

ExtraCredit – RentTrack Integration

BuildCreditModule - AWS and Heroku Infra Setup

Table of Contents

[1. High-level Infra Setup 3](#_Toc21713609)

[a. Non-Prod 3](#_Toc21713610)

[b. Production <TBD> 4](#_Toc21713611)

[2. Environment: Dev 4](#_Toc21713612)

[a. Database 4](#_Toc21713613)

[b. ActiveMQ 8](#_Toc21713614)

[c. Heroku Shield Private Space 10](#_Toc21713615)

[3. Environment: Test 11](#_Toc21713616)

[a. Database 11](#_Toc21713617)

[b. ActiveMQ 12](#_Toc21713618)

[c. Heroku Shield Private Space 12](#_Toc21713619)

[4. Environment: Stage 12](#_Toc21713620)

[a. Database 12](#_Toc21713621)

[b. ActiveMQ 13](#_Toc21713622)

[c. Heroku Shield Private Space 16](#_Toc21713623)

[5. Environment: Production <TBD> 16](#_Toc21713624)

[a. Database 16](#_Toc21713625)

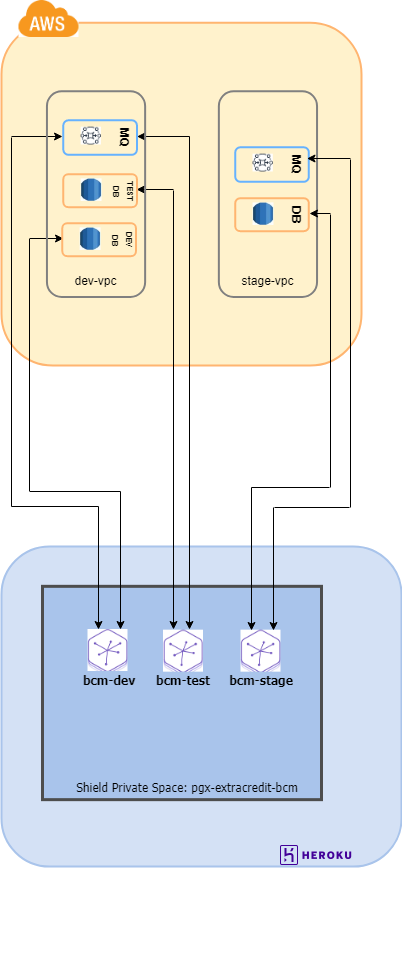
[b. ActiveMQ 16](#_Toc21713626)

[c. Heroku Shield Private Space 16](#_Toc21713627)

# High-level Infra Setup

## Non-Prod

The diagram given below gives a high-level view of the AWS and Heroku Infra setup in Non-Prod. Connectivity between AWS VPC and Heroku Shield Private Space is established through VPC Peering. Dev and Test Database instances are provisioned in the ‘dev-vpc’, and the existing MQ Broker in ‘dev-vpc’ is used for both Dev and Test region Queue. Stage database instances are provisioned in ‘stage-vpc’. The existing stage MQ broker is used for stage environment.



## Production <TBD>

# Environment: Dev

## Database

For BCM, we would be using the Amazon Aurora DB with PostgreSQL compatible version

#### Environment Details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| db cluster name | Endpoints | DB name | Engine version | Multi AZ deployment | VPC |
| dbclusteridentifier-extracresit-bcm-dev-data01 | [Writer]  dbclusteridentifier-extracredit-bcm-dev-data01.cluster-czqcrnu8rq1a.us-west-2.rds.amazonaws.com  [Reader] dbclusteridentifier-extracredit-bcm-dev-data01.cluster-ro-czqcrnu8rq1a.us-west-2.rds.amazonaws.com | BuildCredit | 10.7 | No | extracredit-dev-vpc-vpc (vpc-0ccd8c86dea26ace2) |

#### Configuration Steps

AWS CloudFormation provides a common language for you to describe and provision all the infrastructure resources in your cloud environment. CloudFormation allows you to use programming languages or a simple text file to model and provision, in an automated and secure manner, all the resources needed for your applications across all regions and accounts.

The database in environment is created through CloudFormation, following existing templates used for Rewards module.

BCM Template name: ExtraCredit BCM Dev Deployment

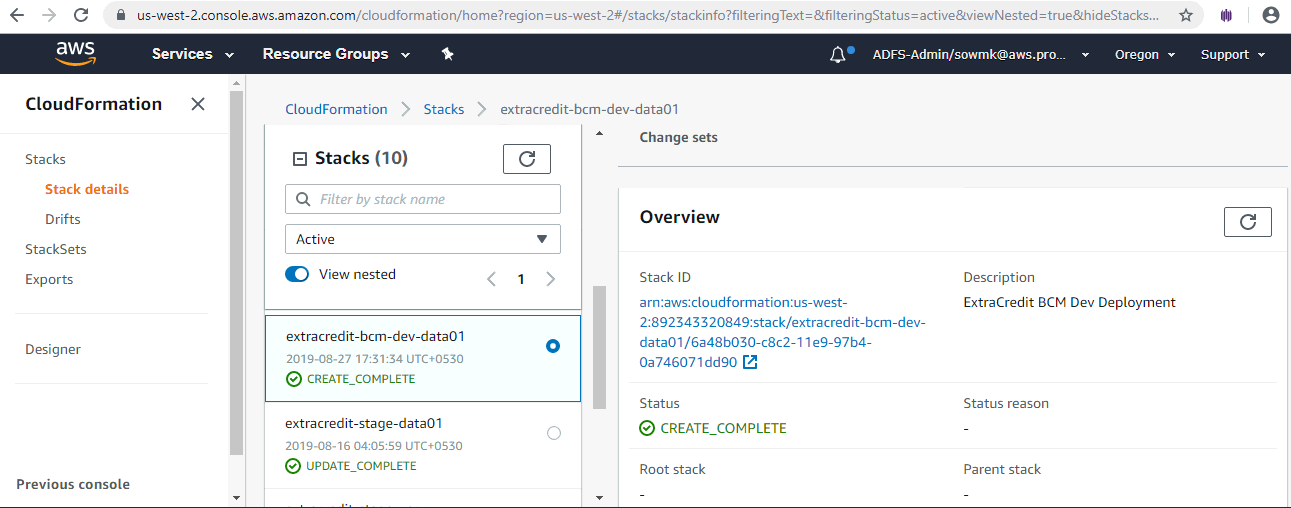
Referred Template: ExtraCredit Dev Deployment

Step by step details related to the same are given below:

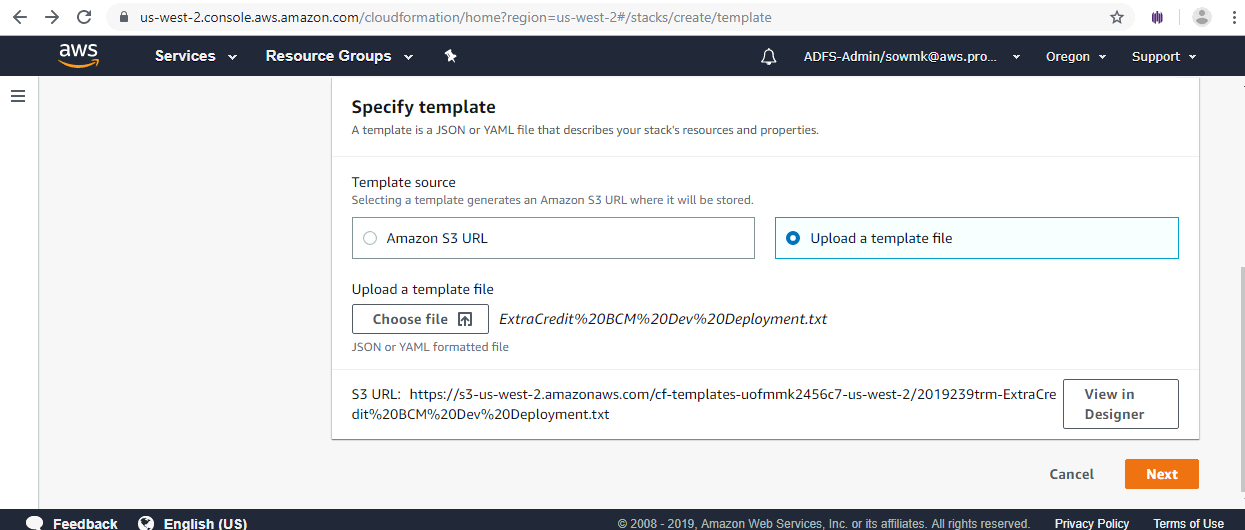
Step1:

Login to the AWS console > Go to CloudFormation > Click on Stacks > Click on Create Stack

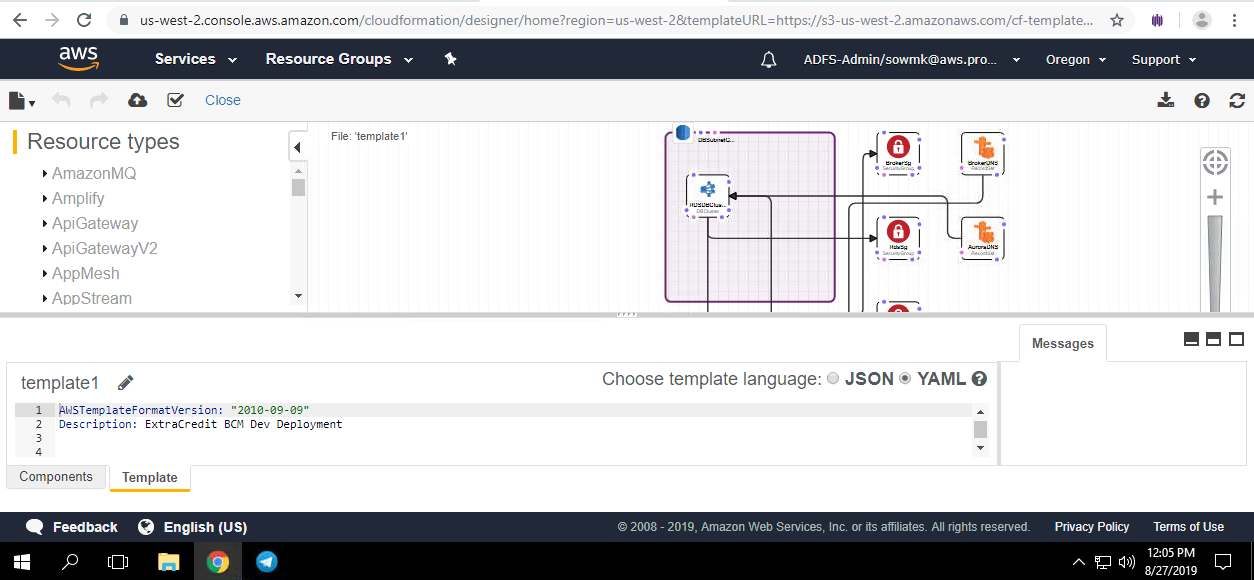
Click on “Upload a template file”



Step2: Upload the template file. The template is written in JSON format. For dev environment we are going with single instance deployment with no read replica, and all the read and write operations will be done on the primary database instance.

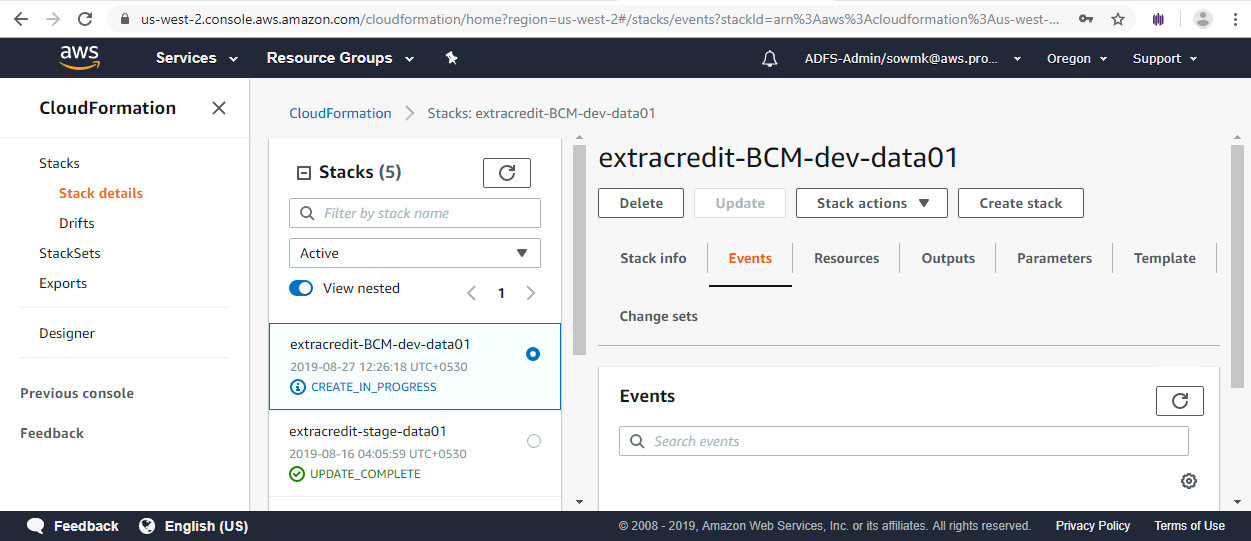


Step 3: After uploading the template, please validate the template by clicking the validate button (“Check mark” icon on the top left). If the template is valid click on Create Stack. If the template is invalid, check the errors and fix them.

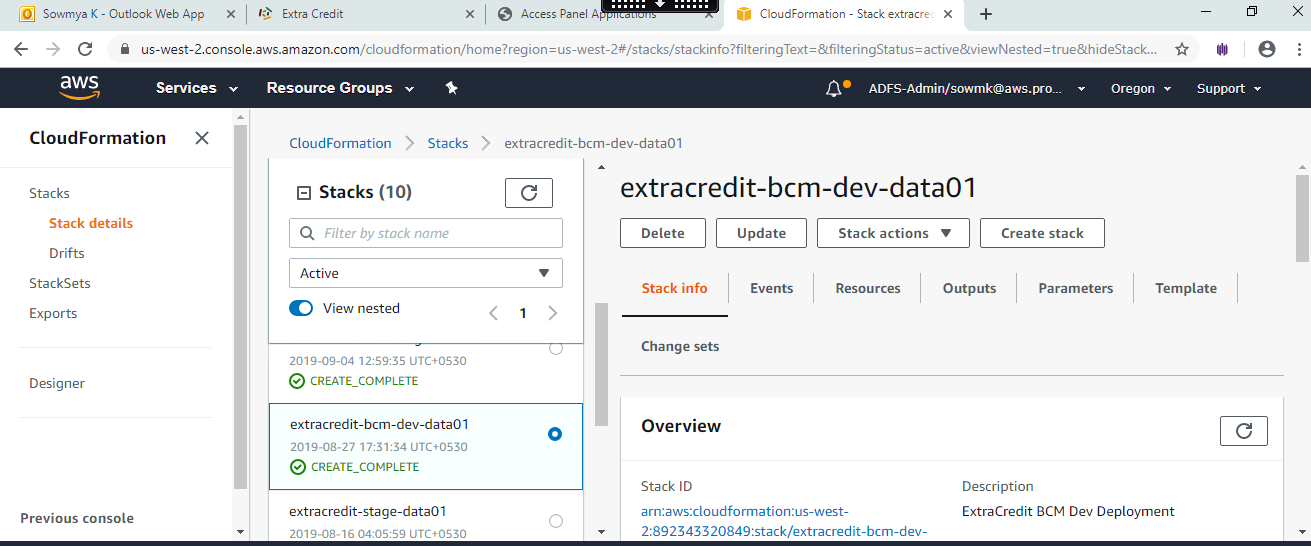


Step 4: Once the template is valid we will be providing parameters such as database name, MasterUsername and MasterUserPassword.

After entering the parameters, review the details and click on create stack. You can see the status CREATE\_IN\_PROGRESS, meanwhile we can see the events taking place - that is, resource creation happening one after the other.



Once the creation is complete, the status can be seen as CREATE\_COMPLETE. In case there are any errors, the stack creation will be rolled back, and we need to fix the errors and create the stack again.



## ActiveMQ

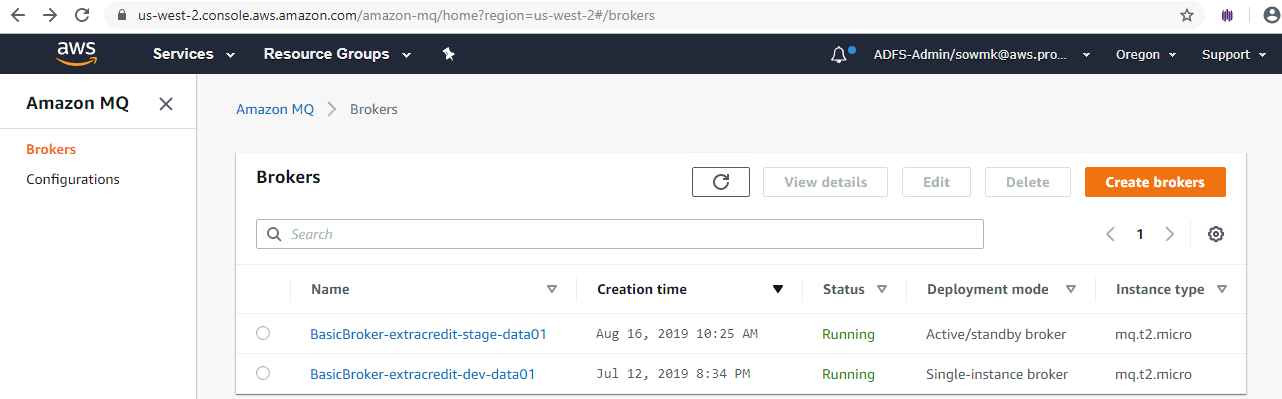
ActiveMQ Brokers have been created already for Dev and Stage (as part of Rewards Module’s setup), and BCM would be using those brokers to host the relevant Queue(s). There is no separate broker for Test environment - we would be using the one in Dev.

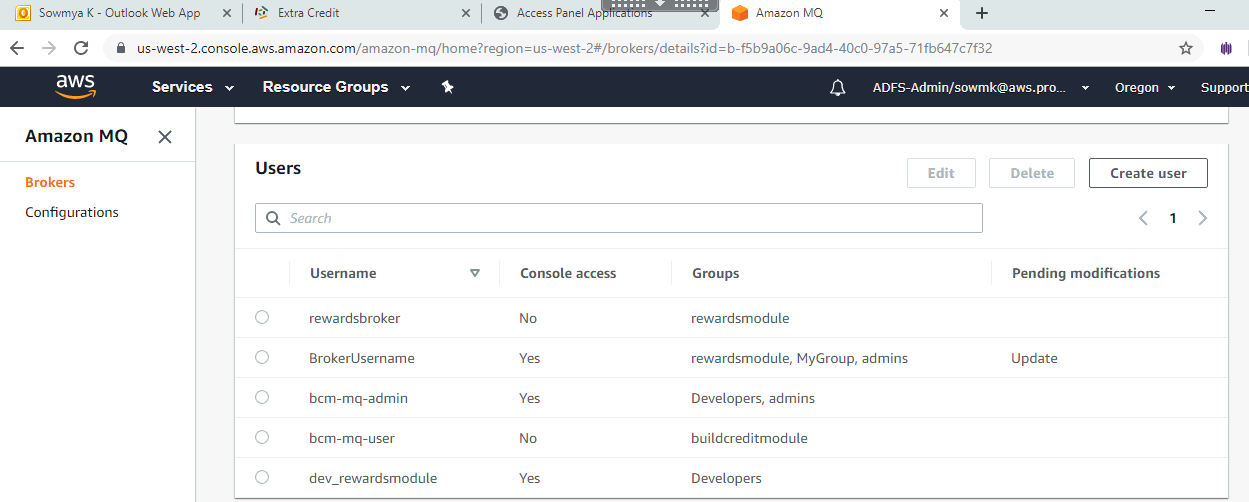
#### Environment Details

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Broker name | Broker instance type | Deployment mode | Active MQ Web console | Endpoint | Users created | Groups created |
| BasicBroker-extracredit-dev-data01 | mq.t2.micro | Single-instance broker | https:///b-f5b9a06c-9ad4-40c0-97a5-71fb647cf32-1.mq.us-west-2amazon.com:8162 | ssl://b-f5b9a06c-9ad4-40c0-97a5-71fb647cf32-1.mq.us-west-2amazon.com:61617 | bcm-mq-admin  bcm-mq-user | Developers,  Admins,buildcreditmodule |

#### Configuration Steps

The dev and test uses a single instance broker.



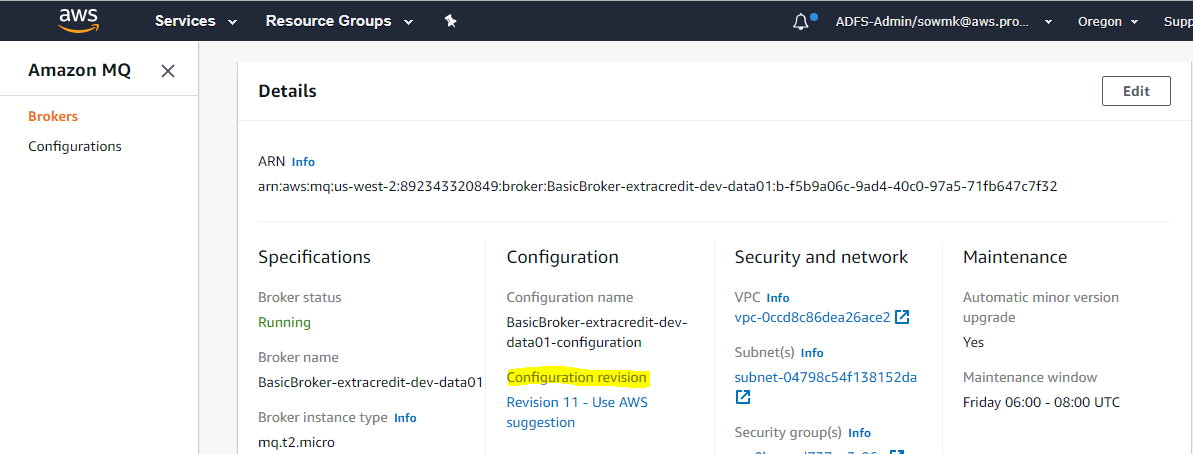


The Authorization plugin info has been updated to add the groups related to BCM

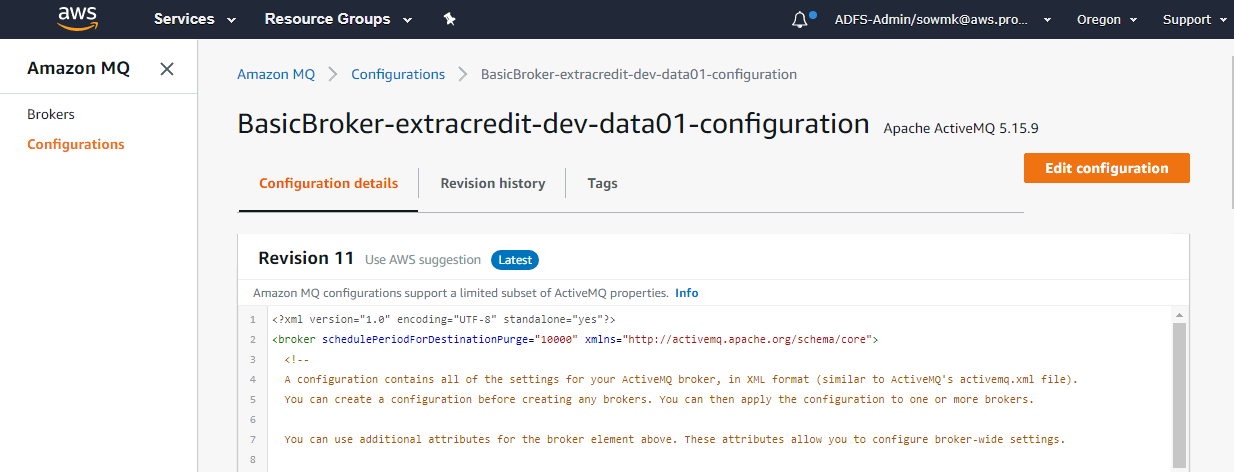


To change the Authorization plugin info:

* + 1. Navigate to Amazon MQ in the AWS console
    2. Select the relevant broker for dev(BasicBroker-extracredit-dev-data01), and
    3. In the details page under configuration revision, click on the revision link



* + 1. Click on edit configuration as shown in the below screen and make the necessary changes and save the changes.



Once the changes are saved, reboot the broker so that the change can take effect immediately.

## Heroku Shield Private Space

#### Environment Details

|  |  |  |  |
| --- | --- | --- | --- |
| Team Name | Shield space name | Region | CIDR range |
| progrexion | pgx-extracredit-bcm | Oregon | 10.99.0.0/16 |

#### Configuration Steps

Command used to create the Shield Private Space:

**> heroku spaces:create --cidr 10.99.0.0/16 --space pgx-extracredit-bcm --shield --team progrexion --region oregon**

NOTE:

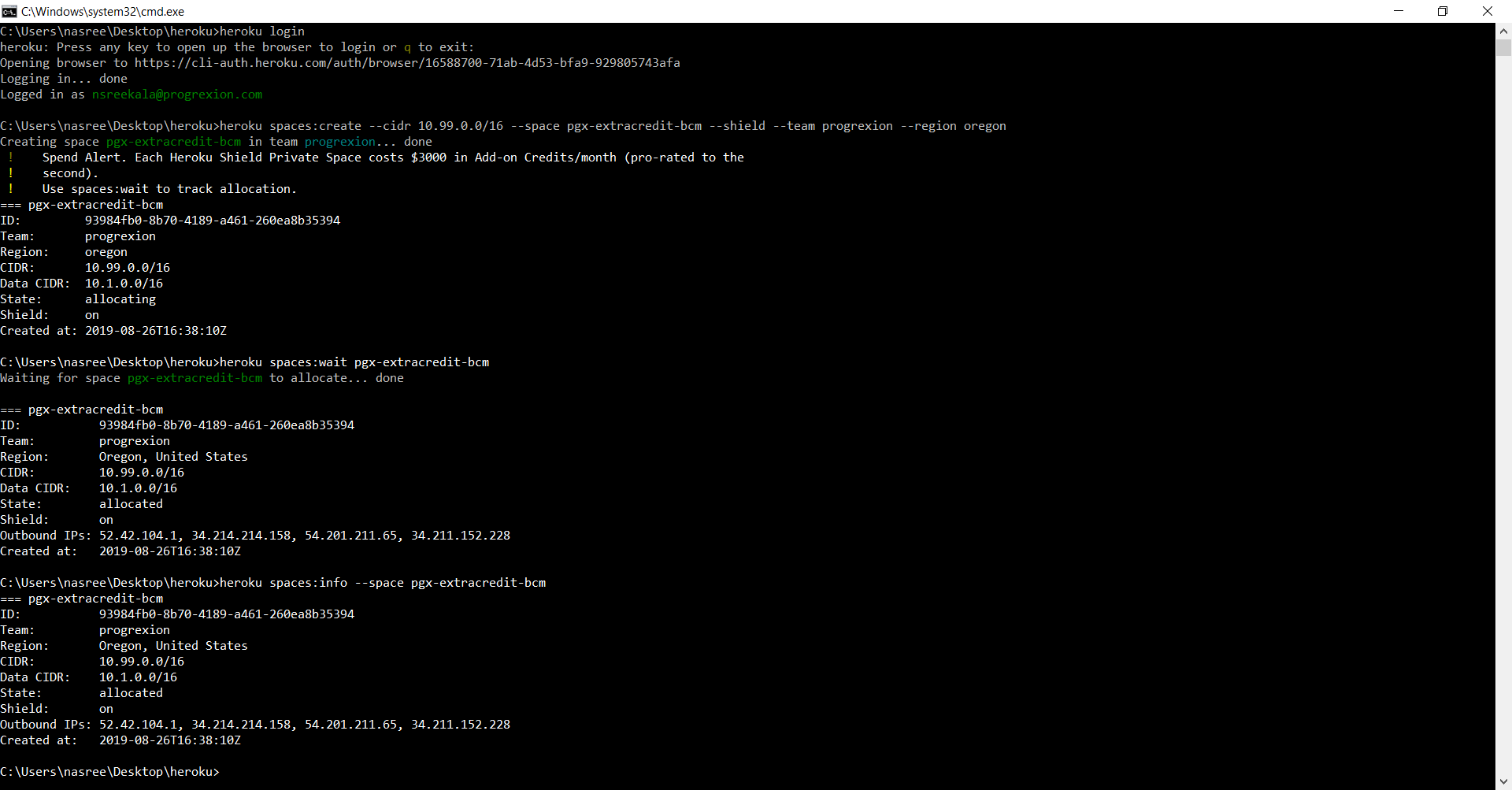
*CIDR*: 10.99.0.0/16 is the CIDR block range provided by Pgx

*Team*: ‘progrexion’ is the one used by Pgx

*Space Name*: ‘pgx-extracredit-bcm’ as chosen for BCM

*Region*: Provided ‘oregon’, mirroring the setting used for the other Private Space ‘pgx-production’ that is already in place under team ‘progrexion’.

Screenshot of the result is given below along with the “spaces:info” command output:



# Environment: Test

## Database

#### Environment Details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| db cluster name | Endpoints | DB name | Engine version | Multi AZ deployment | VPC |
| dbclusteridentifier-extracresit-bcm-test-data01 | [Writer]  dbclusteridentifier-extracredit-bcm-test-data01.cluster-czqcrnu8rq1a.us-west-2.rds.amazonaws.com  [Reader] dbclusteridentifier-extracredit-bcm-test-data01.cluster-ro-czqcrnu8rq1a.us-west-2.rds.amazonaws.com | BuildCredit | 10.7 | No | extracredit-dev-vpc-vpc (vpc-0ccd8c86dea26ace2) |

#### Configuration Steps

Configuration steps will be same as in Dev environment.

BCM Template name: ExtraCredit BCM Test Deployment

Referred Template: ExtraCredit BCM Dev Deployment

## ActiveMQ

The same message broker is used for dev and test, hence the Test region MQ details will be same as in Dev environment.

## Heroku Shield Private Space

#### Environment Details

Same as in Dev environment.

#### Configuration Steps

Same as in Dev environment.

# Environment: Stage

## Database

#### Environment Details

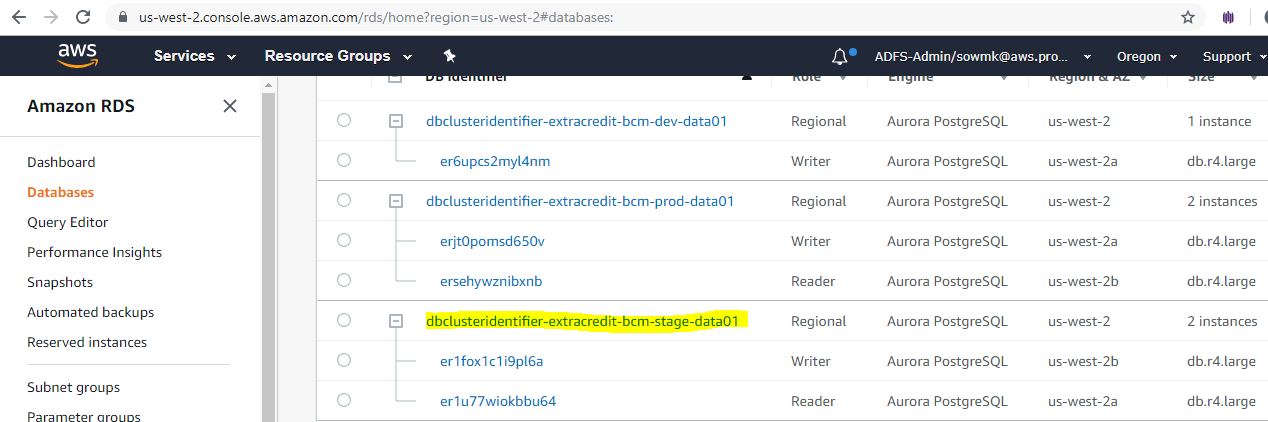
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DB Cluster Name | Endpoints | DB Name | Engine version | Multi AZ deployment | VPC |
| dbclusteridentifier-extracresit-bcm-stage-data01 | [Writer]  dbclusteridentifier-extracredit-bcm-stage-data01.cluster-czqcrnu8rq1a.us-west-2.rds.amazonaws.com  [Reader] dbclusteridentifier-extracredit-bcm-stage-data01.cluster-ro-czqcrnu8rq1a.us-west-2.rds.amazonaws.com | BuildCredit | 10.7 | Yes | extracredit-stage-vpc-vpc (vpc-08de1c25ecd0ac599) |

#### Configuration Steps

Configuration steps would remain same as dev expect that in stage we used Multi AZ deployment with one read replica as shown in the below screenshot.

BCM Template name: ExtraCredit BCM Stage Deployment

Referred Template: ExtraCredit Stage Deployment



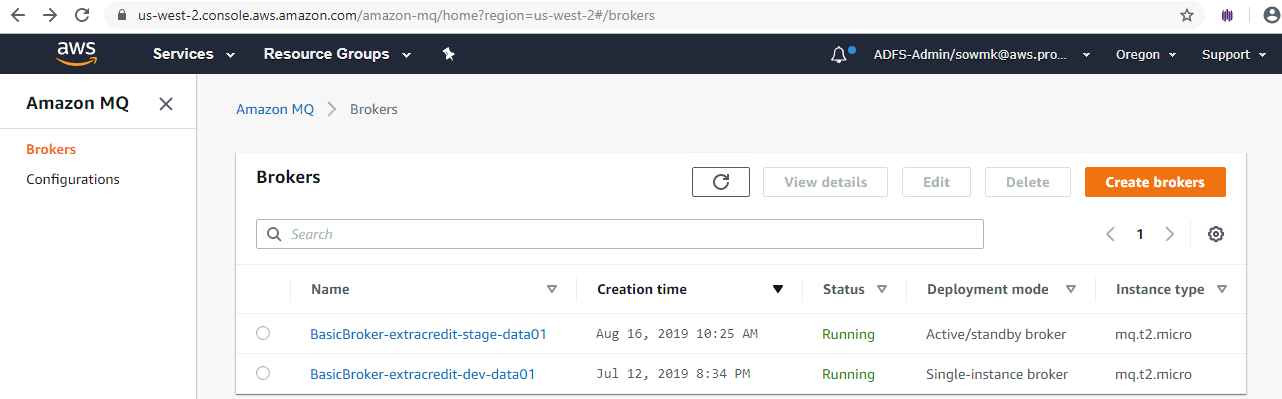
## ActiveMQ

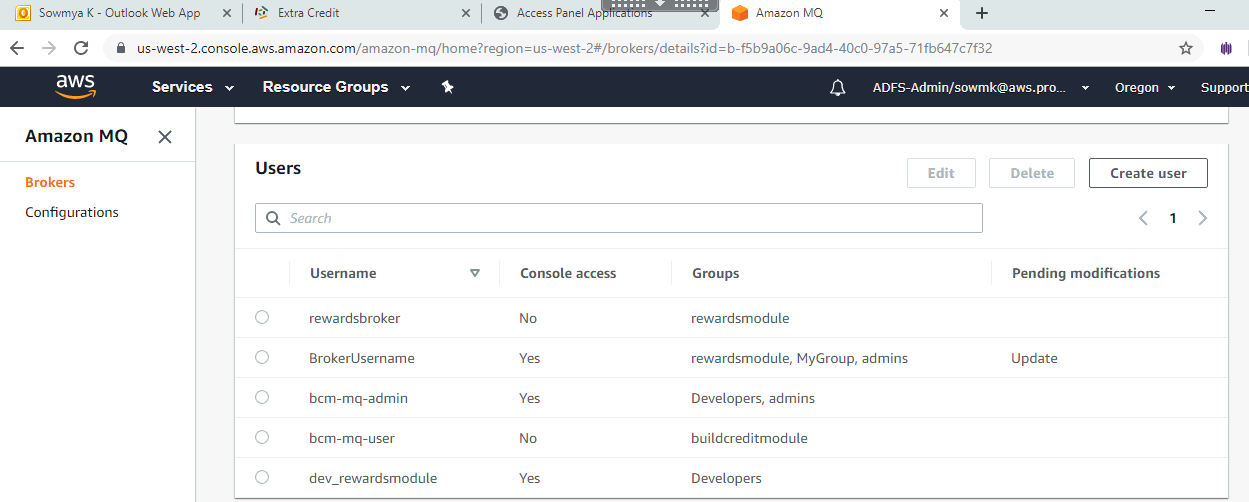
#### Environment Details

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Broker name | Broker instance type | Deployment mode | Active MQ Web console | Endpoint | Users created | Groups created |
| BasicBroker-extracredit-dev-data01 | mq.t2.micro | Active/Standby broker | BasicBroker-extracredit-stage-data01 [Active/Stand-by mode] | ssl://b-84e52d86-83dd-45da-b16d3ebaf755-1.mq.us-west-2.amazonaws.com:61617  ssl://b-84e52d86-83dd-45da-b16d3ebaf755-2.mq.us-west-2.amazonaws.com:61617 | bcm-mq-admin  bcm-mq-user | Developers,admins,buildcreditmodule |

#### Configuration Steps

Amazon MQ for stage environment uses Active/standby broker for high availability. Normally, only one of the broker instances is active at any time, while the other broker instance is on standby. If one of the broker instances malfunctions or undergoes maintenance, it takes Amazon MQ a short while to take the inactive instance out of service, allowing the healthy standby instance to become active and to begin accepting incoming communications. When you reboot a broker, the failover takes only a few seconds.

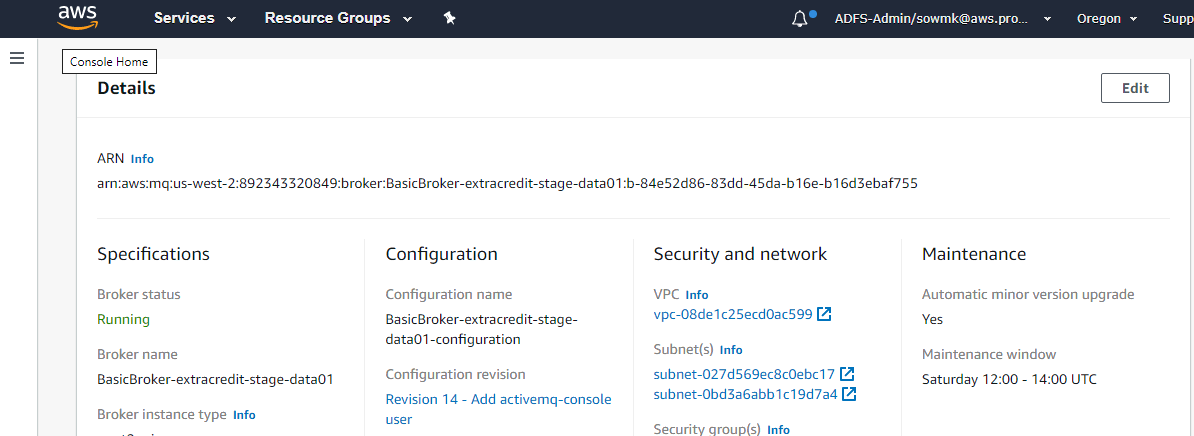




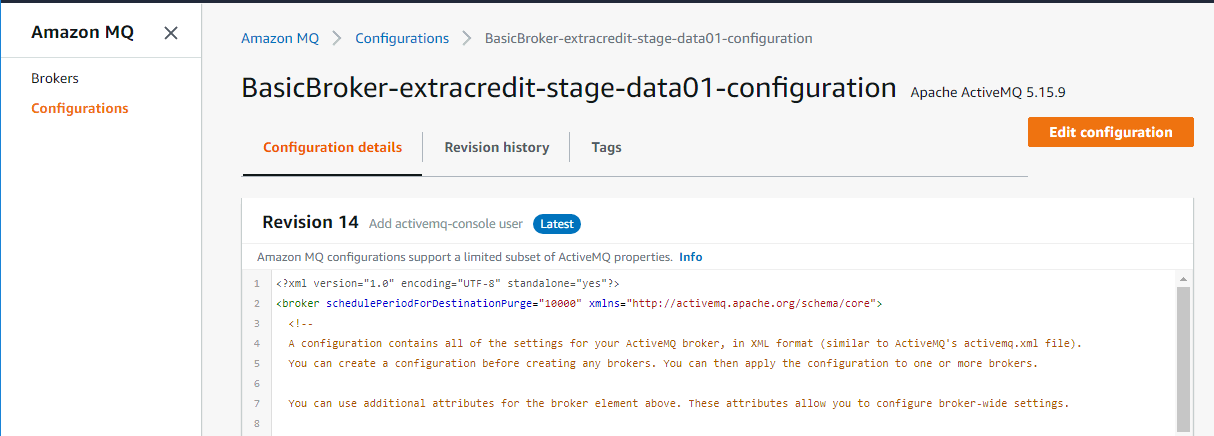
The Authorization plugin info has been modified to add the groups related to bcm module.



To change the Authorization plugin info navigate to Amazon MQ in the AWS console, select the relevant broker for stage (BasicBroker-extracredit-stage-data01) and in the details page under configuration revision click on the revision link.



Click on edit configuration as shown in the below screen and make the necessary changes and save the changes.



Once the changes are saved reboot the broker so that the change can take effect immediately.

## Heroku Shield Private Space

#### Environment Details

Same as dev

#### Configuration Steps

Same as dev

# Environment: Production <TBD>

## Database

#### Environment Details

<TBD>

#### Configuration Steps

<TBD>

## ActiveMQ

#### Environment Details

<TBD>

#### Configuration Steps

<TBD>

## Heroku Shield Private Space

#### Environment Details

<TBD>

#### Configuration Steps

<TBD>