. It is labeled as 'Free tier eligible'. Below the result, there are checkboxes for 'Root device type: ebs', 'Virtualization type: hvm', and 'ENI Enabled: Yes'. To the right, there are buttons for 'Select' and 'Cancel and Exit'. A sidebar on the left lists 'Quick Start (1)' with options for 'My AMIs (0)', 'AWS Marketplace (20)', 'Community AMIs (1048)', and a checked 'Free tier only' checkbox. At the bottom, there are links for 'Feedback', 'English (US)', and copyright information.

Figure 4: Ubuntu AMI

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

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.

Currently selected: t2.micro (Variable ECUs, 1 vCPU, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)								
	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized	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	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

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	1	Launch into Auto Scaling Group
Purchasing option	<input type="checkbox"/> Request Spot instances	
Network	vpc-8b5942e3 (default)	<input type="checkbox"/> Create new VPC
Subnet	No preference (default subnet in any Availability Zone)	<input type="checkbox"/> Create new subnet
Auto-assign Public IP	Use subnet setting (Enable)	
Placement group	<input type="checkbox"/> Add instance to placement group	
Capacity Reservation	Open	
Domain join directory	No directory	<input type="checkbox"/> Create new directory
IAM role	None	<input type="checkbox"/> Create new IAM role

Cancel Previous Review and Launch Next: Add Storage

Figure 6: Configure Instance details

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

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

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	e.g. SSH for Admin Desktop
HTTP	TCP	80	Anywhere	e.g. SSH for Admin Desktop

Add Rule Cancel Previous 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

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

Step 7: Review Instance Launch

- Press Launch

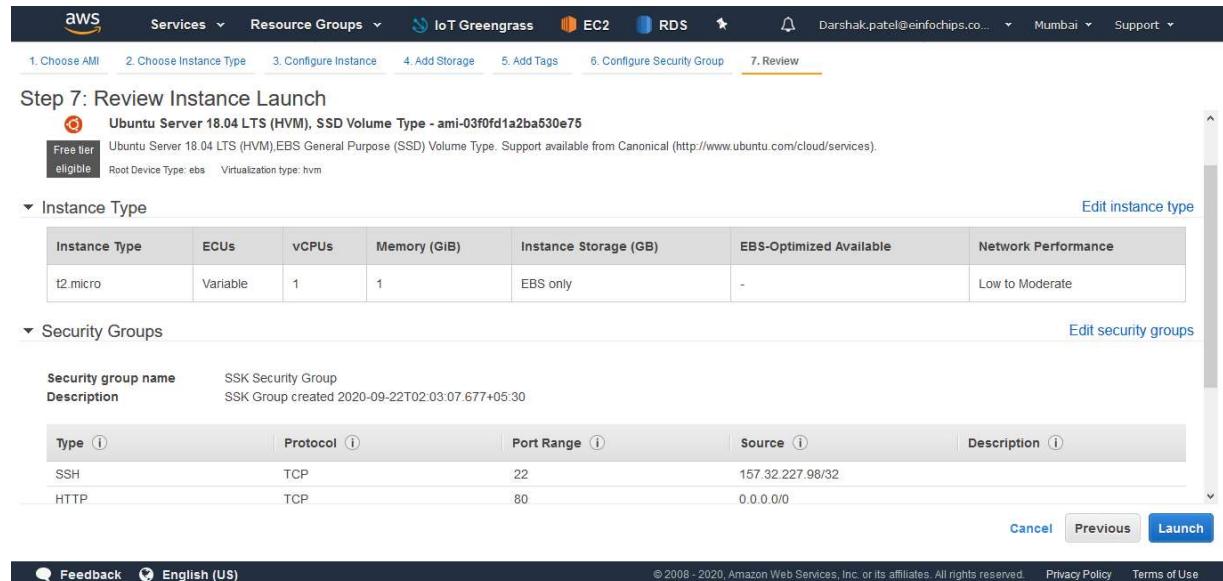


Figure 9: Review Instance

Step 8: Create New key pair

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

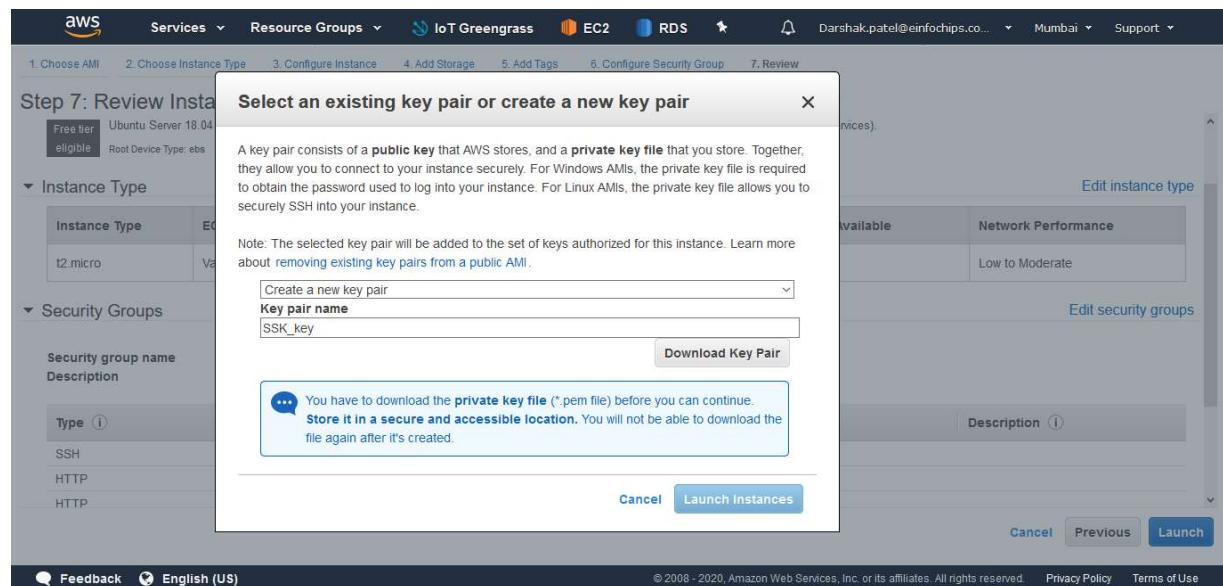


Figure 10: Configure key

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

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”.

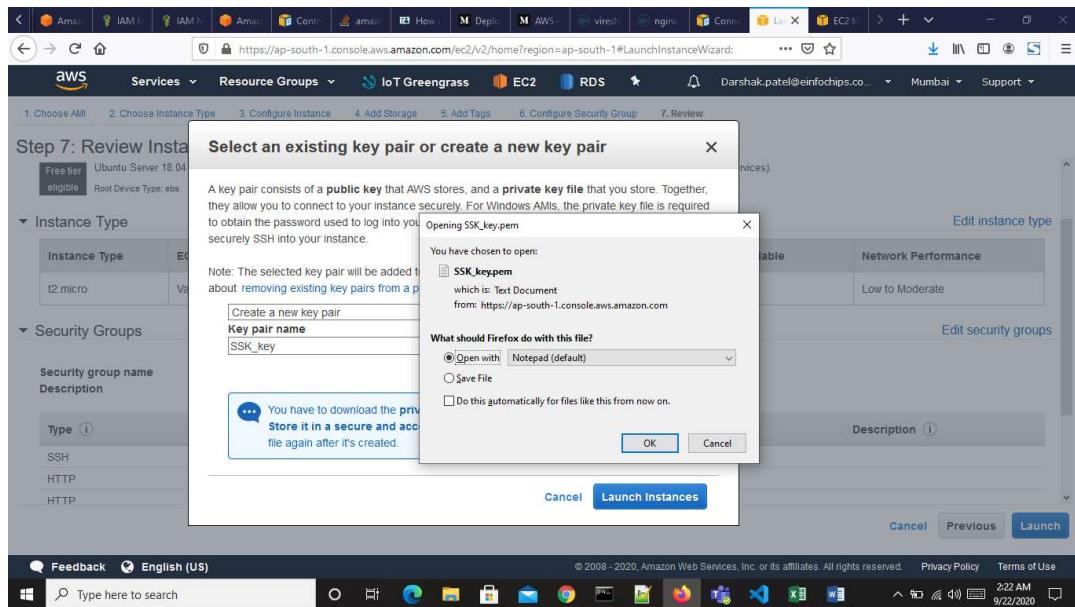


Figure 11: Download key

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.

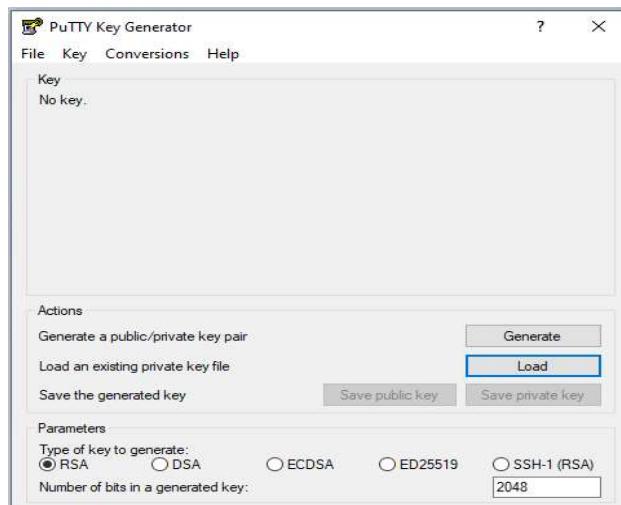


Figure 1: Convert PPK file

- 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](https://13.235.8.114)

Session Name: SSK EC2

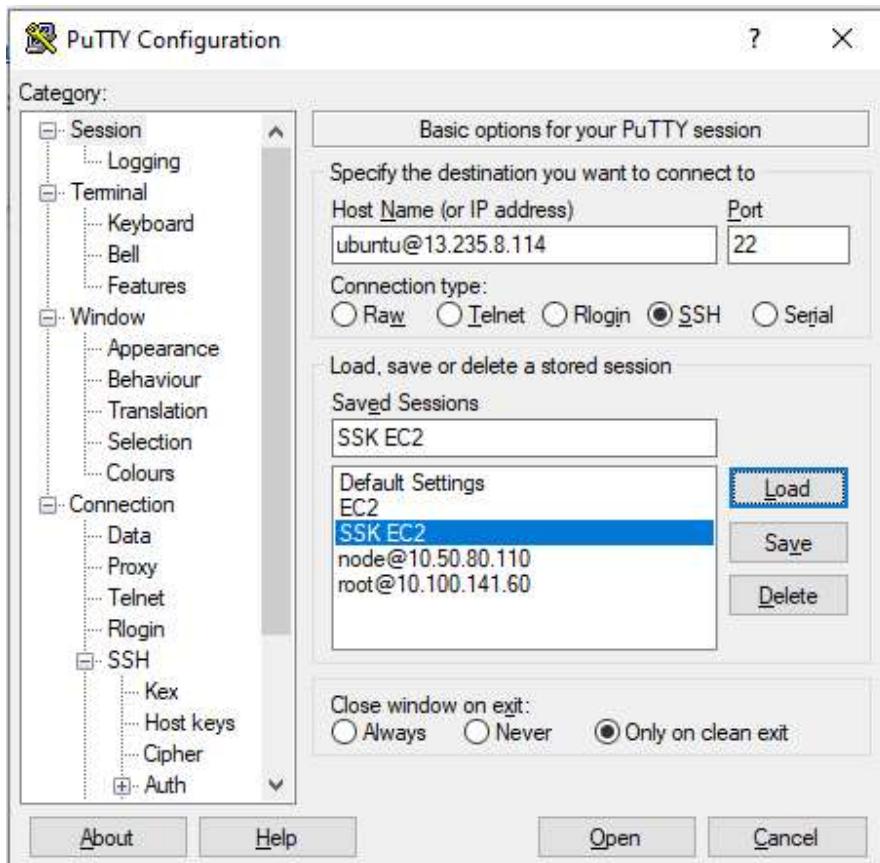


Figure 1: Configure putty

To configure key

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

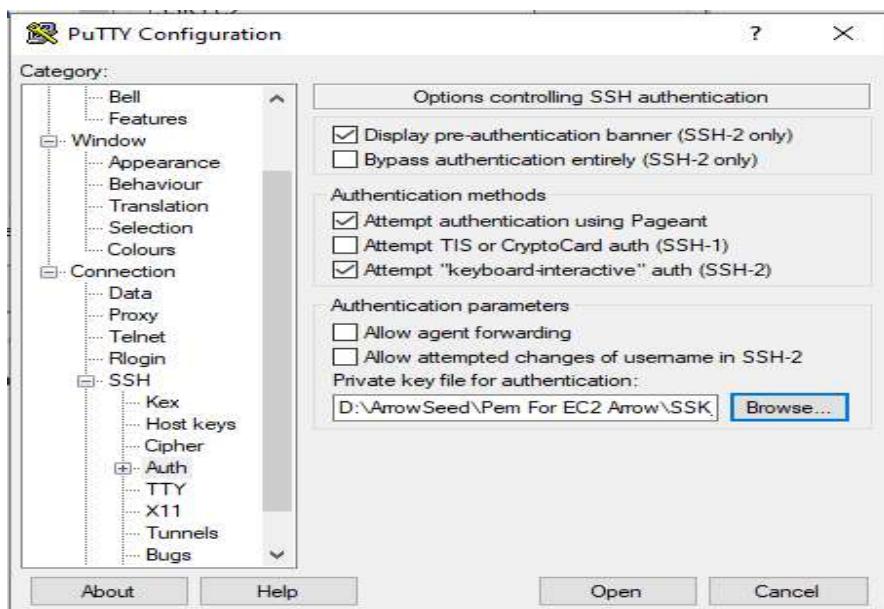


Figure 2: add key to putty

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

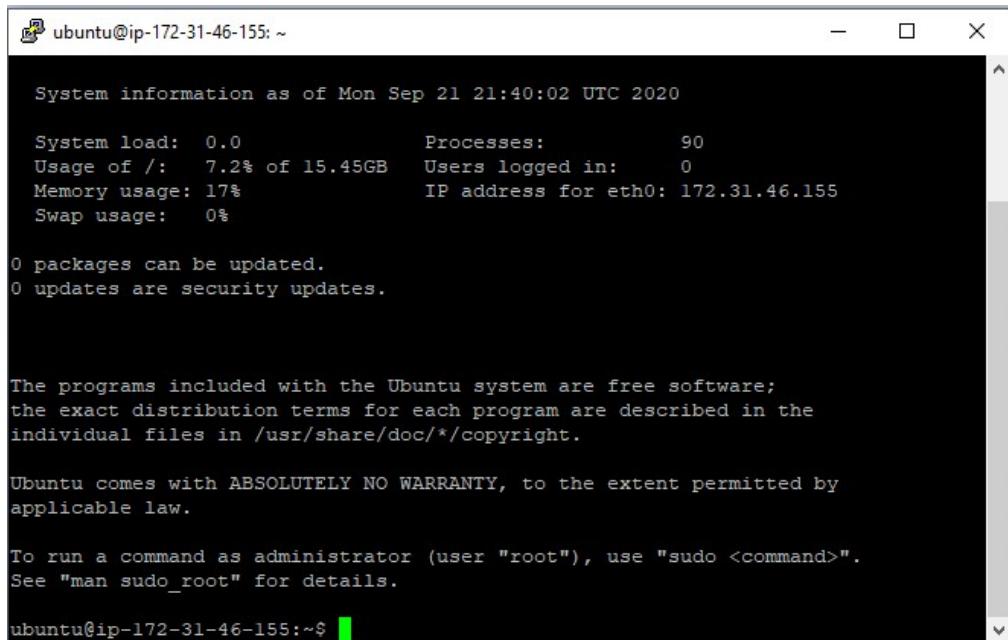


Figure 3: Connected AWS

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

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

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 navigation links like EC2 Dashboard, Instances, Images, and Elastic Block Store. The main area displays the 'Instance summary for i-04edc66ab39475528' for an instance named 'i-04edc66ab39475528'. The instance is listed as 'Running'. Key details shown include its Public IPv4 address (13.235.8.114), Public IPv4 DNS (ec2-13-235-8-114.ap-south-1.compute.amazonaws.com), VPC ID (vpc-8b5942e3), and Subnet ID (subnet-9bfcd9f3). Buttons for 'Connect' and 'Actions' are visible at the top right of the summary card.

Figure 1: Application access

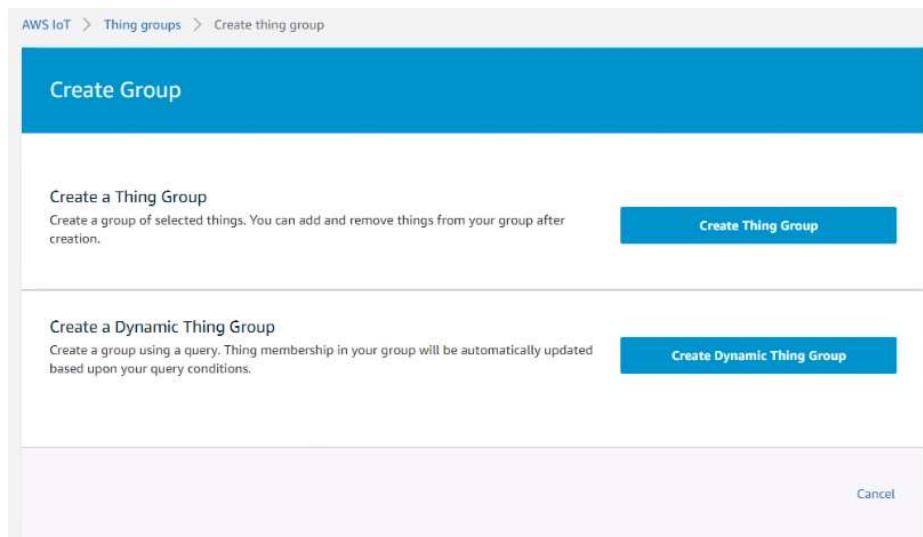
Make note of the Public IPv4 DNS address provided and it needs to be in the following format, with a leading HTTP:// as shown below;

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

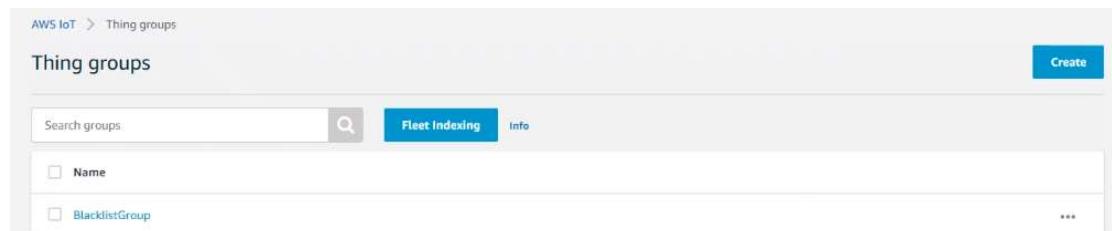
4.2 Application – “Allow” or “Deny” listing

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

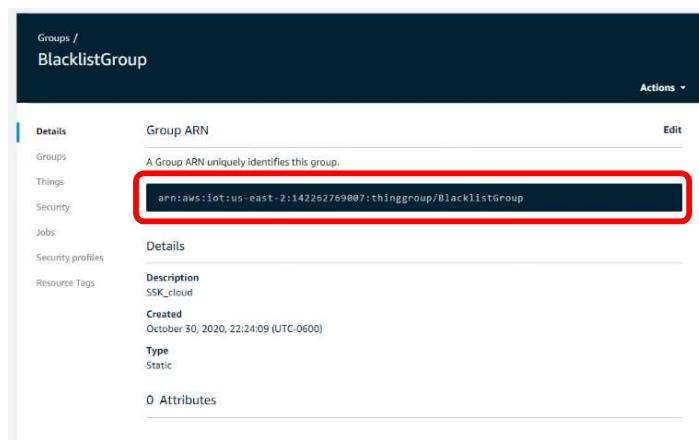
- To create ‘Thing group’ : navigate to IoT Core → Manage → Thing groups → Create



- 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;



SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

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

The screenshot shows the AWS IoT Policies page. At the top right is a blue 'Create' button. Below it is a search bar labeled 'Search policies' with a magnifying glass icon. A table lists several policies:

Name	...
esp32_amazon_freertos_policy	...
blacklistpolicy	...
PSoC64-Policy	...
PSoC64	...
My_IoT_Policy	...
GG_11_SSK_dev	...

- 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 Thing groups page. At the top right is a blue 'Create' button. Below it is a search bar labeled 'Search groups' with a magnifying glass icon. A table lists three groups:

Name	...
BlacklistGroup	...
whitelist	...

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

The screenshot shows the AWS IoT Groups / whitelist page. The navigation bar at the top says "AWS IoT > Thing groups > whitelist". The main title is "Groups / whitelist". On the left, there is a sidebar with links: Details, Groups, Things, Security, Jobs, Security profiles, and Resource Tags. The "Details" link is selected. The main content area has a header "Group ARN" with a value "arn:aws:iot:us-east-2:142262769007:thinggroup/whitelist" and an "Edit" button. Below this are sections for "Description" (with the note "You do not have description for the thing group yet."), "Created" (November 19, 2020, 09:35:55 (UTC-0700)), and "Type" (Static). At the bottom, it says "0 Attributes".

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

The screenshot shows the "Policy" selection dialog for the "whitelist" group. The title is "Groups / whitelist / whitelist". The main section is titled "Policy" with the sub-instruction "Browse and select the policies you want to attach to this group.". A modal window titled "Select a policy to attach to this group" is open, showing a list of policies: "Policies", "Search for a policy" (with a search icon), "esp32_amazon_freertos_policy", "blacklistpolicy", "PSoC64-Policy", and "PSoC64". At the bottom of this list is a "Select" button. Below the modal, there is a "Select another policy to attach to this group" section with a "No policy selected" message and a "Select" button. At the very bottom, there is a "Directly attached" section which is currently empty.

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

- 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" and "Browse and select the policies you want to attach to this group." Below this, there's a section titled "Select a policy to attach to this group" containing a single item: "blacklistpolicy". To the right of this item are "Remove" and "Select" buttons. Below this is another section titled "Select another policy to attach to this group" with a "No policy selected" message and a "Select" button. Underneath these sections is a box labeled "Directly attached" containing a JSON policy document:

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

At the bottom of the editor are "Cancel" and "Save" buttons.

The screenshot shows the AWS Groups page for the "whitelist" group. The top navigation bar includes "Groups / whitelist" and "Actions". The main content area has tabs for "Details", "Policies", and "Edit". The "Policies" tab shows that "blacklistpolicy" is attached to this group. The "Edit" button is to the right. Below this, under the "Security" tab, there's a "Things" section listing "Things in this group have the following permissions:" and a "Jobs" section listing "iot:>". A note indicates "EXPLICITLY DENIED" for the "iot:>" permission, with a link to "arm:aws:iot:us-east-2:142262769007:topic/replaceWithATopic".

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 options like Dashboard, Databases (which is selected), Query Editor, Performance Insights, Snapshots, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Events, Event subscriptions, Recommendations, and Certificate update. The main area is titled 'Databases' and shows a table with one row. The row contains the DB identifier 'seed-server', Role 'Instance', Engine 'MySQL Community', Region & AZ 'ap-south-1b', Size 'db.t2.micro', and a green checkmark icon. At the top of the main area, there are buttons for 'Group resources', 'Modify', 'Actions', 'Restore from S3', and 'Create database'. Below the table, there are navigation arrows and a search bar labeled 'Filter databases'.

4.4 Creating a Database

- Select “Standard Create”

The screenshot shows the 'Choose a database creation method' step in the AWS RDS 'Create New Database' wizard. It has two options: 'Standard Create' (selected) and 'Easy Create'. 'Standard Create' is described as setting all configuration options, including availability, security, backups, and maintenance. 'Easy Create' is described as using recommended best-practice configurations where some options can be changed after creation. Below this, there's a section for 'Engine options' with six engine types: Amazon Aurora, MySQL (selected), MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server. Each engine has its logo and a brief description. At the bottom, there are links for 'Feedback', 'English (US)', and standard footer links for 'Privacy Policy' and 'Terms of Use'.

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

- Select “Free Tier”

The screenshot shows the AWS RDS MySQL 8.0.20 creation wizard. In the 'Templates' section, the 'Free tier' option is selected, highlighted with a blue border. The 'Production' and 'Dev/Test' options are also shown. Below the templates, there are 'Settings' for the DB instance identifier, which is set to 'ssk-server'. The bottom navigation bar includes links for Feedback, English (US), Privacy Policy, and Terms of Use.

- 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. A red arrow points to the 'Include previous generation classes' checkbox, which is also selected. The 'Storage' section below shows 'General Purpose (SSD)' selected for storage type and '20 GiB' allocated storage. The bottom navigation bar includes links for Feedback, English (US), Privacy Policy, and Terms of Use.

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

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

The screenshot shows the AWS Database Connectivity configuration page. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, user info (Darshak.patel@einfochips.com @ 9784-2352-2010), location (Mumbai), and support links. The main section is titled 'Connectivity' under 'Virtual private cloud (VPC)'. It shows a dropdown menu set to 'Default VPC (vpc-8b5942e3)'. A note says 'Only VPCs with a corresponding DB subnet group are listed.' Below this is a message: 'After a database is created, you can't change the VPC selection.' Under 'Database authentication', the 'Password authentication' option is selected, with a note: 'Authenticates using database passwords.' Other options like 'Password and IAM database authentication' and 'Password and Kerberos authentication' are also listed. At the bottom, there are links for 'Feedback', 'English (US)', and legal notices: '© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.', 'Privacy Policy', and 'Terms of Use'.

- Select “Publicly accessible” under Additional Connectivity Configuration

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

Connectivity

Subnet group: default-vpc-c202c1a9

Security group: Choose security groups
default

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.

Cancel **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 a database named 'ssk-server'. The 'Connectivity & security' tab is selected. In the 'Endpoint & port' section, the endpoint is listed as 'ssk-server.cqzdvbumpk6e.us-east-2.rds.amazonaws.com'.

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

The screenshot shows the AWS RDS console for a database named 'ssk-server'. The 'Configuration' tab is selected. In the 'Instance' section, under the 'Availability' heading, the 'Master username' is listed as 'admin_id'.

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

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

- 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 service page for a MySQL database instance. The left sidebar lists various options like Dashboard, Databases, and Snapshots. The main content area is titled "Connectivity & security". It displays endpoint details such as Endpoint (ssk-server.cqzdvbumpk6e.us-east-2.rds.amazonaws.com) and Port (3306). Under "Networking", it shows the Availability zone (us-east-2c), VPC (vpc-c202c1a9), and Subnet group (default-vpc-c202c1a9). In the "Security" section, it lists the VPC security groups (default (sg-9f61e6f8) (active)), Public accessibility (Yes), Certificate authority (rds-ca-2019), and Certificate authority date (Aug 22nd, 2024). Below this, the "Security group rules (2)" section shows two entries: "default (sg-9f61e6f8)" with Type CIDR/IP - Inbound and Rule 0.0.0.0/0, and another entry for "default (sg-9f61e6f8)" with Type CIDR/IP - Outbound and Rule 0.0.0.0/0.

Inbound Rules:

The screenshot shows the AWS EC2 Security Groups "Edit inbound rules" page for the "sg-9f61e6f8 - default" group. The top navigation bar includes links for EC2, Security Groups, sg-9f61e6f8 - default, and Edit inbound rules. The main content area is titled "Inbound rules" and shows two existing rules. Rule 1 is for All traffic (Protocol All, Port range All) and Rule 2 is for All traffic (Protocol All, Port range All). Both rules have a "Delete" button. A "Add rule" button is located below the rules. A note at the bottom states: "⚠️ 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 right are "Cancel", "Preview changes", and a highlighted "Save rules" button.

Outbound Rules:

The screenshot shows the 'Edit outbound rules' section of the AWS EC2 Security Groups configuration. It displays two outbound rules:

- Outbound rule 1:** Type: All traffic, Protocol: All, Port range: All. Destination type: Custom, Destination: 0.0.0.0/0.
- Outbound rule 2:** Type: All traffic, Protocol: All, Port range: All. Destination type: Custom, Destination: ::/0.

A note at the bottom states: "⚠️ 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." Buttons at the bottom include 'Cancel', 'Preview changes', and 'Save rules'.

4.5 Creating an IAM User

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

The screenshot shows the 'Identity and Access Management (IAM)' service in AWS. The 'Users' section is selected. A prominent 'Add user' button is visible at the top left of the main content area. The table below lists three existing users: ADMIN, root_user, and User_Name, along with their group assignments and access key ages.

User name	Groups	Access key age	Password age
ADMIN	Administrator	307 days	307 days
root_user	Administrator	4 days	None
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 2: Set user details. A 'User name*' field contains 'NAME'. Under 'Access type*', the 'Programmatic access' checkbox is checked. Step 3: Set permissions is visible at the bottom.

- 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 3: Set permissions. The 'Add user to group' section shows the 'Administrator' group selected. The 'Attached policies' table shows 'AdministratorAccess' attached. Step 4: Set permissions boundary is visible at the bottom.

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

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

The screenshot shows the 'Create user' wizard in the AWS IAM console. The current step is 'Review'. The user details section shows a User name of 'NAME' and an AWS access type of 'Programmatic access - with an access key'. The Permissions boundary is set to 'Permissions boundary is not set'. The Permissions summary section indicates that the user will be added to the 'Administrator' group. The Tags section shows 'No tags were added.' At the bottom, there are 'Cancel', 'Previous', and 'Create user' buttons.

- 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.

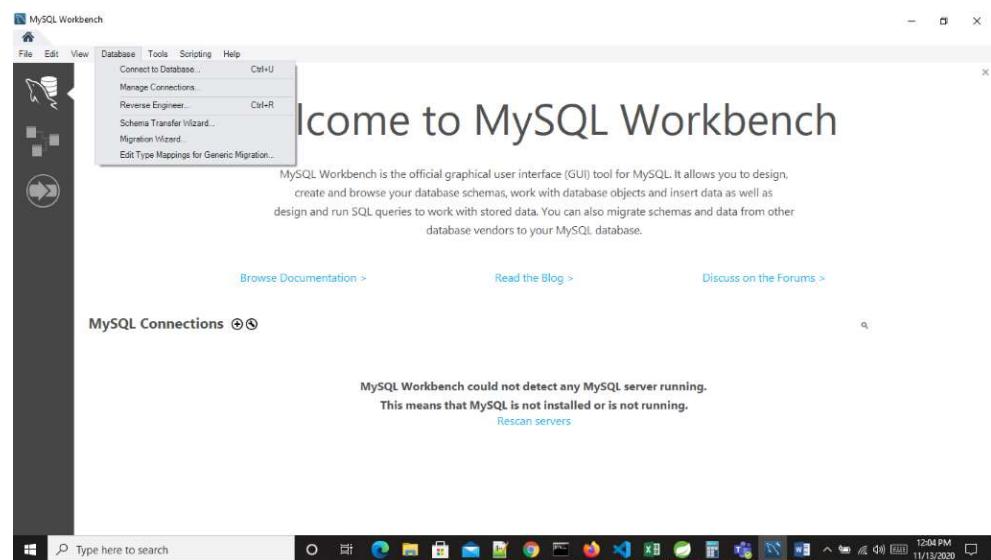
<https://dev.mysql.com/downloads/workbench/>

2. Install Postman

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

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

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE



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

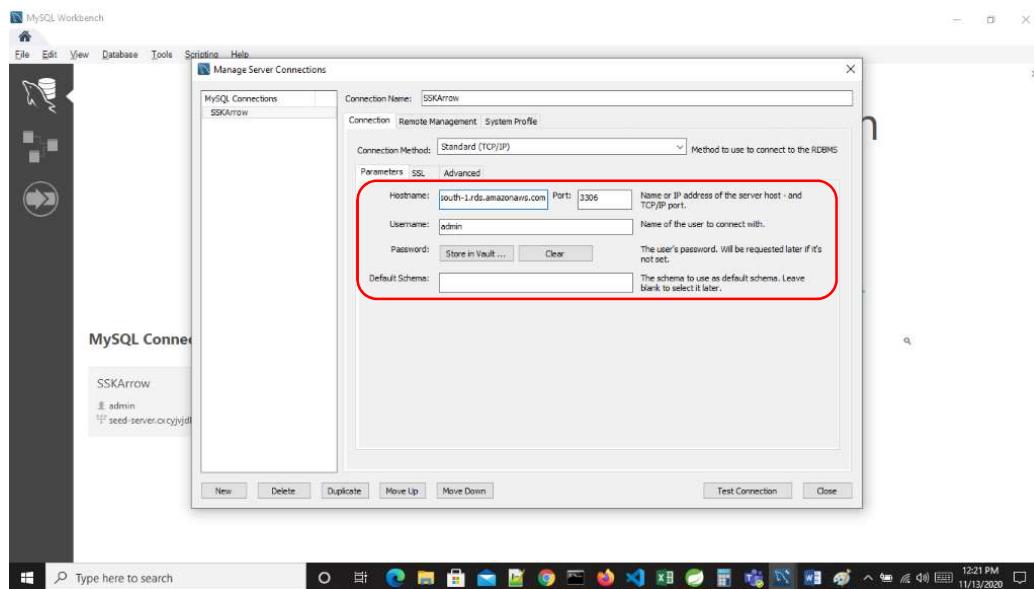
Connection Name: <Name of connection>

Host Name: <AWS RDS HOST URL>

Port: 3306 (Default value)

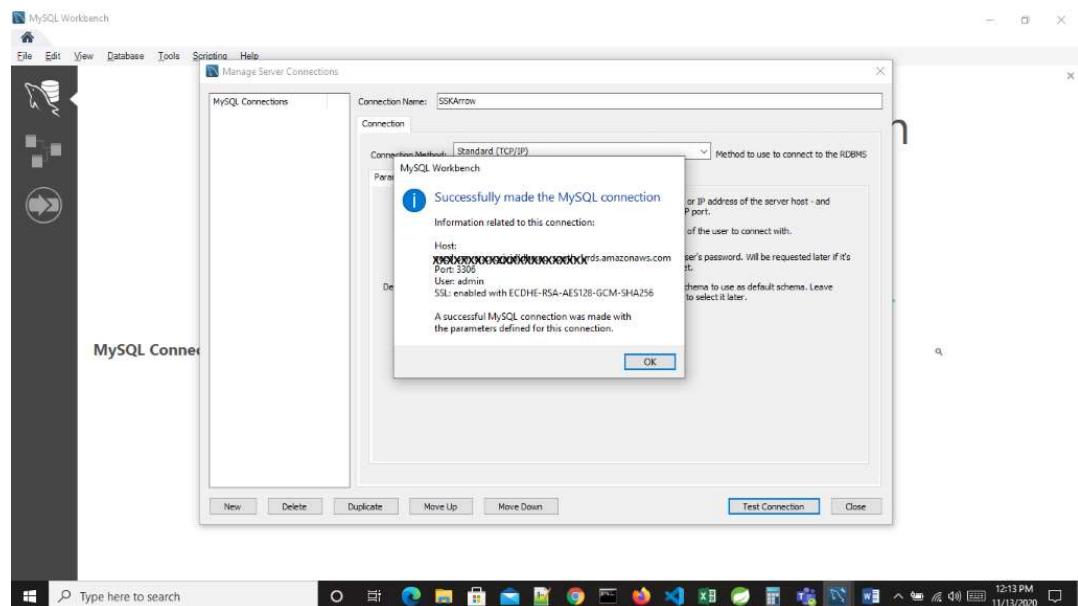
User Name: < AWS RDS User name>

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

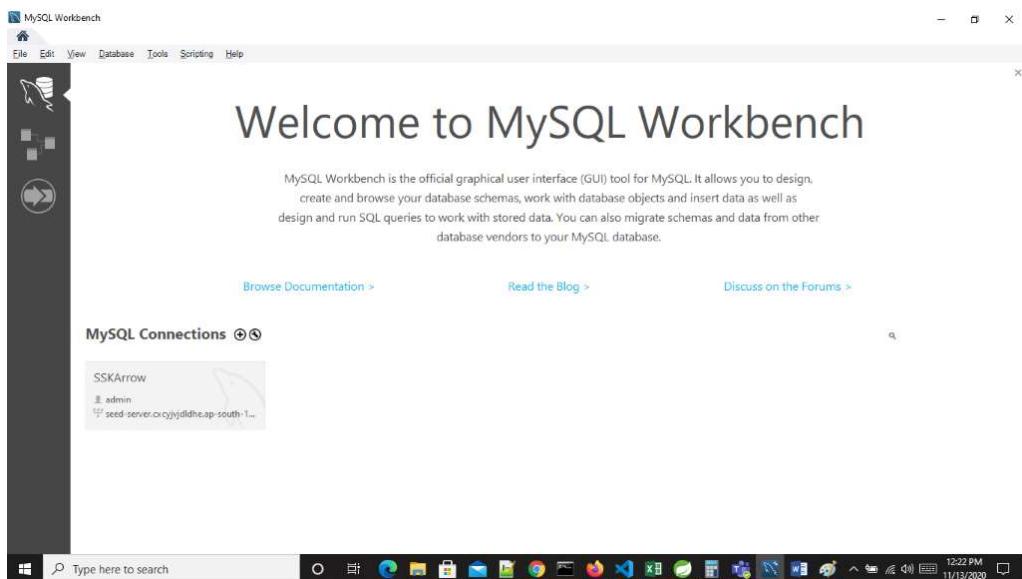


SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

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

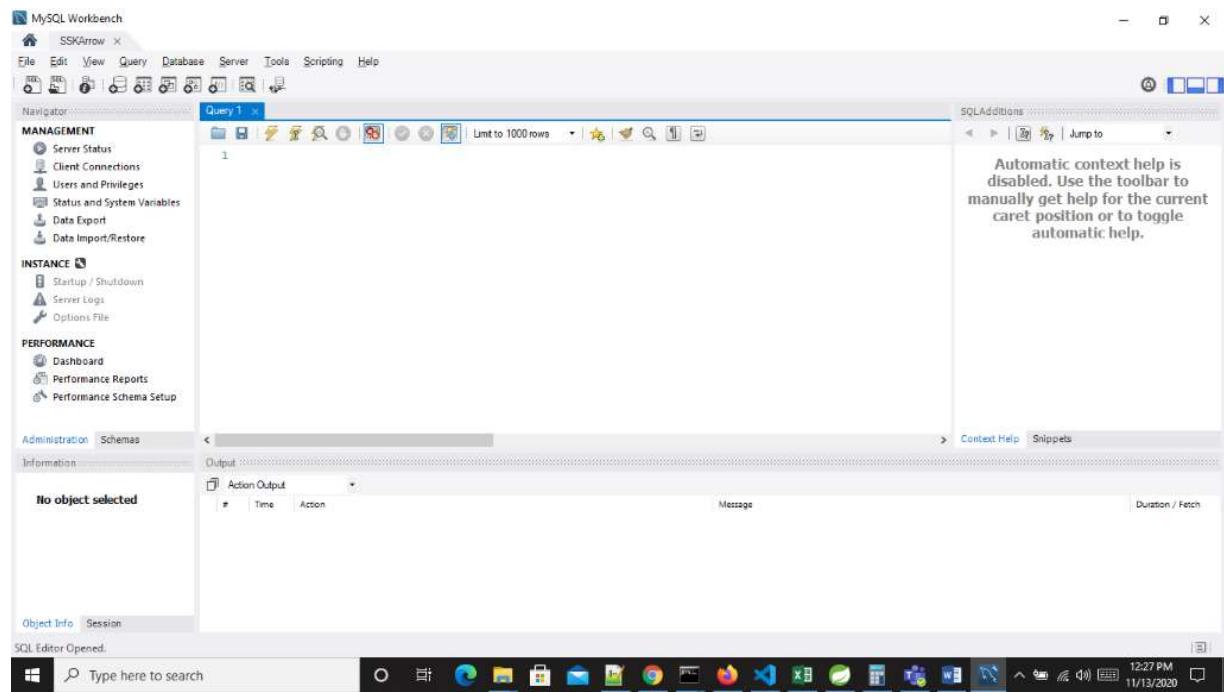


6. Further it will show following details. Click on created connection button.

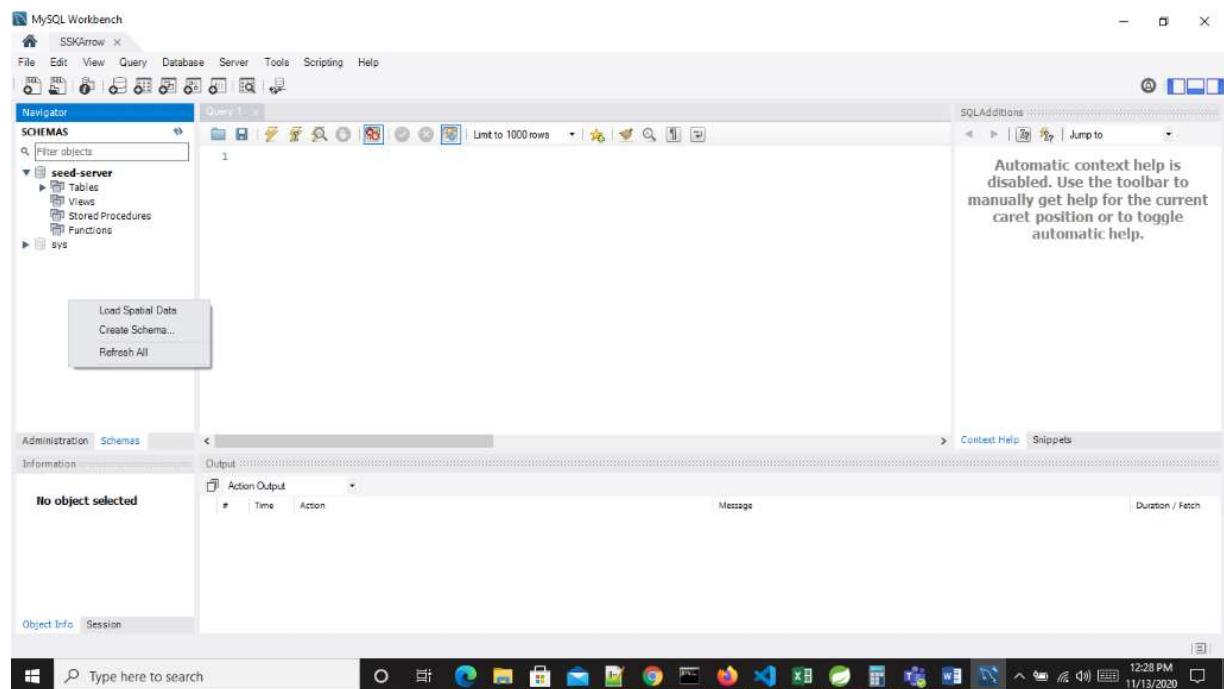


SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

7. It will open Administration tab.

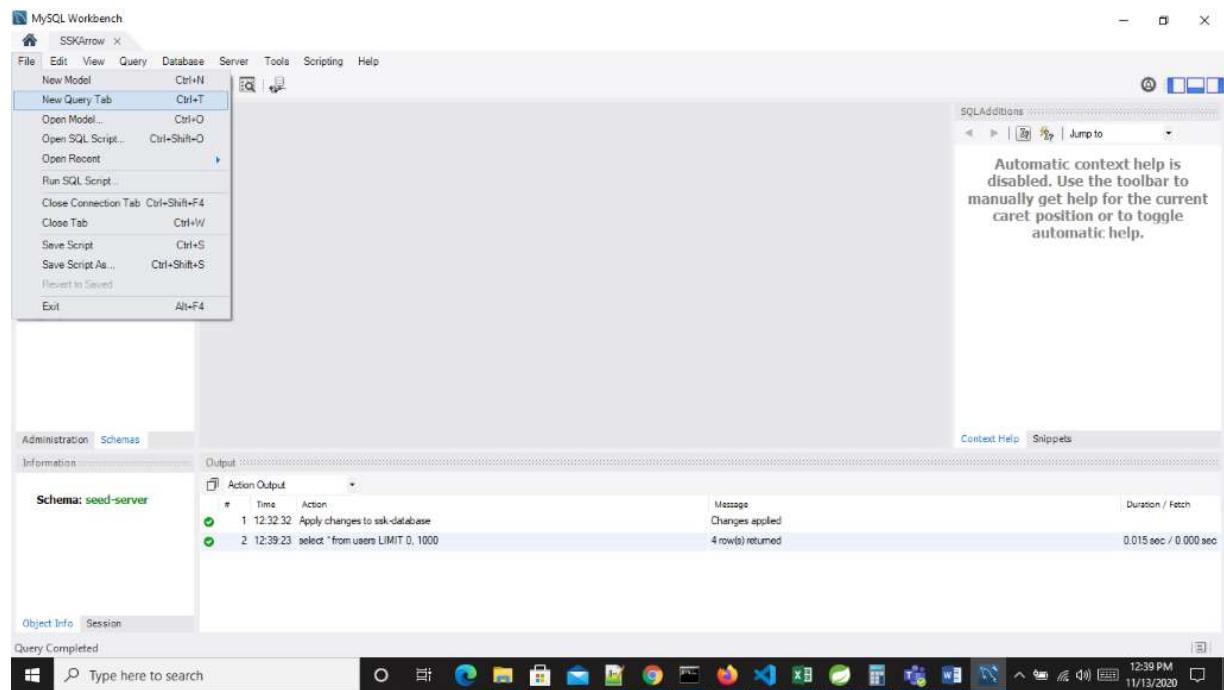


8. Select Schemas Tab & right click on mouse.



SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

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



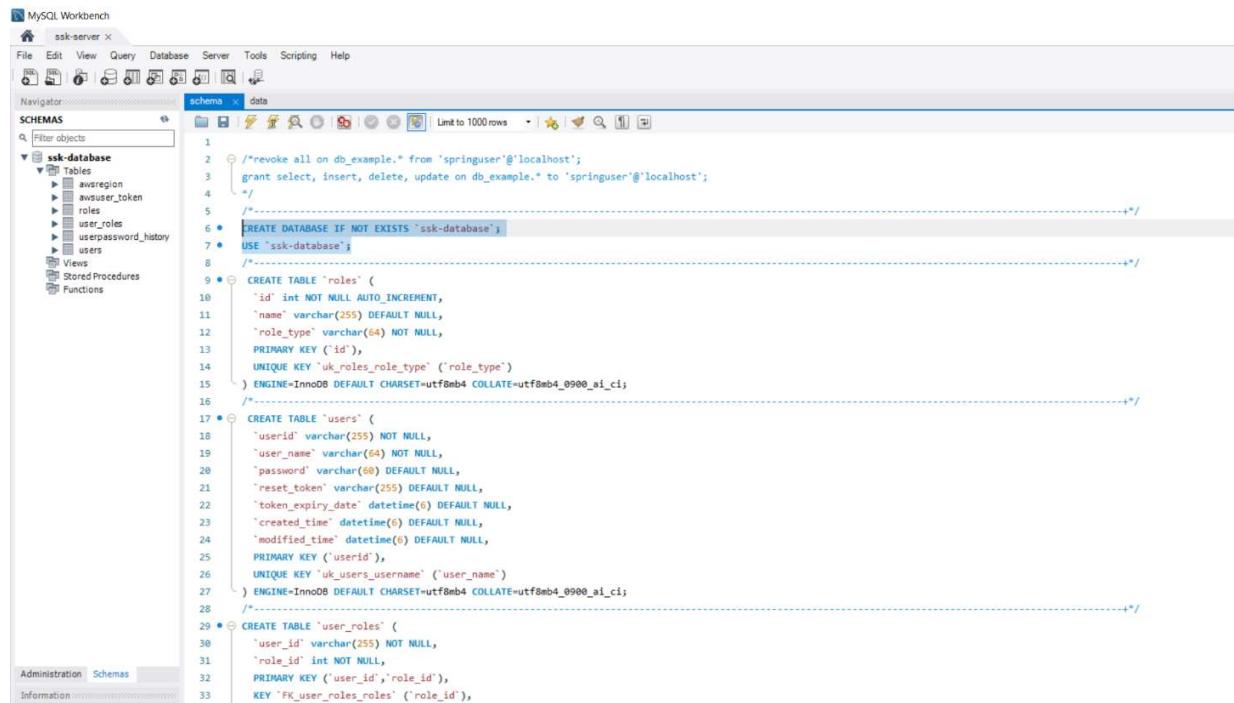
10. Edit and Execute schema.sql

- 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



SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE



```

MySQL Workbench
ssk-server x
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  /*revoke all on db_example.* from 'springuser'@'localhost';
2  grant select, insert, delete, update on db_example.* to 'springuser'@'localhost';
3  */
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(60) 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

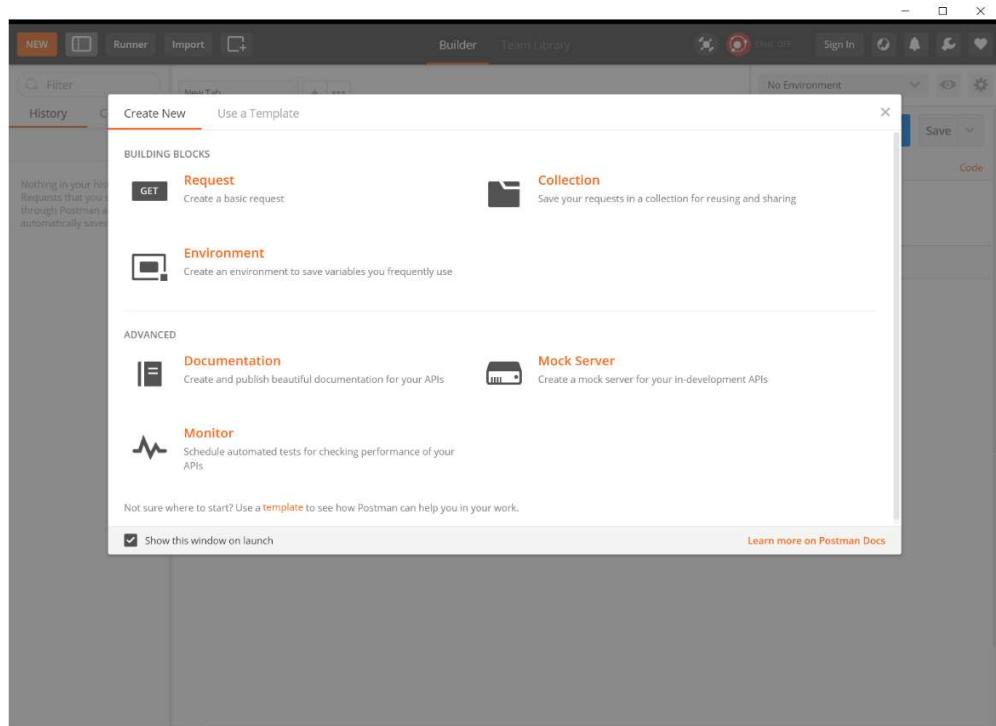
```

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());

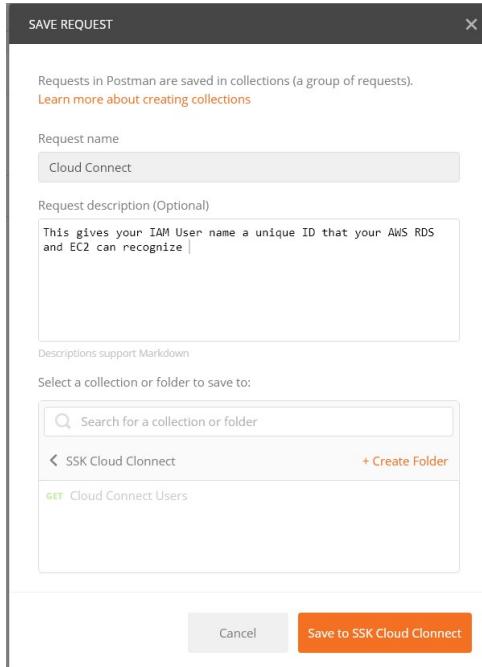
```

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

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE



- 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

SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

<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 application interface. The top navigation bar includes 'NEW', 'Runner', 'Import', 'Builder' (which is highlighted), 'Team Library', 'Sync Off', 'Sign In', and various notification and settings icons. On the left, there's a sidebar with 'History' (selected), 'Collections', 'Clear all', and a 'Today' section containing a recent item: 'GET https://iam.amazonaws.com/?Action= GetUser&UserName=IAM-username&Version=2010-05-08'. The main workspace is the 'Builder' tab, where a GET request is defined to the same URL. The 'Authorization' tab is active, showing 'AWS Signature' selected for the type. The 'Service Name' field is filled with 'iam'. Other tabs like 'Headers (4)', 'Body', 'Pre-request Script', and 'Tests' are visible but inactive.

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

The screenshot shows the Postman application interface with the 'Response' tab selected. The status bar indicates 'Status: 200 OK' and 'Time: 437 ms'. The response body is displayed in XML format. Line 7 of the XML output highlights the User ID 'AIDASCH4FSVXRM7SBDZ5T' in blue, indicating it is the copied value.

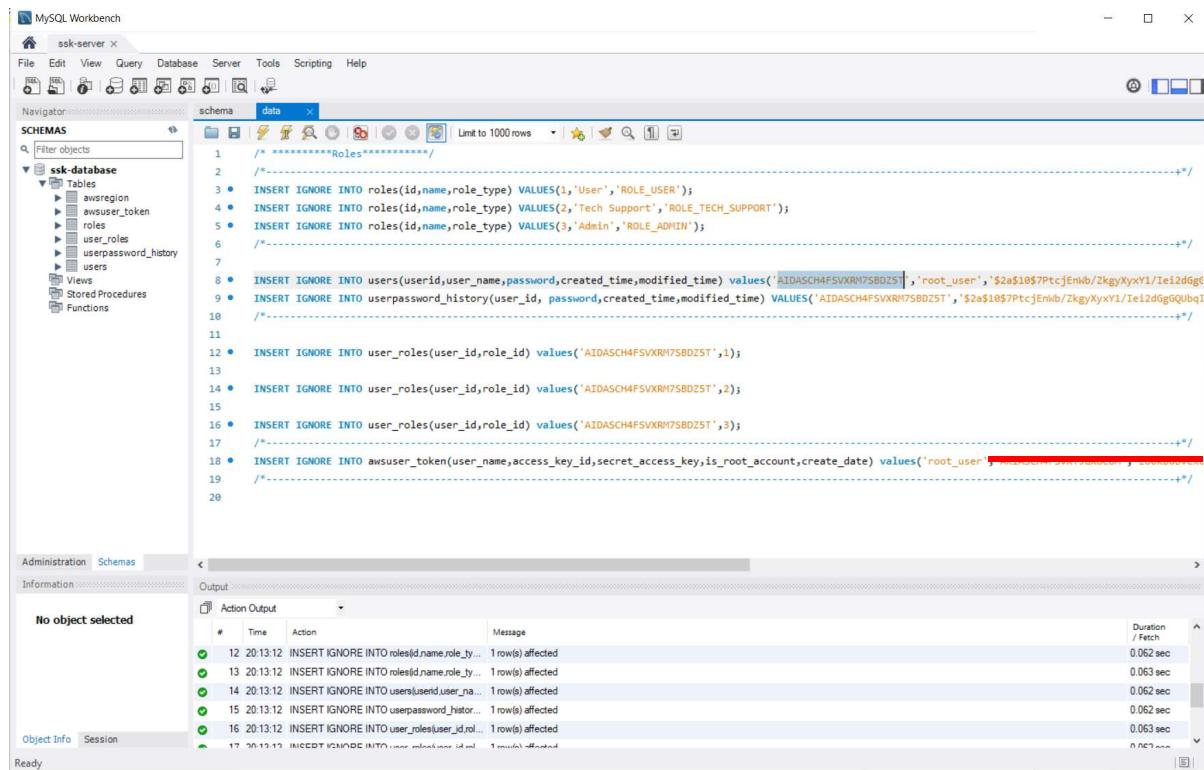
```

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@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(user_id,user_name,password,created_time,modified_time) values('AIDASCH4FSVXRH7SBDZ5T','root_user','$2a$10$7PtcjEnib/ZkgYXyxY1/Iei2dGgC
9  • INSERT IGNORE INTO userpassword_history(user_id, password,created_time,modified_time) VALUES('AIDASCH4FSVXRH7SBDZ5T','$2a$10$7PtcjEnib/ZkgYXyxY1/Iei2dGgGQuJbqI
10 /*
11
12 • INSERT IGNORE INTO user_roles(user_id,role_id) values('AIDASCH4FSVXRH7SBDZ5T',1);
13
14 • INSERT IGNORE INTO user_roles(user_id,role_id) values('AIDASCH4FSVXRH7SBDZ5T',2);
15
16 • INSERT IGNORE INTO user_roles(user_id,role_id) values('AIDASCH4FSVXRH7SBDZ5T',3);
17 /*
18 • INSERT IGNORE INTO awuser_token(user_name,access_key_id,secret_access_key,is_root_account,create_date) values('root_user',...,...,...,...)
19 /*
20

```

No object selected

Action Output

#	Time	Action	Message	Duration / Fetch
12	20:13:12	INSERT IGNORE INTO roles(id,name,role_ty...	1 row(s) affected	0.052 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(user_id,user_n...	1 row(s) affected	0.062 sec
15	20:13:12	INSERT IGNORE INTO userpassword_hist...	1 row(s) affected	0.056 sec
16	20:13:12	INSERT IGNORE INTO user_roles(user_id,...	1 row(s) affected	0.063 sec
17	20:13:12	INNOCET LOADING INITIALLY INDEXES	1 rows(s) affected	0.057 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)

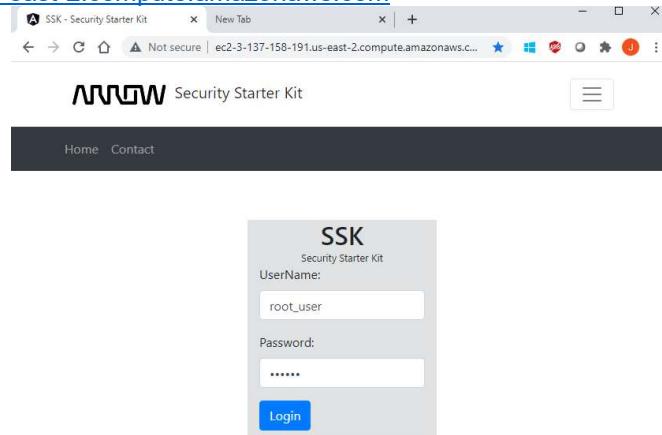
```
$sudo docker run --name ssk-server -e
```

```
MYSQL_URL="jdbc:mysql://<RDS_ENDPOINT>:3306/<SCHEMA_NAME>?useSSL=false&serverTimezone=UTC&useLegacyDatetimeCode=false&allowPublicKeyRetrieval=true" -e MYSQL_UNAME="" -e MYSQL_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>



SECURITY STARTER KIT CLOUD CONNECT – INSTALLATION & SETUP GUIDE

- 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.docx 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/>