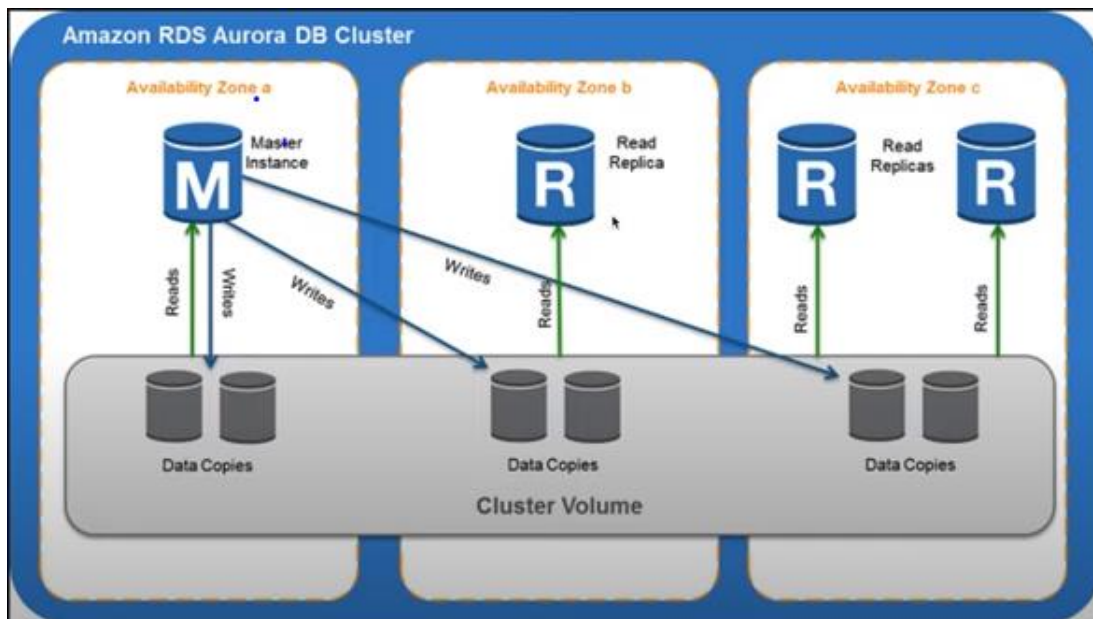


AUTOSCALING IN RDS-AURORA DATABASE CLUSTER

Aurora Database is one of the offerings in Relational Database Service (RDS). AWS highly encourage customers to use Aurora Database due to its High Performance, Scalability, Durability and Highly Available features, but you can opt other Database solutions which are available in AWS such as MySQL, PostgreSQL, MariaDB, Oracle and Microsoft SQL Server.

In this lab, we are going to focus on Aurora Database and see how we can achieve Autoscaling of Database Instances for a Highly Available architecture.



Amazon RDS Aurora Database Cluster includes a Master Instance along with up to 15 Read Replicas. Read Replicas gets deployed in different Availability Zones for High Availability and used for Read operations, while the Master Instance is used for Read as well as for the Write operations.

In this lab, we are going to perform the below tasks:

- Task 1: Create a Security Group to access the Aurora Cluster
- Task 2: Create an RDS-Aurora Cluster
- Task 3: Create Autoscaling for RDS-Aurora Database Cluster
- Task 4: Autoscaling Verification
- Task 5: Scale-In Aurora DB Instances

Task 1: Create a Security Group to access the Aurora Cluster

Login to the AWS Management Console.

Navigate to the EC2 Service, click on Security Groups and click on Create Security Group.

Give the Security Group Name of your choice, select VPC as Custom VPC that we've created in the second lab.

Under Inbound Rules, create a rule for "MYSQL/Aurora" with source as "My IP". Note that we'll later make our RDS-Aurora cluster publicly available and use this Security Group to control the Inbound traffic. Keep the Outbound rule as default.

Basic details

Security group name [Info](#)

Name cannot be edited after creation.

Description [Info](#)

VPC [Info](#)

Inbound rules [Info](#)

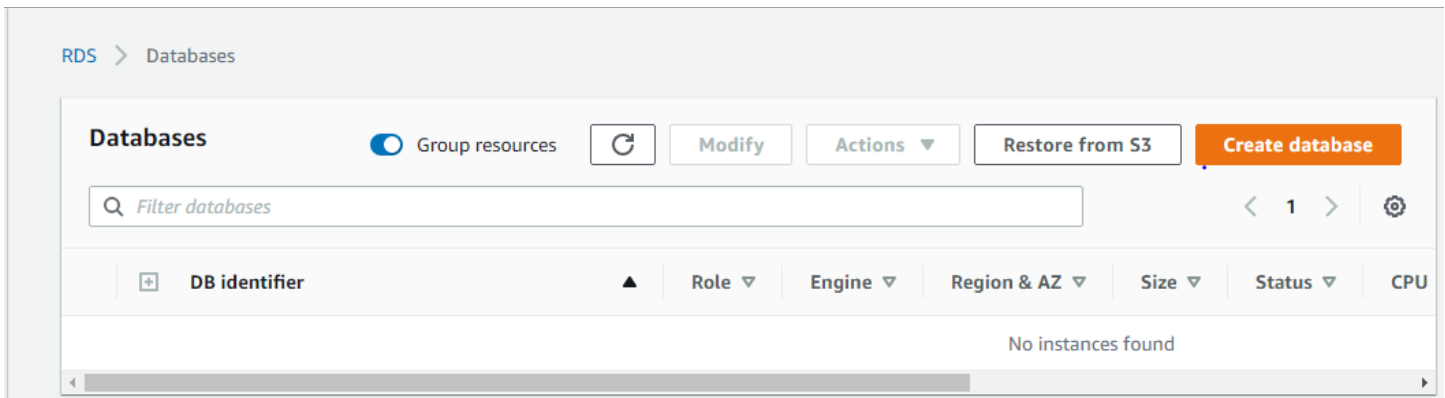
Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
MYSQL/Aurora	TCP	3306	My IP		Delete
			70.123.124.218/32		

Finally, click on Create Security Group.

<input type="checkbox"/>	Security group ID	Security group name	VPC ID
<input type="checkbox"/>	sg-0138fa39c03c2ca04	default	vpc-062814d035612343e
<input type="checkbox"/>	sg-04bd568afe8fa65c4	SG-Windows Servers	vpc-062814d035612343e
<input type="checkbox"/>	sg-0e97cfd310093cb0	no-ingress-sg	vpc-062814d035612343e
➔ <input type="checkbox"/>	sg-0ebe4e8cc2d95786b	Aurora Security Group	vpc-062814d035612343e
<input type="checkbox"/>	sg-0fa905f89ae8dfd9e	SG-Application Load Balancer	vpc-062814d035612343e
<input type="checkbox"/>	sg-431e9d10	default	vpc-a6c288dc

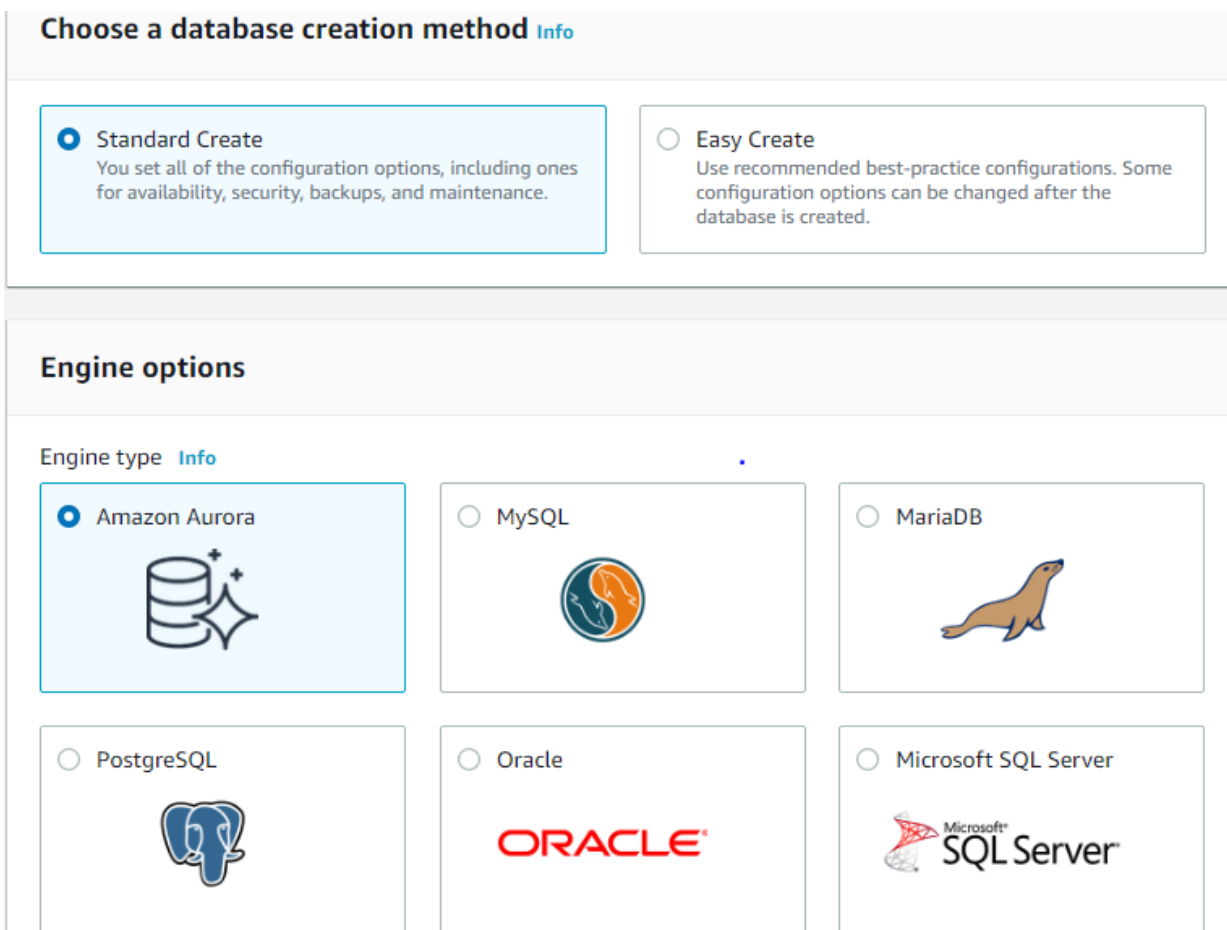
Task 2: Create an RDS-Aurora Cluster

Navigate to Relational Database Service (RDS) and click on Create Database.



Keep the Database Creation method as Standard.

Select Engine Type as Amazon Aurora.



Set the Database Location as Regional. Now, the Read Replicas will be created across multiple Availability Zones within a single Region. This is how you can achieve **High Availability** across Availability Zones for RDS-Aurora Database Instances.

Database Location

☒ **Regional**

You provision your Aurora database in a single AWS Region.

☐ **Global**

You can provision your Aurora database in multiple AWS Regions. Writes in the primary AWS Region are replicated with typical latency of less than 1 sec to secondary AWS Regions.

Keep Database Features as One Writer and Multiple Readers. Under Templates, select Dev/test template.

Database features

☒ **One writer and multiple readers**

Supports multiple reader instances connected to the same storage volume as a single writer instance. This is a good general-purpose option for most workloads.

☐ **One writer and multiple readers - Parallel query**

Improves the performance of analytic queries by pushing processing down to the Aurora storage layer. This is a good option for hybrid transactional/analytic workloads.

☐ **Multiple writers**

Supports multiple writer instances connected to the same storage volume. This is a good option for when continuous writer availability is required.

☐ **Serverless**

- You specify the minimum and maximum amount of resources needed, and Aurora scales the capacity based on database load. This is a good option for intermittent or unpredictable workloads.

Templates

Choose a sample template to meet your use case.

☐ **Production**

Use defaults for high availability and fast, consistent performance.

☒ **Dev/Test**

This instance is intended for development use outside of a production environment.

Give the DB Cluster Identifier Name as Aurora Cluster & set Master Password as “adminadmin”. You can set your preferred DB Cluster Identifier Name and Master Password.

Settings

DB cluster identifier [Info](#)

Type a name for your DB cluster. The name must be unique cross all DB clusters owned by your AWS account in the current AWS Region.

The DB cluster identifier is case-insensitive, but is stored as all lowercase (as in "mydbcluster"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username [Info](#)

Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. First character must be a letter

☐ Auto generate a password

Amazon RDS can generate a password for you, or you can specify your own password

Master password [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), "(double quote) and @ (at sign).

Confirm password [Info](#)

Under DB Instance Size, enable “Include previous generation classes” and select the DB Instance Class as “db.r3.large”. RDS-Aurora Database is not included in Free Tier hence we’ve selected the lowest DB Instance Class.

DB instance size

DB instance class [Info](#)

Choose a DB instance class that meets your processing power and memory requirements. The DB instance class options below are limited to those supported by the engine you selected above.

☒ Memory Optimized classes (includes r and x classes)

☐ Burstable classes (includes t classes)

2 vCPUs 15.25 GiB RAM Not EBS Optimized

☒ Include previous generation classes

Select Create Replica in Different Zone to have Amazon RDS maintain a synchronous standby replica in a different Availability Zone than the DB instance. Amazon RDS will automatically fail over to the standby in the case of a planned or unplanned outage of the primary. This is how we can achieve **High Availability** in Relational Database Service.

Availability & durability

Multi-AZ deployment [Info](#)

- ☒ Create an Aurora Replica/Reader node in a different AZ (recommended for scaled availability)
Creates an Aurora replica for fast failover and high availability.
- ☐ Don't create an Aurora Replica

Under Connectivity, select the Custom VPC.

Connectivity



Virtual Private Cloud (VPC) [Info](#)

VPC that defines the virtual networking environment for this DB cluster.

Custom VPC (vpc-062814d035612343e) ▼

Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change the VPC selection.

► Additional connectivity configuration

Now click on Additional Connectivity Configuration under Connectivity.

In the Lab 4 & 5, we deploy our RDS Instances in Private Subnets by using the Subnet Groups. In this lab, we are going to make the Database Publicly Available and apply the Security Group that we've created in Task Number 1. Make sure to deselect the Default Security Group.

The best practice is to keep the Database Instance in Private Subnets. For this lab, we've made the Database publicly accessible with restriction to Inbound Traffic by applying Security Group. This give you an idea on how to deploy RDS Instances in Public Subnets and apply security with the help of Security Groups.

▼ **Additional connectivity configuration**

Subnet group [Info](#)
DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

Create new DB Subnet Group ▼

Publicly accessible [Info](#)

☒ **Yes**
Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.

☐ **No**
RDS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.

VPC security group
Choose one or more RDS security groups to allow access to your database. Ensure that the security group rules allow incoming traffic from EC2 instances and devices outside your VPC. (Security groups are required for publicly accessible databases.)

☒ **Choose existing**
Choose existing VPC security groups

☐ **Create new** •
Create new VPC security group

Existing VPC security groups

Choose VPC security groups ▼

Aurora Security Group ✕ ←

Database port [Info](#)
TCP/IP port the database will use for application connections.

3306

Keep the Default Settings for the Database Authentication.

Finally, click on Additional Configuration.

Give the Initial Database Name of your choice.

Database options

DB instance identifier [Info](#)

If you do not provide one, a default identifier based on the cluster identifier will be used.

Initial database name [Info](#)

If you do not specify a database name, Amazon RDS does not create a database.

DB cluster parameter group [Info](#)

DB parameter group [Info](#)

Option group [Info](#)

Failover priority

Uncheck “Enable Encryption” and keep rest all settings default.

Encryption

- ☐ **Enable Encryption**
Choose to encrypt the given instance. Master key ids and aliases appear in the list after they have been created using the Key Management Service(KMS) console. [Info](#)

Now click on Create.

RDS-Aurora cluster creation process takes more time than launching an EC2 Instances.

Now the Cluster is ready and DB Instances are Created & Available.

DB identifier	Role	Engine	Region & AZ	Size	Status
auroracluster	Regional	Aurora MySQL	us-east-1	2 instances	Available
auroracluster-instance-1	Writer	Aurora MySQL	us-east-1b	db.r5.large	Available
auroracluster-instance-1-us-east-1a	Reader	Aurora MySQL	us-east-1a	db.r5.large	Available

You'll notice that the Cluster has one DB Instance with role as Writer and other DB Instance with role as Reader. Both the DB Instances are in different Availability Zones and are in same Region.

Click on the Cluster Name (auroracluster). Scroll down till you see Endpoints.

Endpoints (2)

Edit

Delete

Create custom endpoint

Filter endpoint

< 1 >

Endpoint name	Status	Type	Port
<div></div> auroracluster.cluster-c2bo8s6pqhhy.us-east-1.rds.amazonaws.com	Available	Writer	3306
<div></div> auroracluster.cluster-ro-c2bo8s6pqhhy.us-east-1.rds.amazonaws.com	Available	Reader	3306

You'll observe that there are Writer Endpoint and Reader Endpoint.

On left hand side, under Amazon RDS, click on Databases.

Now click on each DB Instances with roles as Writer and Reader and note down the Endpoints.

You'll notice that these Endpoints for Reader & Writer DB Instances are different than the one we noticed at the Cluster Level.

Writer:

DB identifier	Role	Engine	Region & AZ	Size	Status
auroracluster	Regional	Aurora MySQL	us-east-1	2 instances	Available
auroracluster-instance-1	Writer	Aurora MySQL	us-east-1b	db.r5.large	Available
auroracluster-instance-1-us-east-1a	Reader	Aurora MySQL	us-east-1a	db.r5.large	Available

Connectivity & security | Monitoring | Logs & events | Configuration | Maintenance | Tags

Connectivity & security

Endpoint & port	Networking	Security
Endpoint auroracluster-instance-1.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com	Availability zone us-east-1b	VPC security groups Aurora Security Group (sg-0ebe4e8cc2d95786b)

Reader:

DB identifier	Role	Engine	Region & AZ	Size	Status
auroracluster	Regional	Aurora MySQL	us-east-1	2 instances	Available
auroracluster-instance-1	Writer	Aurora MySQL	us-east-1b	db.r5.large	Available
auroracluster-instance-1-us-east-1a	Reader	Aurora MySQL	us-east-1a	db.r5.large	Available

Connectivity & security | Monitoring | Logs & events | Configuration | Maintenance | Tags

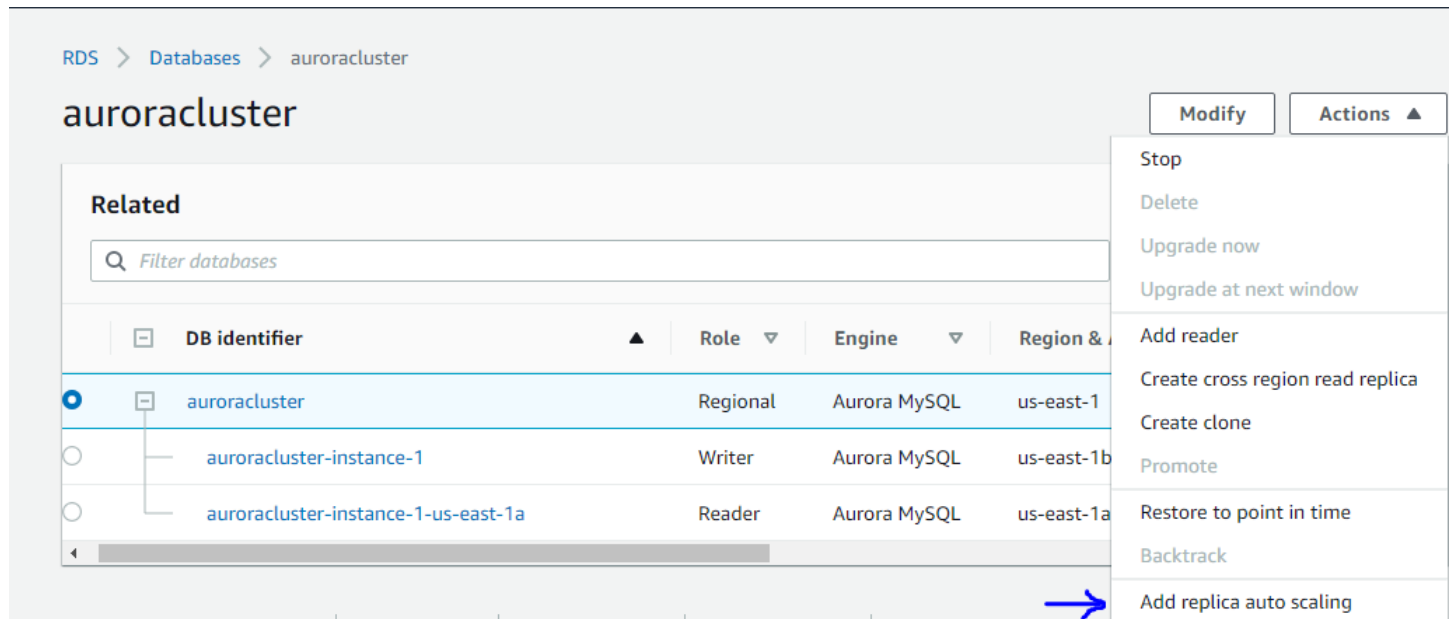
Connectivity & security

Endpoint & port	Networking	Security
Endpoint auroracluster-instance-1-us-east-1a.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com	Availability zone us-east-1a VPC Custom VPC (vpc-062814d035612343e)	VPC security groups Aurora Security Group (sg-0ebe4e8cc2d95786b) (active)

It is always recommended to use the Cluster Level Endpoints of Writer & Reader DB Instances in your Application.

Task 3: Create Autoscaling for RDS-Aurora Database Cluster

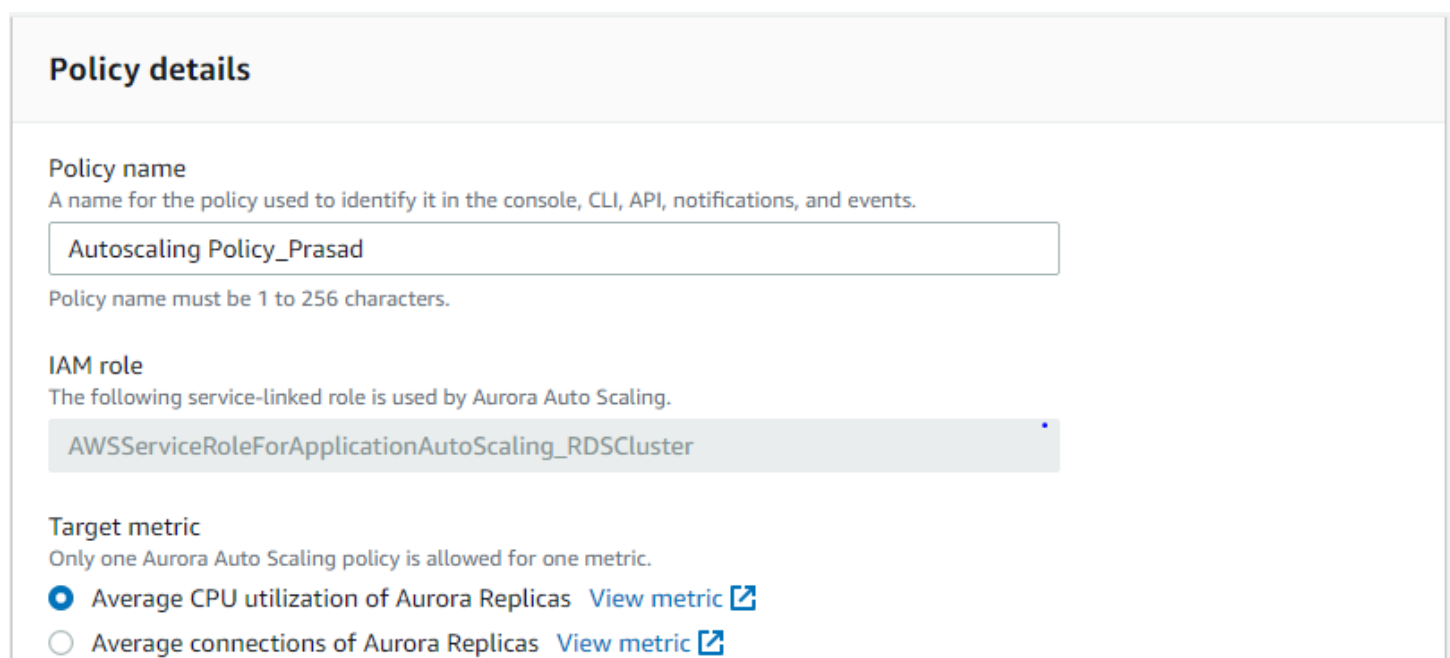
Select the DB Cluster, click on Action and click on Add Replica Auto Scaling.



The screenshot shows the AWS RDS console interface. At the top, the breadcrumb navigation is 'RDS > Databases > auroracluster'. Below this, the title 'auroracluster' is displayed. To the right of the title are two buttons: 'Modify' and 'Actions'. The 'Actions' button is clicked, opening a dropdown menu. The menu contains several options: 'Stop', 'Delete', 'Upgrade now', 'Upgrade at next window', 'Add reader', 'Create cross region read replica', 'Create clone', 'Promote', 'Restore to point in time', 'Backtrack', and 'Add replica auto scaling'. A blue arrow points to the 'Add replica auto scaling' option. In the background, a table titled 'Related' is visible, showing a list of database instances. The first instance, 'auroracluster', is selected and highlighted in blue. It has a role of 'Regional', engine of 'Aurora MySQL', and is located in the 'us-east-1' region. Below it are two other instances: 'auroracluster-instance-1' (Writer role) and 'auroracluster-instance-1-us-east-1a' (Reader role).

DB identifier	Role	Engine	Region &
auroracluster	Regional	Aurora MySQL	us-east-1
auroracluster-instance-1	Writer	Aurora MySQL	us-east-1b
auroracluster-instance-1-us-east-1a	Reader	Aurora MySQL	us-east-1a

Give the Policy Name of your choice. Select the Target Metric as per your requirement. In our case we will select the Target Metric as Average CPU Utilization of Aurora Replicas.



The screenshot shows the 'Policy details' form in the AWS console. The form has three main sections: 'Policy name', 'IAM role', and 'Target metric'. The 'Policy name' section has a text input field containing 'Autoscaling Policy_Prasad' and a note that the policy name must be 1 to 256 characters. The 'IAM role' section shows a dropdown menu with 'AWSServiceRoleForApplicationAutoScaling_RDSCluster' selected. The 'Target metric' section has two radio button options: 'Average CPU utilization of Aurora Replicas' (which is selected) and 'Average connections of Aurora Replicas'. Both options have a 'View metric' link next to them.

Policy details

Policy name
A name for the policy used to identify it in the console, CLI, API, notifications, and events.

Autoscaling Policy_Prasad

Policy name must be 1 to 256 characters.

IAM role
The following service-linked role is used by Aurora Auto Scaling.

AWSServiceRoleForApplicationAutoScaling_RDSCluster

Target metric
Only one Aurora Auto Scaling policy is allowed for one metric.

☒ Average CPU utilization of Aurora Replicas [View metric](#)

☐ Average connections of Aurora Replicas [View metric](#)

Set the Target Value as 2%. It means if the CPU Utilization of DB Instances goes above the 2% then add specified number of Read Replicas to your RDS-Aurora Cluster.

Target value


Specify the desired value for the selected metric. Aurora Replicas will be added or removed to keep the metric close to the specified value.

 %**► Additional configuration**

Under Cluster Capacity Details, you'll notice that the Maximum Capacity as 15. Aurora is the only database which supports up to 15 Read Replicas. Hence AWS is highly recommending its customer to make best out of it.

Cluster capacity details
Configure the minimum and maximum number of Aurora Replicas you want Aurora Auto Scaling to maintain.


Minimum capacity
Specify the minimum number of Aurora Replicas to maintain.
 Aurora Replicas

Maximum capacity
Specify the maximum number of Aurora Replicas to maintain. Up to 15 Aurora Replicas are supported.
 Aurora Replicas

For our case, we'll set the Maximum Capacity as 5. Hence if the average CPU Utilization of DB Instances or Read Replicas goes above 2%, then the Autoscaling policy will launch 5 Read Replicas in the Cluster.

Cluster capacity details
Configure the minimum and maximum number of Aurora Replicas you want Aurora Auto Scaling to maintain.

Minimum capacity
Specify the minimum number of Aurora Replicas to maintain.
 Aurora Replicas

Maximum capacity
Specify the maximum number of Aurora Replicas to maintain. Up to 15 Aurora Replicas are supported.
 Aurora Replicas 

Finally, click on Add Policy.

To see your defined autoscaling policy, select your Cluster and click on Logs & Events. There you'll notice your Autoscaling policy.

Connectivity & security	Monitoring	Logs & events	Configuration	Maintenance & backups	Tags
Auto scaling policies (1)					
<input type="text" value="Filter name"/> < 1 >					
Name	Scaling action	Target metric	Target value		
Autoscaling Policy_Prasad	Add or remove Aurora Replicas	Average CPU utilization of Aurora Replicas	2 Percent		

Wait for a while, grab a cup of tea. This is now going to take 15-30 minutes.

Task 4: Autoscaling Verification

You'll now notice that the Cluster Includes total 5 Read Replicas along with a Master DB Instance with Writer role.

Related				
<input type="text" value="Filter databases"/>				
<input type="checkbox"/> DB identifier	Role	Engine	Region & AZ	
<input type="radio"/> auroracluster	Regional	Aurora MySQL	us-east-1	
<input type="radio"/> auroracluster-instance-1	Writer	Aurora MySQL	us-east-1b	
<input checked="" type="radio"/> application-autoscaling-316d1239-1283-4d61-a93a-5c02596f4f12	Reader	Aurora MySQL	us-east-1b	
<input type="radio"/> application-autoscaling-9da2516a-5c43-42e5-94c0-f463df2b0b7d	Reader	Aurora MySQL	us-east-1a	
<input type="radio"/> application-autoscaling-b50a68f1-1fb3-4012-8436-98a60ebb3972	Reader	Aurora MySQL	us-east-1a	
<input type="radio"/> application-autoscaling-be8f5097-f9f8-4e3d-8750-a14d3923c2ec	Reader	Aurora MySQL	us-east-1b	
<input type="radio"/> auroracluster-instance-1-us-east-1a	Reader	Aurora MySQL	us-east-1a	

All the Read Replicas are in different Availability Zones, which makes our architecture Highly Available. Replicas which includes AUTOSCALING in its name are launched by the Autoscaling Policy.

Task 5: Scale-In Aurora DB Instances

Now let's Scale-In Read Replicas.

Navigate to the Scaling Policy, and now set the Target Value as 75%.

Policy details

Policy name
A name for the policy used to identify it in the console, CLI, API, notifications, and events.

Autoscaling Policy_Prasad

Policy name must be 1 to 256 characters.

IAM role
The following service-linked role is used by Aurora Auto Scaling.

AWSServiceRoleForApplicationAutoScaling_RDSCluster

Target metric
Only one Aurora Auto Scaling policy is allowed for one metric.

☒ Average CPU utilization of Aurora Replicas [View metric](#)

☐ Average connections of Aurora Replicas [View metric](#)

Target value
Specify the desired value for the selected metric. Aurora Replicas will be added or removed to keep the metric close to the specified value.

75 %

► **Additional configuration**

Set the Maximum limit to 2 Read Replicas.

Cluster capacity details
Capacity values specified below apply to all the Aurora Auto Scaling policies for the DB cluster.

Minimum capacity
Specify the minimum number of Aurora Replicas to maintain.

1 Aurora Replicas

Maximum capacity
Specify the maximum number of Aurora Replicas to maintain. Up to 15 Aurora Replicas are supported.

2 Aurora Replicas

⚠ Changes to the capacity values will be applied to all the Auto Scaling policies for this DB cluster.

It means if the average CPU Utilization of Read Replicas goes beyond 75 %, then the autoscaling policy maintains two Read Replicas in the Cluster.

Now if you look at our Aurora Cluster, you'll notice that Read Replicas which were launched earlier now starts getting DELETED.

In the below snapshot, you'll notice that the Cluster has one Master DB Instance with role as Writer along with the 2 Read Replicas. Note that out of these 2 Read Replicas, one was an initial Read Replica which was launched during the Cluster creation, while the another Read Replica was launched by the Autoscaling Policy.

auroracluster Modify Actions ▼

Related

DB identifier	Role	Engine	Region & AZ	Size	Status
auroracluster	Regional	Aurora MySQL	us-east-1	6 instances	Available
auroracluster-instance-1	Writer	Aurora MySQL	us-east-1b	db.r5.large	Available
auroracluster-autoscaling-316d1239-1283-4d61-a93a-5c02596f4f12	Reader	Aurora MySQL	us-east-1b	db.r5.large	Deleting
auroracluster-autoscaling-9da2516a-5c43-42e5-94c0-f463df2b0b7d	Reader	Aurora MySQL	us-east-1a	db.r5.large	Deleting
auroracluster-autoscaling-b50a68f1-1fb3-4012-8436-98a60ebb3972	Reader	Aurora MySQL	us-east-1a	db.r5.large	Deleting
auroracluster-autoscaling-be8f5097-f9f8-4e3d-8750-a14d3923c2ec	Reader	Aurora MySQL	us-east-1b	db.r5.large	Available
auroracluster-instance-1-us-east-1a	Reader	Aurora MySQL	us-east-1a	db.r5.large	Available

After deletion of the additional Read Replicas, your Cluster will maintain one Master DB Instance & one Read Replica which was launched during the Cluster creation. In-case if the average CPU utilization goes above 75 %, then the Autoscaling Policy will add one more Read Replica in the Cluster which would be in another Availability Zone.

Databases Group resources Refresh Modify Actions ▼ Restore from S3 Create database

DB identifier	Role	Engine	Region & AZ	Size	Status
auroracluster	Regional	Aurora MySQL	us-east-1	2 instances	Available
auroracluster-instance-1	Writer	Aurora MySQL	us-east-1b	db.r5.large	Available
auroracluster-instance-1-us-east-1a	Reader	Aurora MySQL	us-east-1a	db.r5.large	Available

This completes the lab on Autoscaling in RDS-Aurora Database.

If you have any questions, contact me on pbhavsar@smu.edu.