

Automate RDS Snapshots

In this Lab, we are going to design a Highly Available architecture wherein Snapshots of RDS Instances gets copied to another Region using Lambda function. The entire process doesn't require any manual intervention as Lambda Function automates the tasks. Since, we've Snapshot of RDS Instances available in another Region, there would be less application downtime in-case of primary region failure.

Below is the architecture design:



When Snapshot of an RDS Instance is taken, it triggers the CloudWatch Rule which then triggers the Lambda Function. Lambda Function runs a script to copy this Snapshot to another Region to maintain a Highly Availability.

List of Tasks:

- Task 1: Create IAM Policy
- Task 2: Create IAM Role
- Task 3: Configure Lambda Function
- Task 4: Configure CloudWatch Rule
- Task 5: Verify the Automation

Task 1: Create IAM Policy

Login to the AWS Management Console.

Navigate to the IAM Service and on left-hand side click on Policies.

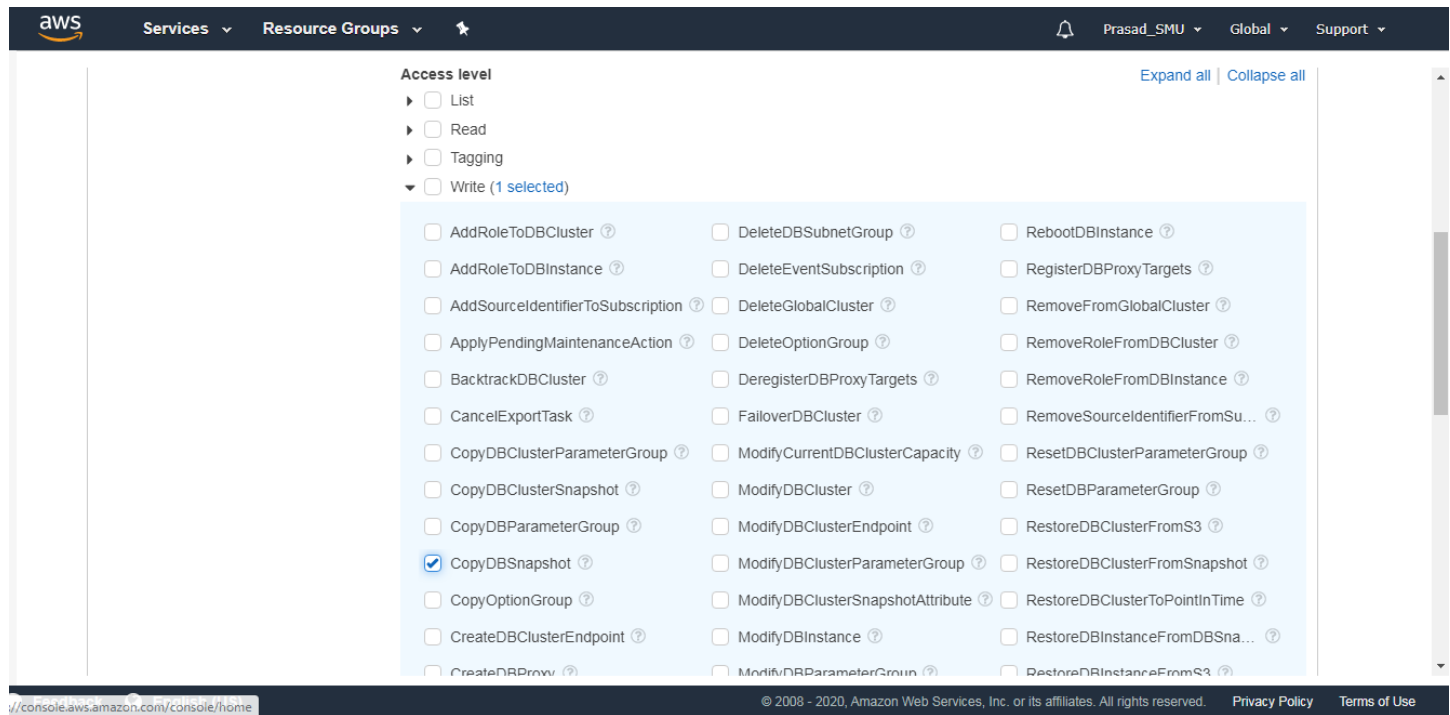
The screenshot shows the AWS IAM console dashboard. The left-hand navigation pane is expanded, showing the 'Identity and Access Management (IAM)' section. The main content area displays the 'Welcome to Identity and Access Management' page. It includes a sign-in link for IAM users, a list of IAM resources (Users: 2, Roles: 10, Groups: 1, Identity Providers: 0, Customer Managed Policies: 2), and a 'Security Status' section with a progress bar indicating 5 out of 5 complete. The 'Security Status' section lists five tasks: 'Delete your root access keys', 'Activate MFA on your root account', 'Create individual IAM users', 'Use groups to assign permissions', and 'Apply an IAM password policy'. On the right side, there is a 'Feature Spotlight' section with a video player titled 'Introduction to AWS IAM' and an 'Additional Information' section with links to 'IAM best practices', 'IAM documentation', 'Web Identity Federation Playground', 'Policy Simulator', and 'Videos, IAM release history and additional resources'.

Click on Create Policy.

Under Service, select the **Relational Database Service (RDS)**.

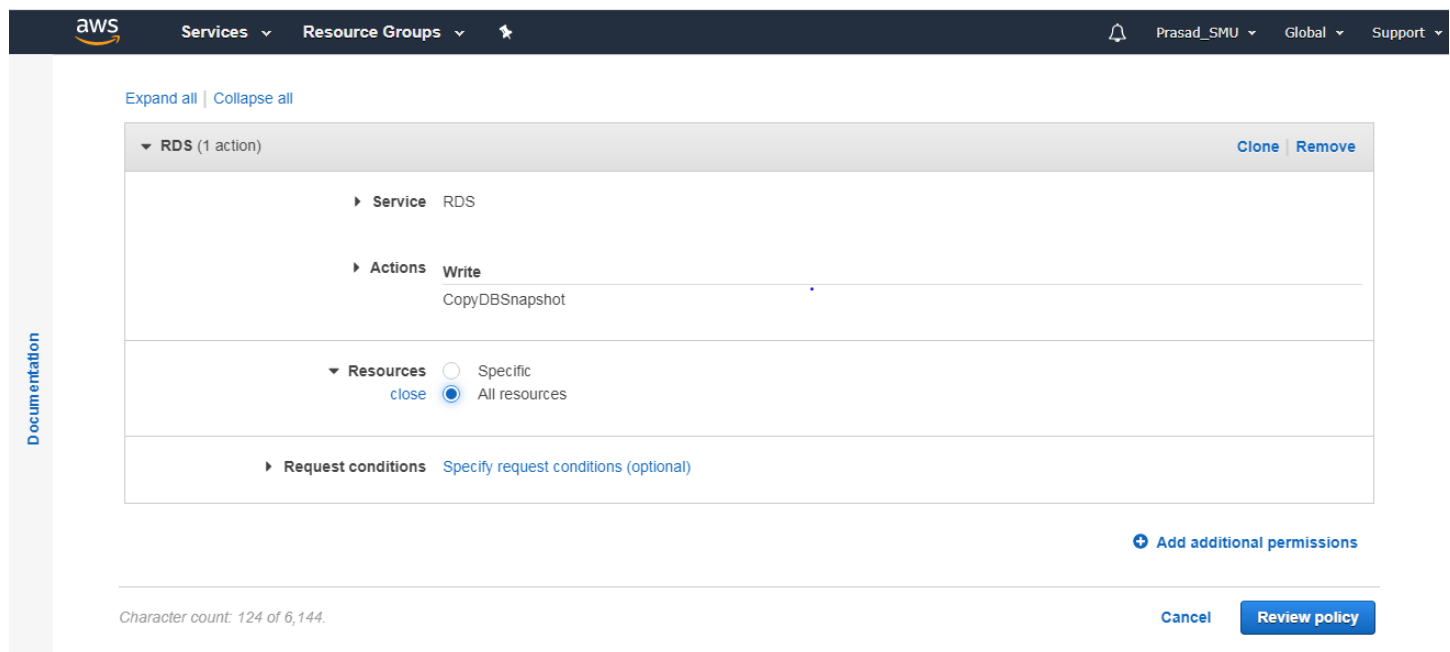
The screenshot shows a portion of the AWS Services list. The 'RDS (1 action)' service is highlighted, and a warning icon with the text '1 warning' is visible next to it. Below the service name, the text 'Service RDS' is displayed.

Expand Access Level and select **CopyDBSnapshot**.



The screenshot shows the AWS IAM console interface. At the top, there's a navigation bar with the AWS logo, 'Services', 'Resource Groups', and a star icon. On the right, there's a user profile 'Prasad_SMU', 'Global', and 'Support'. The main content area is titled 'Access level' with links for 'Expand all' and 'Collapse all'. Under 'Access level', there are four options: 'List', 'Read', 'Tagging', and 'Write (1 selected)'. The 'Write' option is expanded, showing a list of permissions. 'CopyDBSnapshot' is checked, while others are unchecked. The footer shows the URL 'v/console.aws.amazon.com/console/home' and copyright information '© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.' along with 'Privacy Policy' and 'Terms of Use' links.

Also, select the **All Resources**.



The screenshot shows the AWS IAM console interface for configuring a policy. The 'RDS (1 action)' policy is selected, with 'Clone' and 'Remove' options. The 'Service' is 'RDS'. The 'Actions' section is set to 'Write', with 'CopyDBSnapshot' listed. The 'Resources' section is set to 'All resources' (selected with a radio button), with a 'close' link. The 'Request conditions' section is set to 'Specify request conditions (optional)'. At the bottom, there's a 'Character count: 124 of 6,144.' and 'Cancel' and 'Review policy' buttons. A 'Documentation' link is visible on the left sidebar.

Click on Review Policy.

Give Policy Name and Description as per your choice and click on Create Policy.

Review policy

Name* CopyRDSsnapshots
Use alphanumeric and '+,=, @, _' characters. Maximum 128 characters.

Description Permissions to copy RDS Automated Snapshots.
Maximum 1000 characters. Use alphanumeric and '+,=, @, _' characters.

Summary

Service	Access level	Resource	Request condition
Allow (1 of 228 services) Show remaining 227			
RDS	Limited: Write	All resources	None

* Required

Cancel Previous Create policy

Task 2: Create IAM Role

Navigate to IAM Service, click on Roles and click on Create Role.

Identity and Access Management (IAM)

Dashboard

Access management

- Groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings

Access reports

- Access analyzer
- Archive rules
- Analyzers
- Settings

Credential report

Organization activity

Roles

What are IAM roles?

IAM roles are a secure way to grant permissions to entities that you trust. Examples of entities include the following:

- IAM user in another account
- Application code running on an EC2 instance that needs to perform actions on AWS resources
- An AWS service that needs to act on resources in your account to provide its features
- Users from a corporate directory who use identity federation with SAML

IAM roles issue keys that are valid for short durations, making them a more secure way to grant access.

Additional resources:

- [IAM Roles FAQ](#)
- [IAM Roles Documentation](#)
- [Tutorial: Setting Up Cross Account Access](#)
- [Common Scenarios for Roles](#)

Create role Delete role

Showing 10 results

Choose the use case as **Lambda** and click Next.

Services ▾ **Resource Groups** ▾ ⭐

Prasad_SMU ▾ Glo

AWS service
EC2, Lambda and others

Another AWS account
Belonging to you or 3rd party

Web identity
Cognito or any OpenID provider

SAML 2.0 federation
Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose a use case

Common use cases

EC2
Allows EC2 instances to call AWS services on your behalf.

Lambda
Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

API Gateway	CodeDeploy	EMR	KMS	RoboMaker
AWS Backup	CodeGuru	ElastiCache	Kinesis	S3
AWS Chatbot	CodeStar Notifications	Elastic Beanstalk	Lambda	SMS
AWS Support	Comprehend	Elastic Container Service	Lex	SNS
Amplify	Config	Elastic Transcoder	License Manager	SWF

* Required

Cancel **Next: Permissions**

Now select the **AWSLambdaBasicExecutionRole** to give Lambda access to the CloudWatch.

Services ▾ **Resource Groups** ▾ ⭐

Prasad_SMU ▾ Global ▾

Choose one or more policies to attach to your new role.

Create policy

Filter policies ▾ Showing 13 results

	Policy name ▾	Used as
<input checked="" type="checkbox"/>	AWSLambdaBasicExecutionRole	None
<input type="checkbox"/>	AWSLambdaBasicExecutionRole-a69274c2-1a4f-443d-b7de-b00d893d9c2f	Permissions policy (1)
<input type="checkbox"/>	AWSLambdaDynamoDBExecutionRole	None
<input type="checkbox"/>	AWSLambdaENIManagementAccess	None
<input type="checkbox"/>	AWSLambdaExecute	None
<input type="checkbox"/>	AWSLambdaFullAccess	None
<input type="checkbox"/>	AWSLambdaInvocation-DynamoDB	None
<input type="checkbox"/>	AWSLambdaKinesisExecutionRole	None

▸ Set permissions boundary

* Required

Cancel Previous **Next: Tags**

Also, select the new Policy which you've created in Task1.

The screenshot shows the AWS IAM console interface. At the top, the navigation bar includes the AWS logo, 'Services', 'Resource Groups', a star icon, a notification bell, 'Prasad_SMU', and 'Global'. Below the navigation bar, a message states: 'Choose one or more policies to attach to your new role.' There is a 'Create policy' button on the left and a refresh icon on the right. A search bar labeled 'Filter policies' contains the text 'copyrds'. To the right of the search bar, it says 'Showing 1 result'. Below the search bar is a table with two columns: 'Policy name' and 'Used as'. The table contains one row with a checked checkbox, the policy name 'CopyRDS Snapshots', and the value 'None'.

	Policy name	Used as
<input checked="" type="checkbox"/>	CopyRDS Snapshots	None

Add Tags as per your choice.

The screenshot shows the 'Add tags' step in the AWS IAM console. The navigation bar is the same as in the previous screenshot. Below it, the title 'Create role' is followed by four numbered steps: 1, 2, 3 (which is highlighted in blue), and 4. The main heading is 'Add tags (optional)'. Below this, a paragraph explains: 'IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this role. [Learn more](#)'. Below the text is a table with three columns: 'Key', 'Value (optional)', and 'Remove'. The first row has 'Desc' in the 'Key' column, 'Automated Snapshot Copy' in the 'Value' column, and a remove icon (an 'x' in a square) in the 'Remove' column. Below this row is a row with 'Add new key' in the 'Key' column and an empty text box in the 'Value' column. Below the table, it says 'You can add 49 more tags.' At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next: Review' (which is highlighted in blue).

Key	Value (optional)	Remove
Desc	Automated Snapshot Copy	
Add new key		

Click on Next: Review.

Give the Role Name as per your choice and click on Create Role.

The screenshot shows the 'Create role' wizard in the AWS IAM console, specifically the 'Review' step (Step 4 of 4). The role name is 'Lambda_CopyRDS Snapshots'. The role description is 'Allows Lambda functions to call AWS services on your behalf.' The trusted entities are 'AWS service: lambda.amazonaws.com'. The policies attached are 'AWSLambdaBasicExecutionRole' and 'CopyRDS Snapshots'. The permissions boundary is 'Permissions boundary is not set'. At the bottom, there are buttons for 'Cancel', 'Previous', and 'Create role'.

aws Services Resource Groups

Create role

1 2 3 4

Review

Provide the required information below and review this role before you create it.

Role name* Lambda_CopyRDS Snapshots

Use alphanumeric and '+-._@-' characters. Maximum 64 characters.

Role description Allows Lambda functions to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+-._@-' characters.

Trusted entities AWS service: lambda.amazonaws.com

Policies AWSLambdaBasicExecutionRole CopyRDS Snapshots

Permissions boundary Permissions boundary is not set

* Required Cancel Previous Create role

Task 3: Configure Lambda Function

Navigate to Lambda Service and click on Create Function.

The screenshot shows the 'Functions' page in the AWS Lambda console. The left sidebar shows the navigation menu with 'Functions' highlighted. The main content area shows a list of functions with columns for 'Function name', 'Description', 'Runtime', 'Code size', and 'Last modified'. There is a search bar and a 'Create function' button.

aws Services Resource Groups

Prasad_SMU N. Virginia Support

AWS Lambda

Lambda > Functions

Functions (5)

Filter by tags and attributes or search by keyword

Function name Description Runtime Code size Last modified

Create function

Give the Function Name as per your choice, select the Runtime as Python 3.7 and select the existing role as the role which you created in Task 2. Click on Create Function.

Function name
Enter a name that describes the purpose of your function.
CopyAutomatedSnapshotsCrossRegion

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function.
Python 3.7

Permissions [Info](#)
Lambda will create an execution role with permission to upload logs to Amazon CloudWatch Logs. You can configure and modify permissions further when you add triggers.

▼ **Choose or create an execution role**

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☐ Create a new role with basic Lambda permissions

☒ Use an existing role

☐ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

Lambda_CopyRDS Snapshots

[View the Lambda_CopyRDS Snapshots role on the IAM console.](#)

Under Function Code, paste the Python Code that I've provided. In Line 4, modify the Destination Region as per your accordance.

CopyAutomatedSnapshotsCrossRegion Throttle Qualifiers Actions Select a test event Test Save

File Edit Find View Go Tools Window Save Test

```

1 import json
2 import boto3
3
4 destinationRegion = "us-west-1"
5
6 def lambda_handler(event, context):
7
8     sourceRegion = event['region']
9
10    rds = boto3.client('rds', region_name=destinationRegion)
11
12    #Build the Snapshot ARN - Always use the ARN when copying snapshots across region.
13    sourceSnapshotARN = event['detail']['SourceArn']
14    sourceSnapshotARN = sourceSnapshotARN.replace("db:", "snapshot:")
15
16    #build a new snapshot name
17    sourceSnapshotIdentifier = event['detail']['SourceIdentifier']
18    targetSnapshotIdentifier = "{}-ManualCopy".format(sourceSnapshotIdentifier)
19    targetSnapshotIdentifier = targetSnapshotIdentifier.replace(":", "-")
20
21    #Execute copy
22    try:
23        copy = rds.copy_db_snapshot(SourceDBSnapshotIdentifier=sourceSnapshotARN, TargetDBSnapshotIdentifier=targetSnapshotIdentifier)
24        print("Started Copy of Snapshot {0} in {2} to {1} in {3} ".format(sourceSnapshotIdentifier, targetSnapshotIdentifier, sourceRegion, destinationRegion))
25
26    except ClientError as ex:
27        if ex.response['Error']['Code'] == 'DBSnapshotAlreadyExists':
28            print("Snapshot {0} already exist".format(targetSnapshotIdentifier))
29        else:
30            print("ERROR: {0}".format(ex.response['Error']['Code']))

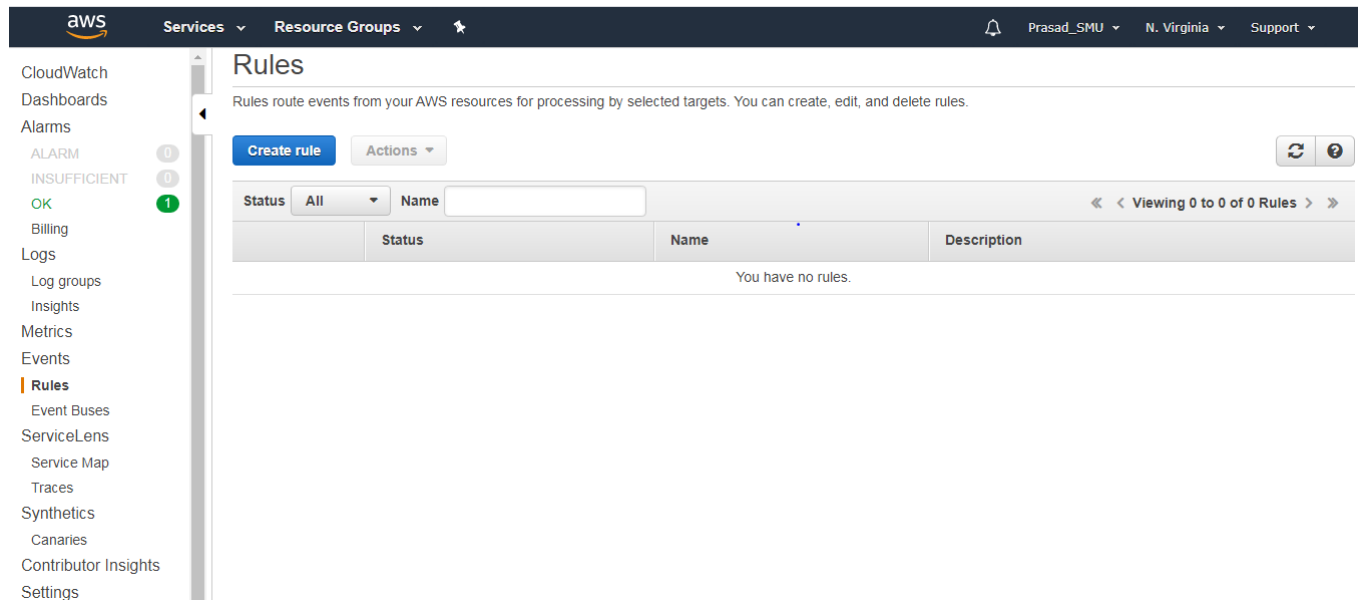
```

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Click on SAVE.

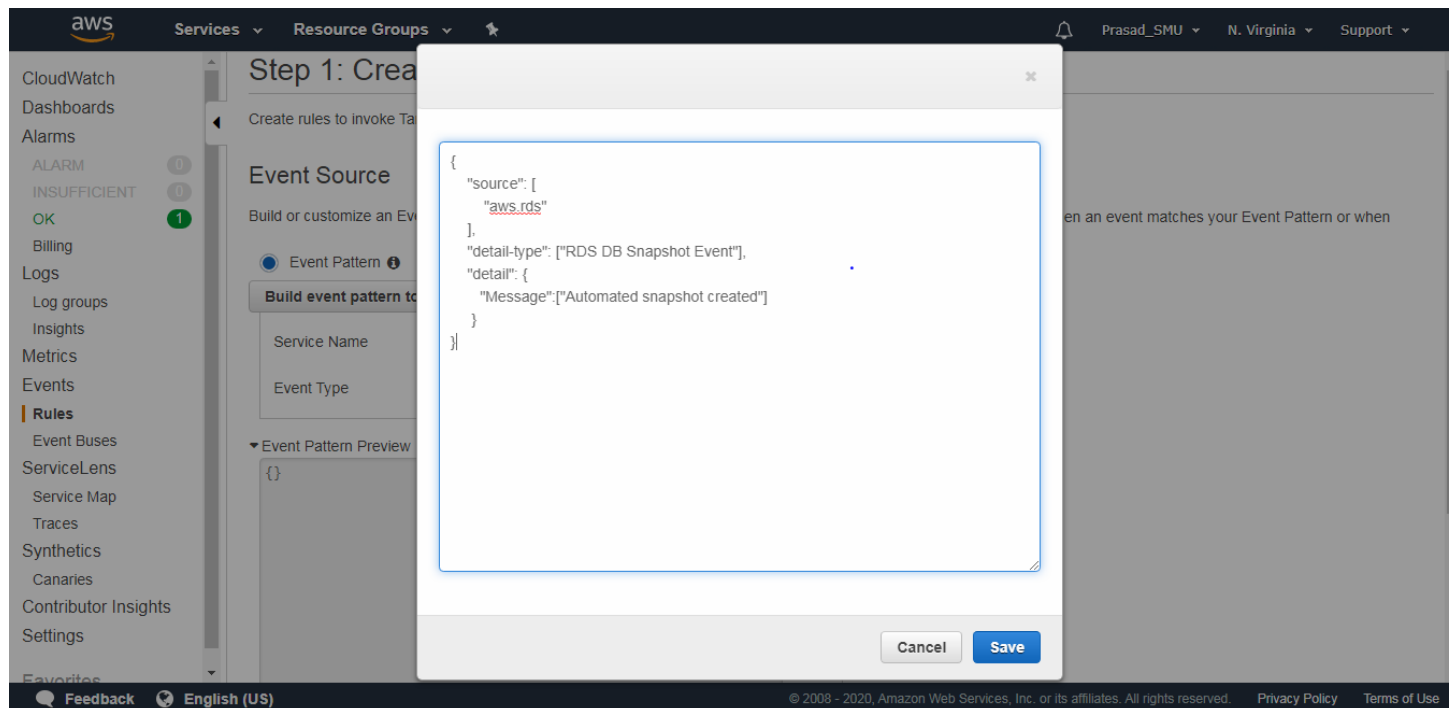
Task 4: Configure CloudWatch Rule

Navigate to the CloudWatch Service, Click on Rules and click on Create Rule.



The screenshot shows the AWS CloudWatch console. The left sidebar contains navigation links for CloudWatch, Dashboards, Alarms, Billing, Logs, Metrics, Events, Rules (highlighted), ServiceLens, Service Map, Traces, Synthetics, Canaries, Contributor Insights, and Settings. The main content area is titled 'Rules' and includes a 'Create rule' button and an 'Actions' dropdown. Below this is a table with columns 'Status', 'Name', and 'Description'. The table is empty, displaying the message 'You have no rules.' The top navigation bar shows the user is logged in as 'Prasad_SMU' in the 'N. Virginia' region.

In the Event Pattern Preview, click on EDIT and paste the filter code which I've provided.



The screenshot shows the 'Step 1: Create Rule' wizard in the AWS CloudWatch console. The 'Event Source' section is selected, and the 'Event Pattern Preview' section is visible. The 'Event Pattern Preview' section shows a JSON filter code that has been pasted into the 'Event Pattern' field. The filter code is:

```
{
  "source": [
    "aws.rds"
  ],
  "detail-type": ["RDS DB Snapshot Event"],
  "detail": {
    "Message": ["Automated snapshot created"]
  }
}
```

The 'Event Pattern Preview' section also includes a 'Build event pattern to match' button. The 'Event Source' section includes fields for 'Service Name' and 'Event Type'. The 'Event Pattern Preview' section includes a 'Filter' field. The bottom of the screen shows the 'Feedback' and 'English (US)' links, along with the copyright notice: '© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use'.

Expand the show sample events, make sure to match the highlighted pattern.

The screenshot shows the AWS CloudWatch console. On the left sidebar, the 'Rules' section is highlighted. The main area displays a sample event under the heading 'Show sample event(s)'. The event is a JSON object with the following details:

```

{
  "version": "0",
  "id": "844e2571-85d4-695f-b930-0153b71dcb42",
  "detail-type": "RDS DB Snapshot Event",
  "source": "aws.rds",
  "account": "123456789012",
  "time": "2018-10-06T12:26:13Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:rds:us-east-1:123456789012:db:mysql-instance-2018-10-06-12-24"
  ],
  "detail": {
    "EventCategories": [
      "creation"
    ],
    "SourceType": "SNAPSHOT",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:mysql-instance-2018-10-06-12-24",
    "Date": "2018-10-06T12:26:13.882Z",
    "SourceIdentifier": "rds:mysql-instance-2018-10-06-12-24",
    "Message": "Automated snapshot created"
  }
}

```

At the bottom of the event viewer, there is a 'Required' section with a 'Cancel' button and a 'Configure details' button.

On the Right-Hand side, under Targets, click on Add Targets and select the Lambda Function which you've created in Task 3.

The screenshot shows the 'Step 1: Create rule' wizard in the AWS CloudWatch console. The main area is divided into two sections: 'Event Source' and 'Targets'.

Event Source: The 'Event Pattern' radio button is selected. Below it, the 'Build custom event pattern' dropdown is open, showing a JSON event pattern:

```

{
  "source": [
    "aws.rds"
  ],
  "detail-type": ["RDS DB Snapshot Event"],
  "detail": {
    "Message": ["Automated snapshot created"]
  }
}

```

Targets: The 'Lambda function' dropdown is selected. Below it, the 'Function*' dropdown is open, showing the selected function: 'CopyAutomatedSnapshotsCrossRegion'. There are links to 'Configure version/alias' and 'Configure input'. At the bottom, there is an 'Add target*' button.

Click Next.

Give CloudWatch Rule Name and Description as per your accordance and create on Create Rule.

aws Services Resource Groups

CloudWatch Dashboards Alarms 0 0 1 Billing Logs Log groups Insights Metrics Events Rules

Step 2: Configure rule details

Rule definition

Name* InvokeOnRDSSnapshot

Description Execute on the creation of RDS Snapshot.

State ☒ Enabled

CloudWatch Events will add necessary permissions for target(s) so they can be invoked when this rule is triggered.

* Required

Cancel Back Create rule

Comeback to Lambda Function now, you can see that CloudWatch Rule has been added as a trigger to Lambda Function.

aws Services Resource Groups

Lambda > Functions > CopyAutomatedSnapshotsCrossRegion ARN - arn:aws:lambda:us-east-1:616399057

CopyAutomatedSnapshotsCrossR...

Throttle Qualifiers Actions

Configuration Permissions Monitoring

▼ Designer

CopyAutomatedSnapshotsCrossRegion

Layers (0)

CloudWatch Events/EventBridge

+ Add trigger

Click on CloudWatch Events/EventBridge, you'll notice the same filter code.

The screenshot shows the AWS CloudWatch Events console. The breadcrumb navigation is 'CopyAutomatedSnapshotsCrossR...'. The event rule 'InvokeOnRDSSnapshot' is selected, with its ARN: `arn:aws:events:us-east-1:616399057974:rule/InvokeOnRDSSnapshot`. The rule is enabled. The event pattern is as follows:

```
{
  "detail-type": [
    "RDS DB Snapshot Event"
  ],
  "source": [
    "aws.rds"
  ],
  "detail": {
    "Message": [
      "Automated snapshot created"
    ]
  }
}
```

Task 5: Verify the Automation

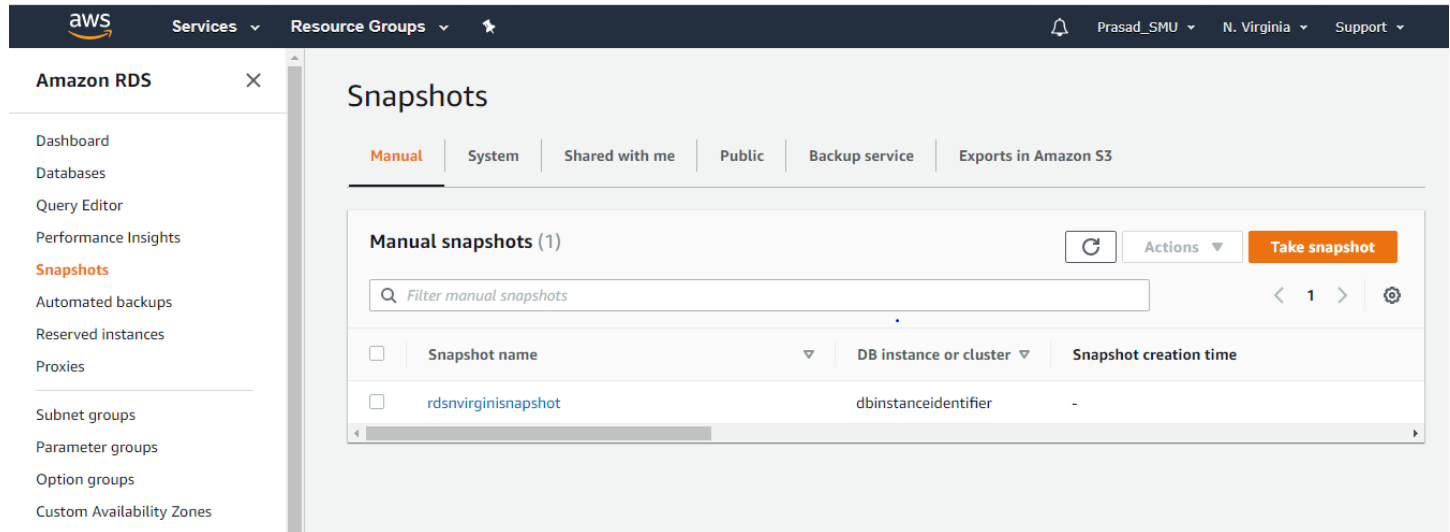
Navigate to US-WEST-1 region, and go to RDS Service.

Click on Snapshots, you'll notice that there is not a single RDS Instance Snapshot.

The screenshot shows the AWS Amazon RDS Snapshots console. The breadcrumb navigation is 'RDS > Snapshots'. The 'Manual' tab is selected. The 'Manual snapshots (0)' section shows a search bar with the placeholder 'Filter manual snapshots' and a 'Take snapshot' button. Below the search bar is a table with columns: 'Snapshot name', 'DB instance or cluster', and 'Snapshot creation time'. The table is currently empty. The footer shows 'Feedback' and 'English (US)'.

Comeback to the Primary Region i.e. N. Virginia (US-EAST-1) wherein you've RDS Instances running.

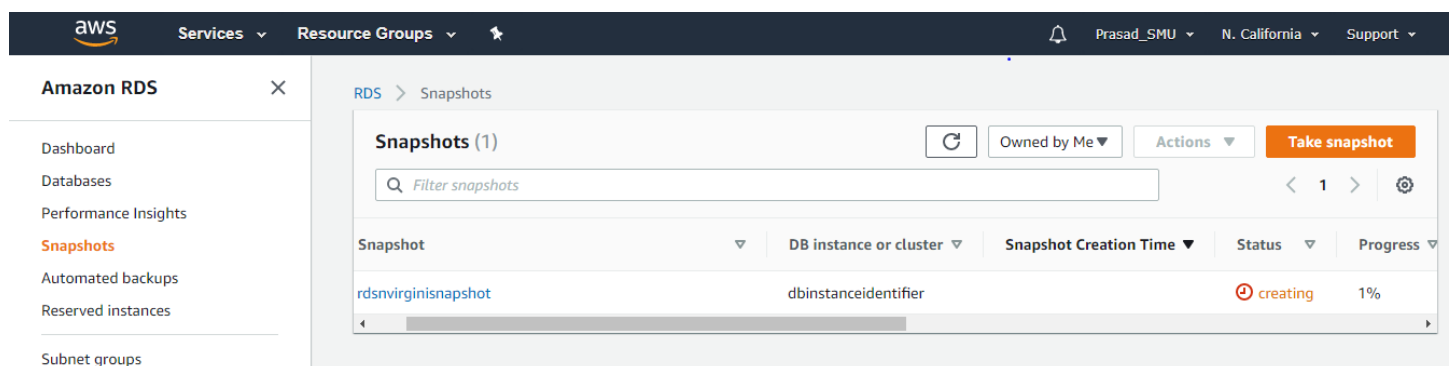
Navigate to the RDS Service, click on Snapshots and click on **Take Snapshot**.



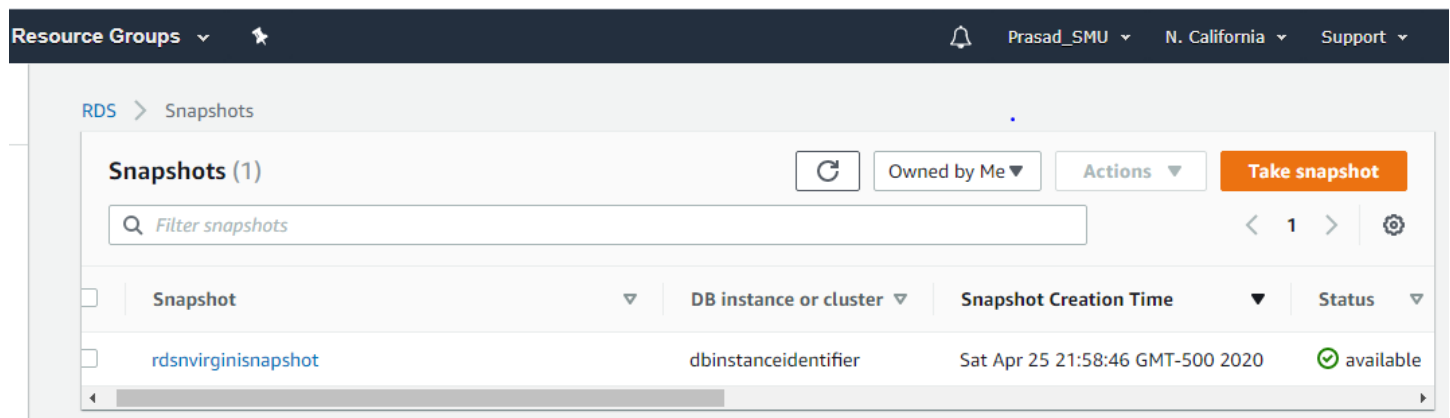
Now again change the region to N. California (US-WEST-1).

Navigate to RDS Service and Click on Snapshots.

You'll notice that the Snapshot creating process has been initiated.



Wait for some time, the snapshot has been successfully copied to another region N. California (US-WEST-1).



This completes the lab on Automate RDS Snapshots.

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