**Telematics BE Project**

Steps to setup manually deployed AWS Services

Document Control Information

Document Edit History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Additions/Modifications | Prepared/Revised By |
| V0.1 | 01/12/2022 | First Draft | Deloitte |
|  |  |  |  |
|  |  |  |  |

Document Review/Approval History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Name | Organization/Title | Comments/Approval Email |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Acronym

|  |  |
| --- | --- |
| Acronym | Description |
| RDS | Relational Databases |
| AWS | Amazon Web Services |
| API | Application Programming Interfaces |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Contents

[1 Overview 4](#_Toc96439434)

[2 MSK Connector 4](#_Toc96439435)

[3 EKS Cluster 7](#_Toc96439436)

[4 Adding Secrets in Secrets Manager 8](#_Toc96439437)

[5 Infra As code Services. 9](#_Toc96439438)

[6 Microservices. 9](#_Toc96439439)

# Overview

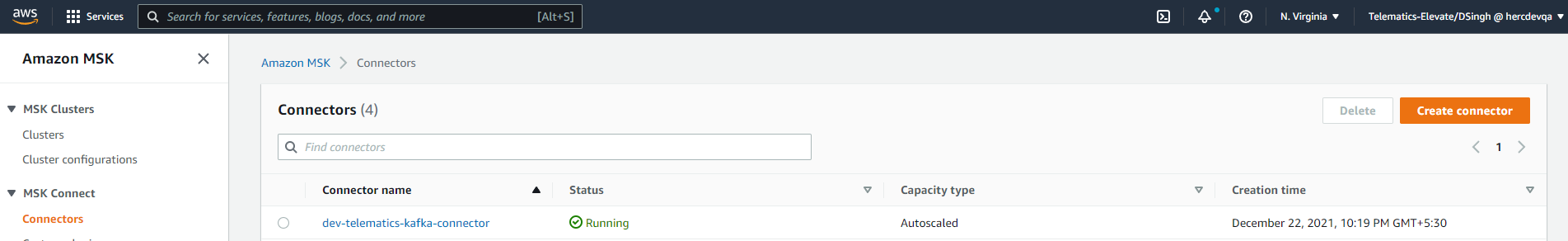
The purpose of this document is to list down steps to setup manually deployed AWS Services while moving the AWS services into the stage and production environments. The following services will be deployed and setup manually.

* MSK Connector
* EKS Cluster
* S3 Buckets
* Kubernetes Dashboard Deployment
* Secrets Manager secrets setup

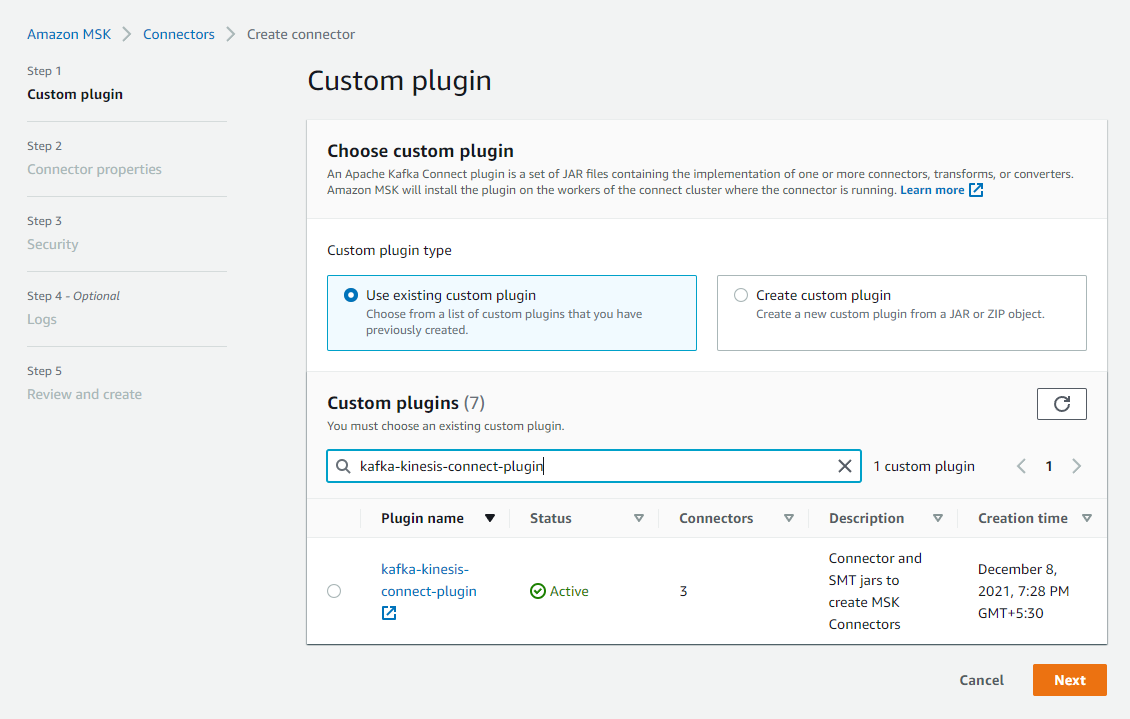
# MSK Connector

1. Log into AWS Environment and search for MSK

(Link: <https://console.aws.amazon.com/msk/home?region=us-east-1#/connectors>)



1. Click on option - **Create Connector**
2. Next select option to create from custom plugin and search for - **kafka-kinesis-connect-plugin**



1. Select the plugin and click next. Enter suitable connector details as follows:

**Connector name: qa/dev-telematics-kafka-connecto**

**Description: Task Description**

**Cluster Type: MSK Cluster**

**Cluster Name: qa/dev-elevaterewrite-kakfa**

**Connector configuration**

**Configuration settings**:

**connector.class=com.amazon.kinesis.kafka.AmazonKinesisSinkConnector**

**tasks.max=2**

**transforms=insertuuid**

**maxBufferedTime=1500**

**metricsGranuality=shard**

**aggregration=false**

**pauseConsumption=true**

**metricsNameSpace=KafkaKinesisStreamsConnector**

**usePartitionAsHashKey=false**

**singleKinesisProducerPerPartition=true**

**maxConnections=1**

**metricsLevel=detailed**

**rateLimit=100**

**flushSync=true**

**topics=qa.tu-simulator**

**transforms.insertuuid.uuid.field.name=Id**

**sleepPeriod=1000**

**streamName=qa-telemetry-raw-data-stream**

**ttl=60000**

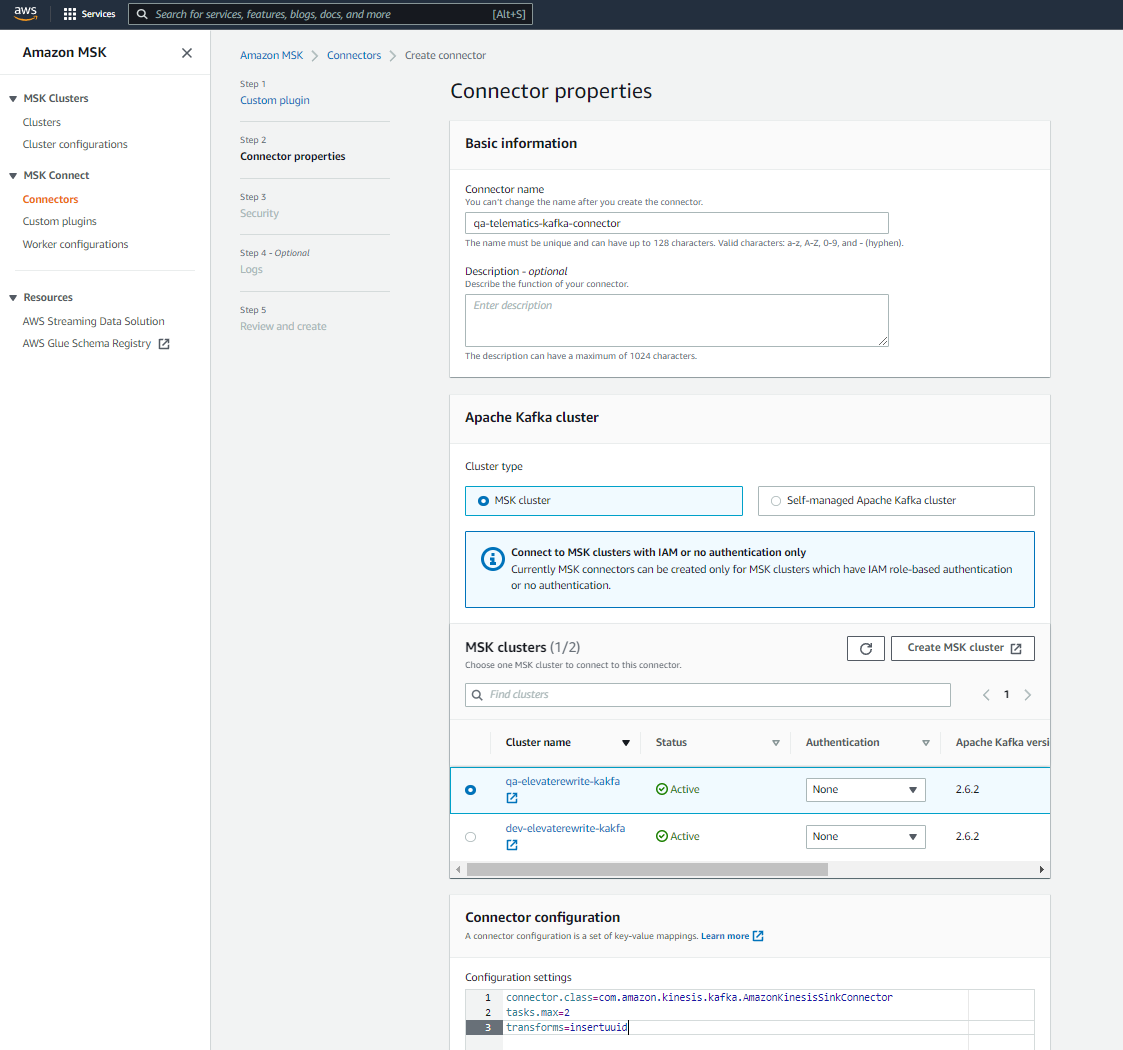
**sleepCycles=10**

**outstandingRecordsThreshold=500000**

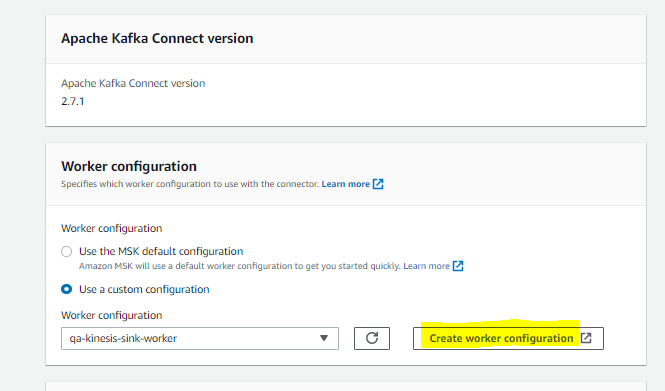
**avroParser=true**

**region=us-east-1**

**transforms.insertuuid.type=com.github.cjmatta.kafka.connect.smt.InsertUuid$Value**



1. Let the other values to be default and next under **Worker Configuration** select “**Use a custom configuration”**
2. Click on create worker configuration:



1. Enter the following details in Worker Configuration:

**Worker configuration name: qa-kinesis-sink-worker**

**Description: For kinesis sink connector**

**Worker configuration:**

**key.converter=org.apache.kafka.connect.storage.StringConverter**

**value.converter=io.confluent.connect.avro.AvroConverter**

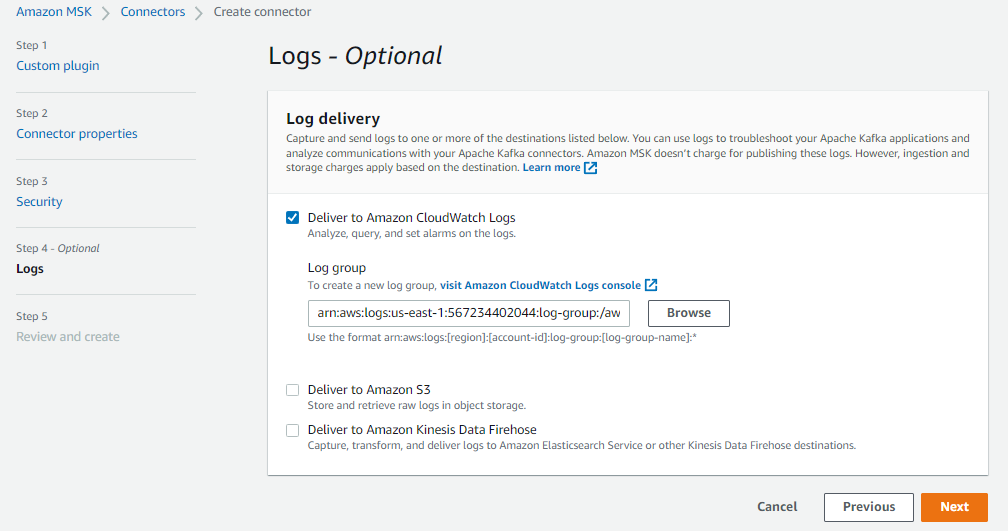
**value.converter.schema.registry.url=http://10.219.194.225:8080**

**key.converter.schemas.enable=false**

**value.converter.schemas.enable=true**

**schemas.enable=false**

1. Click create worker configuration
2. Go back to the previous tab
3. Select the created worker configuration: **qa-kinesis-sink-worker**
4. Next, select the Identity and Access Management (IAM) role: Msk-KafkaRole
5. Click Next -> Next
6. Check Deliver to Amazon CloudWatch Logs box & browse the log group: /aws/mskconnect



1. Click Next and the connector is created successfully

# EKS Cluster

1. Log into AWS Environment and search for EKS

(Link: <https://console.aws.amazon.com/eks/home?region=us-east-1#/clusters>)

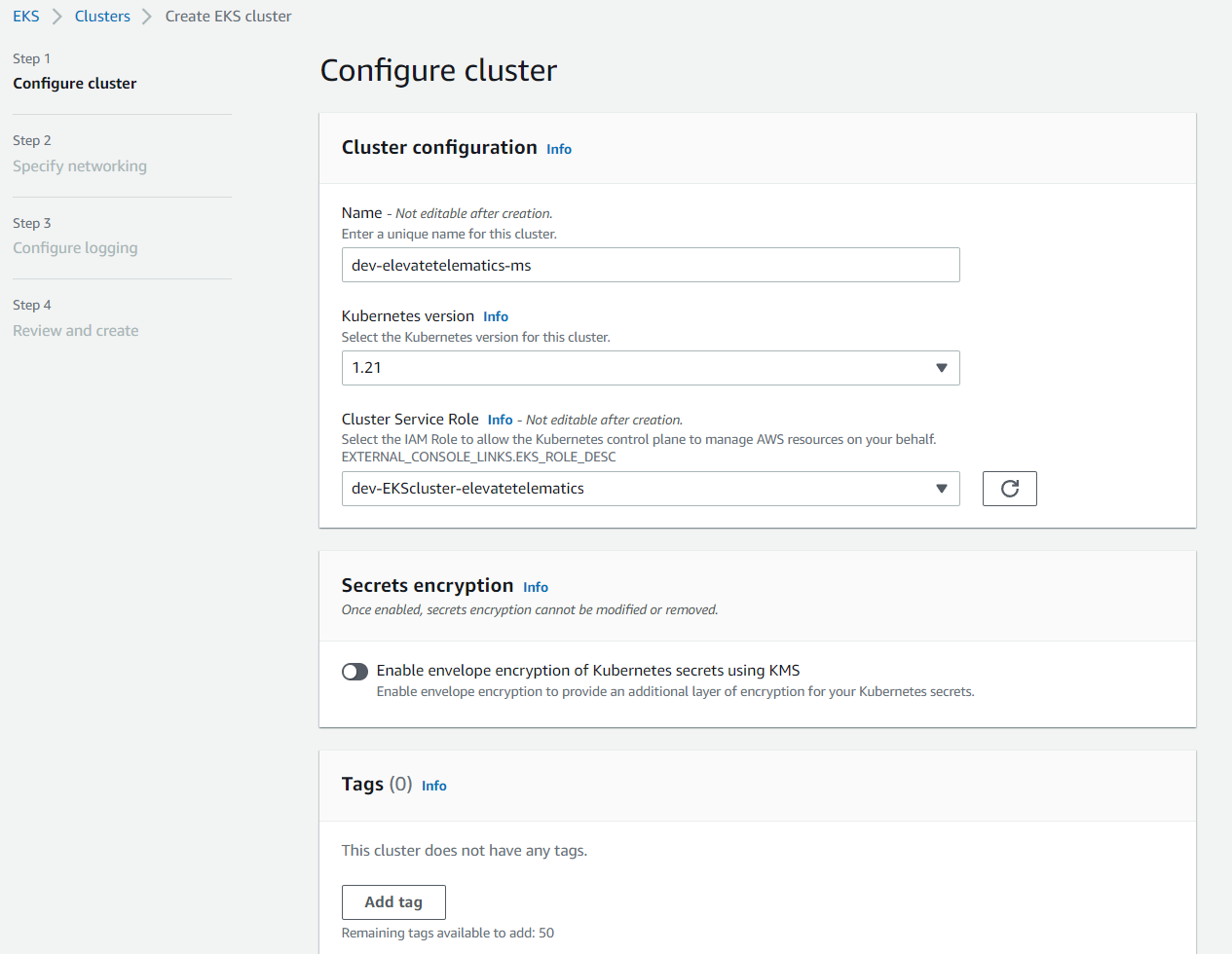
1. Click on option – **Add Cluster -> Create**
2. Next Configure cluster using suitable details:

**Cluster Name: dev/qa-elevatetelematics-ms**

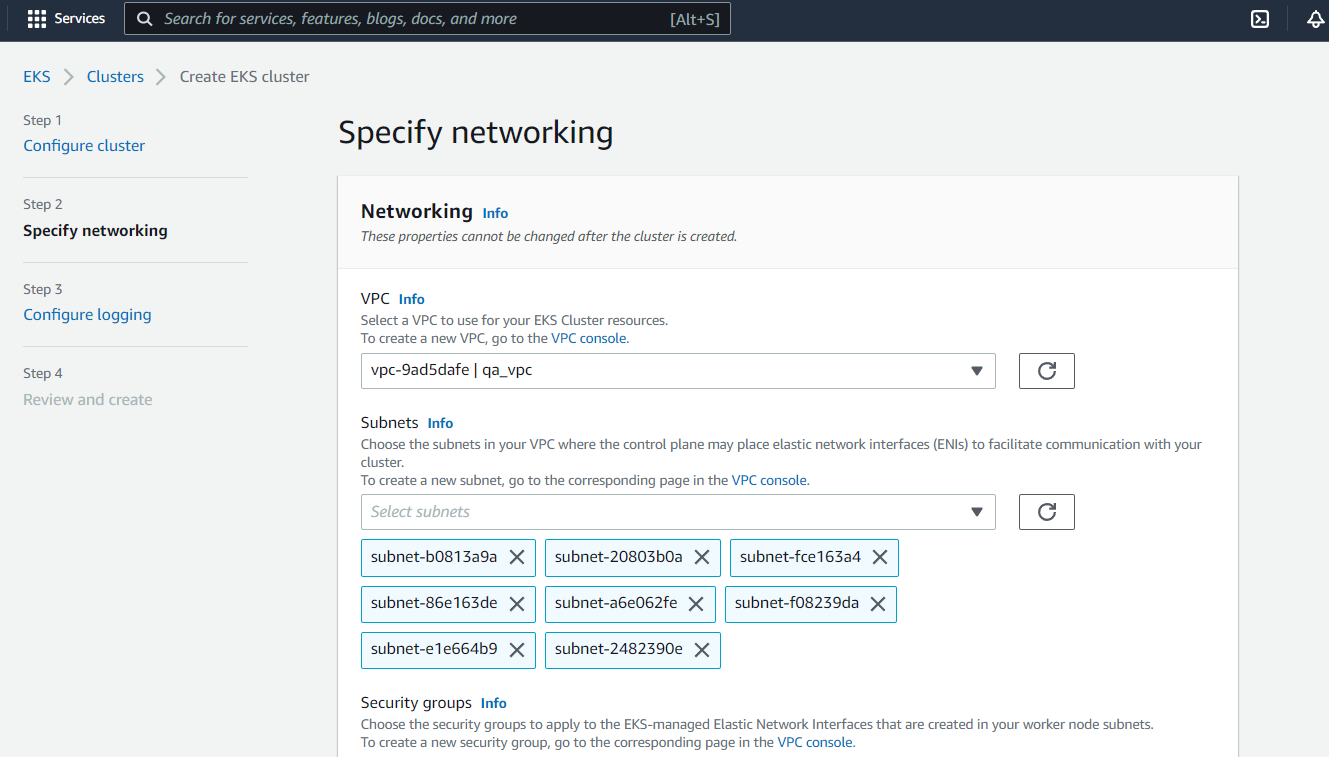
**Kubernetes version: 1.21**

**Cluster Service Role: dev-EKScluster-elevatelematics**

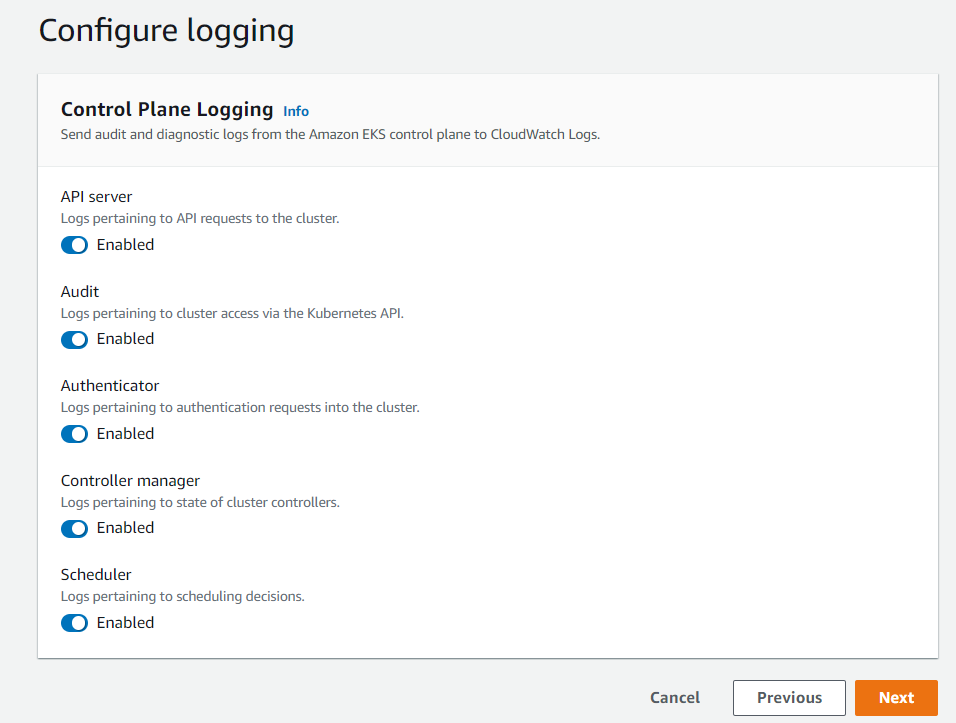
**Tags : Add Suitable tags**



1. Click on Next and Select VPC and Sg configurations



1. Let other values stay as default values and Click Next, Enable All Logging Parameters



1. Review the cluster and the cluster creation is successful.

# Adding Secrets in Secrets Manager

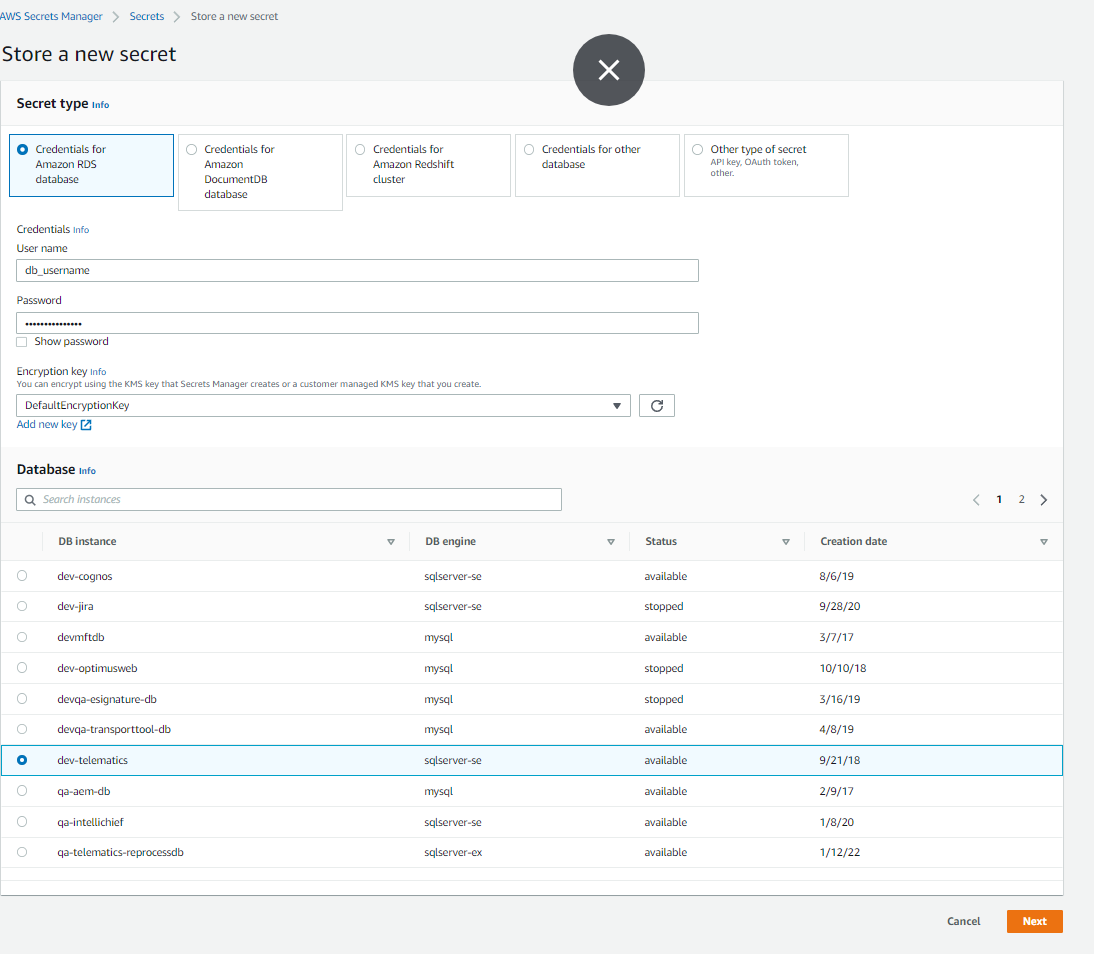
1. Log into AWS Environment and search for AWS Secrets Manager

(Link: <https://console.aws.amazon.com/secretsmanager/home?region=us-east-1#!/listSecrets/>)

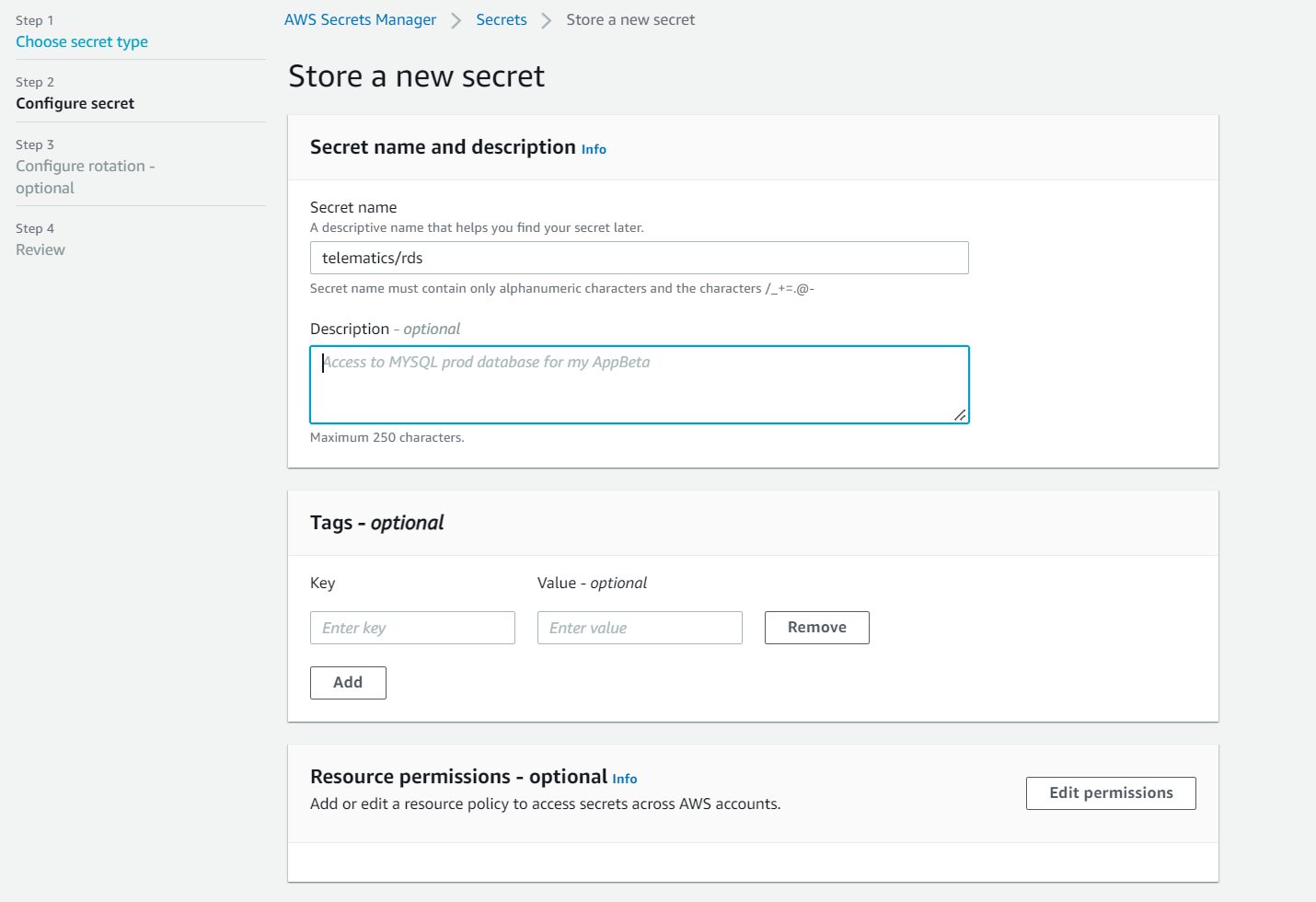
1. Click on option – **Store a new secret**
2. **Select Secret Type: Credentials for RDS**

**Enter the Secret Values (Username & Password/Secret text)**

**Select Encryption key and then the Database to configure the secrets**



1. Click on Next and enter Secret Name,Description,Tags



1. Click on Next and Review the details. Click on Store and the Secret is created successfully for the configured AWS Service

# S3 Bucket

1. Log into AWS Environment and search for AWS Secrets Manager

(Link: <https://s3.console.aws.amazon.com/s3/home?region=us-east-1#>)

1. Click on option – **Create Bucket**
2. **Fill the following details:**

**Bucket Name, Select Region, Versioning(Enable/Disable), Encryption(Disable/Enable), Object Lock(Enable/Disable)**

***Note- All the features except Object Lock can be enabled/disabled even after bucket creation***

1. Once filled click on create bucket and then upload the required object in bucket and the S3 bucket is setup successfully