Steps to Create database in AWS and connect to SQL:

**To create an Aurora DB cluster using the console**

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the upper-right corner of the AWS Management Console, choose the AWS Region in which you want to create the DB cluster.

Aurora is not available in all AWS Regions. For a list of AWS Regions where Aurora is available, see [Region availability](https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Concepts.RegionsAndAvailabilityZones.html#Aurora.Overview.Availability).

1. In the navigation pane, choose Databases.
2. Choose Create database.
3. In Choose a database creation method, choose Standard Create. In Engine options, choose Amazon Aurora.


                                Engine options
                            

1. In Edition , choose one of the following:
   * Amazon Aurora MySQL-Compatible Edition-We chose this.
   * Amazon Aurora PostgreSQL-Compatible Edition

We chose Amazon Aurora MySQL-Compatible Edition.

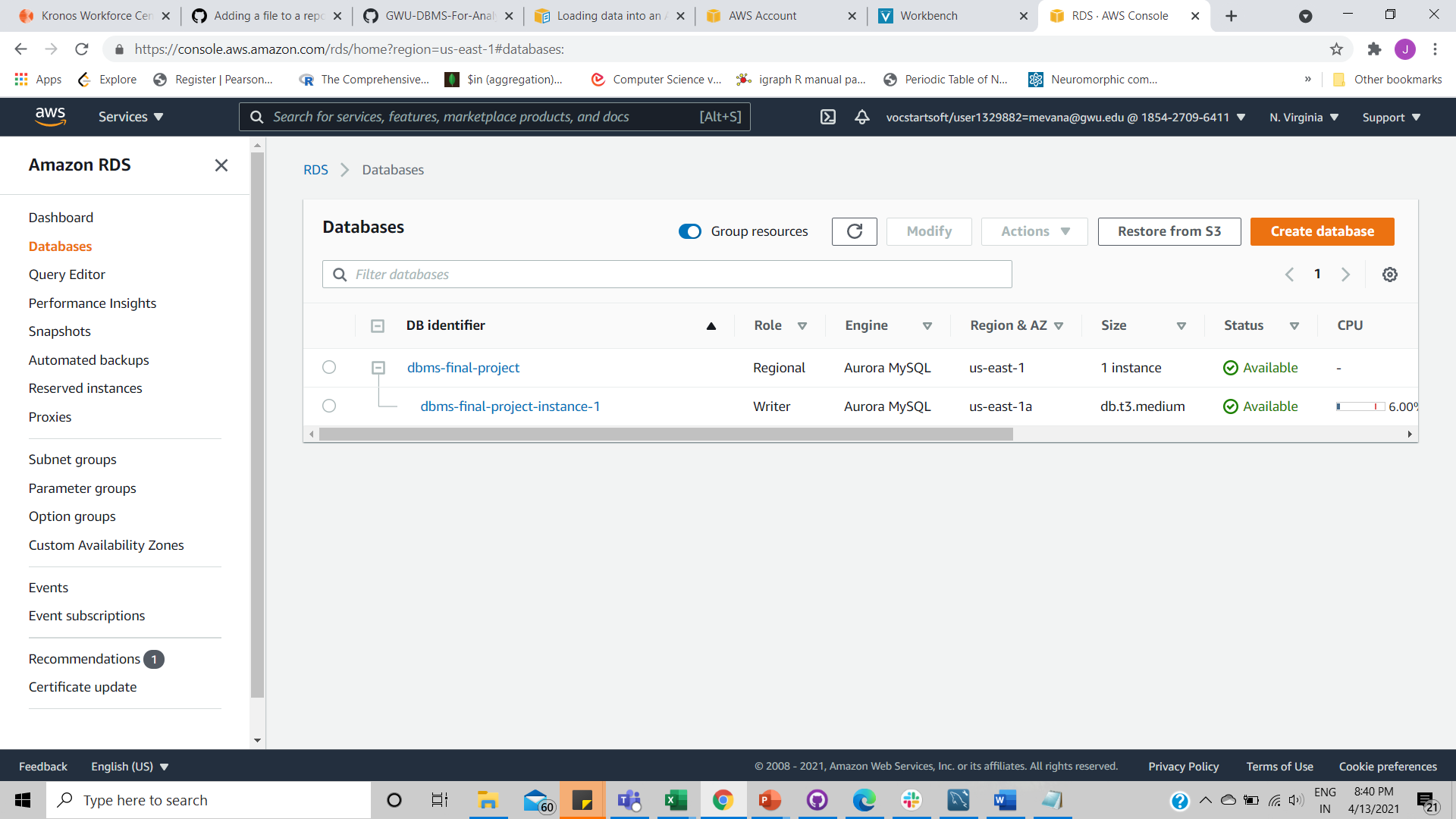
1. \*For Amazon Aurora MySQL-Compatible Edition, choose one of the following in Database features: Serverless
2. To enter the master password, do the following:
   * 1. In the Settings section, open Credential Settings.
     2. Clear the Auto generate a password check box.
     3. (Optional) Change the Master username value and enter the same password in Master password and Confirm password.

By default, the new DB instance uses an automatically generated password for the master user.

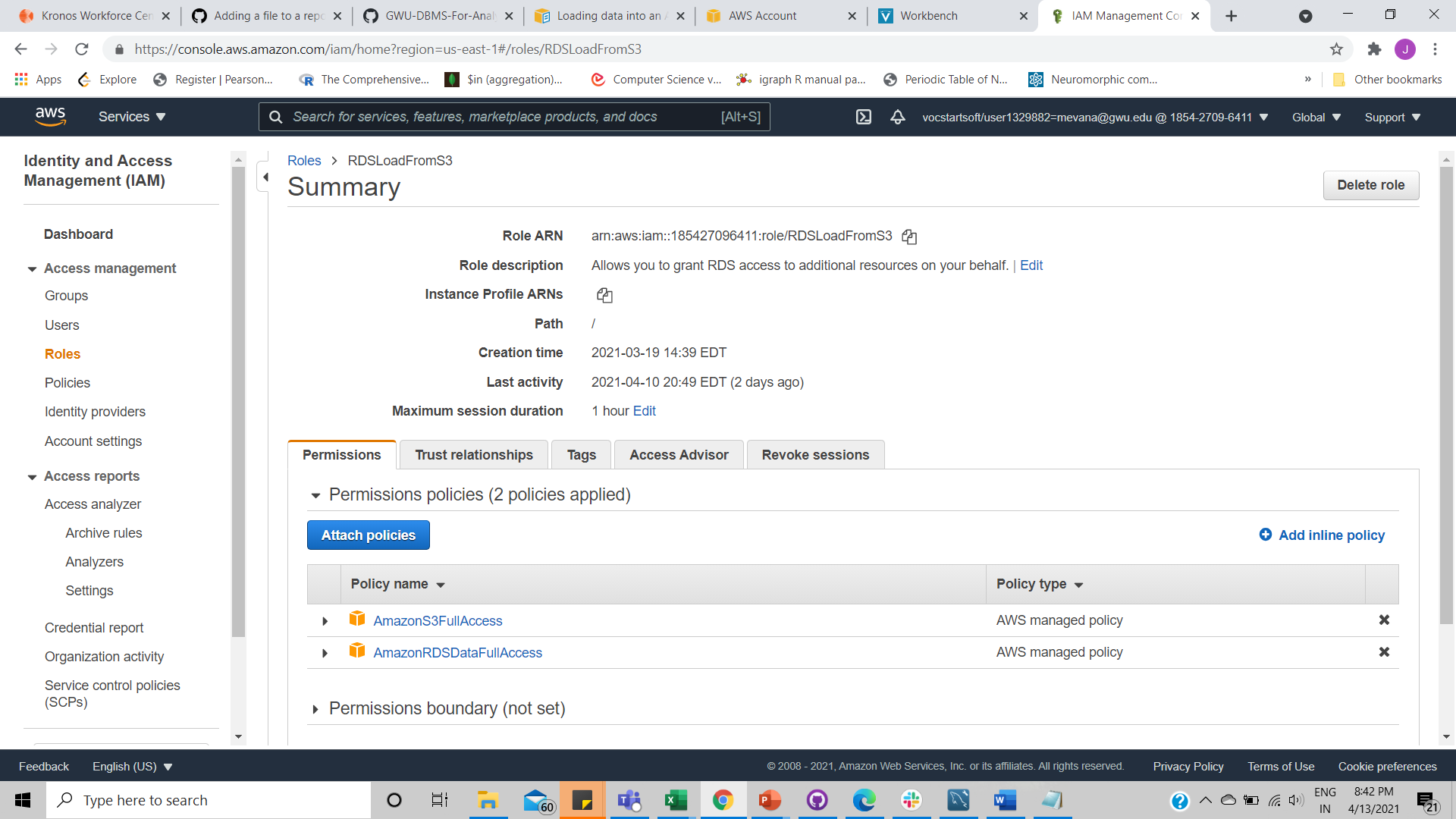
1. Choose Create database.

To view the master user name and password for the DB cluster, choose View credential details.

1. For Databases, choose the name of the new Aurora DB cluster.
2. On the RDS console, the details for new DB cluster appear. The DB cluster and its DB instance have a status of creating until the DB cluster is ready to use. When the state changes to available for both, we connected to the DB cluster. Depending on the DB instance class and the amount of storage, it can take up to 20 minutes before the new DB cluster is available.
3. **Open AWS RDS**: Create an AWS Identity and Access Management (IAM) policy that provides the bucket and object permissions that allow your Aurora MySQL DB cluster to access Amazon S3.



1. Create an IAM role, and attach the IAM policy you created in [Creating an IAM policy to access Amazon S3 resources](https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraMySQL.Integrating.Authorizing.IAM.S3CreatePolicy.html) to the new IAM role.

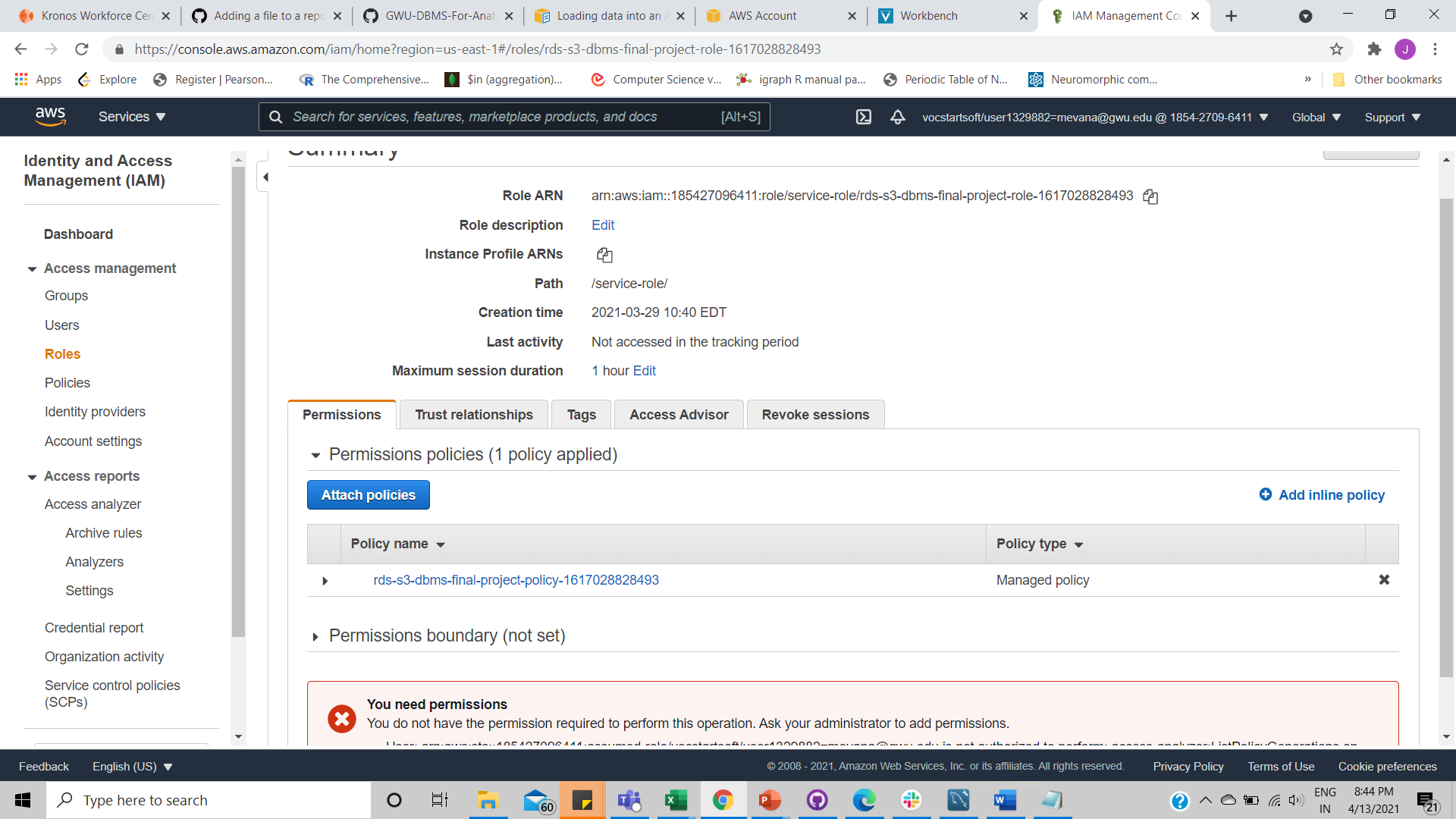


1. .To create an IAM policy to grant access to your Amazon S3 resources
2. Open the [IAM Management Console](https://console.aws.amazon.com/iam/home?#home).
3. In the navigation pane, choose Policies.
4. Choose Create policy.
5. On the Visual editor tab, choose a service, and then choose S3.
6. For Actions, choose Expand all, and then choose the bucket permissions and object permissions needed for the IAM policy.

Object permissions are permissions for object operations in Amazon S3, and need to be granted for objects in a bucket, not the bucket itself. Choose Resources, and choose Add ARN for bucket.

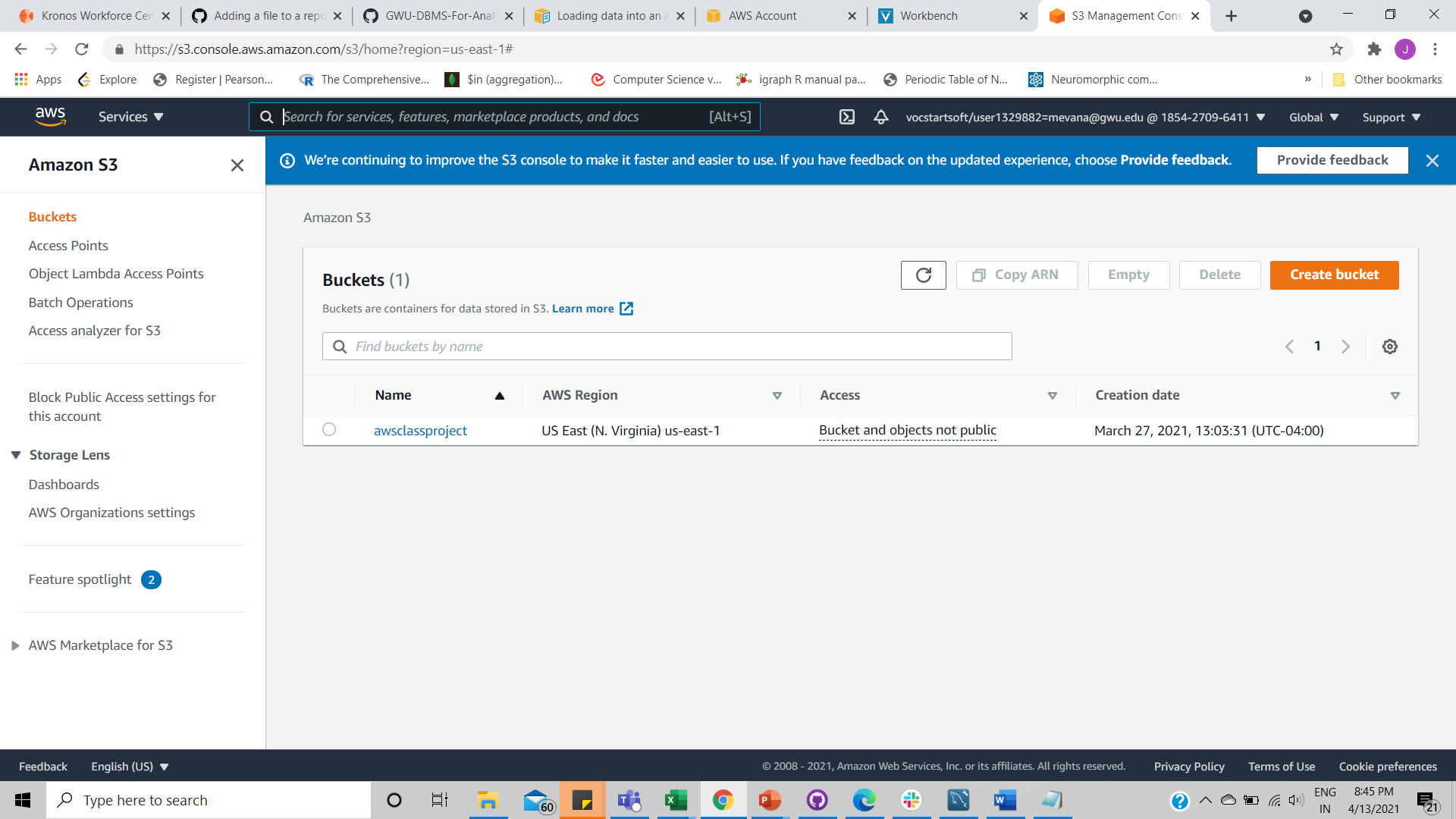
1. In the Add ARN(s) dialog box, provide the details about your resource, and choose Add.

Specify the Amazon S3 bucket to allow access to. We wanted to allow Aurora to access the Amazon S3 bucket named example-bucket, so we set the Amazon Resource Name (ARN) value to arn:aws:s3:::example-bucket.



1. If the object resource is listed, choose Add ARN for object.
2. In the Add ARN(s) dialog box, provide the details about your resource.
3. Choose Review policy.
4. For Name, enter a name for your IAM policy, for example AllowAuroraToExampleBucket.
5. Choose Create policy.

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.



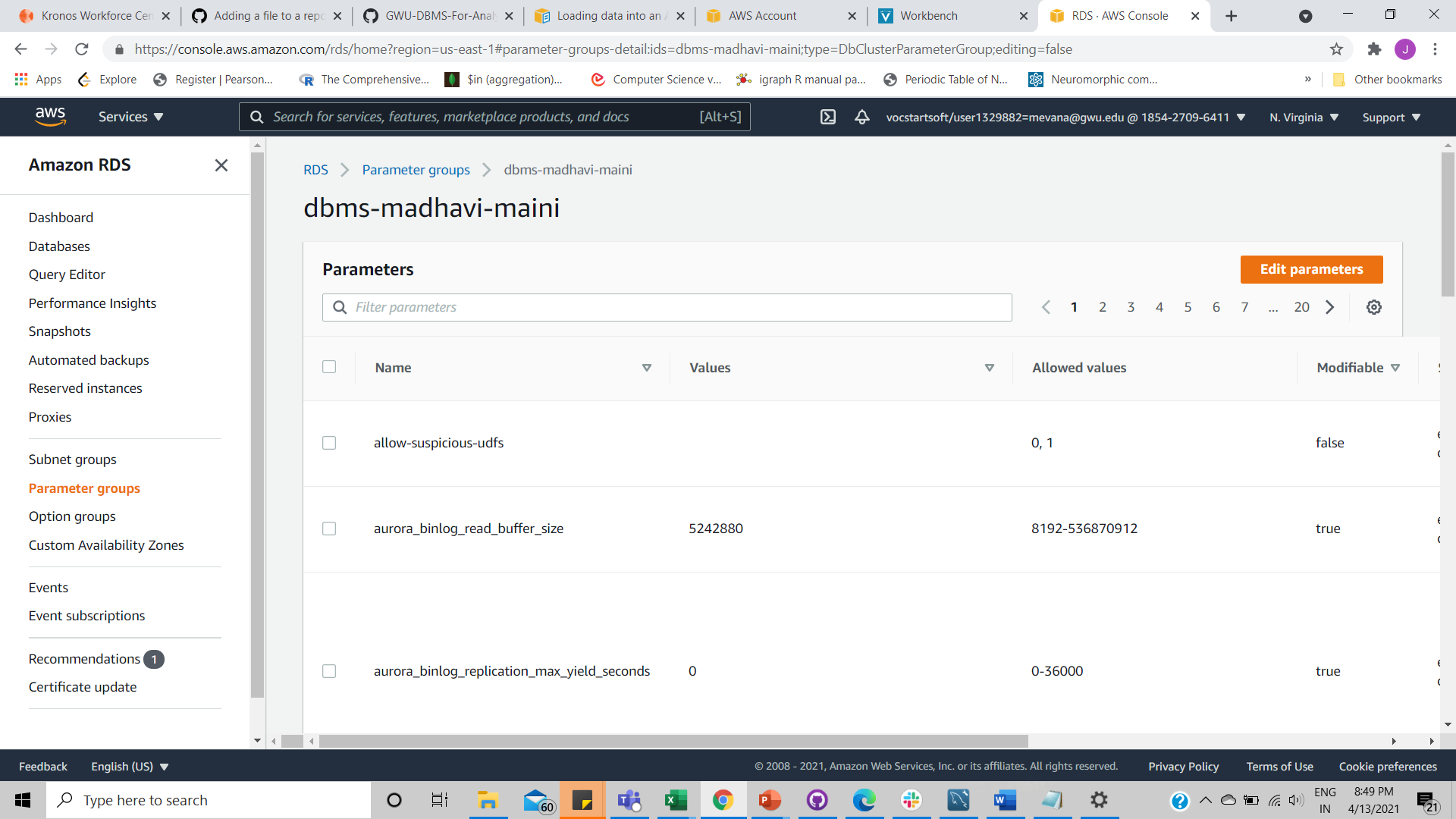
1. Make sure the DB cluster is using a custom DB cluster parameter group.

For more information about creating a custom DB cluster parameter group.

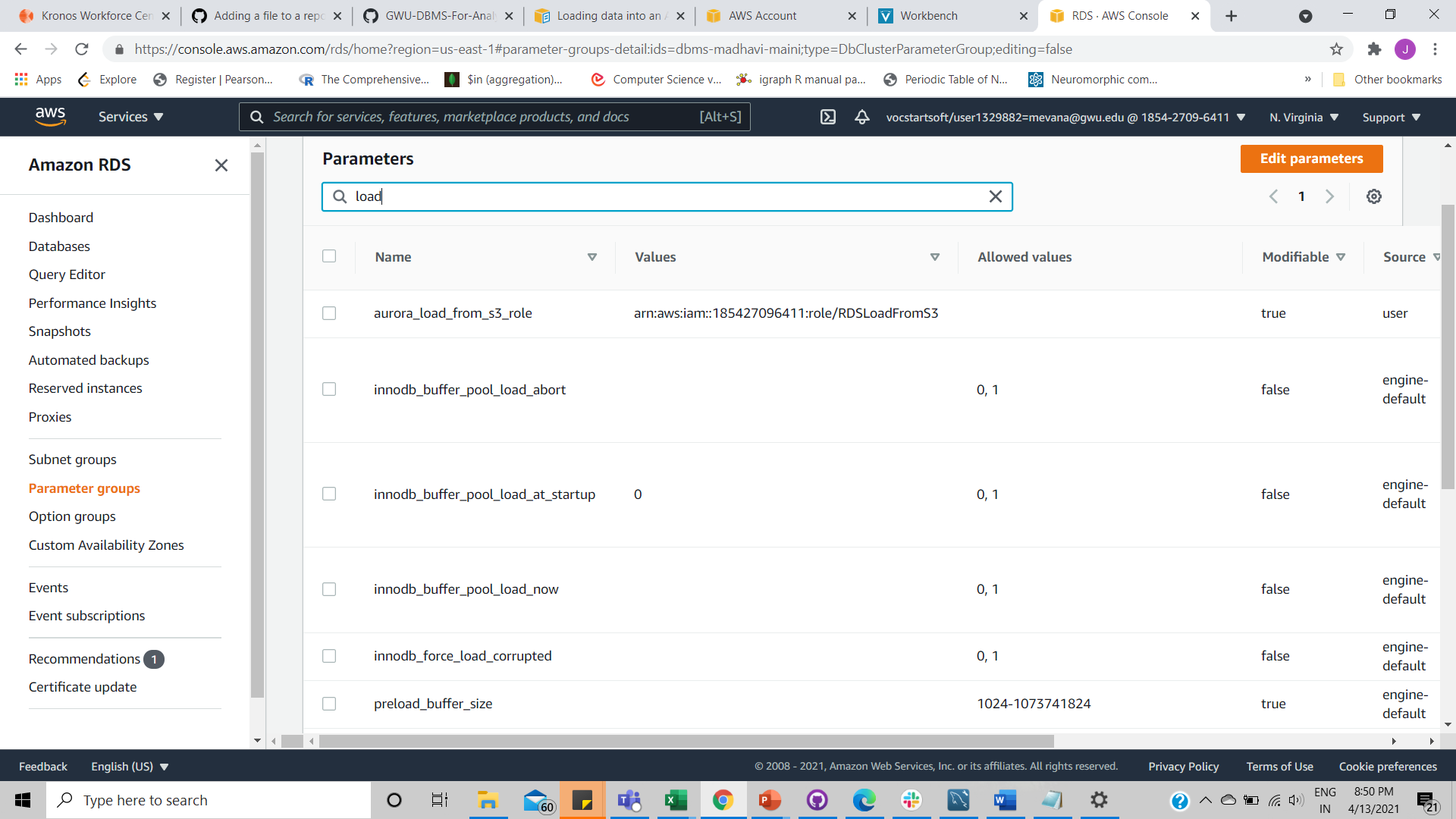
1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose Parameter groups.
3. Choose Create parameter group.

The Create parameter group window appears.

1. In the Parameter group family list, select a DB parameter group family
2. In the Type list, select DB Cluster Parameter Group.
3. In the Group name box, enter the name of the new DB cluster parameter group.
4. In the Description box, enter a description for the new DB cluster parameter group.
5. Choose Create.



1. Set either the aurora\_load\_from\_s3\_role or aws\_default\_s3\_role DB cluster parameter to the Amazon Resource Name (ARN) of the new IAM role. If an IAM role isn't specified for aurora\_load\_from\_s3\_role, Aurora uses the IAM role specified in aws\_default\_s3\_role.



1. To permit database users in an Aurora MySQL DB cluster to access Amazon S3, associated the role that we created in [Creating an IAM role to allow Amazon Aurora to access AWS services](https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraMySQL.Integrating.Authorizing.IAM.CreateRole.html) with the DB cluster. For an Aurora global database, associate the role with each Aurora cluster in the global database.
2. Configure the Aurora MySQL DB cluster to allow outbound connections to Amazon S3
3. Specifying a path to an Amazon S3 bucket:

s3-region://bucket-name/file-name-or-prefix

1. LOAD DATA FROM S3
2. The Database in mysql:

