**Steps to integrate S3 with Amazon Oracle RDS**

You can transfer files between an Amazon RDS for Oracle DB instance and an Amazon S3 bucket. You can use Amazon S3 integration with Oracle features such as Data Pump. For example, you can download Data Pump files from Amazon S3 to the DB instance host.

**Prerequisites for Amazon RDS Oracle Integration with Amazon S3**

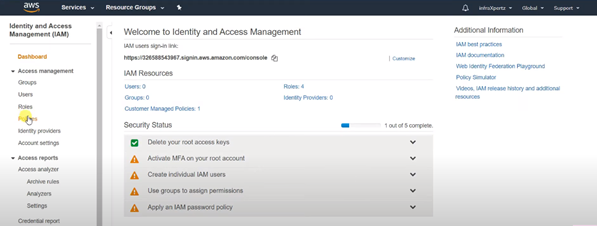
To work with Amazon RDS for Oracle integration with Amazon S3, the Amazon RDS DB instance must have access to an Amazon S3 bucket. For this, you create an AWS Identity and Access Management (IAM) policy and an IAM role. The Amazon VPC used by your DB instance doesn’t need to provide access to the Amazon S3 endpoints.

To add a role to a DB instance, the status of the DB instance must be available.

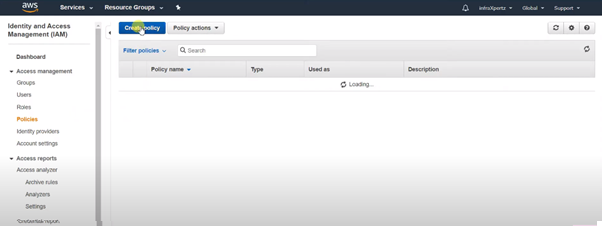
Below are the step by step procedure to create Amazon RDS to S3 Integration :

**To create an IAM policy to allow Amazon RDS access to an Amazon S3 bucket**

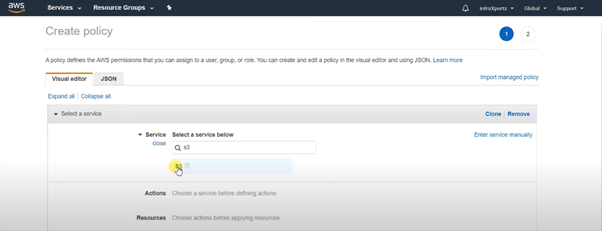
1. Open the [IAM Management Console](https://console.aws.amazon.com/iam/home?#home).
2. In the navigation pane, choose **Policies**.



3. Choose Create Policy



4. On the **Visual editor** tab, choose **Choose a service**, and then choose **S3**.

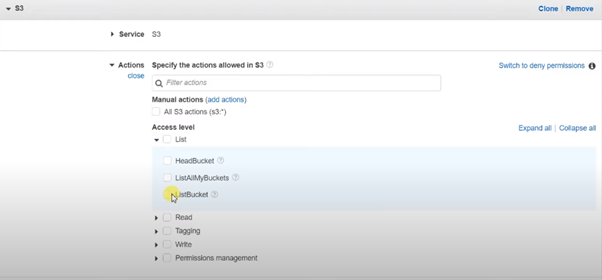


5. For **Actions**, choose **Expand all**, and then choose the bucket permissions and object permissions needed for the IAM policy.

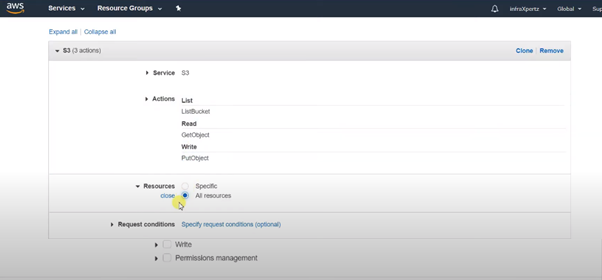
Include the appropriate actions in the policy based on the type of access required:

* GetObject– Required to transfer files from an Amazon S3 bucket to Amazon RDS.
* ListBucket– Required to transfer files from an Amazon S3 bucket to Amazon RDS.
* PutObject– Required to transfer files from Amazon RDS to an Amazon S3 bucket.

*Object permissions* are permissions for object operations in Amazon S3, and need to be granted for objects in a bucket, not the bucket itself. For more information about permissions for object operations in Amazon S3, see Permissions for Object Operations.

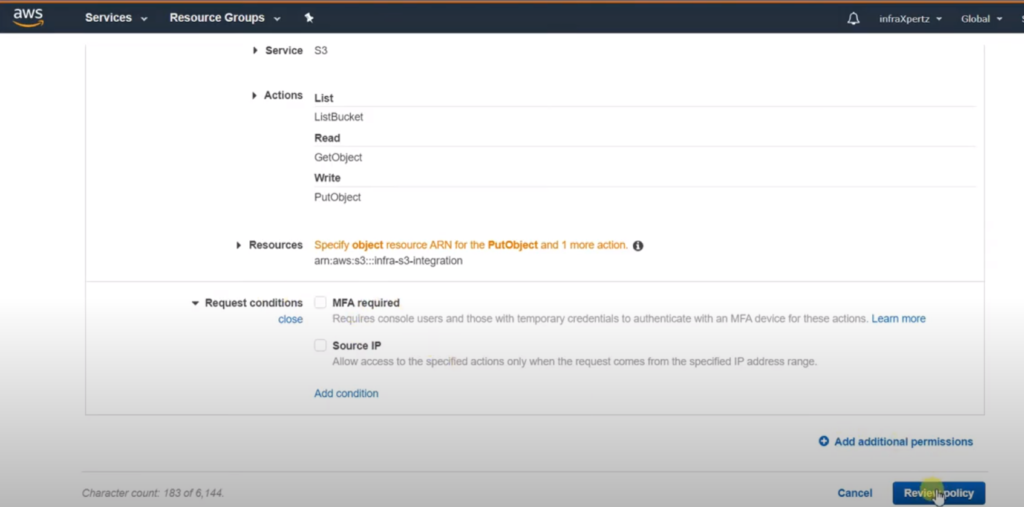


6. Choose **Resources**, and choose **Add ARN**for **bucket**.

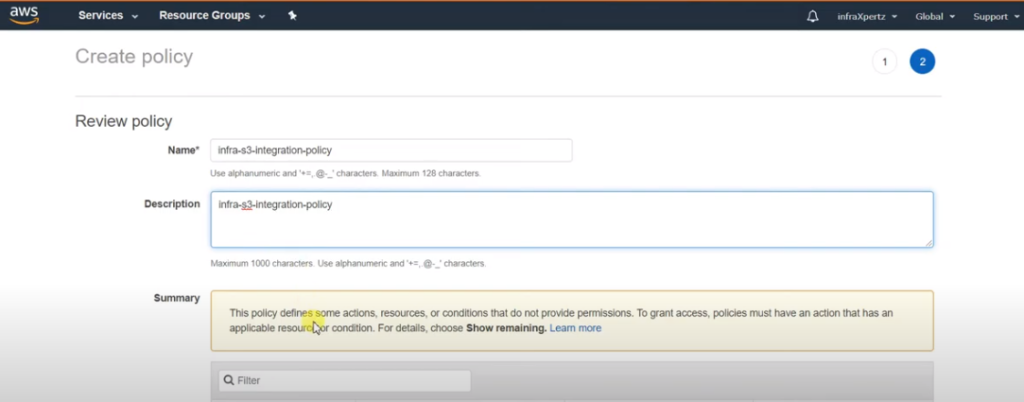


7. For the Amazon S3 bucket, specify the Amazon S3 bucket to allow access to. For the object, you can choose **Any** to grant permissions to any object in the bucket

8. Choose **Review policy**



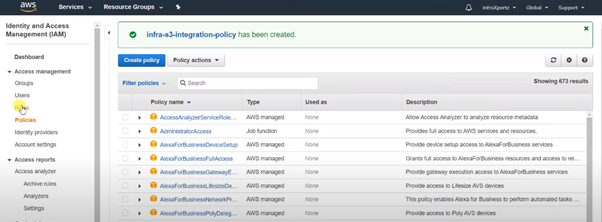
9. For **Name**, enter a name for your IAM policy, for example infra-s3-integration-policy. You use this name when you create an IAM role to associate with your DB instance. You can also add an optional **Description**

****

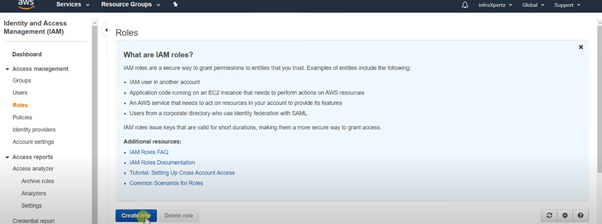
10. Choose **Create policy**.

**To create an IAM role to allow Amazon RDS access to an Amazon S3 bucket**

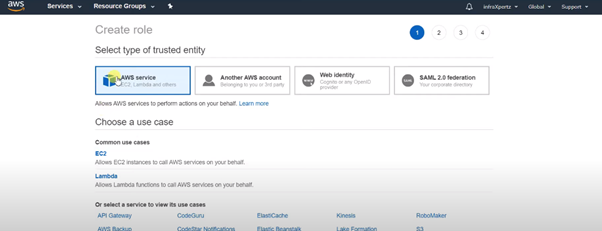
1. In the navigation pane, choose **Roles**.



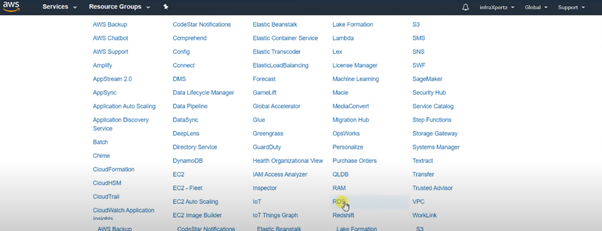
2. Choose Create Role

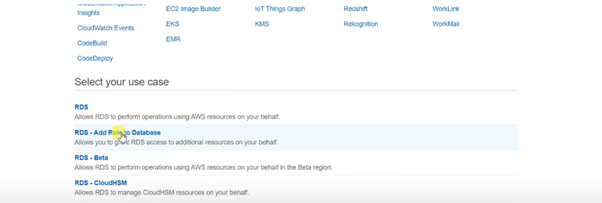


3. For AWS Service, choose RDS



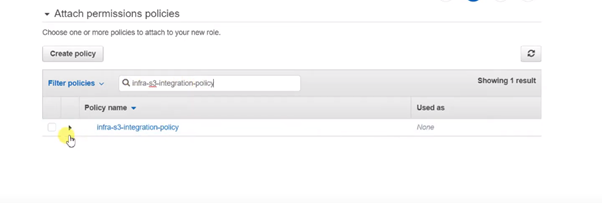
4. For **Select your use case**, choose **RDS – Add Role to Database**.





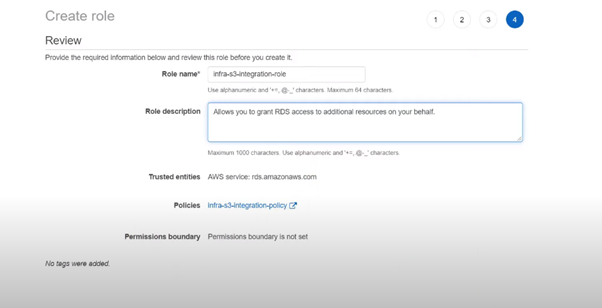
5.  Choose **Next: Permissions**.

6. For **Search**under **Attach permissions policies**, enter the name of the IAM policy you created, and choose the policy when it appears in the list.



7. Choose **Next: Tags**and then **Next: Review**.

8. Set **Role name**to a name for your IAM role, for example infra-s3-integration-role. You can also add an optional **Role description**

****

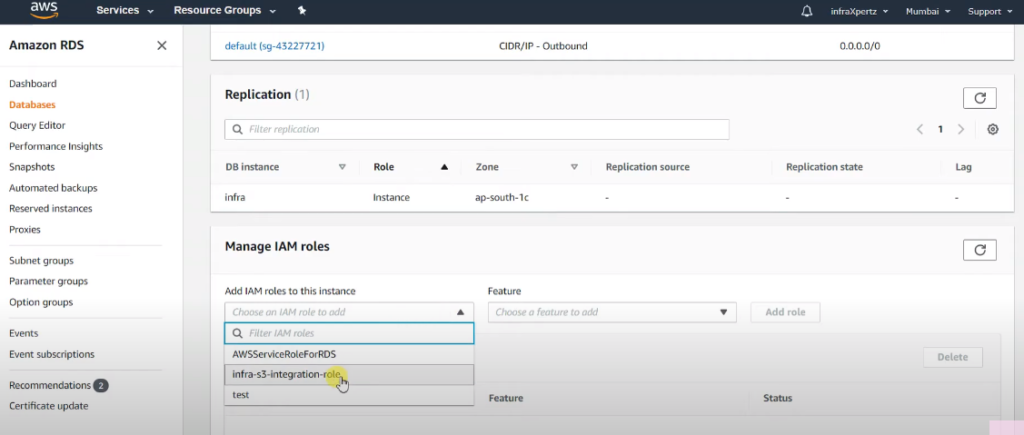
9. Choose **Create Role**.

**To associate your IAM role with your DB instance**

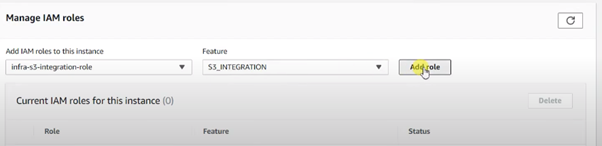
1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.

2. Choose the Oracle DB instance name to display its details.

3. On the **Connectivity & security** tab, in the **Manage IAM roles** section, choose the role to add under **Add IAM roles to this instance**

****

4. For **Feature**, choose **S3\_INTEGRATION**.



5. Choose **Add role**.

**Adding the Amazon S3 Integration Option**

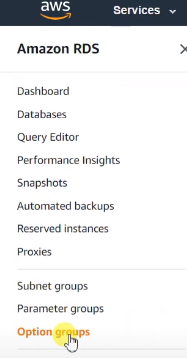
To use Amazon RDS for Oracle Integration with Amazon S3, your Amazon RDS Oracle DB instance must be associated with an option group that includes the S3\_INTEGRATION option.

**To configure an option group for Amazon S3 integration**

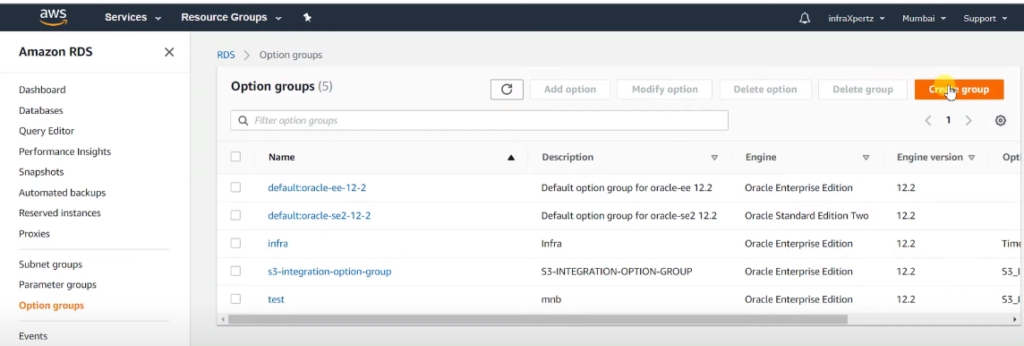
**Create a new option group or identify an existing option group to which you can add the S3\_INTEGRATION**

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.

2. In the navigation pane, choose **Option groups**.



3. Choose **Create group**.



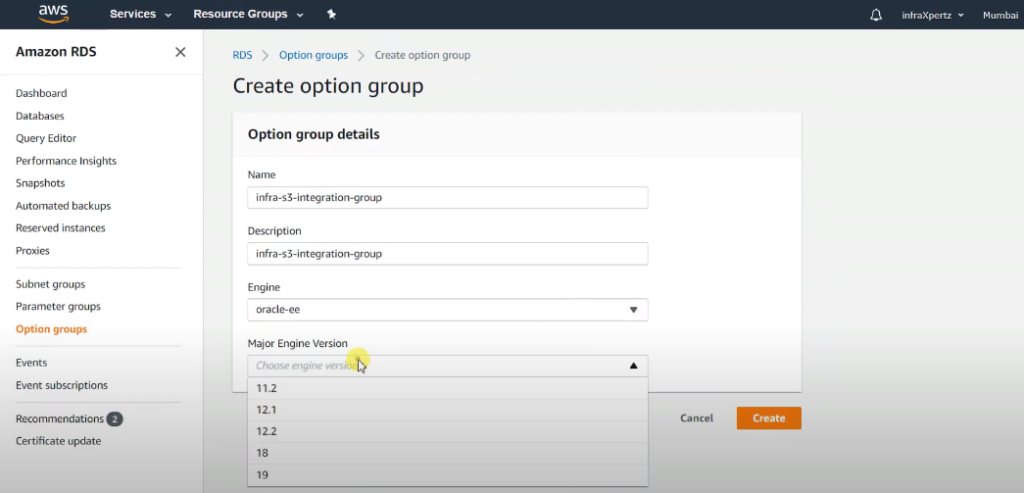
4. In the **Create option group** window, do the following:

a. For **Name**, type a name for the option group that is unique within your AWS account. The name can contain only letters, digits, and hyphens.

b. For **Description**, type a brief description of the option group. The description is used for display purposes.

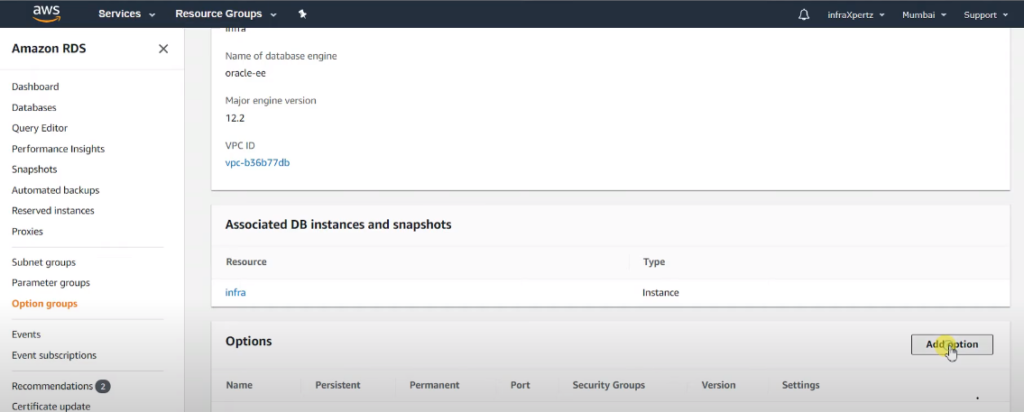
c. For **Engine**, choose the DB engine that you want. The Engine should be the same as your database

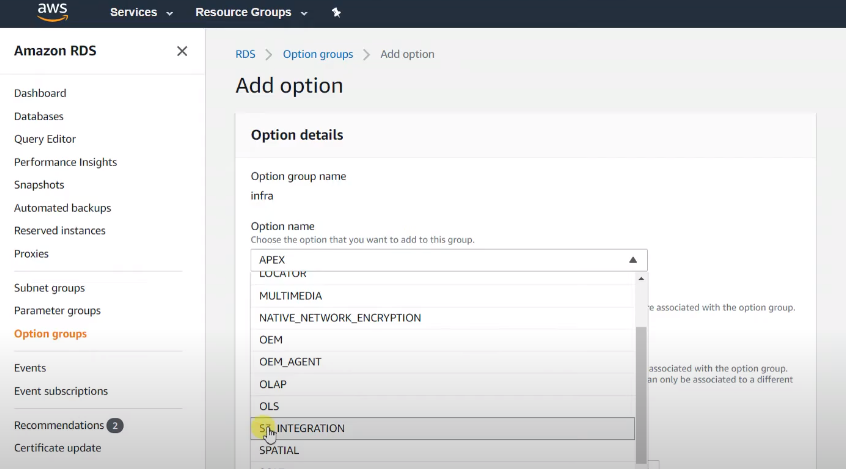
d. For **Major engine version**, choose the major version of the DB engine that you want.

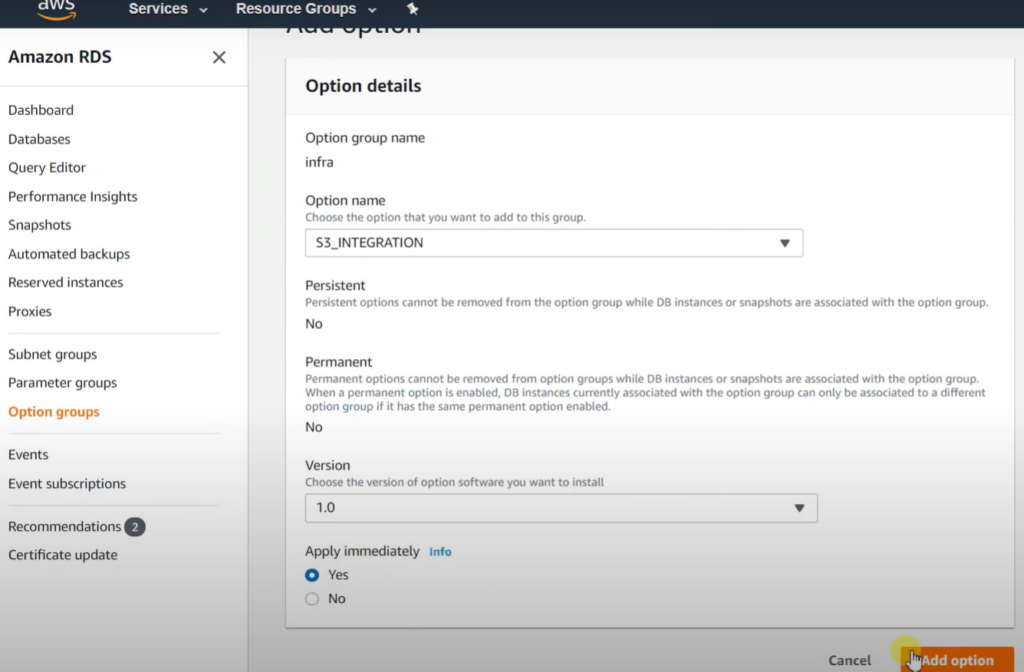


5. To continue, choose **Create**. To cancel the operation instead, choose **Cancel**.

6. Add the S3\_INTEGRATION option to the option group.







This marks the process of AWS RDS with S3 Integration

**Transferring Files Between Amazon RDS for Oracle and an Amazon S3 Bucket**

 Now let us test whether we are able to upload or download using the Integration

You can use Amazon RDS procedures to upload files from an Oracle DB instance to an Amazon S3 bucket. You can also use Amazon RDS procedures to download files from an Amazon S3 bucket to an Oracle DB instance.

**Uploading Files from an Oracle DB Instance to an Amazon S3 Bucket**

To upload files from an Oracle DB instance to an Amazon S3 bucket, use the Amazon RDS procedure rdsadmin.rdsadmin\_s3\_tasks.upload\_to\_s3.

The following example uploads all of the files in the *RDS\_DIRECTORY\_NAME* directory to the Amazon S3 bucket named *your\_bucket\_name :*

SELECT rdsadmin.rdsadmin\_s3\_tasks.upload\_to\_s3(

                    p\_bucket\_name    =>  ‘*your\_bucket\_name*‘,

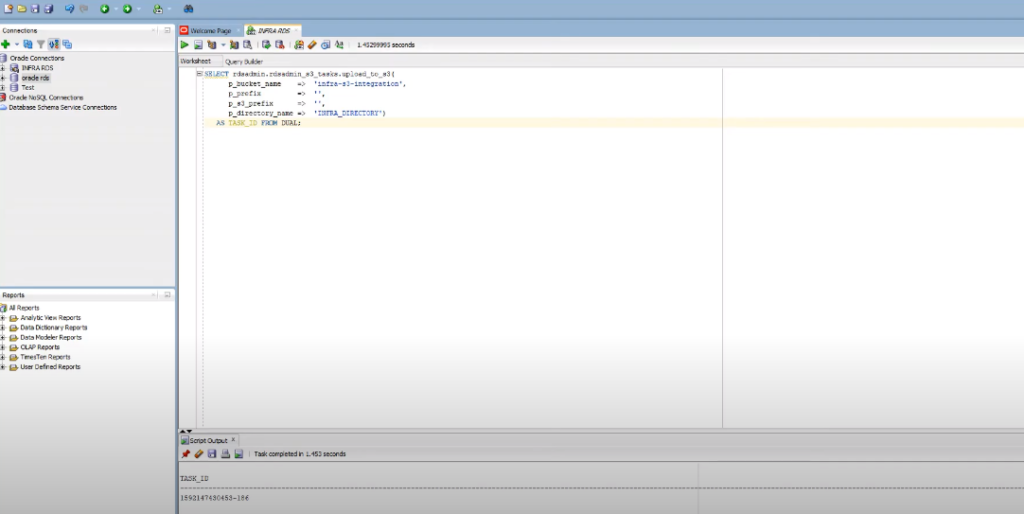
                    p\_prefix         =>  ”,

                    p\_s3\_prefix      =>  ”,

                    p\_directory\_name =>  *‘RDS\_DIRECTORY\_NAME‘*)

AS TASK\_ID FROM DUAL;

Copy the Task ID as mentioned in the script output to check whether the files have been uploaded :



**You can view the result by displaying the task’s output file.**

SELECT text FROM table(rdsadmin.rds\_file\_util.read\_text\_file(‘BDUMP’,‘dbtask-*task-id*.log’))

Replace *task-id* with the task ID returned by the procedure.

The following example uploads all of the files in the *RDS\_DIRECTORY\_NAME* directory to the Amazon S3 bucket named *your\_bucket\_name*. The files are uploaded to a dbfiles folder and ora is added to the beginning of each file name :

SELECT rdsadmin.rdsadmin\_s3\_tasks.upload\_to\_s3(

                    p\_bucket\_name    =>  ‘*your\_bucket\_name*‘,

                    p\_prefix         =>  ”,

                    p\_s3\_prefix      =>  ‘*dbfiles/ora*‘,

                    p\_directory\_name =>  ‘*RDS\_DIRECTORY\_NAME*‘)

AS TASK\_ID FROM DUAL;

**Downloading Files from an Amazon S3 Bucket to an Oracle DB Instance**

To download files from an Amazon S3 bucket to an Oracle DB instance, use the Amazon RDS procedure rdsadmin.rdsadmin\_s3\_tasks.download\_from\_s3.

The return value for the rdsadmin.rdsadmin\_s3\_tasks.download\_from\_s3 procedure is a task ID.

The following example downloads all of the files in the Amazon S3 bucket named *your\_bucket\_name*to the *RDS\_DIRECTORY\_NAME* directory :

SELECT rdsadmin.rdsadmin\_s3\_tasks.download\_from\_s3(

                              p\_bucket\_name    =>  ‘*your\_bucket\_name*‘,

                              p\_directory\_name =>  ‘*RDS\_DIRECTORY\_NAME*‘)

AS TASK\_ID FROM DUAL;

The following example downloads all of the files with the prefix *db* in the Amazon S3 bucket named *your\_bucket\_name*to the *RDS\_DIRECTORY\_NAME* directory :

SELECT rdsadmin.rdsadmin\_s3\_tasks.download\_from\_s3(

                               p\_bucket\_name    =>  ‘*your\_bucket\_name*‘,

                               p\_s3\_prefix      =>  ‘*db*‘,

                               p\_directory\_name =>  ‘*RDS\_DIRECTORY\_NAME*‘)

AS TASK\_ID FROM DUAL;

 To check whether the file has been downloaded from you S3 bucket :

SELECT  \* from table (rdsadmin.rds\_file\_util.read\_text\_file(

            p\_directory => ‘*RDS\_DIRECTORY\_NAME‘,*

*p\_filename   => ‘Downloaded\_file\_name‘));*