RocketCX Installation And Configuration Guide

RocketCX

ECS-RCX-IG-1.0.0

Template version: 1.0

Template effective date: February 20th, 2015

**Table of Contents**

1 Overview 4

1.1 Features 4

1.2 Benefits 4

1.3 References 5

2 Intended Audience 5

3 Application Dependencies 5

4 Prerequisites 5

4.1 ServiceNow Requirements 5

4.2 AWS Requirements 6

4.3 Supported Browsers 6

4.4 PC Requirements 6

4.5 Browser Configurations 6

5 Installation Steps 7

5.1 Overview of Installation process 7

5.2 RocketCX Registration 8

5.2.1 Amazon Marketplace Subscription 8

5.2.2 RocketCX portal registration 9

5.2.3 RocketCX Instance Configuration 10

5.2.4 Capture required data from RocketCX portal 11

5.3 AWS Configuration 12

5.3.1 Amazon Connect 12

5.3.1.1 Setup Amazon Connect Instance 12

5.3.1.2 Configure Call Recordings to S3 bucket 14

5.3.1.3 Configure CORS for S3 bucket 15

5.3.1.4 Whitelist ServiceNow Instance URL in Amazon Connect 16

5.3.2 Configure IAM users 17

5.3.2.1 Create IAM user 17

5.3.2.2 Create IAM Policy 18

5.3.2.3 Attach IAM policy to IAM user 21

5.3.3 Capture AWS data required for RocketCX 22

5.4 ServiceNow Configuration 28

5.4.1 Install Openframe plugin 28

5.4.2 Request & Install RocketCX Enterprise Connector for ServiceNow 28

5.4.2.1 Request RocketCX application from ServiceNow store 28

5.4.2.2 Install RocketCX on ServiceNow instance 29

5.4.3 RocketCX Roles and Groups 29

5.4.3.1 Create Groups 30

5.4.4 Restricted Caller Access Privileges 32

5.4.5 Configure Openframe plugin 33

5.4.6 Configuration for Click to Dial 34

5.4.7 Configure RocketCX 35

5.4.7.1 Amazon Connect Queue 35

5.4.7.2 Admin Console 36

5.4.7.2.1 Amazon Connect Configuration 36

5.4.7.2.2 ServiceNow Configuration 37

5.4.7.2.3 RocketCX Portal Configuration 38

5.4.8 Setup Service account for RocketCX API Access 41

6 Post Install & Testing 42

6.1 Set up Lambda and Amazon Connect Contact flows 42

7 External systems connection 42

8 Demo Data 42

9 Developer Section 43

9.1 User details API endpoint 43

9.2 Record details API endpoint 44

9.3 Active Module API endpoint 46

9.4 Incident record details API endpoint 47

9.5 HR case record details API endpoint 48

9.6 CSM case record details API endpoint 48

9.7 Survey API endpoint 48

10 Support and Troubleshooting 49

10.1 Support 49

10.2 Troubleshooting 49

10.2.1 Amazon Connect Troubleshooting Issues with the CCP 49

10.2.2 Connectivity Tool 49

# Overview

RocketCX Enterprise Connector for ServiceNow (RocketCX) is a computer telephony integration (CTI) between Amazon Connect and ServiceNow. It provides service centre agents with access to a comprehensive set of telephony controls within the ServiceNow interface meaning they do not need to switch screens to make and receive calls.

RocketCX includes a number of agent productivity features listed below

## Features

* **ServiceNow Integration** – All standard call handling features available natively within the ServiceNow interface – Answer, Ignore, Place on hold, Consult, Transfer, Conference and Manage Agent Status.
* **Intelligent Dialling** – Click on any telephone number or contact field within ServiceNow to make an outbound call. RocketCX intelligently handles country and dialling codes providing a one click dialling experience for outbound contact centres. Create favourites for even faster access.
* **Screen Pop** – Display caller details along with any open incidents or cases before the call is answered, enabling the agent to provide a warm start to the call.
* **Automated Call Logging & Recording** – Call logs are created automatically capturing contact and agent details, call duration, summary, notes, disposition codes, customer satisfaction score and access to a full recording of the call.
* **Create Record** – Create ServiceNow Incidents and Cases from within the RocketCX interface linking contacts, call logs and call recordings automatically.
* **Reporting & Dashboards** – Real-time service centre dashboard and historical trend reports provide the visibility supervisors need to manage their teams all within ServiceNow.
* **Easy to Configure –** Customisable disposition codes and related record lookups allow the agent experience to be fully customised.

## Benefits

* Telephony system integration with ServiceNow eliminating frequent screen switching to make and receive calls
* Outbound calling by clicking on the user in ServiceNow eliminating manual dialling effort.
* Reports, Metrics and Dashboard to assess Service Centre performance.
* Call records available on ServiceNow to review agent performance.

## References

| # | Document Identifier | Document Title |
| --- | --- | --- |
| 1 | ECS-RCX-AD-1.0.0 | RocketCX Application Design 1.0.0 |
| 2. | ECS-RCX-UG-1.0 | RocketCX User Guide v1.0 |
| 3. | ECS-RCX-TP-1.0.0 | RocketCX Test Plan 1.0.0 |
| 4. | Not Applicable | [RocketCX Quick Start Guide](https://rocketcx.s3.amazonaws.com/EnterpriseConnectorforServiceNow/v1/Documents/Public/RocketCX+Quick+Start+Guide.docx) |

# Intended Audience

The intended audience of this document is ServiceNow Administrators who are familiar with Amazon Web Services and looking to explore telephony integration between ServiceNow and Amazon Connect.

Readers are not required to know all the details of Amazon Connect service. However, some knowledge of Amazon Connect basic concepts would be advantageous for using this guide.

# Application Dependencies

* Openframe plugin
* RocketCX creates its own Tables. There are no dependencies on the system tables.
* JS Scripts (these scripts are installed on ServiceNow instance as a part of RocketCX):

1. x\_ecsd\_amazon\_conn.angular.min
2. x\_ecsd\_amazon\_conn.aws\_common
3. x\_ecsd\_amazon\_conn.connect-streams

# Prerequisites

## ServiceNow Requirements

ServiceNow instance with London or Madrid release.

A platform engine subscription that includes custom table entitlements, as per the latest ServiceNow licensing model.

RocketCX Supports Incident Management, Customer Service and Human Resource Core plugins. It is recommended that at least one plugin should be active on ServiceNow instance before installing RocketCX.

## AWS Requirements

AWS account with console access having full permissions on all AWS services or at least services like IAM, S3, Amazon connect, CloudFormation and Lambda.

This AWS account hereinafter will be referred to as AWS Admin.

**Note**: If you do not already have an AWS account, create one at [https://aws.amazon.com](https://aws.amazon.com/) by following the on-screen instructions.

## Supported Browsers

Google Chrome – latest release – Version 78.0

**Note:** Please open “Help->About Google Chrome” to check your browser version.

## PC Requirements

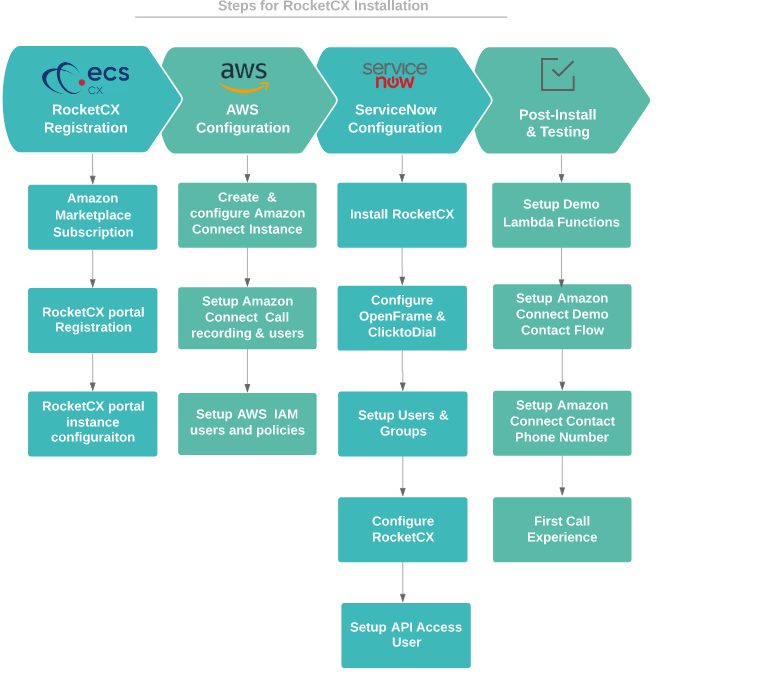
1. Windows 7 or above with 4GB RAM
2. USB headsets are recommended for best audio quality

## Browser Configurations

1. Allow Popups (add domain)
2. Allow Mic
3. Allow Sound

# Installation Steps

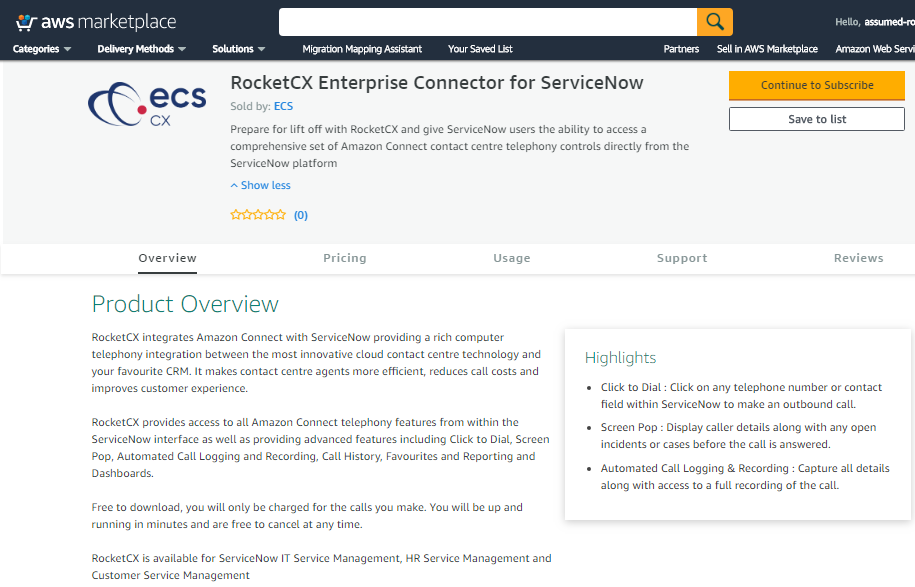
## Overview of Installation process



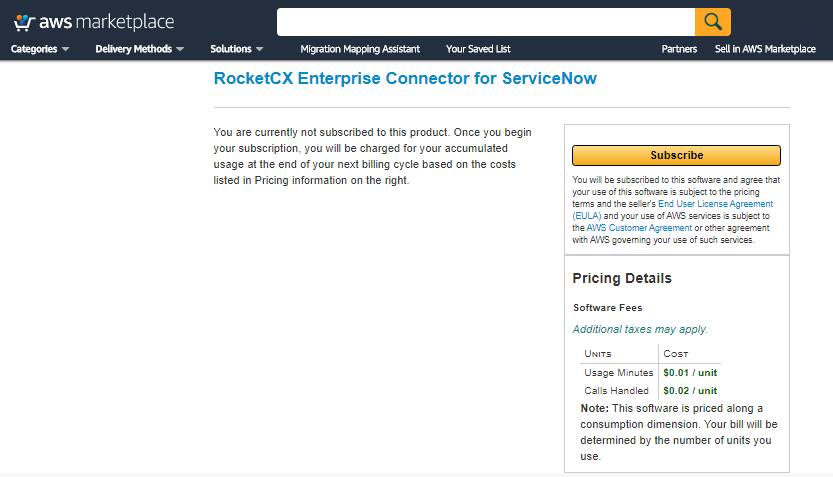
## RocketCX Registration

### Amazon Marketplace Subscription

1. Login to AWS console <https://console.aws.amazon.com/console/home> using [AWS Admin](#AWS_Admin) account.
2. Go to Amazon marketplace <https://aws.amazon.com/marketplace> and search for “RocketCX Enterprise Connector for ServiceNow”
3. Click **Continue to Subscribe** on RocketCX product page.

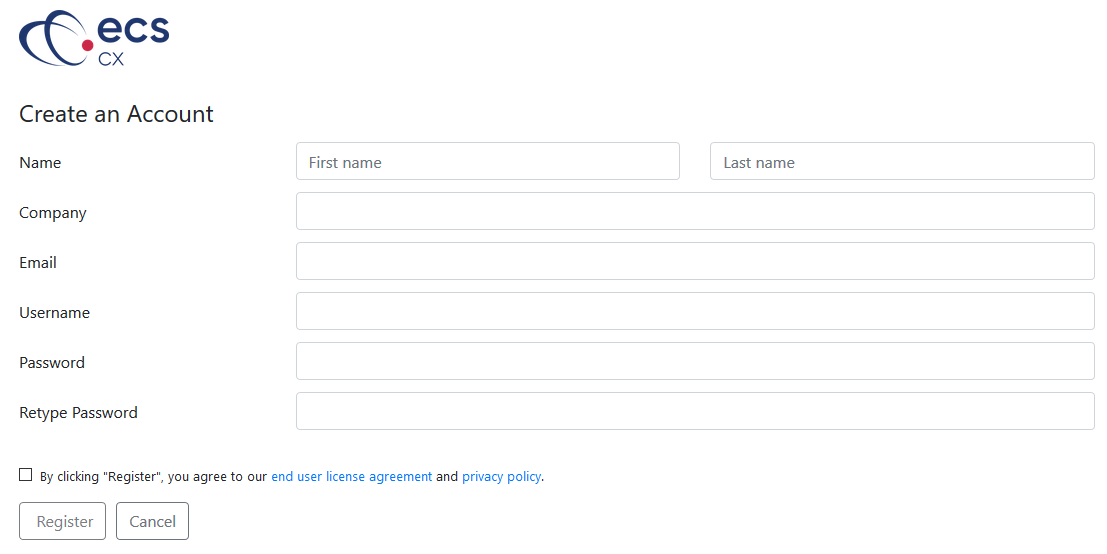


1. Click **subscribe** button and you will be redirected to RocketCX portal.

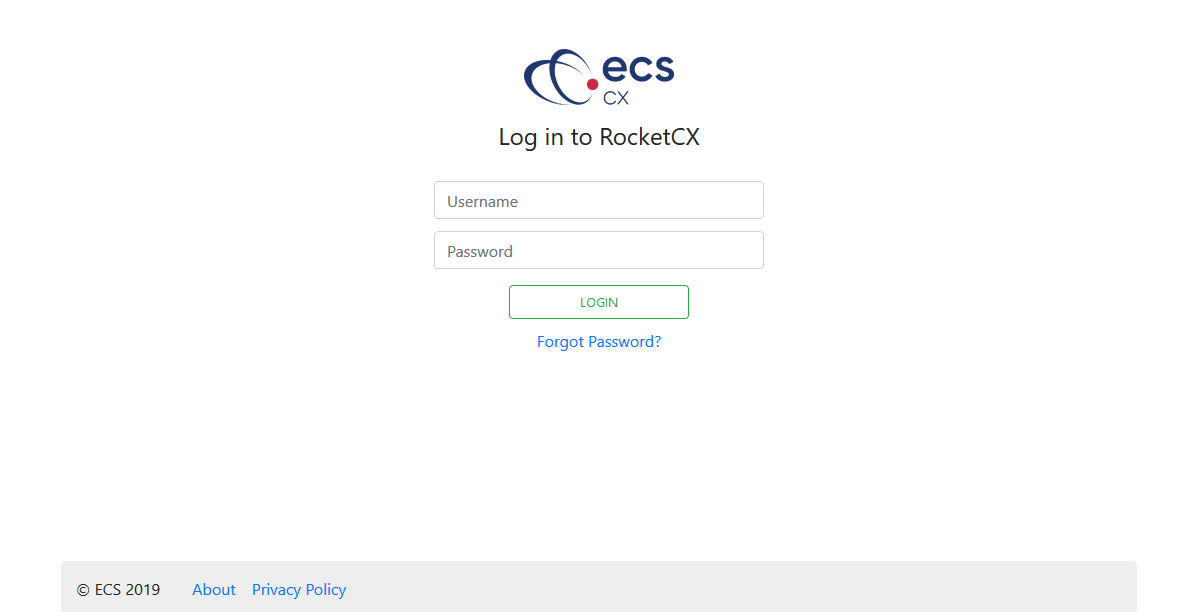


### RocketCX portal registration

1. After redirection from Amazon marketplace on RocketCX portal, sign-up to create your account. To sign-up, click Register after filling all the required details.

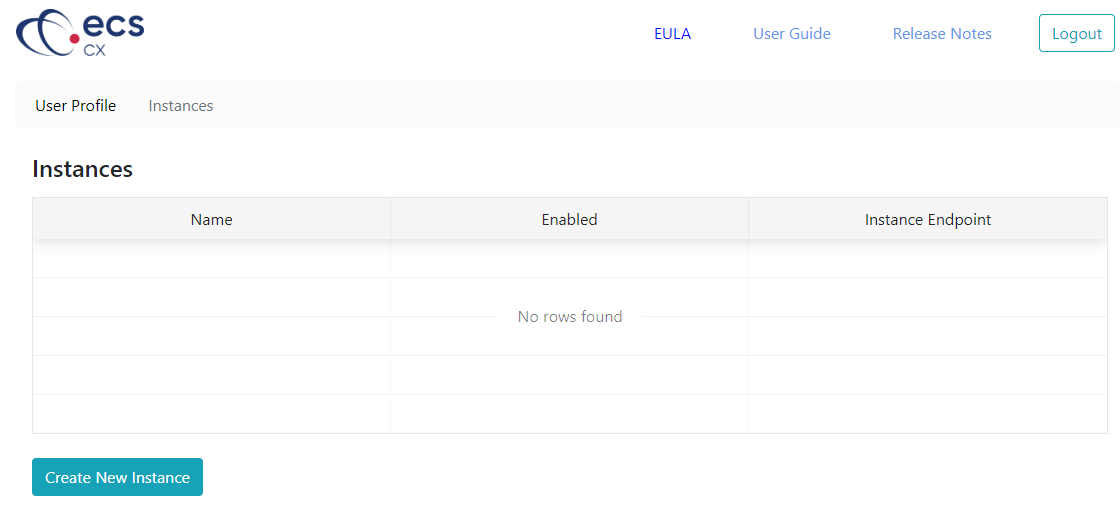


1. After successful signup, you are required to Login with the account created in previous step.



### RocketCX Instance Configuration

1. Successful login on RocketCX Portal opens home page as shown below. Click **Instance Tab** and create a new instance configuration using “**Create New Instance**” button.



1. Provide following details on Instance page to create a new instance.   
     
   **Name:** Unique name

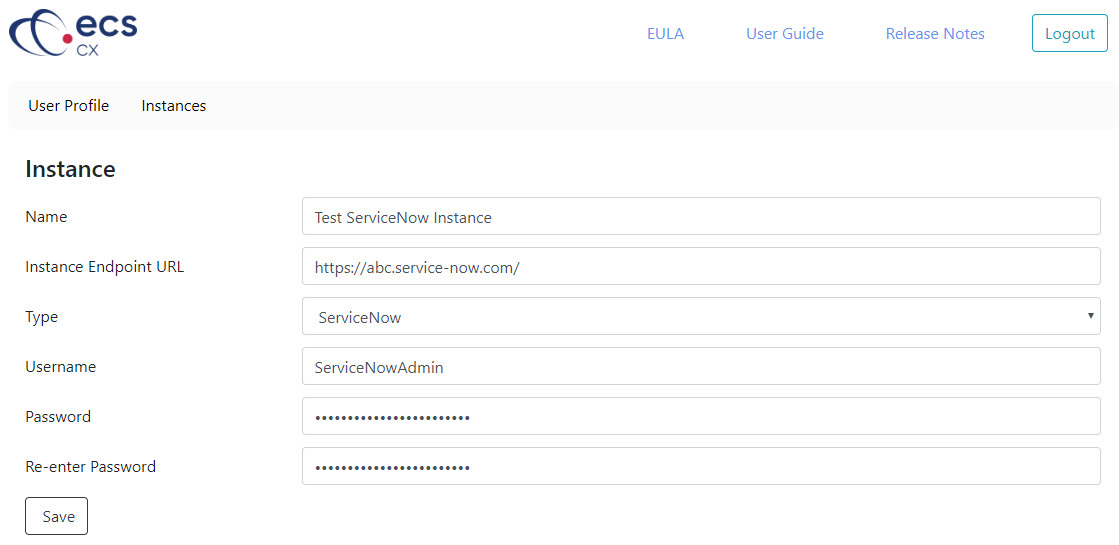
**Instance Endpoint URL:** ServiceNow instance URL

**Type:** ServiceNow

**Username:** Valid User Name

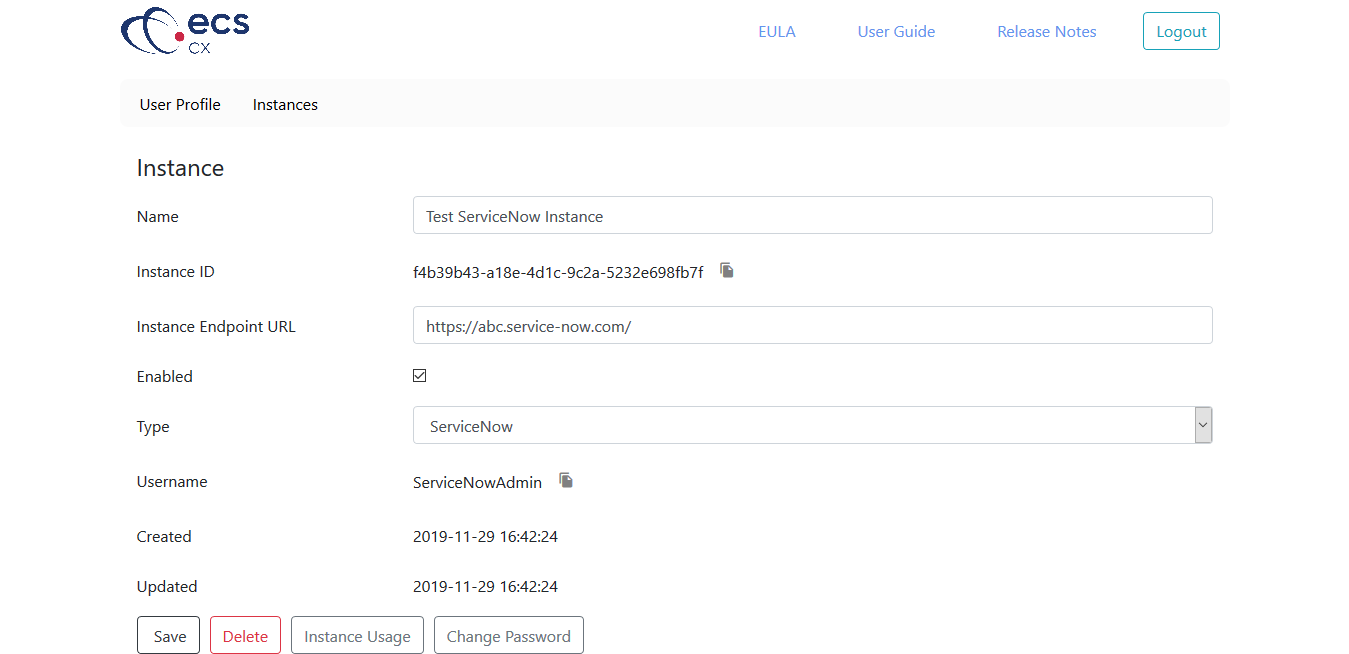
**Password:** Valid Password

**Note:** RocketCX uses Username and Password configured in this step as a service account to authenticate and establish connection between ServiceNow and RocketCX portal.



1. After successful instance configuration, RocketCX portal generates a unique **Instance ID** as shown in the following screenshot.
2. Check the Enabled checkbox to activate the Instance.

**Note:** RocketCX uses Instance ID generated above to update usage data on RocketCX portal.



### Capture required data from RocketCX portal

RocketCX requires following list of properties for maintaining communication with RocketCX portal:

* RocketCX User and Password created in step 2 from [previous section](#rockercx_portal_instance_configuration)
* Instance ID generated in step 3 from [previous section](#rockercx_portal_instance_configuration)

## AWS Configuration

All the steps related to AWS in this section requires login using [AWS Admin](#AWS_Admin) account.

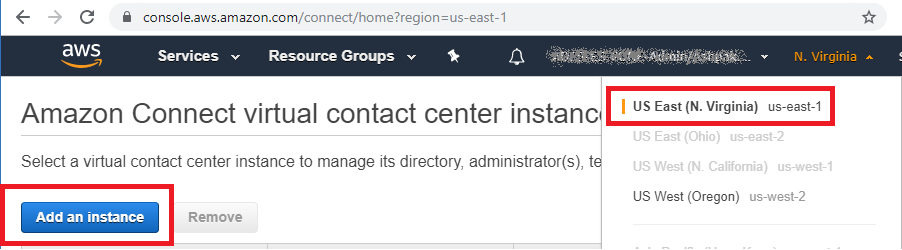
### Amazon Connect

#### Setup Amazon Connect Instance

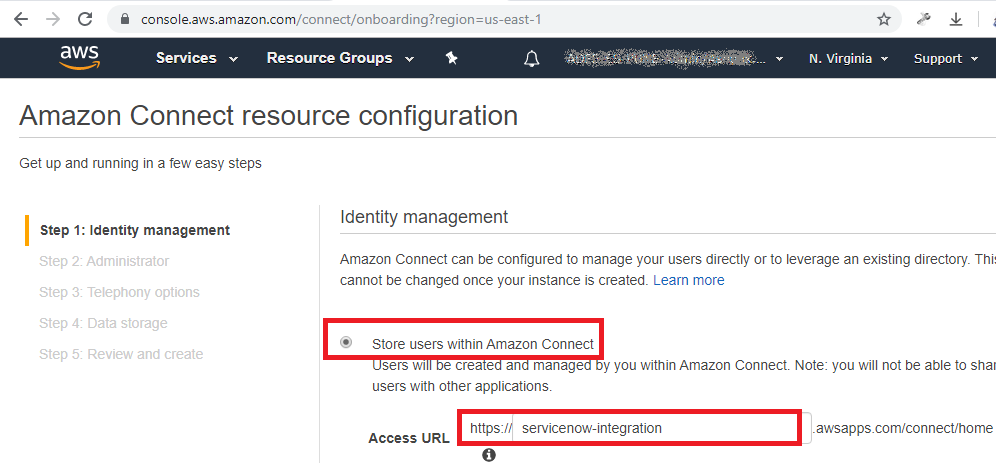
The first step in setting up your Amazon Connect contact centre is to create a virtual contact centre instance. Each instance contains all the resources and settings related to your contact centre.

1. Sign in to AWS console <https://console.aws.amazon.com/connect/>
2. Open the Amazon Connect console at <https://console.aws.amazon.com/connect/>
3. Choose **Get started.** If you have previously created an instance, choose **Add an instance** instead. Make sure to select **US East (N. Virginia)** region for this example.

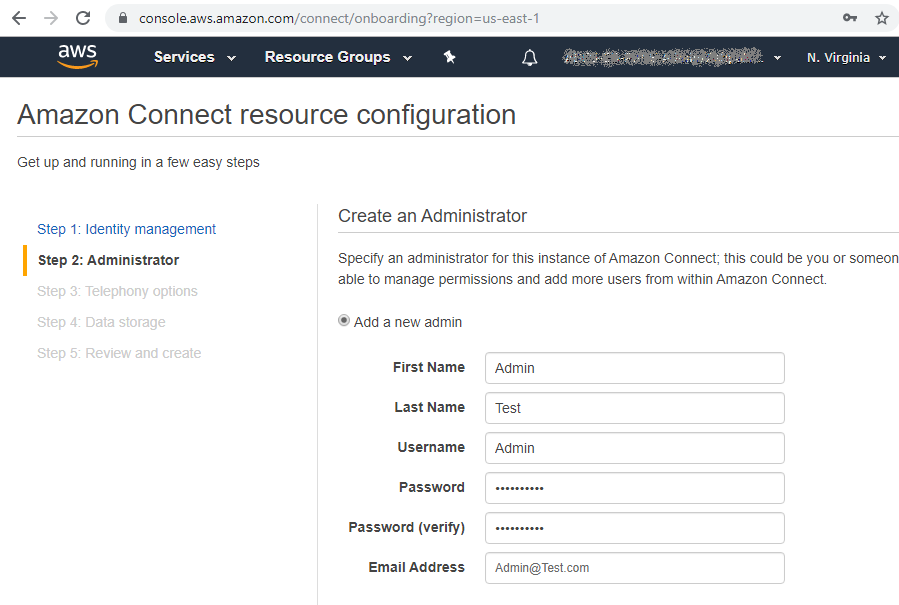
This document hereinafter assumes US East as a default region for all the related AWS services.



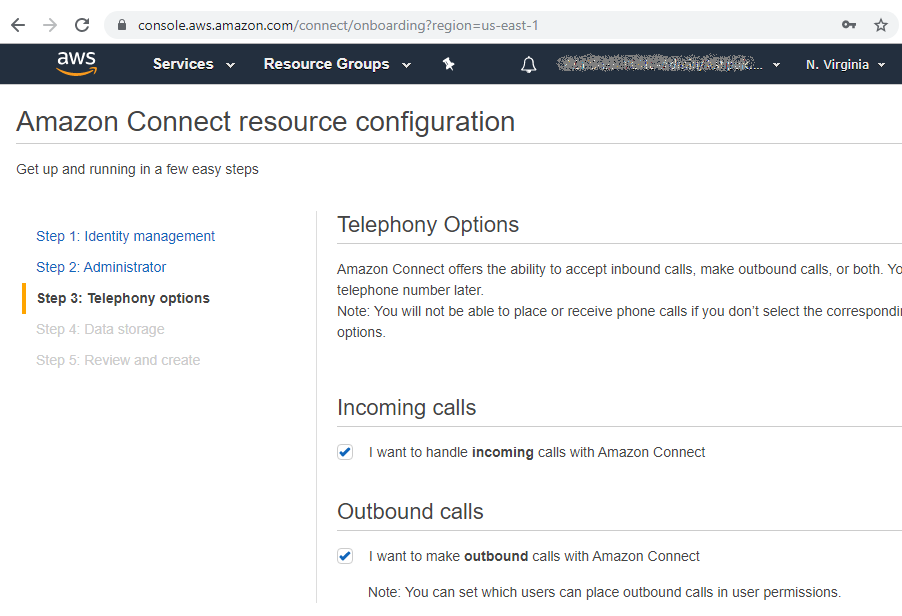
1. Choose **Store users within Amazon Connect** and provide a unique Amazon connect instance name. Click **Next** on this page.



1. After you specify the user name of the administrator for the Amazon Connect instance, a user account is created in Amazon Connect and the user is assigned the **Admin** security profile. Click **Next** after filling details for following fields.



1. Select both the telephony options and click **Next.**

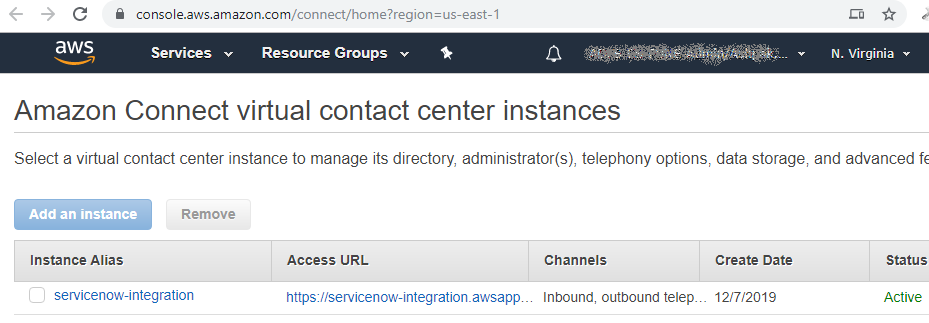


1. Click **Next** on **Data Storage** step without changing any settings
2. Click **Create Instance** on **Review and Create** page.
3. This will create an Amazon Connect instance which might take few minutes.

#### Configure Call Recordings to S3 bucket

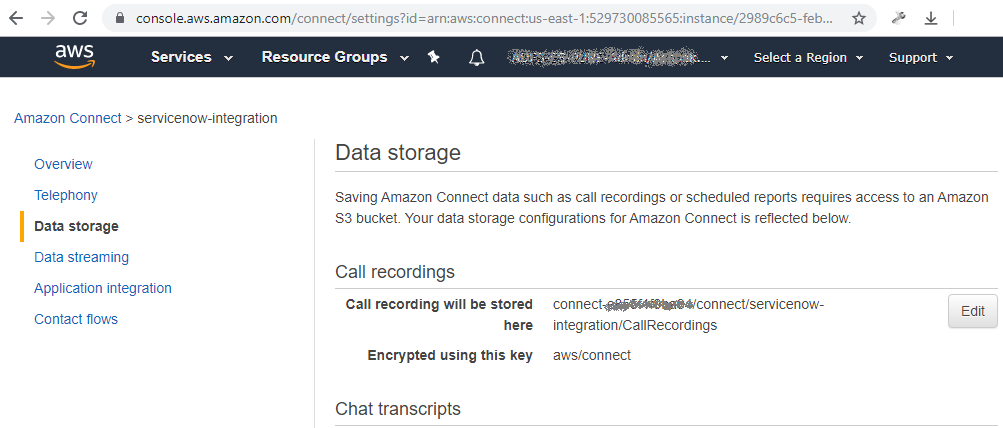
It is recommended to configure call recordings on s3 storage.

1. After successful instance creation, go to Amazon Connect Console and select the instance to configure additional resources. <https://console.aws.amazon.com/connect/home?region=us-east-1>



1. Navigate to **Data Storage** and confirm Call recording is configured to be stored on S3 bucket. Call recording storage is enabled by default for new Instance.

Call Recording settings can be changed with the help of edit button.

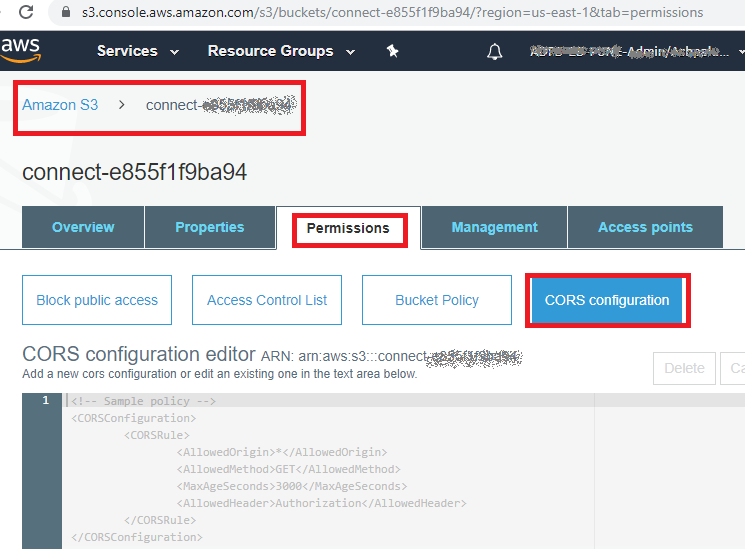


S3 bucket name field is mandatory for RocketCX configuration on ServiceNow. If call recording is not enabled, then a blank S3 bucket can be created providing its access to IAM user.

#### Configure CORS for S3 bucket

CORS Configuration on S3 allows RocketCX to read stored call recording files.

1. Log in to [AWS S3 Console](https://s3.console.aws.amazon.com/s3/home?region=us-east-1)
2. Selectthe S3 bucket used for storing Amazon Connect instance’s [call recordings.](#Data_Storage)
3. Open the Permissions tab and click CORS configuration.



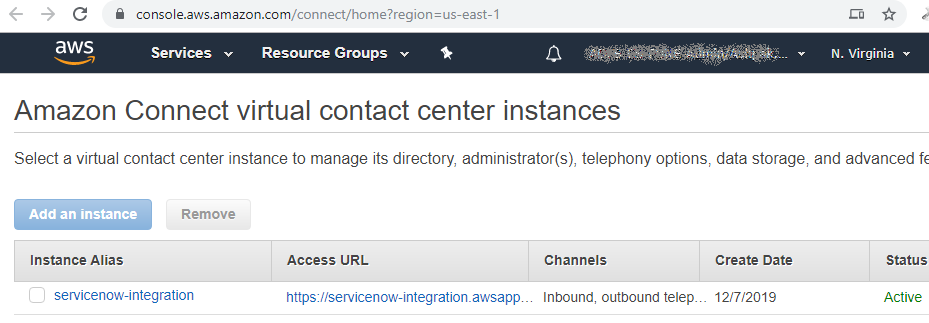
**Note:** Value of ARN from above screenshot is required in [upcoming section](#create_iam_policy). ARN value in above screenshot is “arn:aws:s3:::connect-xxxxxxxxxxx”

1. Inside the CORS Configuration editor, copy and paste the policy details from below CORSRule.txt file and click **Save**. Before saving the configuration, replace “*{servicenow domain}*” string in CORS Configuration editor with your ServiceNow instance URL e.g. https://<my-instance>.service-now.com/

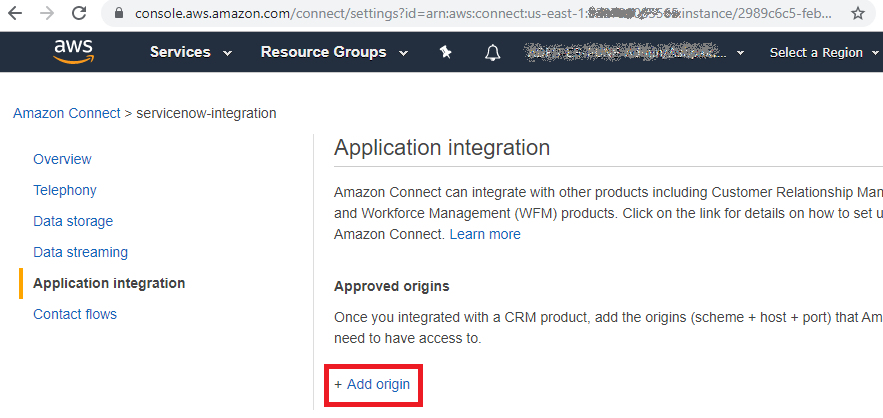


#### Whitelist ServiceNow Instance URL in Amazon Connect

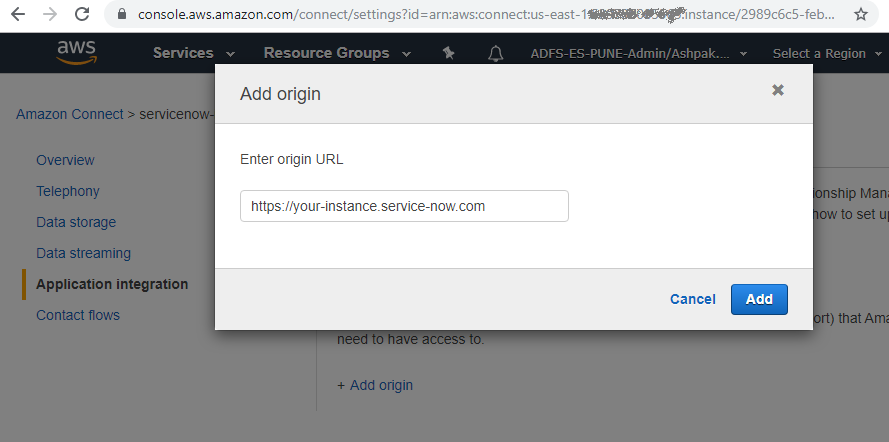
1. Log into Amazon Connect console and select Amazon Connect instance created in previous sections. <https://console.aws.amazon.com/connect/home?region=us-east-1>



1. Select **Application Integration** and click **Add Origin**



1. Add ServiceNow instance URL to allow traffic between ServiceNow Instance and Amazon Connect instance. Click **Add** after adding ServiceNow URL.



### Configure IAM users

RocketCX requires AWS IAM user created in this section for following operations.

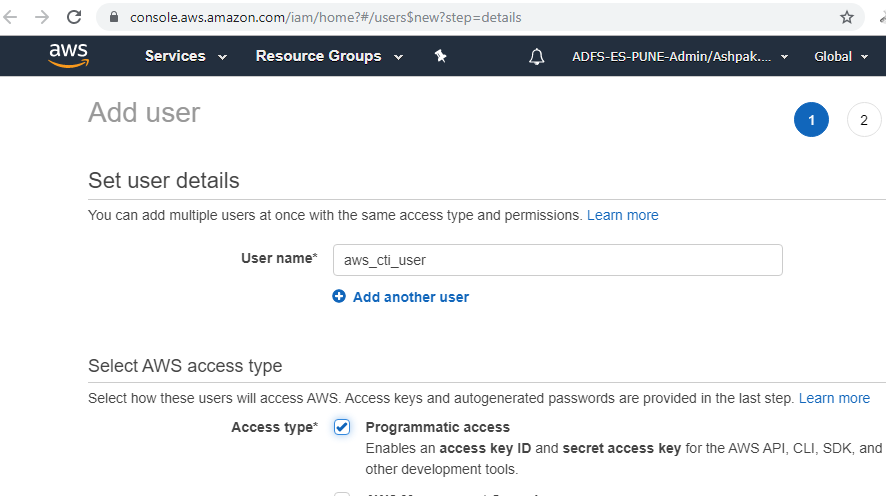
* Read Amazon Connect call recording from S3 bucket
* Get Amazon connect metric data for Reporting

#### Create IAM user

1. Log into AWS console with following link to start creating a new IAM user

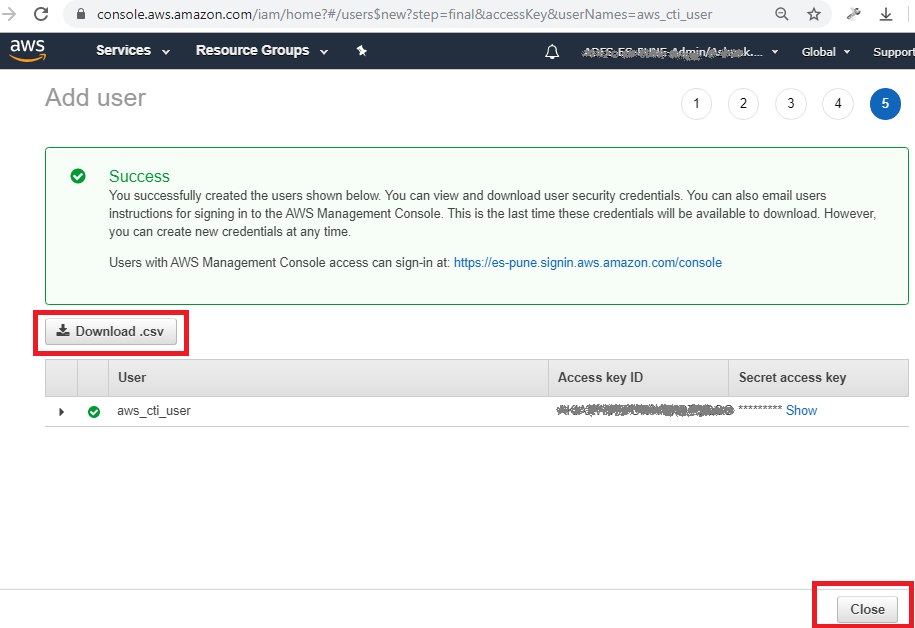
<https://console.aws.amazon.com/iam/home?#/users$new>

1. Add the details as per following screenshot and click **Next: Permissions**



1. Click **Next: Tags** on **Set Permissions** window without changing any settings.
2. Click **Next: Review** on **Add Tags** window without adding any details.
3. Click **Create User** on **Review** page by ignoring warning “This user has no permissions”.
4. Click **Download .csv** and **close** as per following screenshot.

RocketCX requires IAM user details from downloaded csv file in later sections.



#### Create IAM Policy

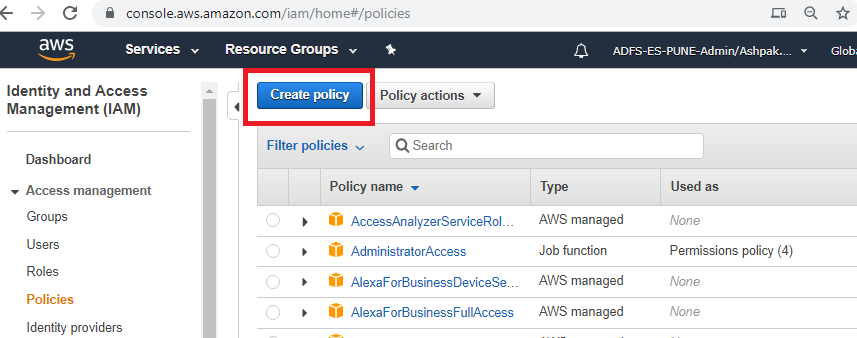
IAM user created in previous step requires relevant permissions provided by IAM policies to read call recordings from S3 bucket and get Amazon connect metrics data.

It is essential to perform steps 1 to 6 below, for every policy mentioned in **Permission policy details table** ([listed below](#permission_policy_details))

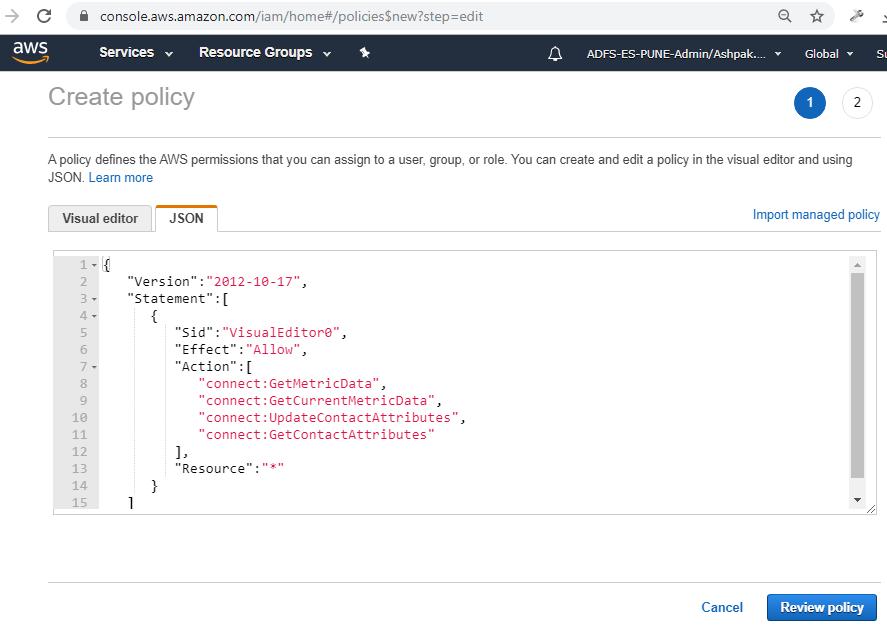
1. Open IAM policy console

<https://console.aws.amazon.com/iam/home#/policies>

1. Click **Create Policy**



1. Navigate to JSON Tab under Create policy page.
2. Open JSON file embedded in **Permission Policy Details** tables ([listed below](#permission_policy_details)) and copy-paste the contents under JSON tab. Click **Review Policy**.



1. On ‘Review Policy’ page, please enter the policy name (corresponding to the JSON file opened in Step 4)
2. Click **Create Policy** button.

**Permission Policy Details:**

|  |  |
| --- | --- |
| **Policy Name** | cti\_connect\_api |
| **Policy JSON** |  |
| **Description** | Required for fetching metric data from Amazon Connect for Reporting and Analytics  Please ignore unrecognized Amazon connect action warning during import. |

|  |  |
| --- | --- |
| **Policy Name** | cti\_s3\_api |
| **Policy JSON** |  |
| **Replace** | Replace the <[bucket\_arn](#s3_arn)> string in the above JSON with ARN of the S3 bucket that stores the call recordings from Connect.  Refer AWS documentation for more information: <https://docs.aws.amazon.com/AmazonS3/latest/dev/s3-arn-format.html> |
| **Description** | Required for Listening to Call Recordings from Amazon S3 |

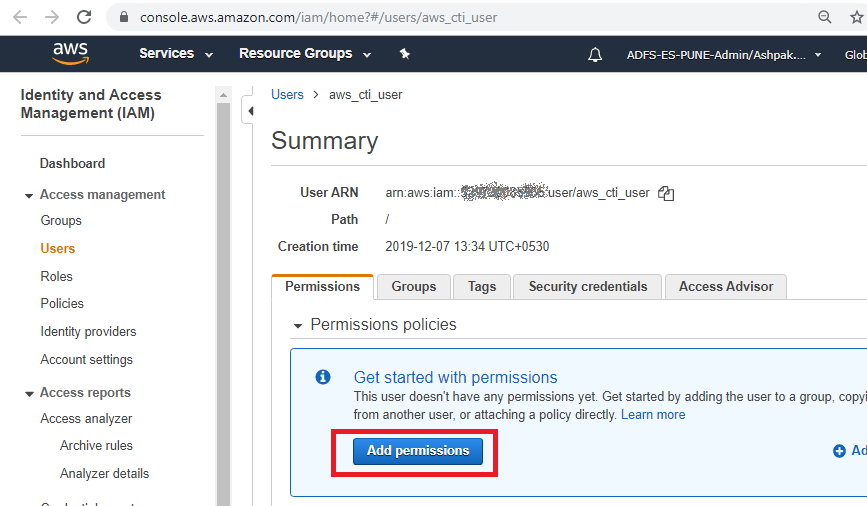
|  |  |
| --- | --- |
| **Policy Name** | cti\_sts\_api |
| **Policy JSON** |  |
| **Description** | To create temporary credentials that would be used to access data from S3 and Amazon Connect |

#### Attach IAM policy to IAM user

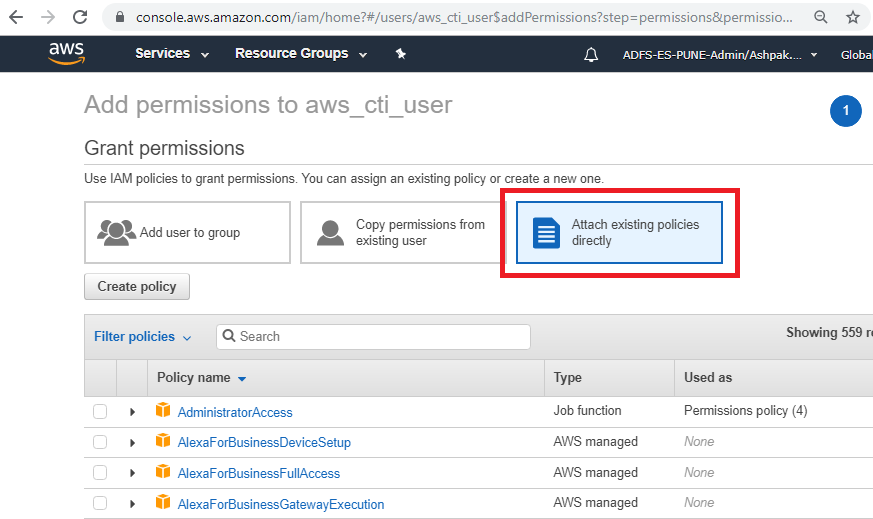
1. Open ‘aws\_cti\_user’ IAM user (created in [previous section](#create_iam_user)) through the following link

<https://console.aws.amazon.com/iam/home?#/users/aws_cti_user>

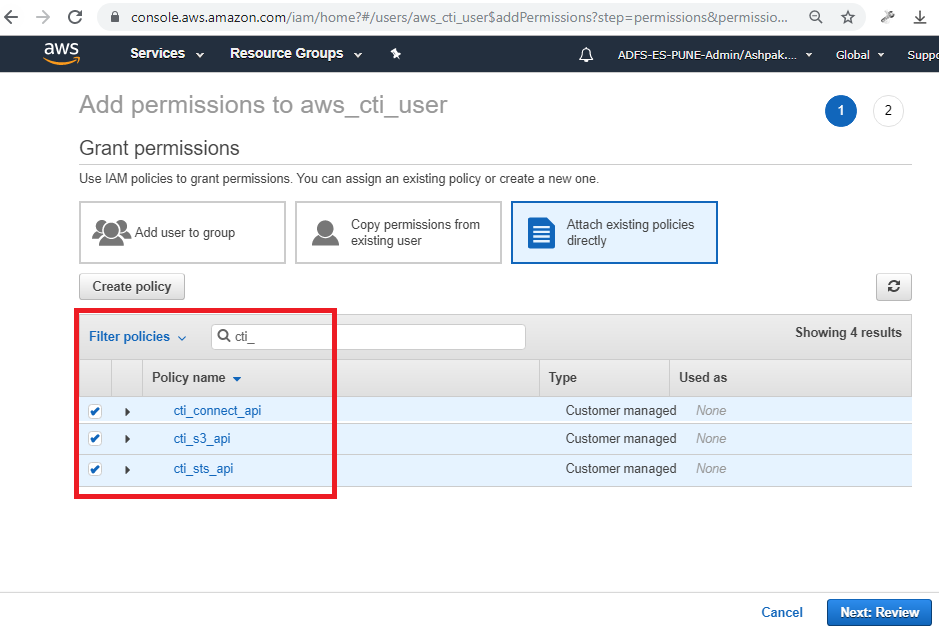
1. Under permissions tab click **Add Permissions**.



1. On ‘Grant Permissions’ page, select **Attach existing policies directly**.



1. Search and select every individual policy created in the [previous section](#create_iam_policy) to attach permissions to aws\_cti\_user. Click **Next:Review**



1. Click **Add Permission** on permission summary page to attach selected policies to IAM user.

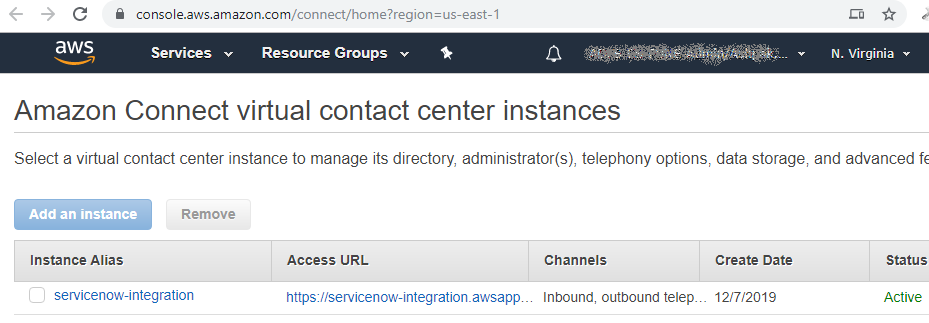
### Capture AWS data required for RocketCX

RocketCX requires following list of properties for maintaining communication with AWS services. ([RocketCX configuration](#rocketcx_admin_console) has explained in further sections.)

1. **Amazon Connect CCP URL**

Open the Amazon Connect console at [https://console.aws.amazon.com/connect/ home?region=us-east-1](https://console.aws.amazon.com/connect/home?region=us-east-1)

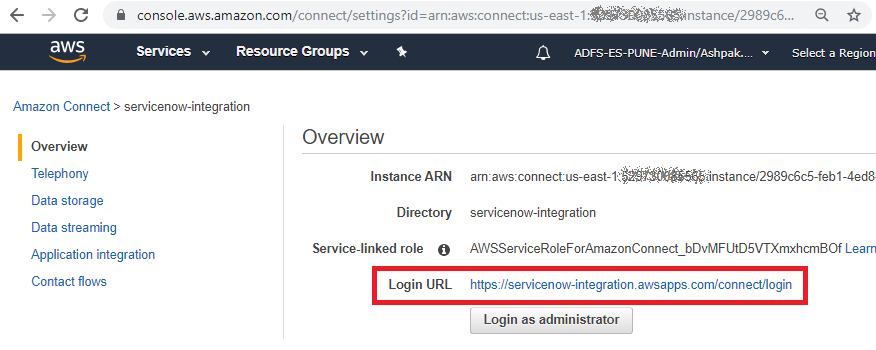
Select Amazon Connect instance



Copy **Login URL** and replace “login” with “ccp#/” at the end of the string.

Note down this updated URL.

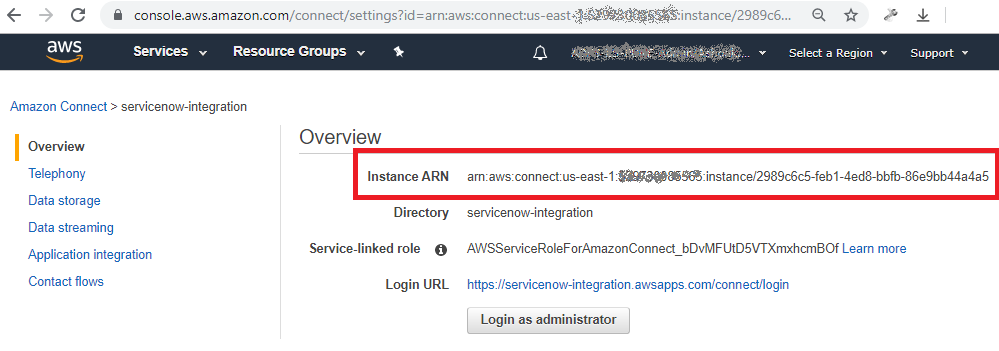
In this example, CCP URL is <https://servicenow-integration.awsapps.com/connect/ccp#/>



1. **Amazon Connect ARN & Instance ID**

Your instance ID is the 36-character alphanumeric string at the end of the instance ARN.

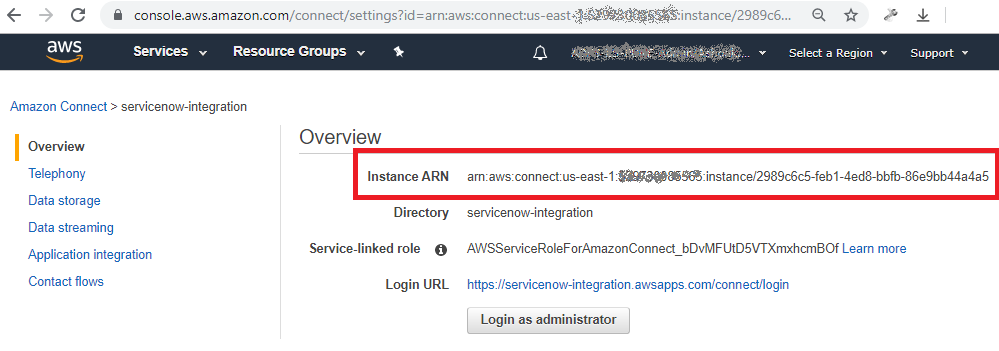
In this example instance ID is 2989c6c5-feb1-4ed8-bbfb-86e9bb44a4a5

****

1. **Amazon Connect Instance region**

Instance region is available in Instance ARN string at the beginning, after ‘arn:aws:connect:’

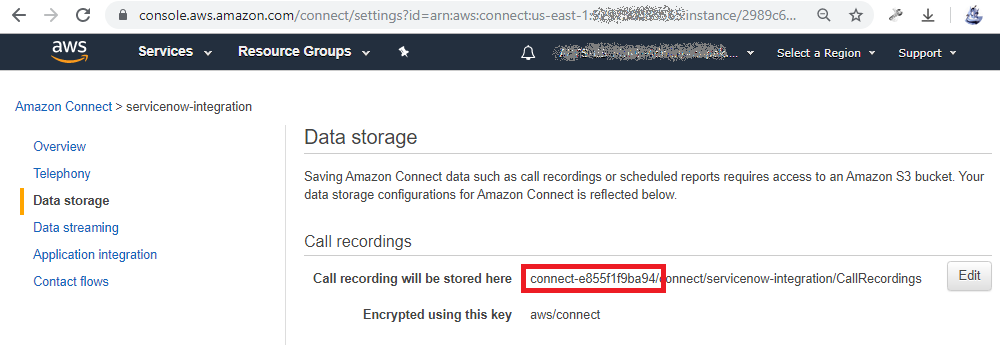
In this example instance region is us-east-1



1. **S3 bucket name and region**

Amazon connect store call recordings on s3 bucket in same region, specified in Data storage section.

S3 bucket name field is mandatory for RocketCX configuration on ServiceNow. If call recording is not enabled, then a blank S3 bucket can be created providing its access to IAM user.



In this example, s3 bucket name is connect-e855f19ba94 and region is same as Amazon connect instance region us-east-1

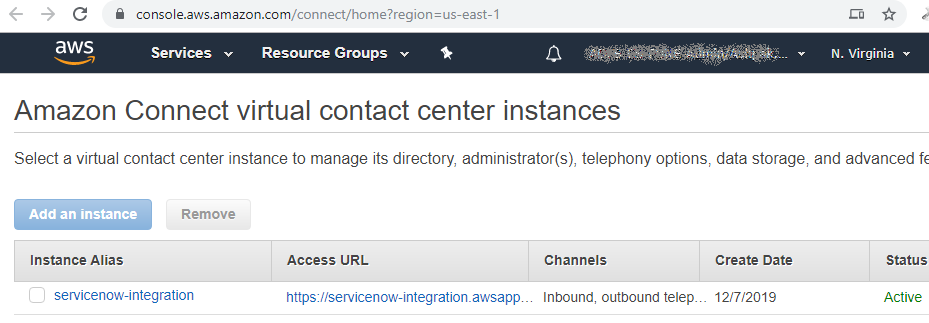
1. **IAM user access key and secret access key**

This information is present in [downloaded csv](#csv_iam) file during IAM user creation.

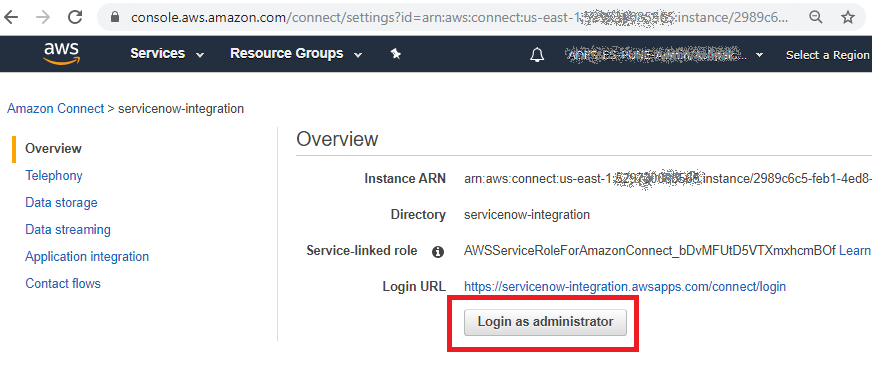
1. **Queue ID, Name and ARN**

Open the Amazon Connect console at <https://console.aws.amazon.com/connect/>

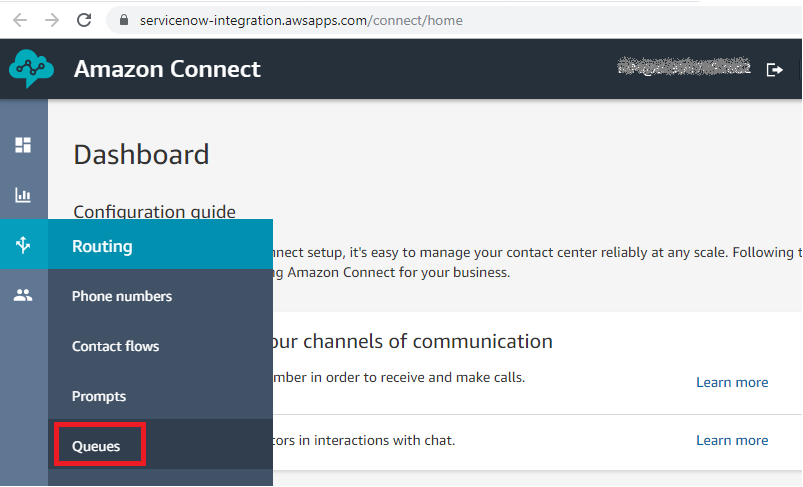
Select Amazon Connect instance



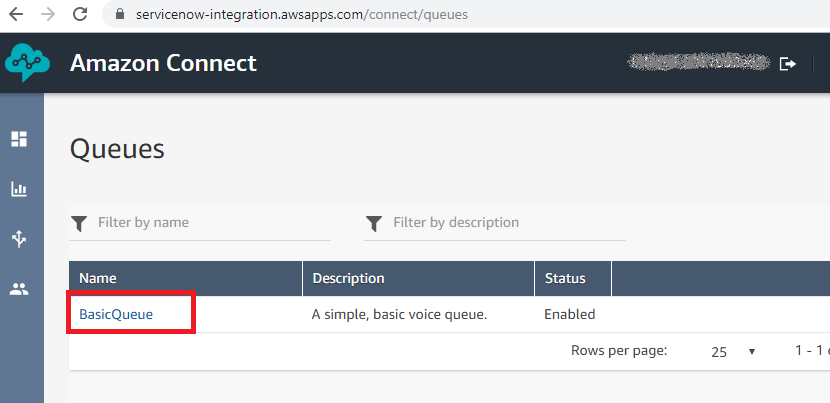
Click **Login as administrator**



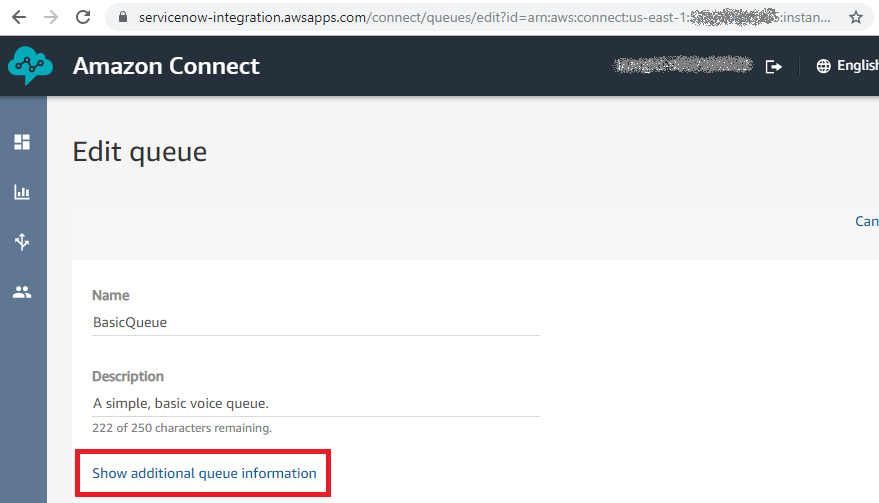
Select **Queues** as mentioned in following screenshot.



Select the queue you want to configure in RocketCX. In this example, we are using default Basic queue.



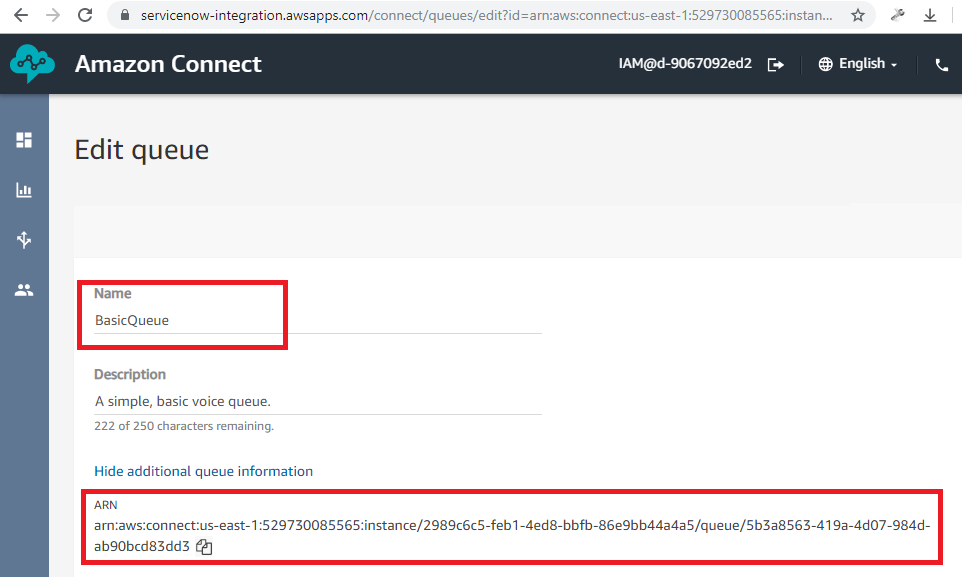
Click **Show additional queue information**



Note down the Queue name, Queue ARN and Queue ID

Queue ID is the 36-character alphanumeric string at the end of the Queue ARN.

In this example Queue ID is 5b3a8563-419a-4d07-984d-ab90bcd83dd3



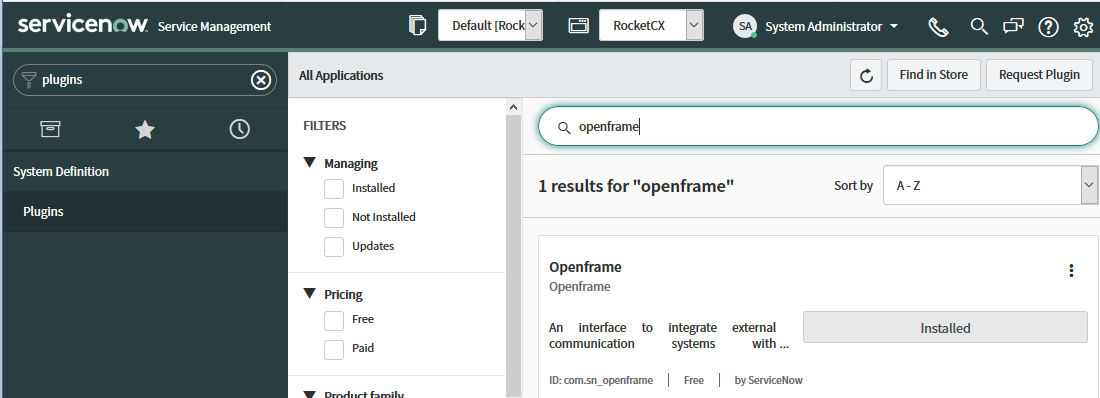
## ServiceNow Configuration

### Install Openframe plugin

Log into ServiceNow instance using **Administrator** account.

Search **Plugins** on left navigation pane in ServiceNow and search for **Openframe** under plugins page.

Click **Install** Openframe plugin from search results (if Openframe is not already installed)



### Request & Install RocketCX Enterprise Connector for ServiceNow

#### Request RocketCX application from ServiceNow store

**HI Service Portal account** is required to download and install application from ServiceNow.

For more information on Hi account, please visit [Hi account KB article](https://hi.service-now.com/kb_view.do?sysparm_article=KB0547103).

1. Click [here](https://store.servicenow.com) to open ServiceNow store.
2. Log in with your ServiceNow Hi account credentials.
3. Search for “RocketCX Enterprise Connector for ServiceNow” in the search box and open the application page.
4. Click **“Get App”** button on the ServiceNow store page.

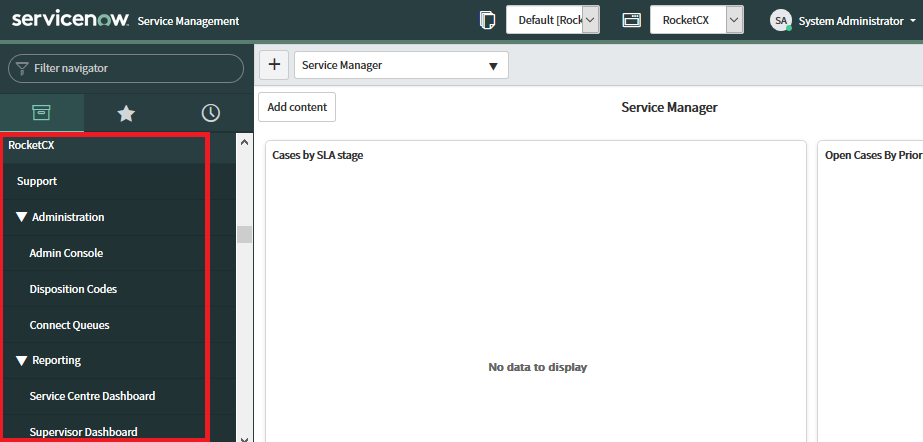
#### Install RocketCX on ServiceNow instance

Once approved, install the application to your ServiceNow instance.

1. Login to ServiceNow instance as **Administrator** and navigate go to

**System Applications > All Available Applications** > **All**

1. Find RocketCX under **Downloads** tab and click **Install**.
2. RocketCX application modules should be available in navigation pane after successful installation.



**Note**: After successful installation of RocketCX, please refresh the page to see RocketCX application modules.

### RocketCX Roles and Groups

After installing [RocketCX](#Install_RocketCX) and [Openframe plug-in](#Install_OpenFrame) on ServiceNow, following roles are automatically created.

|  |  |  |
| --- | --- | --- |
| Role Label | Role Name | Description |
| Open Frame User | sn\_openframe\_user | This role is not part of the RocketCX application but part of the Openframe module, which allows Administrator to provide CTI operations within a ServiceNow instance. |
| Plug-In Administrator | x\_ecsd\_amazon\_conn.Plug-In Administrator | This role must be given to all Agents in order to access CCP along with Agent role. |
| Contact Centre Supervisor | x\_ecsd\_amazon\_conn.Contact Centre Supervisor | Users with this role will have access to all functions on the Admin Console. This is accessed through the ServiceNow back end. |
| Contact Centre Agent | x\_ecsd\_amazon\_conn.Contact Centre Agent  (along with the sn\_openframe\_user role) | Users with this role will have access to the queue and agent reporting functions (for all agents), and the Service Centre Dashboard. |

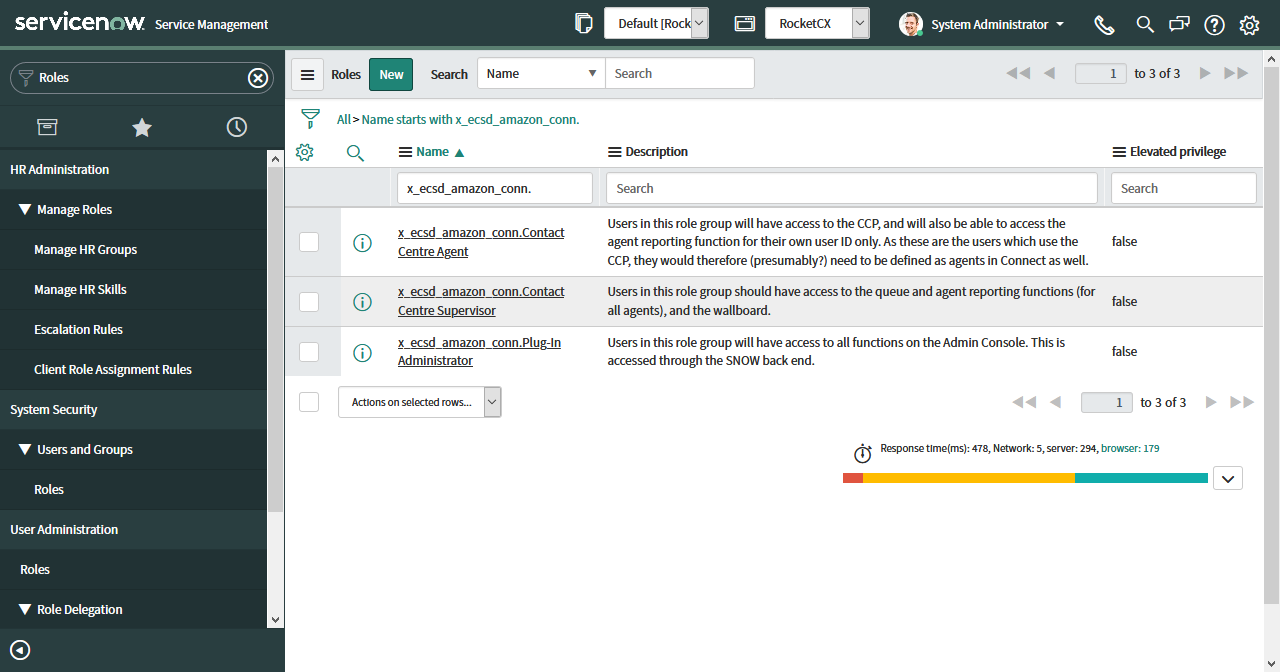
To create new Users and Groups, Log into ServiceNow instance using **Administrator** account.

#### Create Groups

Provided below is a guidance on the groups that could be created and roles to be assigned.

|  |  |
| --- | --- |
| Group Name | Role |
| CTIAgent | * sn\_openframe\_user * x\_ecsd\_amazon\_conn.Contact Centre Agent |
| CTISupervisor | * x\_ecsd\_amazon\_conn.Contact Centre Supervisor |
| CTIPlugin Admin | * x\_ecsd\_amazon\_conn.Plug-in Administrator |

Provided below is a screenshot with details of the roles in RockectCX



**Steps to Create group:**

1. Navigate to User **Administration -> Groups** and create a new record

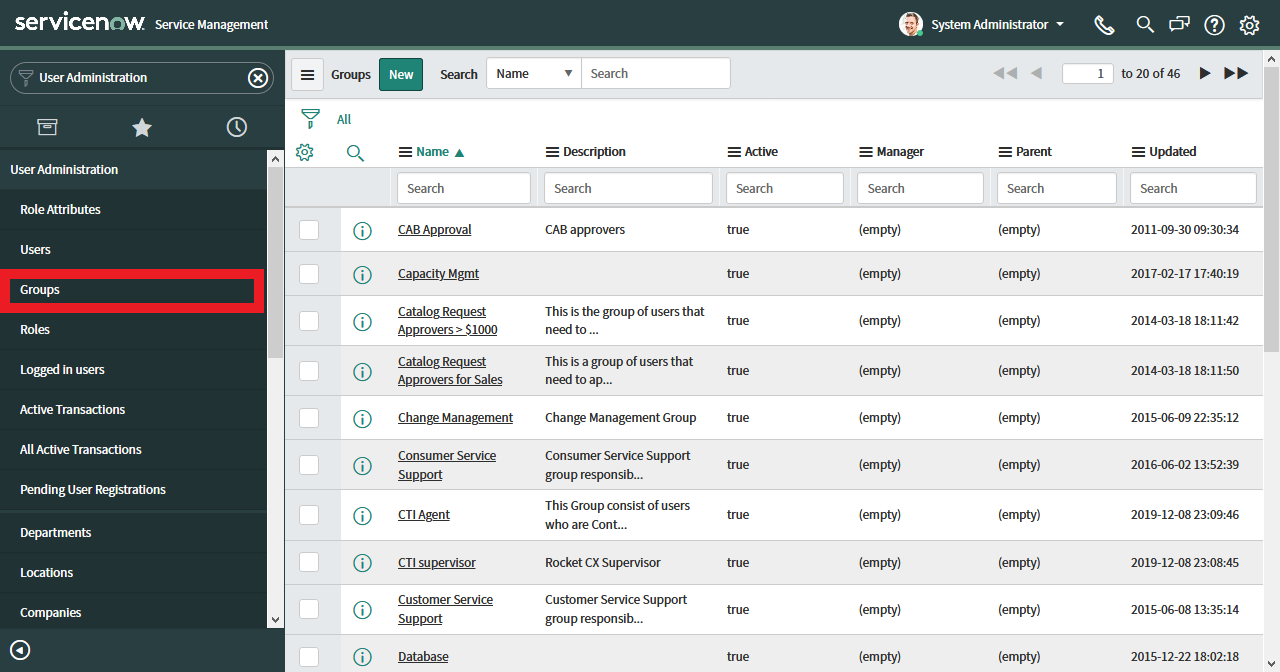
**Steps to Assign role to group:**

1. Navigate to User **Administration -> Groups**.
2. Click the group to assign a role.
3. In the Roles related list, click **Edit**.
4. Use the slushbucket to add the desired roles to the group.
5. Click **Save**.

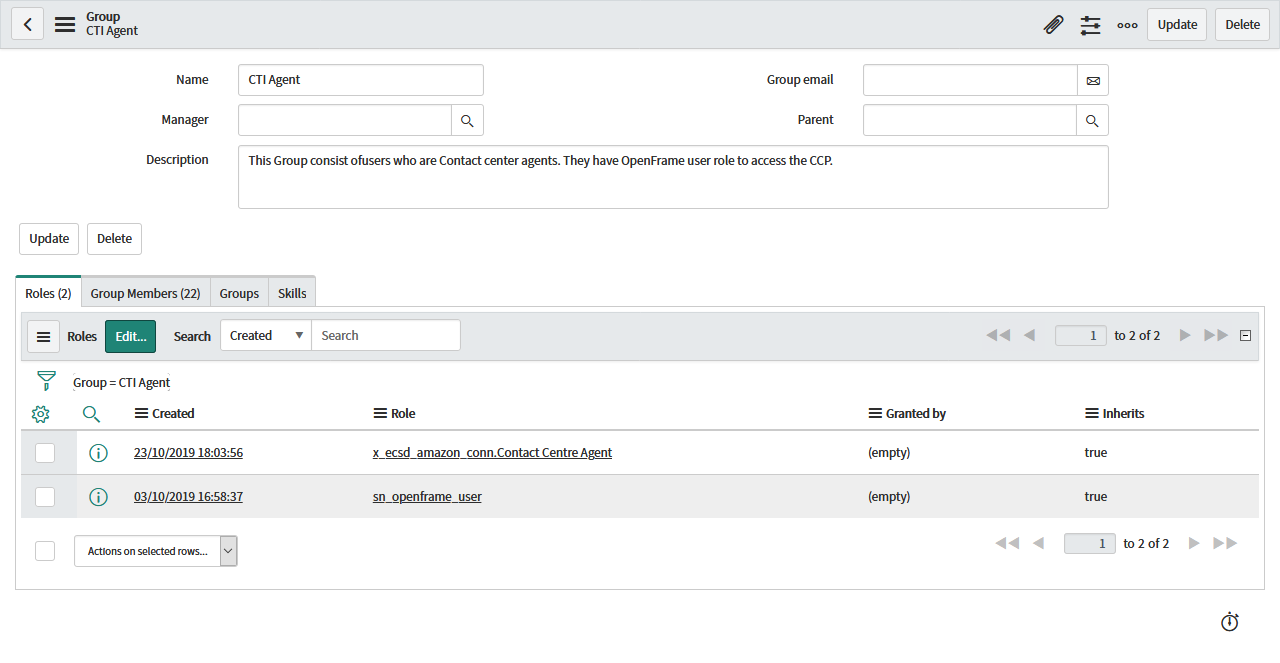
**Steps to Add users to group:**

1. Navigate to User **Administration > Groups**.
2. Click a group Name.
3. In the Group Members related list, click **Edit**.
4. Select one or more names in the Collection list.
5. Click **Add**
6. Click **Save**

Provided below is a screenshot of sample Groups



Provided below is a screenshot of a sample Group with assigned roles.



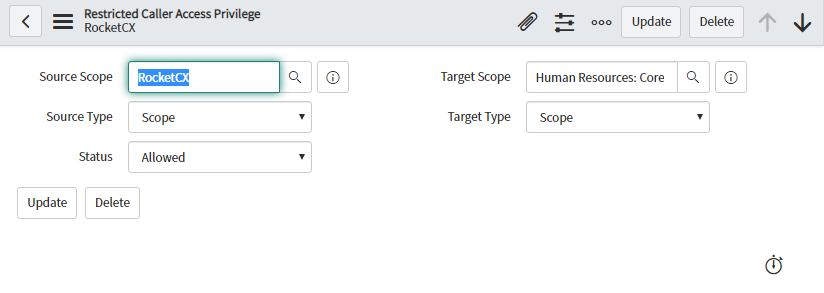
**Note:** The ServiceNow system administrator manages Role group memberships; this is not under purview of RocketCX. A user can be a member of more than one role group. Users with Contact Centre Agent role must be assigned roles or membership of ServiceNow groups which provide access to the ServiceNow applications modules supported by RocketCX which includes Incident, Customer Service and HR Core. For e.g. If you have CSM or HR modules enabled on ServiceNow instance, assign *sn\_customerservice.consumer\_agent* and *sn\_hr\_core.admin* roles to both the groups *CTI Agent* and *CTI Supervisor.*

### Restricted Caller Access Privileges

RocketCX also supports “Human Resources: Core” module. In case if it is active or is activated in the future and you want to use the module, you need to create a new Restricted Caller access privilege record using following steps.

Log into ServiceNow instance using **Administrator** account.

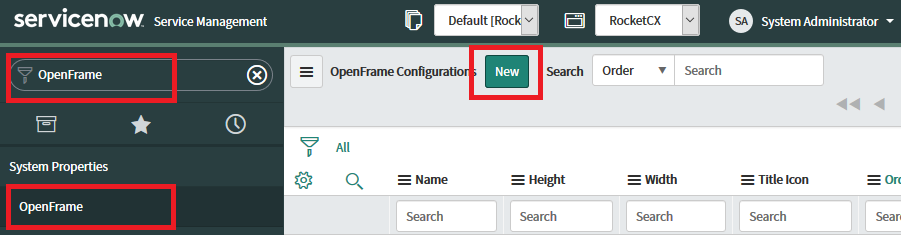
1. Search and select **Application Restricted Caller Access** module and click **New** to create **Restricted Caller Access** **Privilege (sys\_restricted\_caller\_access)** configuration.
2. Enter the following details and Click **Submit**
   1. Source Scope**:** **RocketCX**
   2. Target Scope: **Human Resources: Core**
   3. Source Type: **Scope**
   4. Target Type: **Scope**
   5. Status: **Allowed**



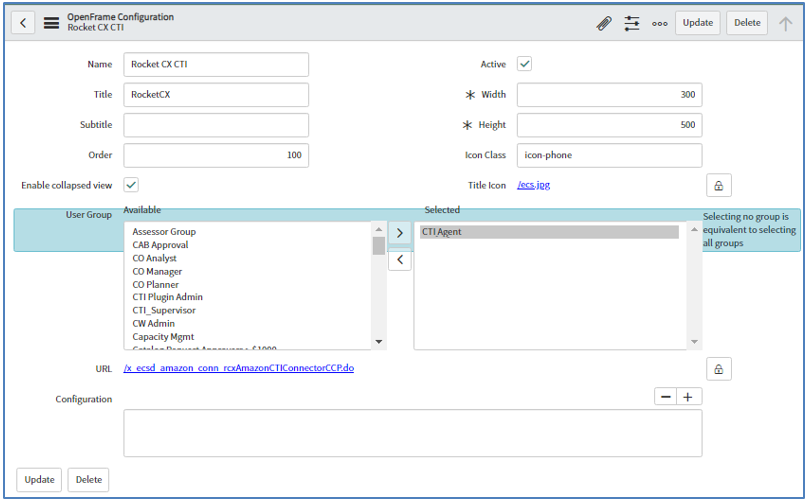
### Configure Openframe plugin

Log into ServiceNow instance using **Administrator** account.

1. Search and select **Openframe** module and click **New** to create Openframe configuration



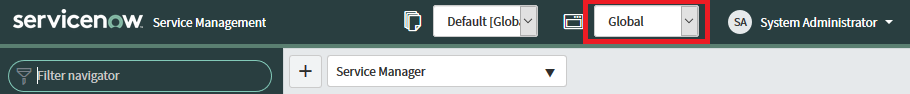
1. Enter following details in Openframe configuration and click **Submit**.
2. Name: “RocketCX”
3. Title: “RocketCX”
4. Active: true (checked)
5. URL: /x\_ecsd\_amazon\_conn\_rcxAmazonCTIConnectorCCP.do
6. Width: 300
7. Height: 500
8. Icon Class: icon-phone
9. Title Icon: /ecs.jpg
10. Enable collapsed view: checked if you want to enable minimize CCP feature.
11. Order: less than other Openframe configuration records if any.
12. Add CTI Agent User group created in last section.



**Note:** You can add respective ServiceNow groups as per requirement to allow access to RocketCX

### Configuration for Click to Dial

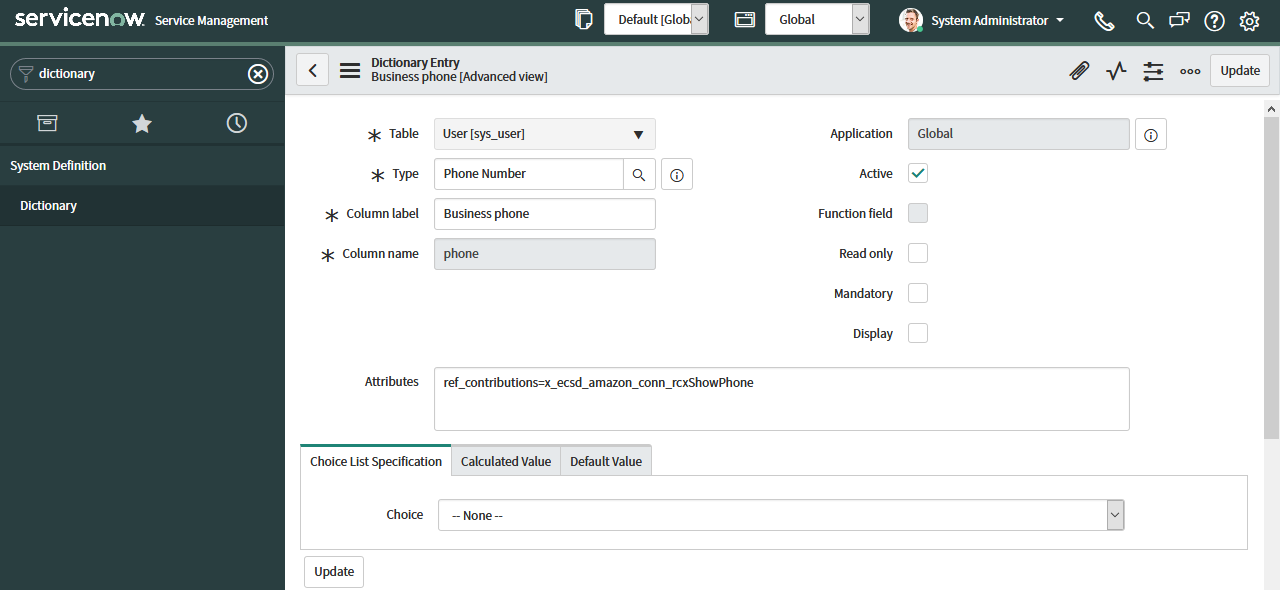
Log into ServiceNow instance using **Administrator** account and change application scope to Global/Customer Service/HR Core.



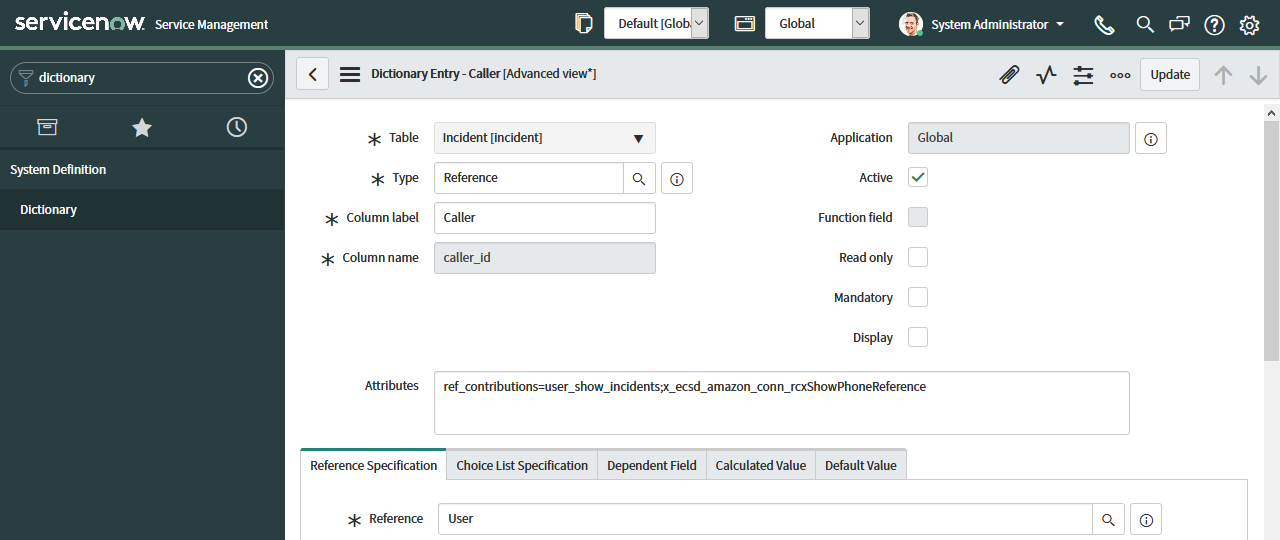
Following configurations enable relevant macros to activate click to dial functionality.

**Note**: Follow the link for more details on modifying the dictionary entry for any field <https://docs.servicenow.com/bundle/madrid-platform-administration/page/administer/data-dictionary-tables/task/t_ModifyADictionaryEntryFromAForm.html>

* Configure dictionary for phone number field and add “ref\_contributions= x\_ecsd\_amazon\_conn\_rcxShowPhone” to attributes field.



* Configure dictionary for User fields on Incident/Case form and add “ref\_contributions=x\_ecsd\_amazon\_conn\_rcxShowPhoneReference” to attributes field.



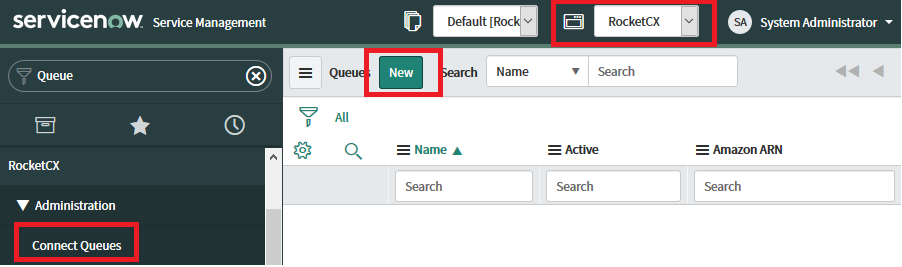
### Configure RocketCX

Log into ServiceNow instance using **Plug-In Administrator** account.

#### Amazon Connect Queue

RocketCX requires Amazon connect queue details in ServiceNow for reporting functionality.

1. Open **Connect Queues** configuration in RocketCX and click **New.**

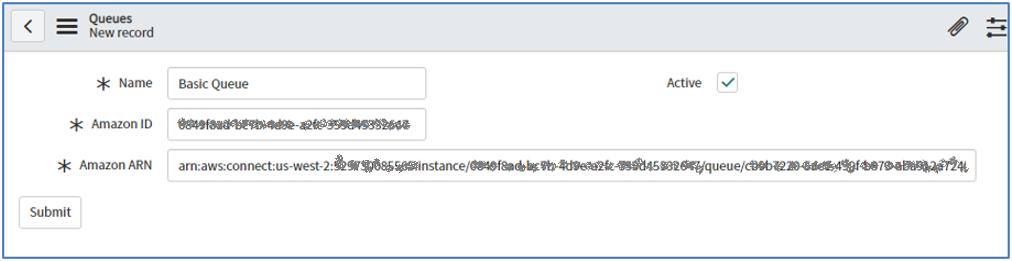


1. Add following details and click **Submit**.

Name: [Queue name](#Queue_Name_ARN) from Amazon connect instance.

Amazon ID: Amazon connect [Queue ARN](#Queue_Name_ARN).

Amazon ARN: Amazon connect [Queue ID](#Queue_Name_ARN).



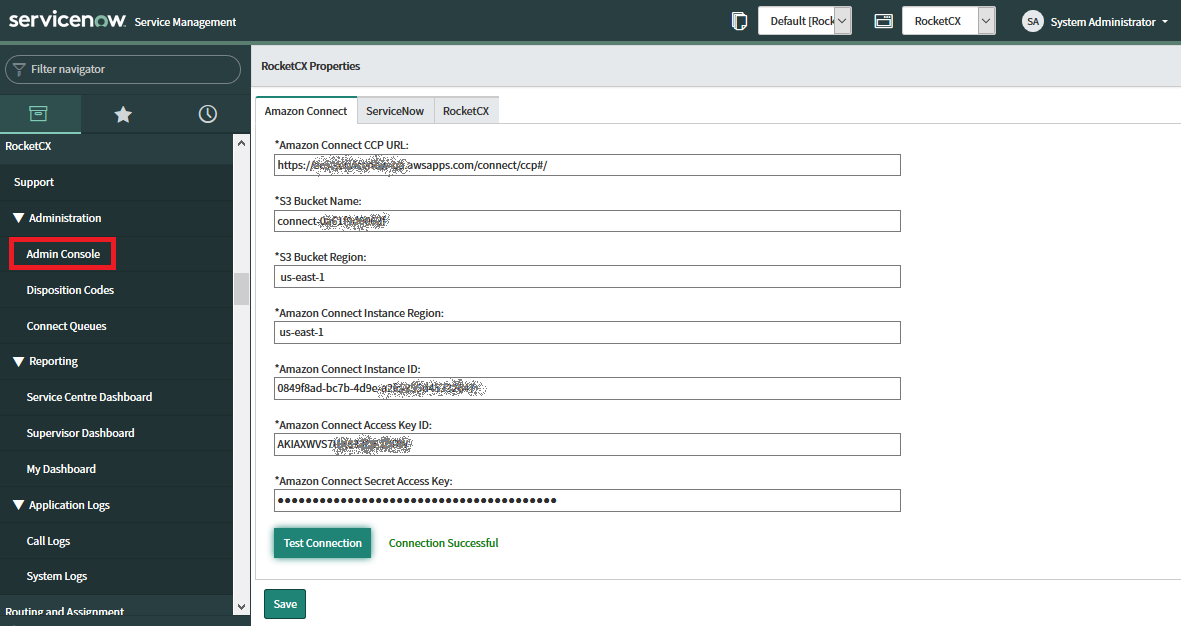
#### Admin Console

Log into ServiceNow instance using **Plug-In Administrator** account.

Select **Admin Console** from RocketCX menu.

##### Amazon Connect Configuration

Select Amazon Connect tab and configure all the fields with [information collected](#data_amazon_connect) during Amazon Connect instance setup.



Click **Test Connection** to check if provided information is correct and RocketCX is able to communicate with AWS services with a **Connection Successful** message.

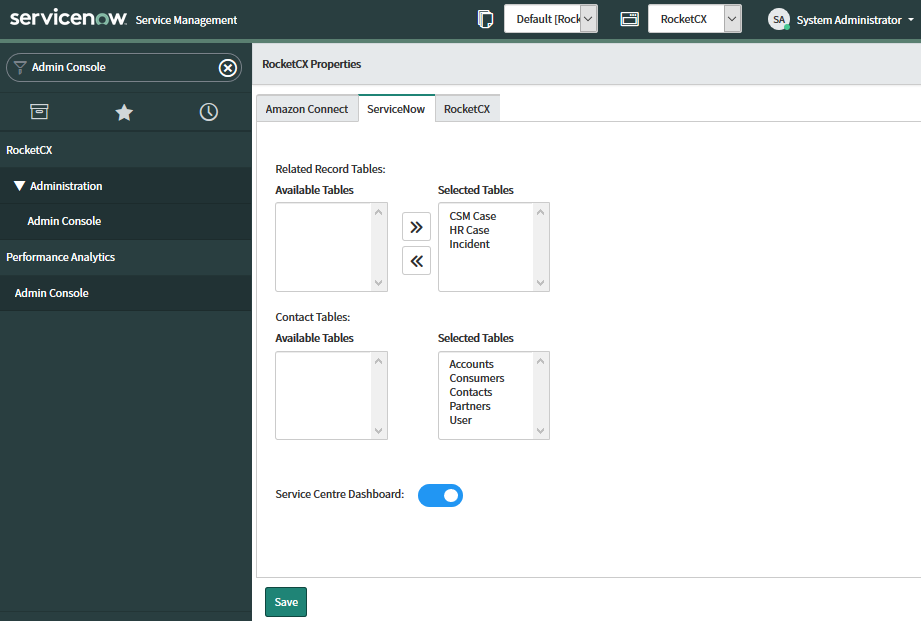
Click **Save** after connection successful message.

**Note**: In case of connection failure, please re-check the details of data collected during Amazon Connect instance creation step and try again.

##### ServiceNow Configuration

Select ServiceNow tab and configure related record and contact table properties.

1. **Related** **Record tables:** RocketCX supports Incident Management, Customer Service and Human Resource Core modules. Select individual tables to activate RocketCX for handling related records from respective modules.
2. **Contact Tables:** Contact tables become available for selection based on the modules selected in previous step. Based on this selection, corresponding contacts are displayed in RocketCX under contacts tab.

****

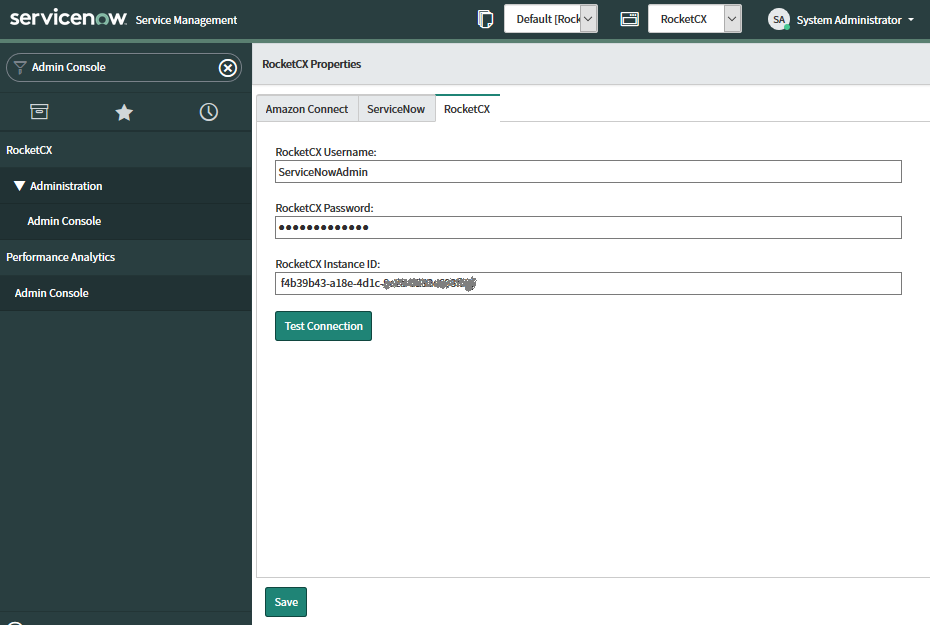
1. **Service Centre Dashboard**

Service Centre Dashboard can be enabled or disabled using toggle button under ServiceNow tab.

Click **Save** once required selection for tables and reporting is done.

##### RocketCX Portal Configuration

Select RocketCX tab and configure all the fields with the [information collected](#data_rocketcx_registration) during RocketCX registration process.



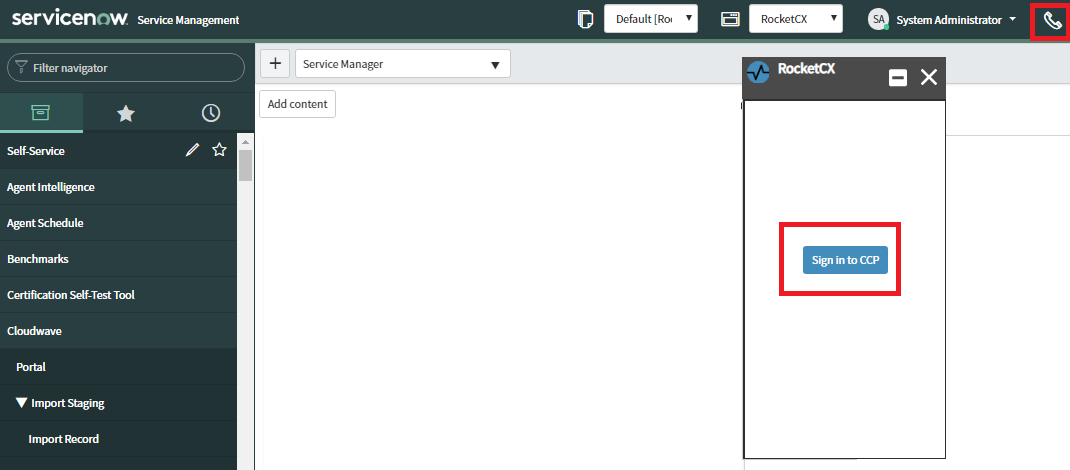
Click **Test Connection** to check if provided information is correct and RocketCX is able to communicate with RocketCX portal with a **Connection Successful** message.

Click **Save** after connection successful message.

**Note**: In case of connection failure, please re-check the details of data collected during RocketCX portal registration step and try again.

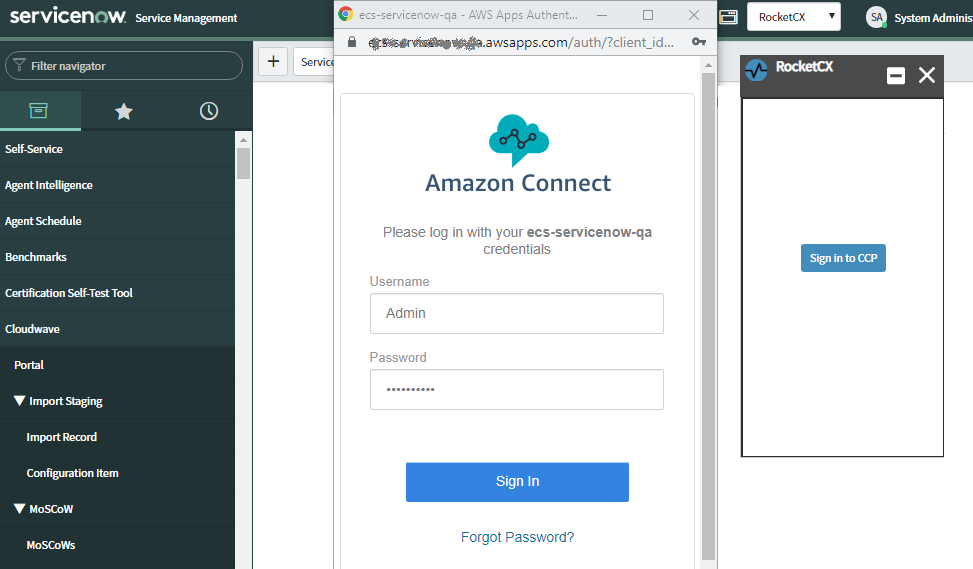
After configuring all the tabs in RocketCX admin console, please open RocketCX CCP to confirm that call configuration is correct.

To open RocketCX CCP click on the phone icon on top right corner and click **Sign In to CCP.**This will open Amazon Connect login window.

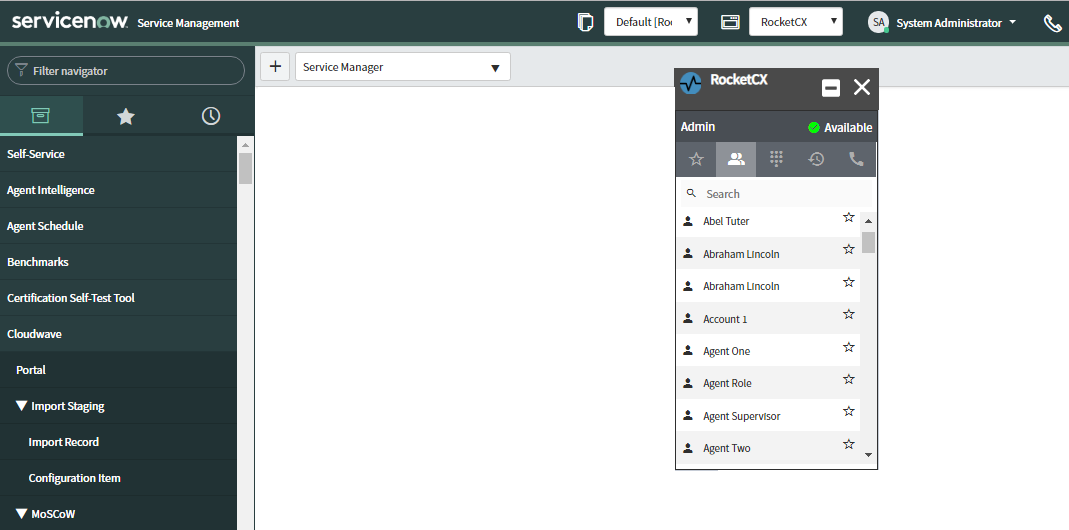


Enter [Admin user credentials](#AmazonConnect_Admin) which are created during Amazon Connect instance setup.

In this example we are using Amazon Connect admin user for sign in but it is recommended to create different users for different roles like Agent, Admin and Supervisor etc. on Amazon connect instance.



Successful login should automatically close Amazon connect login window and open RocketCX.

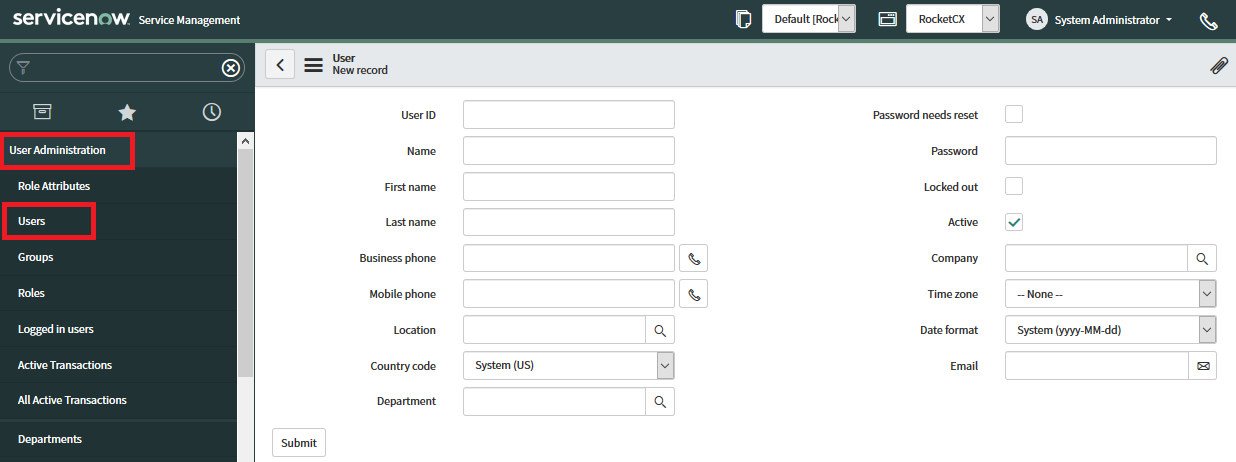


### Setup Service account for RocketCX API Access

Service account is required to use RocketCX API’s through AWS lambda functions. This Service account should have all the roles defined in RocketCX.

**Steps to create Service Account**

1. Navigate to User **Administration > Users**.
2. Click **New** and fill the details shown in the screenshot below.



1. Click **submit**
2. Add roles x\_ecsd\_amazon\_conn.Contact Centre Agent, x\_ecsd\_amazon\_conn.Contact Centre Supervisor and x\_ecsd\_amazon\_conn.Plug-in Administrator to the account.
3. Note the User ID and Password for usage in [RocketCX Quick Start Guide](https://rocketcx.s3.amazonaws.com/EnterpriseConnectorforServiceNow/v1/Documents/Public/RocketCX+Quick+Start+Guide.docx) setup on AWS.

**Note**: The ServiceNow system administrator manages Role group memberships and User accounts; this is not under purview of RocketCX. You could also use an existing user account having the above roles to execute the API’s.

**This concludes installation and configuration of RocketCX on ServiceNow.**

# Post Install & Testing

## Set up Lambda and Amazon Connect Contact flows

After installation, RocketCX could be tested by designing your own call flows and lambda functions.

For lambda function development, please refer to [Developer section](#_Developer_Section).

ECS CX provides demo call flows and lambda functions for expedited testing, please refer [RocketCX Quick Start Guide](https://rocketcx.s3.amazonaws.com/EnterpriseConnectorforServiceNow/v1/Documents/Public/RocketCX+Quick+Start+Guide.docx) to deploy them on AWS.

# External systems connection

* [AWS IAM user](#create_iam_user) to fetch Amazon Connect metrics and call recordings from AWS.
* [RocketCX Portal service account](#rockercx_portal_instance_configuration) for usage data transfer from ServiceNow to RocketCX portal.
* RocketCX supports several API endpoints and access to these API endpoints should be allowed through a service account in ServiceNow. The mentioned service account should be configured and used for API endpoint access.

# Demo Data

The demo data records are required for proper functioning of the application. It has 4 disposition code that are used in RocketCX and few report layers.

# Developer Section

RocketCX provides following web service APIs that can be used while developing AWS lambda functions.

## User details API endpoint

HTTP Method: GET

Resource Path: /api/x\_ecsd\_amazon\_conn/ connect\_cti\_api/user.

Resource URI: /api/x\_ecsd\_amazon\_conn/ connect\_cti\_api/user?contact-number={contact\_number}

Description: This endpoint accepts contact\_number as input and returns customer details from ServiceNow.

Python Example:

    snow\_instance = "https://myinstance.service-now.com"

    #Accept caller phone number from contactflow using contact attrobutes.

    customer\_phone = event['Details']['Parameters']['Phone']

    # prepare API endpoint URL to get user details using phone number

    api\_endpoint\_url = snow\_instance + 'api/x\_ecsd\_amazon\_conn/'

                    + f'connect\_cti\_api/user?contact\_number={customer\_phone}'

    #prepare headers

    headers = {"Accept": "application/json"}

#Call API endpoint

response = requests.get(api\_endpoint\_url, auth=(servicenow\_user,

Response Format:

{

"result": {

"duplicateContact": "false",

"userCount": 1,

"usersDetails": [

{

"callerName": "Stephen Seiters",

"id": "1a826bf03710200044e0bfc8bcbe5da3"

}

]

}

}

servicenow\_password),headers=headers)

## Record details API endpoint

HTTP Method: GET

Resource Path: /api/x\_ecsd\_amazon\_conn/connect\_cti\_api/record

Resource URI: api/x\_ecsd\_amazon\_conn/connect\_cti\_api /record?active={true/false}&contact-number={contact\_number}&user-name={caller\_name}

Description: This resource accepts contact\_number and caller\_name as input parameters and returns active records.

Python Example:

    snow\_instance = "https://myinstance.service-now.com"

    #Accept caller phone number and customer name from contactflow

    customer\_phone = event['Details']['Parameters']['Phone']

    customer\_name =  event['Details']['Parameters']['Name']

    # prepare API endpoint URL to get active record details

    Api\_endpoint\_url = snow\_instance + 'api/x\_ecsd\_amazon\_conn/connect\_cti\_api/'\

                    + f'record?active=true&contact\_number={customer\_phone}'\

                    + f'&caller\_name={customer\_name}'

    #prepare headers

    headers = {"Accept": "application/json"}

    #Call API endpoint

    response = requests.get(url\_with\_extension, auth=(servicenow\_user,

                            servicenow\_password), headers=headers)

Response Format:

{

            "usersDetails": {

"callerName": "Alissa Mountjoy",

"incident": 6,

"hrcase": 0,

"csmcase": 2,

"recordDetails": {

"Customer Service": [

{

"number": "0001919",

"short\_description": "",

"open\_date": "2019-09-05 10:15:14",

"assigned\_to": "",

"assigned\_to\_id": null,

"active": true,

"id": "b503c640db33b7008d4d9098f496194e"

}

],

"Incident": [

{

"number": "0012066",

"short\_description": "User cannot login",

"open\_date": "2019-09-10 13:22:59",

"assigned\_to": "Prasad Dhule",

"assigned\_to\_id": "ec504f3adb1553000da6e536ca961990",

"active": true,

"id": "70fdd4f11bbbfb009eb8ed776e4bcbab"

}

]

}

}

}

## Active Module API endpoint

HTTP Method**:** GET

Resource Path**:** /api/x\_ecsd\_amazon\_conn/connect\_cti\_api/module

Description: This resource returns list of modules selected on Admin Console.

Python Example:

    snow\_instance = "https://myinstance.service-now.com"

    #prepare API endpoint URL to get user details using phone number

    api\_endpoint\_url = snow\_instance + '/api/x\_ecsd\_amazon\_conn/connect\_cti\_api/module'

    #prepare headers

    headers = {"Accept": "application/json"}

    #call API endpoint

    response = requests.get(api\_endpoint\_url, auth=(servicenow\_user,

                                servicenow\_password),headers=headers)

Response Format:

{

            "modules": [

{

"name": "CS",

"id": "sn\_customerservice\_case"

},

{

"name": "HRC",

"id": "sn\_hr\_core\_case"

},

{

"name": "INC",

"id": "incident"

}

]

}

## Incident record details API endpoint

HTTP Method: GET

Resource Path: /api/x\_ecsd\_amazon\_conn/connect\_cti\_api/incident/{record\_number}

Description: This resource returns incident details for incident number entered by caller.

Python Example:

    snow\_instance = "https://myinstance.service-now.com"

    #Accept record number form contact flow

    record\_number = event['Details']['Parameters']['record\_number']

    #Prepare headers

    headers = {"Accept": "application/json"}

    #Prepare API endpoint URL to get user details using phone number

    api\_endpoint\_url = snow\_instance + f'api/x\_ecsd\_amazon\_conn/'

                        +f'connect\_cti\_api/incident/{record\_number}'

    #Call API endpoint

    response = requests.get(api\_endpoint\_url,auth=(servicenow\_user,

Response format:

{

            "number": "INC0011347",

            "short\_description": "Test IT Incident",

            "open\_date": "14-Aug-2019",

            "last\_update": "10-Sep-2019",

"Assigned\_to": "Sachin Gawas",

"Assigned\_to\_ID": "1234",

"Phone\_number": "1234678960",

"active": "false"

}

servicenow\_password), headers=headers)

## HR case record details API endpoint

HTTP Method: GET

Resource Path: /api/x\_ecsd\_amazon\_conn/connect\_cti\_api/hr\_case/{record\_number}

Description: This resource returns HR Case details for HR case number entered by caller.

Response Format: Same as Incident API.

## CSM case record details API endpoint

HTTP Method: GET

Resource Path: /api/x\_ecsd\_amazon\_conn/connect\_cti\_api/csm\_case/{record\_number}

Description: This resource returns CSM Case details for CSM case number entered by caller.

Response Format: Same as Incident API.

Request syntax and response for this API endpoint is similar to Incident details API endpoint.

## Survey API endpoint

HTTP Method: PATCH

Resource Path: /api/x\_ecsd\_amazon\_conn/connect\_cti\_api/survey

Request Data: {

point: INTEGER,

call\_id: STRING (PREVIOUS CALL ID)

}

Description: Amazon Connect consumes this resource to send survey feedback points once call is ended and customer submit survey feedback though call.

Python Example:

    snow\_instance = "https://myinstance.service-now.com"

    #Prepare API endpoint URL

    api\_endpoint\_url = snow\_instance + 'api/x\_ecsd\_amazon\_conn/connect\_cti\_api/survey'

    #Prepare header

    headers = {"Accept": "application/json"}

    #prepare API endpoint input data header

    survey\_data = {"phone\_no": phone, "point" : survey\_points, "call\_id" : call\_id}

    #Call API endpoint

    response = requests.patch(url\_with\_extension, data=json.dumps(survey\_data), auth=(servicenow\_user, servicenow\_password), headers=headers)

# Support and Troubleshooting

## Support

For technical support, please contact ECS support at RocketCX.support@ecs.co.uk

|  |  |
| --- | --- |
| **Aspect** | **Details** |
| Support Hours of Operation | 09:00 - 17:00 (GMT) |
| Support Days of Operation | Monday - Friday (excluding U.K Public Holidays) |
| Promised Response Time | Within 1 Business day of received support request |
| Promised Resolution Time | Within 15 Business Days of response |
| Contact Method | Email |
| Contact Details | RocketCX.support@ecs.co.uk |

## Troubleshooting

### Amazon Connect Troubleshooting Issues with the CCP

<https://docs.aws.amazon.com/connect/latest/adminguide/troubleshooting.html>

### Connectivity Tool

AWS Provides a Tool to test connectivity issues with the CCP, the tool can be accessed here: -

<https://s3.amazonaws.com/connectivitytest/checkConnectivity.html>

Can't Hear Caller or Caller Can't Hear Agent? <https://docs.aws.amazon.com/connect/latest/adminguide/cant-hear-caller.html>

End of Document