

# Databases

**RDS - Relational Database Service**

# AWS RDS

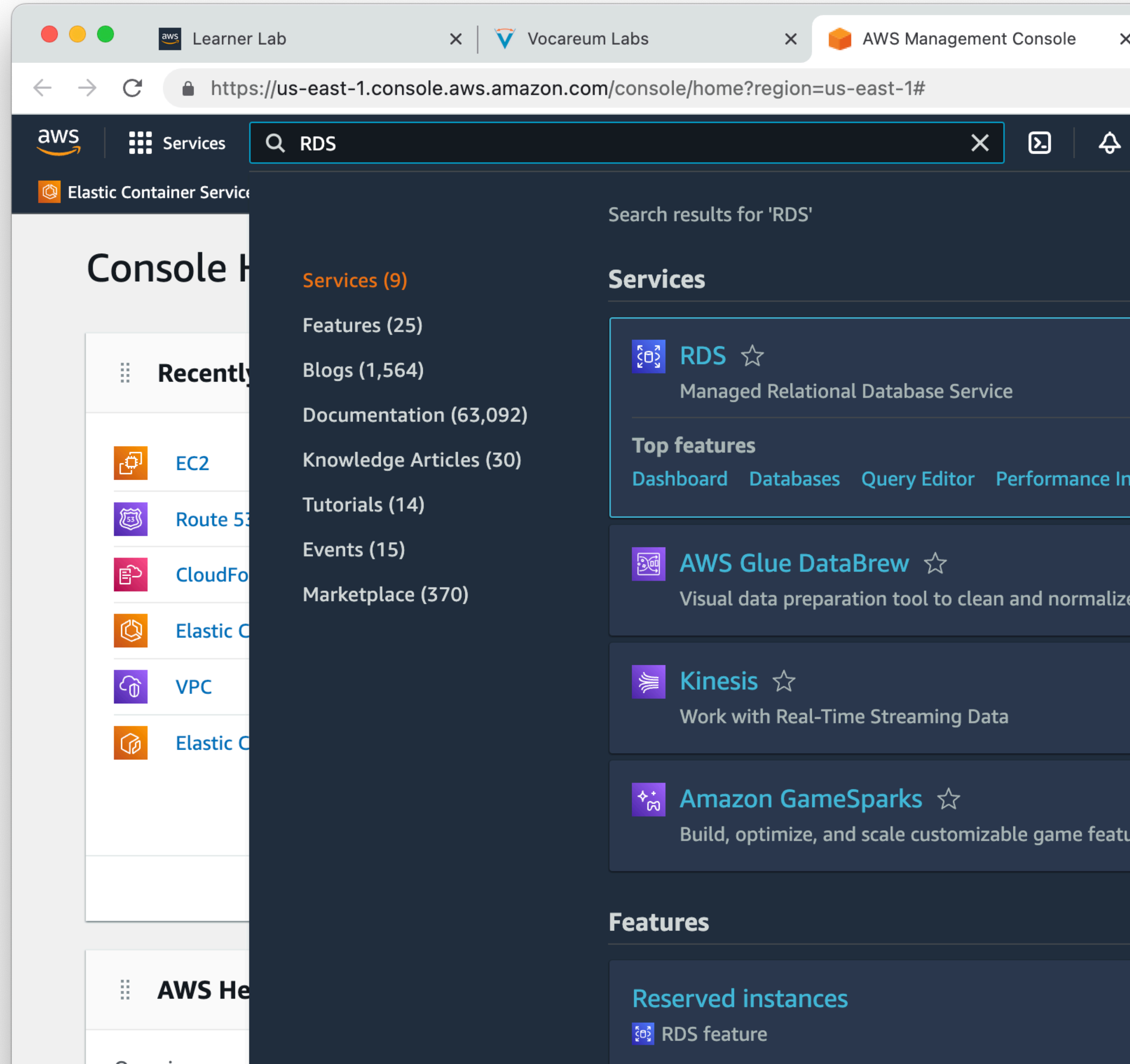
## Managed Database Service

- RDS (Relational Database Server) is Amazon's SQL offering
- Of course, you can always run your own database server, on an instance anywhere...
  - <https://www.mysql.com/>
- But why go to all that trouble?

# AWS RDS

## Managed Database Service

- Search for RDS in the top search bar.
- Click on RDS in the Services results.

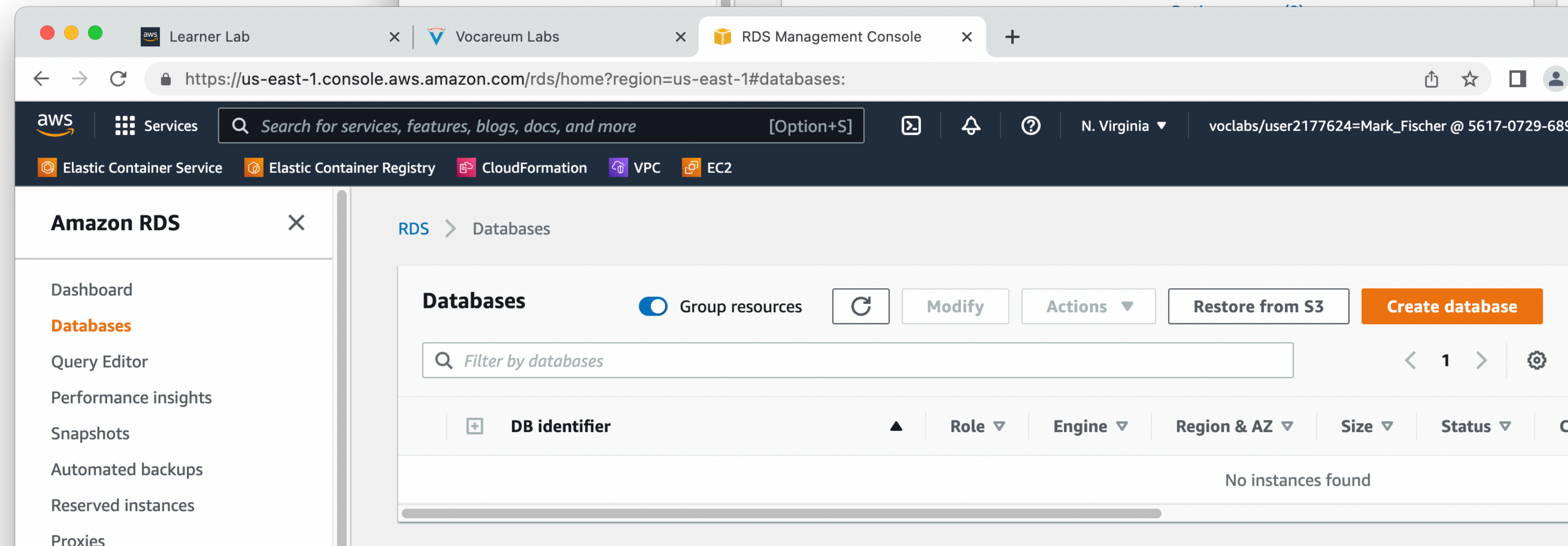
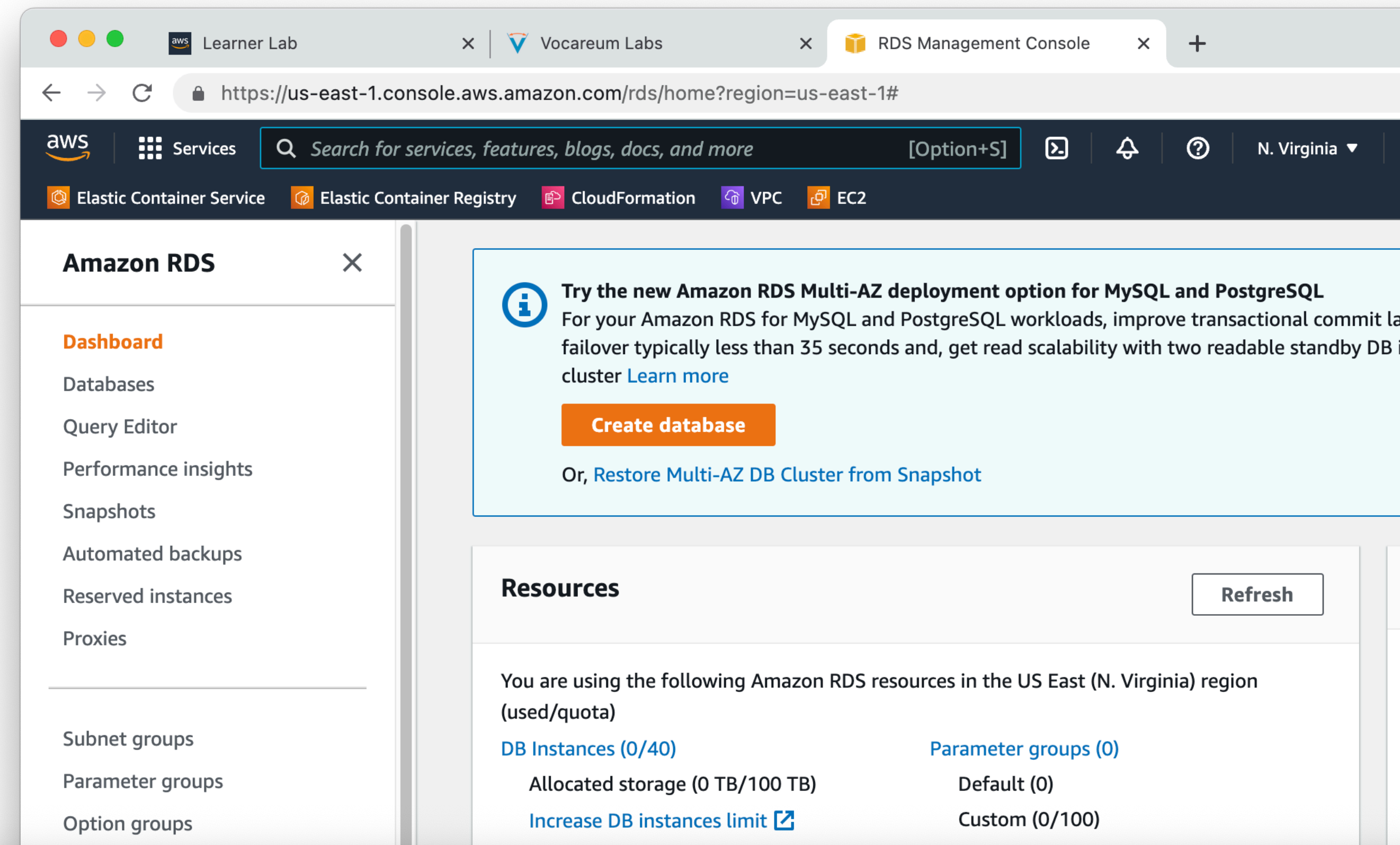




# AWS RDS

## Managed Database Service

- From the dashboard, you may see a “Create Database” button in an announcement.
- If not, click on “Databases” in the left sidebar.
- Click “Create Database”





# AWS RDS

## Managed Database Service

- Select “Standard Create.” We have to turn off some features which aren’t allowed in AWS Academy.
- Select MySQL
- Leave the Edition and version as default.
  - MySQL Community
  - Version 8.0.x

The screenshot shows the AWS RDS console interface for creating a new database. The browser tabs include 'Learner Lab', 'Vocareum Labs', and 'RDS Management Console'. The URL is 'https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:gdb=false;s3-import=fa'. The page title is 'Create database' under the 'RDS > Create database' breadcrumb. The main heading is 'Create database'. Below this, there is a section 'Choose a database creation method' with two options: 'Standard create' (selected) and 'Easy create'. The 'Standard create' option is described as: 'You set all of the configuration options, including ones for availability, security, backups, and maintenance.' The 'Easy create' option is described as: 'Use recommended best-practice configurations. Some configuration options can be changed after the database is created.' Below this is the 'Engine options' section. Under 'Engine type', there are six options: 'Amazon Aurora' (not selected), 'MySQL' (selected), 'MariaDB' (not selected), 'PostgreSQL' (not selected), 'Oracle' (not selected), and 'Microsoft SQL Server' (not selected). Each option has a corresponding icon. Below the engine options is the 'Edition' section, with 'MySQL Community' selected. At the bottom of the console, there is a footer with 'Feedback', a link to 'Looking for language selection? Find it in the new Unified Settings', and a copyright notice: '© 2022, Amazon Web Services, Inc. or i'.



# AWS RDS

## Managed Database Service

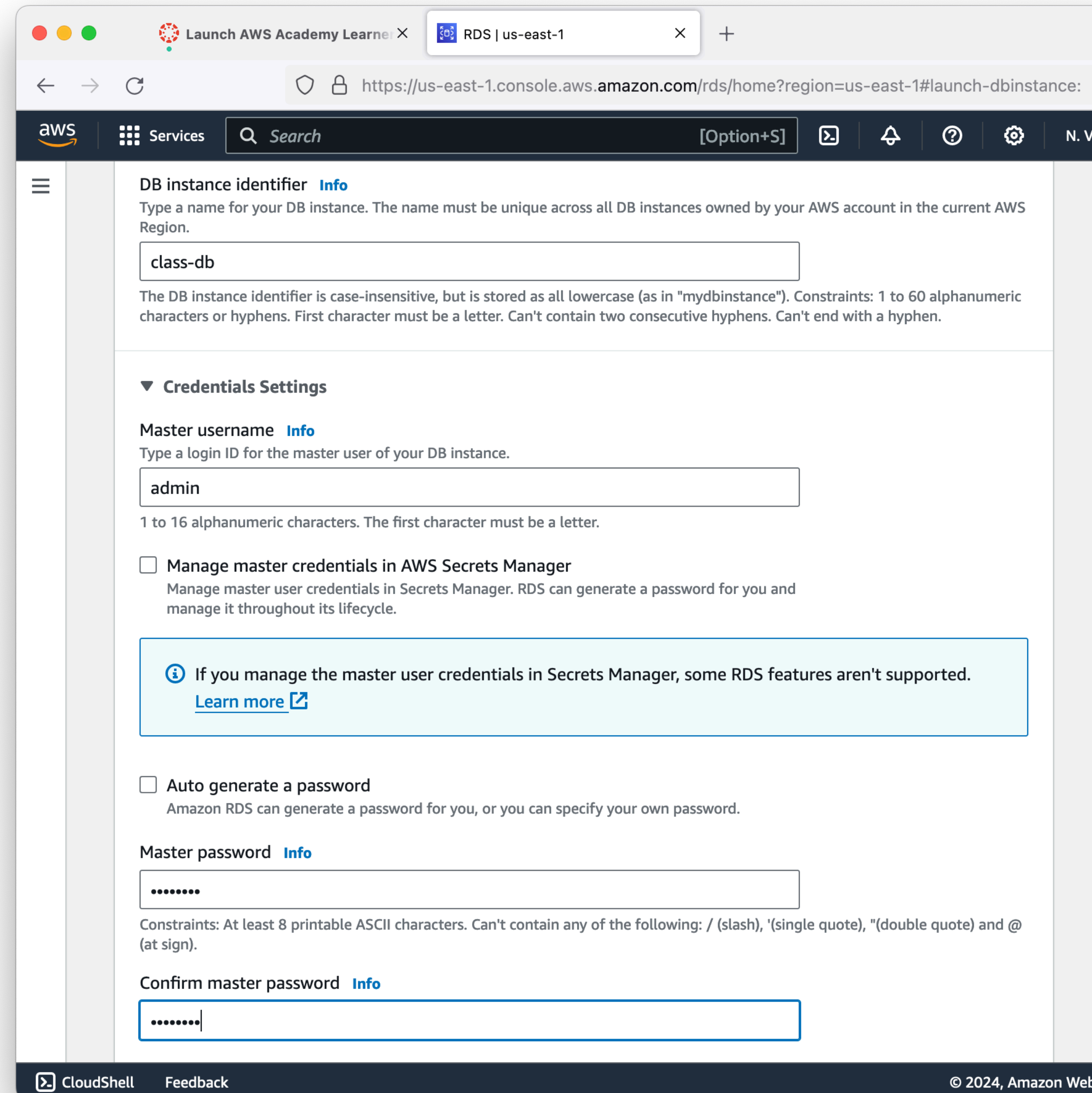
- Make sure to select “Free tier”

The screenshot shows the AWS RDS console interface. At the top, there are browser tabs for 'Launch AWS Academy Learn...' and 'RDS | us-east-1'. The address bar shows the URL: 'https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-database-instance:'. The navigation bar includes the AWS logo, 'Services', a search bar, and the region 'N. Virginia'. The main content area is divided into sections: 'Engine version' with a sub-section 'Hide filters' containing two radio button options for 'Multi-AZ DB cluster' and 'Amazon RDS Optimized Writes'. Below this is a dropdown menu for 'Engine Version' set to 'MySQL 8.0.35'. The 'Templates' section offers three radio button options: 'Production', 'Dev/Test', and 'Free tier' (which is selected and highlighted in blue). The 'Availability and durability' section includes a 'Deployment options' sub-section with two radio button options: 'Single DB instance' and 'Multi-AZ DB instance'. The footer contains 'CloudShell', 'Feedback', and a copyright notice for Amazon Web Services, Inc. © 2024.

# AWS RDS

## Managed Database Service

- Set a name for your DB Instance. This only shows up in the AWS console, its not used for connecting to the database
- Choose a good password, and keep it somewhere safe and memorable.
- If you forget it, you can reset this later.

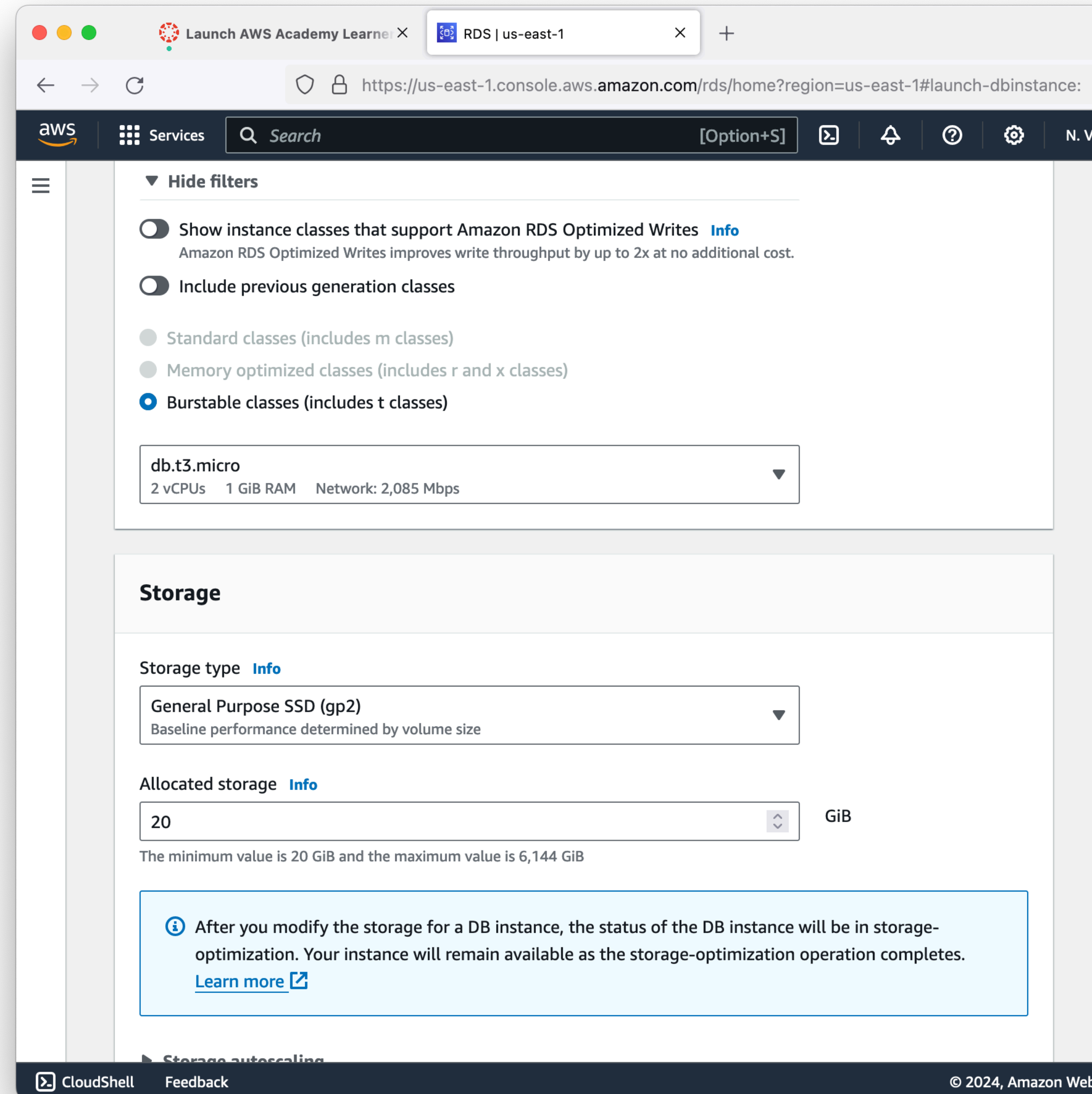




# AWS RDS

## Managed Database Service

- Change the instance class to `db.t3.micro`
- Change the Allocated storage to the smallest allowed: 20 GiB
- Un-check “Enable storage autoscaling”
  - We won’t use anywhere near that much space.



# AWS RDS

## Managed Database Service

- Don't connect to an EC2 resource (we'll configure that ourselves)
- Make sure you have "No" selected for Public access.
- Create a new security group
- Name your security group "database-sg"
  - If you delete this RDS instance and create a new one later, you can re-use this VPC security group

Launch AWS Academy Learner X RDS | us-east-1 X +

← → ↻ <https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:>

aws Services Search [Option+S] N. Virginia

Don't connect to an EC2 compute resource  
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource  
Set up a connection to an EC2 compute resource for this database.

**Virtual private cloud (VPC)** [Info](#)  
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

Default VPC (vpc-0059b159ca91e292b)  
6 Subnets, 6 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

**After a database is created, you can't change its VPC.**

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

default

**Public access** [Info](#)

No  
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

Yes  
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

**VPC security group (firewall)** [Info](#)  
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

Choose existing  
Choose existing VPC security groups

Create new  
Create new VPC security group

New VPC security group name  
database-sg

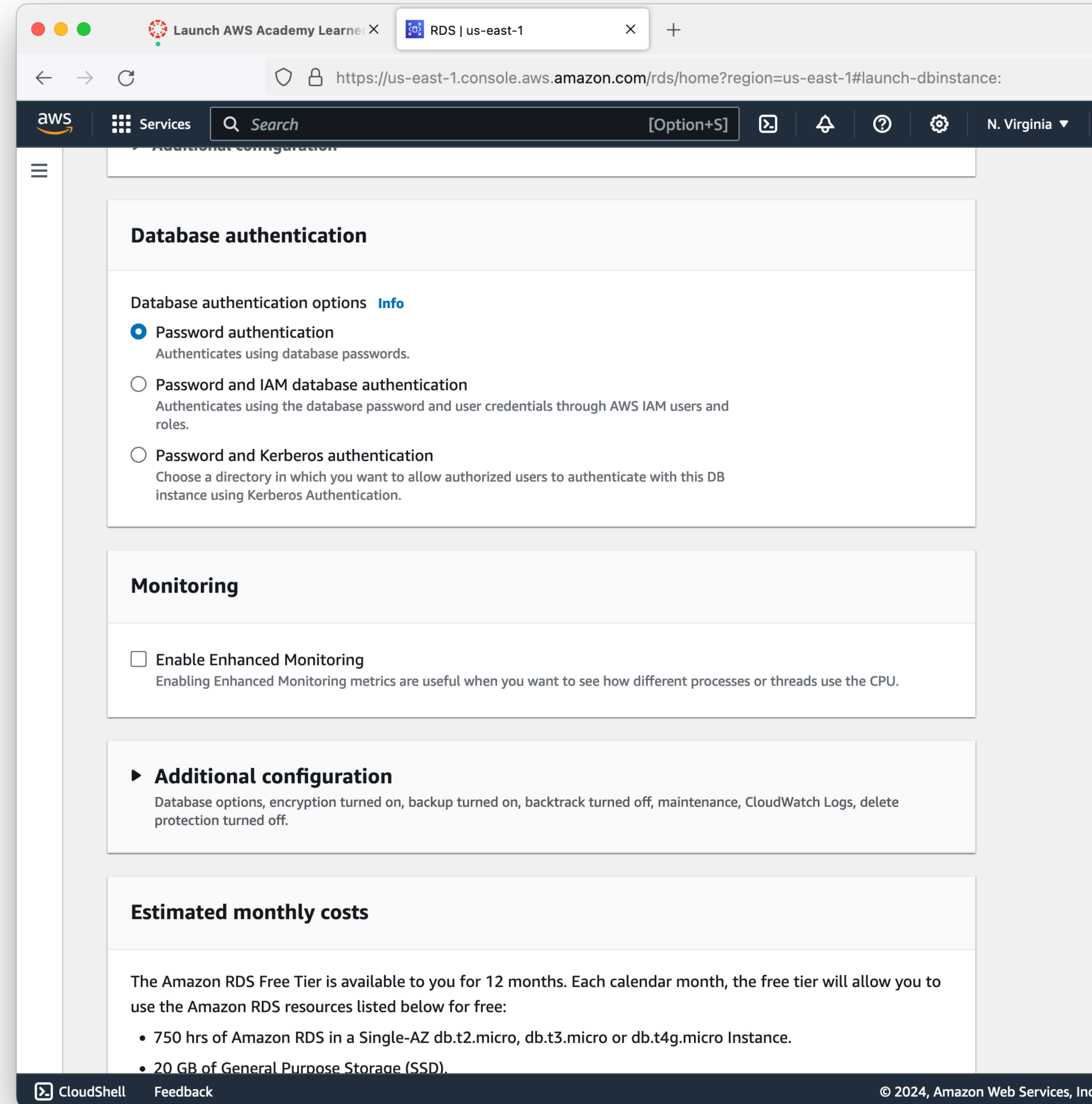
CloudShell Feedback © 2024, Amazon Web Services, Inc.



# AWS RDS

## Managed Database Service

- Leave “Password authentication” selected



The screenshot shows the AWS RDS console interface for a database instance in the us-east-1 region. The 'Database authentication' section is expanded, showing three radio button options: 'Password authentication' (selected), 'Password and IAM database authentication', and 'Password and Kerberos authentication'. Below this, the 'Monitoring' section has 'Enable Enhanced Monitoring' unchecked. The 'Additional configuration' section is collapsed. The 'Estimated monthly costs' section is expanded, showing the Amazon RDS Free Tier details.

**Database authentication**

Database authentication options [Info](#)

- Password authentication  
Authenticates using database passwords.
- Password and IAM database authentication  
Authenticates using the database password and user credentials through AWS IAM users and roles.
- Password and Kerberos authentication  
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

**Monitoring**

- Enable Enhanced Monitoring  
Enabling Enhanced Monitoring metrics are useful when you want to see how different processes or threads use the CPU.

**Additional configuration**  
Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

**Estimated monthly costs**

The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

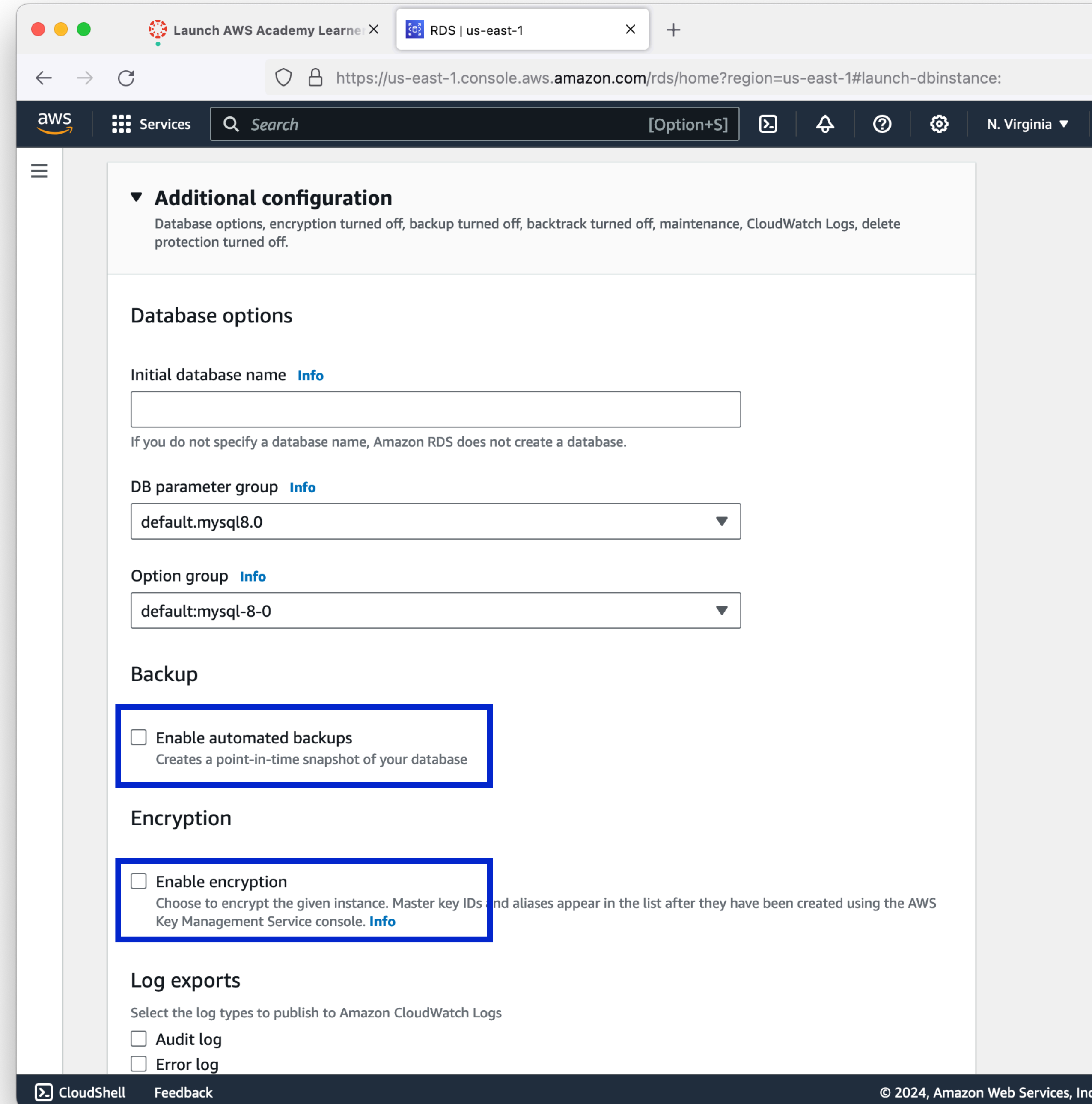
- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro, db.t3.micro or db.t4g.micro Instance.
- 20 GB of General Purpose Storage (SSD).



# AWS RDS

## Managed Database Service

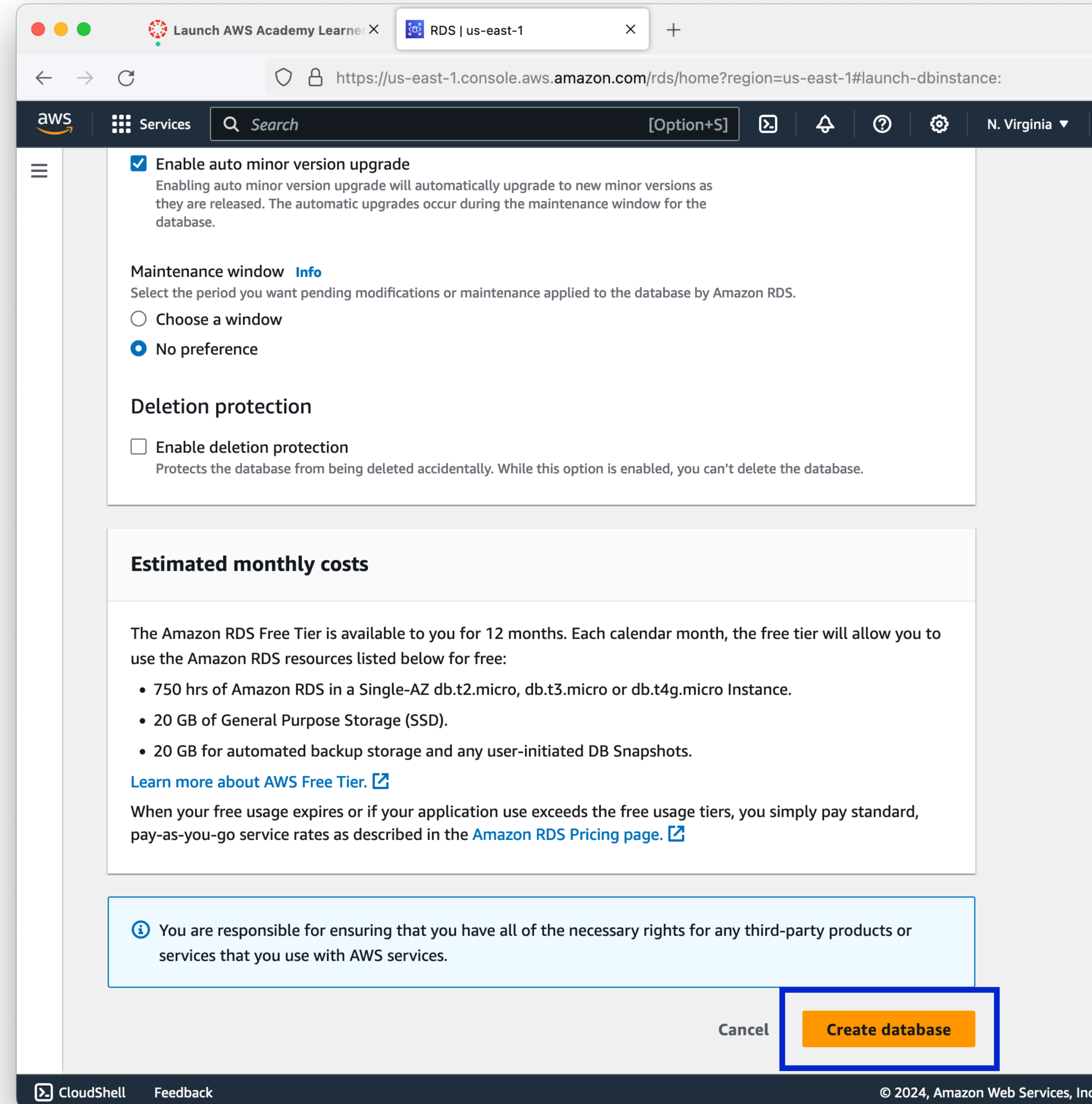
- Expand “Additional configuration”
- Disable automated backups
  - Automated backups are usually the correct default for things, but we really want to minimize costs for the class, and daily backups really add up!
- Disable encryption
  - Usually a good idea, keep it simple for class.



# AWS RDS

## Managed Database Service

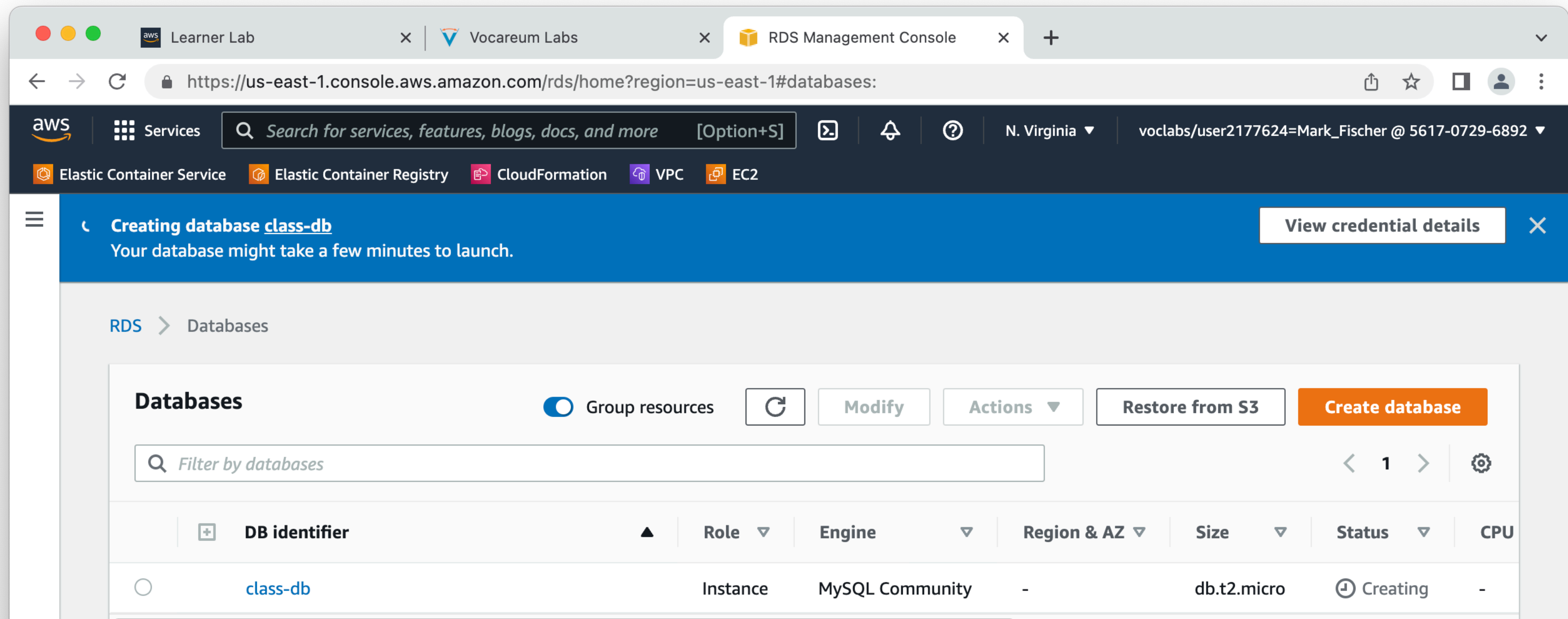
- Click “Create database”



# AWS RDS

## Managed Database Service

- Your database may take several minutes to be ready for use. The cloud is not instant 🤪



The screenshot shows the AWS RDS Management Console interface. At the top, there are browser tabs for 'Learner Lab', 'Vocareum Labs', and 'RDS Management Console'. The address bar shows the URL: `https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#databases:`. The AWS navigation bar includes the 'Services' search bar and the user profile 'voclabs/user2177624=Mark\_Fischer @ 5617-0729-6892'. Below the navigation bar, a blue banner indicates 'Creating database class-db' with the message 'Your database might take a few minutes to launch.' and a 'View credential details' button. The main content area shows the 'Databases' section with a 'Group resources' toggle, 'Modify', 'Actions', 'Restore from S3', and 'Create database' buttons. A search bar for 'Filter by databases' is present. Below this is a table with the following columns: DB identifier, Role, Engine, Region & AZ, Size, Status, and CPU. The table contains one entry for 'class-db' with the status 'Creating'.

DB identifier	Role	Engine	Region & AZ	Size	Status	CPU
class-db	Instance	MySQL Community	-	db.t2.micro	Creating	-



# AWS RDS

## Managed Database Service

- Dark Patterns:
  - AWS now tries to up-sell you when creating things
  - Can only “hide” for 30 days!
  - You can avoid this by deploying resources through automation

The screenshot shows the AWS RDS console interface. At the top, there's a browser tab for 'Databases | RDS | us-east-1' and a URL: 'https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#databases:'. Below the navigation bar, a modal window titled 'Suggested add-ons for class-db' is displayed. The modal contains two main sections:

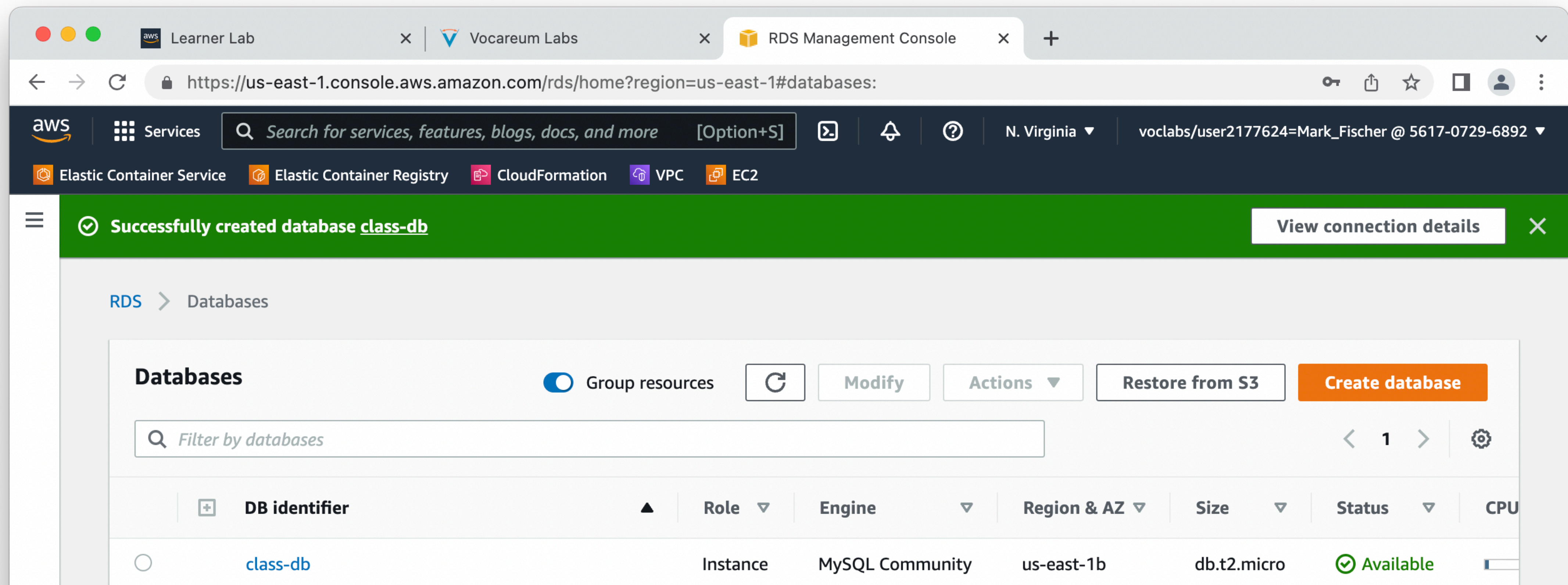
- Create an ElastiCache cluster from RDS using your DB settings - new**: This section includes an icon of a database with a cache layer, a description stating 'You can save up to 55% in cost and gain up to 80x faster read performance using ElastiCache with RDS for MySQL (vs. RDS for MySQL alone)', a 'Learn more' link, and a 'Create ElastiCache cluster' button.
- Use RDS Proxy**: This section includes an icon of a database with a proxy layer, a description stating 'Using a proxy allows your applications to pool and share database connections to help them scale. A proxy simplifies connection management and makes applications more resilient to database failures.', a 'Learn more' link, and a 'Create proxy' button.

At the bottom of the modal, there is an information box: 'You can hide these suggestions so they don't appear after database creation. All these actions can be taken from the database list page or database details page.' Below this, there is a checkbox labeled 'Hide add-ons for 30 days' which is checked, and a 'Close' button.

# AWS RDS

## Managed Database Service

- Eventually your RDS instance will complete. This may take 5 minutes or more.
- Click on your database name to get details on it.



The screenshot shows the AWS RDS Management Console interface. At the top, there's a navigation bar with the AWS logo, a search bar, and the current region (N. Virginia) and user information. Below the navigation bar, a green banner displays a success message: "Successfully created database class-db" with a "View connection details" button. The main content area is titled "Databases" and includes a "Group resources" toggle, a "Filter by databases" search box, and a table of database instances. The table has columns for "DB identifier", "Role", "Engine", "Region & AZ", "Size", "Status", and "CPU". One instance is listed: "class-db" with a role of "Instance", engine of "MySQL Community", region of "us-east-1b", size of "db.t2.micro", and status of "Available".

DB identifier	Role	Engine	Region & AZ	Size	Status	CPU
<a href="#">class-db</a>	Instance	MySQL Community	us-east-1b	db.t2.micro	Available	



# AWS RDS

- You will need to copy down the Endpoint domain name. This is how you will connect to your database from a server.

The screenshot displays the AWS Management Console interface for an Amazon RDS database instance. The browser address bar shows the URL: `https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#database:id=class-db;is-cluster=false`. The console header includes the AWS logo, a search bar, and the user's location (N. Virginia) and account ID (voclabs/user3067805=Test\_Student @ 5332-6741-4).

The main content area is titled "class-db" and includes a "Summary" section with the following details:

DB identifier	Status	Role	Engine	Recommendations
class-db	Available	Instance	MySQL Community	
CPU	Class	Current activity	Region & AZ	
3.06%	db.t3.micro	0	us-east-1f	
		Connections		

Below the summary, there are tabs for "Connectivity & security", "Monitoring", "Logs & events", "Configuration", "Zero-ETL integrations", and "Maintenance". The "Connectivity & security" tab is active, showing the following details:

Endpoint & port	Networking	Security
Endpoint class-db.cb2k68ew0n3c.us-east-1.rds.amazonaws.com	Availability Zone us-east-1f	VPC security groups database-sg (sg-025e5e60516cdacf1) Active
Port 3306	VPC vpc-0059b159ca91e292b	Publicly accessible No
	Subnet group default-vpc-0059b159ca91e292b	Certificate authority rds-ca-rsa2048-g1

The endpoint value is highlighted with a blue box. The footer of the console includes "CloudShell", "Feedback", and copyright information for Amazon Web Services, Inc. or its affiliates.



# AWS RDS

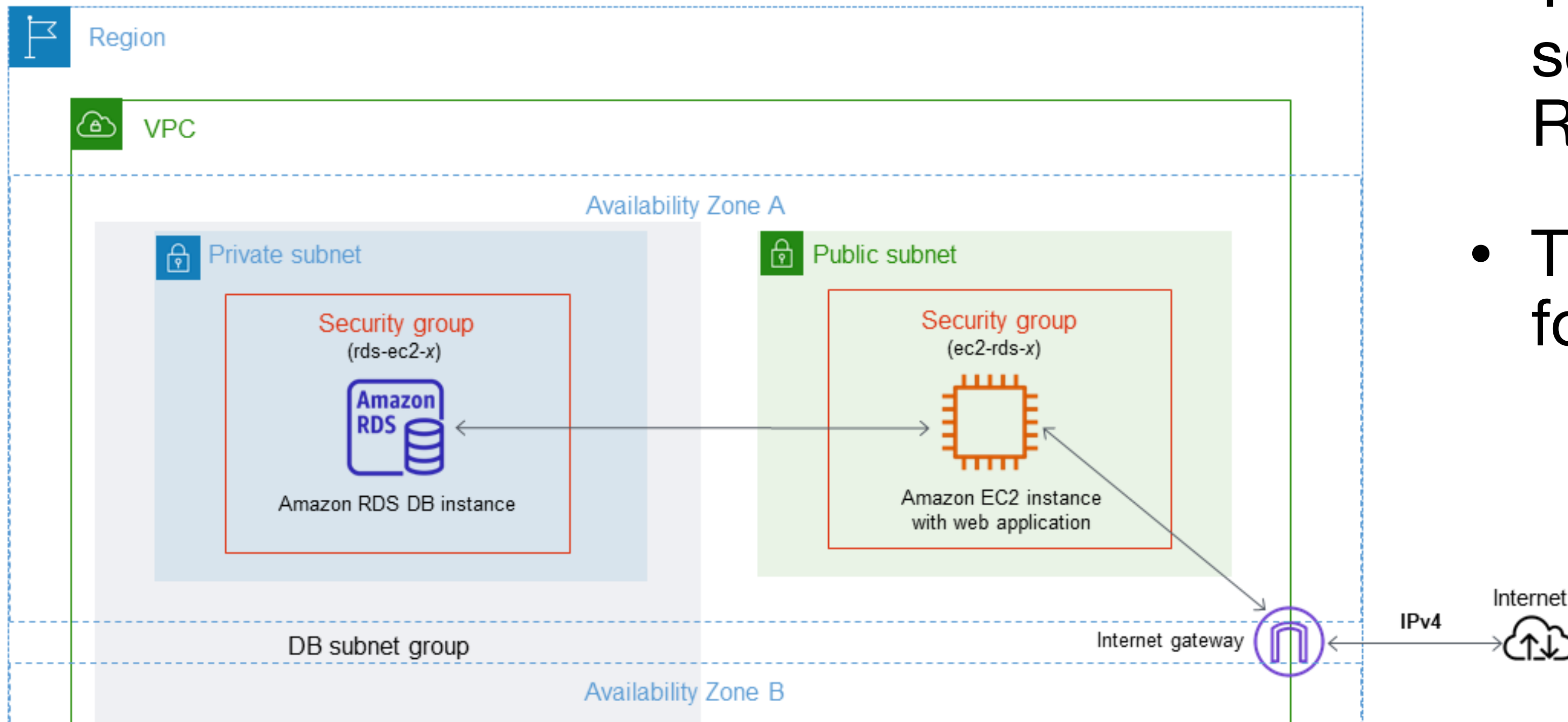
## Security Groups

- Our new RDS instance does NOT have a public IP address
  - Because RDS is a fully managed service, you cannot ssh into it
  - With no Public IP you cannot connect to it directly from your laptop
- Our new RDS instance has a private IP address, and is listening on port 3306
  - We need to give our EC2 instance access

# AWS RDS

## Security Groups

- With no public access, we must allow our EC2 instance access to the RDS instance
- This is done by allowing the security group attached to the RDS instance
- This is a very common pattern for cloud applications

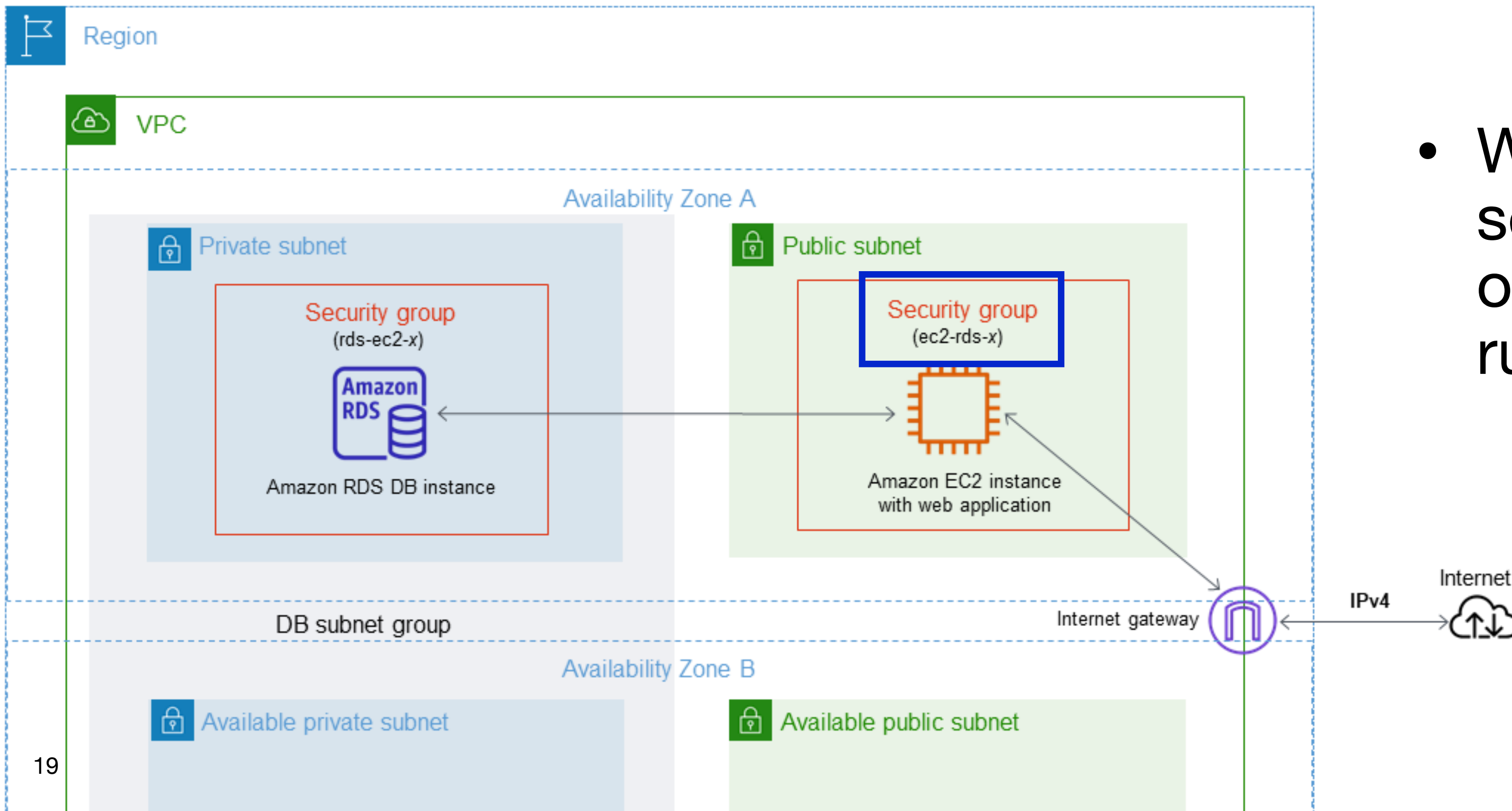


[https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER\\_VPC.Scenarios.html](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_VPC.Scenarios.html)

# AWS RDS

## Security Groups

- We'll do this by adding the security group ID attached to our EC2 instance to the ingress rules of the RDS security group

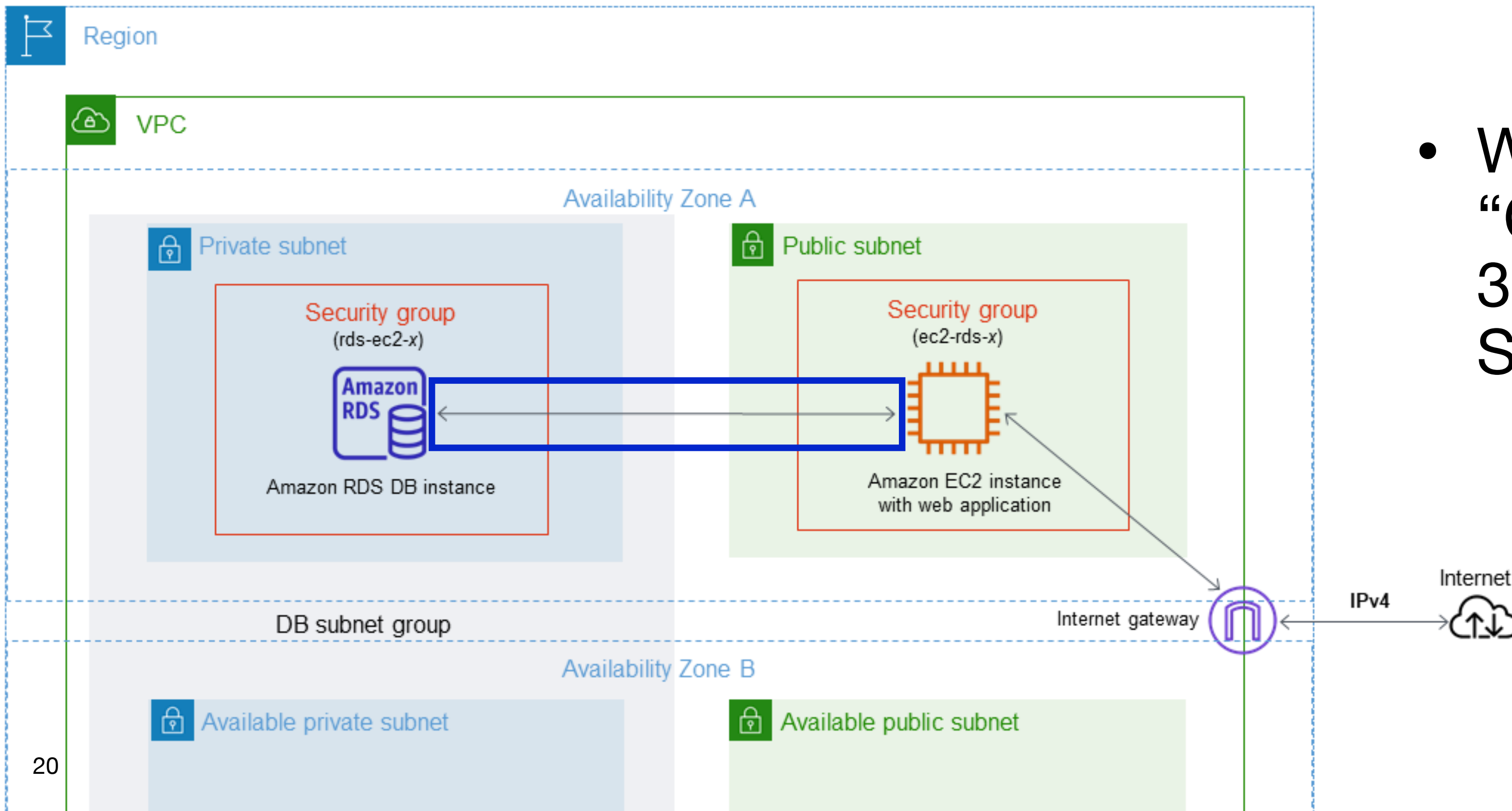




# AWS RDS

## Security Groups

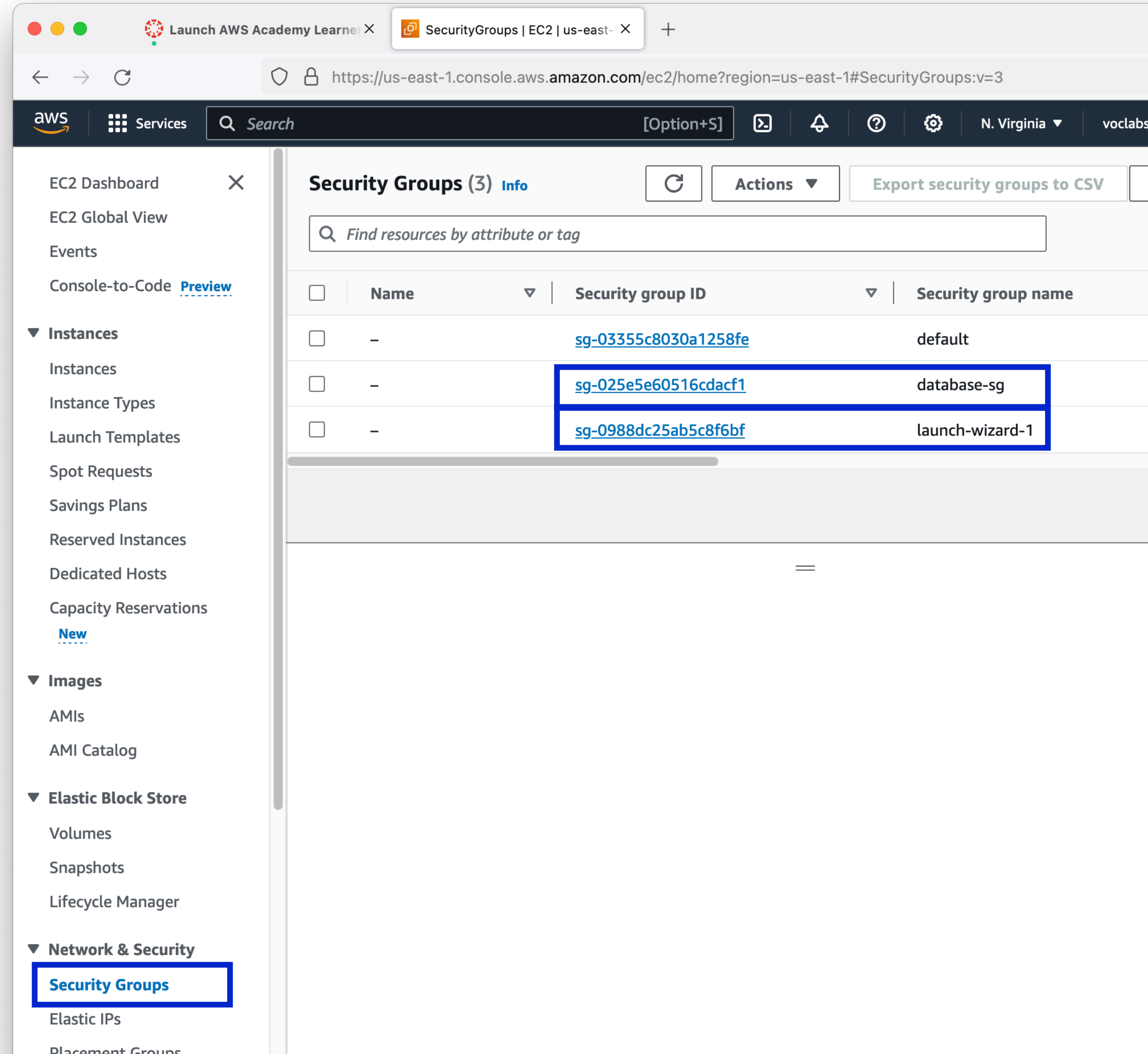
- We want to add a rule that says, “Connections to the DB [port 3306] are allowed from the EC2 Security Group.”



# AWS RDS

## Security Groups

- In the EC2 console, select “Security Groups” from the left sidebar
- the launch-wizard-1 SG is the one attached to our EC2 instance
- The database-sg SG is attached to our database



# AWS RDS Security Groups

- We need to update the database security group, so select that one
- Then click “Edit inbound rules”

The screenshot shows the AWS Management Console interface for Security Groups. The left sidebar contains navigation options like EC2 Dashboard, Instances, Images, Elastic Block Store, and Network & Security. The main content area displays a list of Security Groups with columns for Name, Security group ID, Security group name, and VPC ID. The 'database-sg' group is selected. Below the list, the 'Inbound rules' tab is active, showing a table with columns for Name, Security group rule..., IP version, Type, and Protocol. A rule is listed with Type 'MYSQL/Aurora' and Protocol 'TCP'. The 'Edit inbound rules' button is highlighted.

Name	Security group ID	Security group name	VPC ID
-	<a href="#">sg-03355c8030a1258fe</a>	default	<a href="#">vpc-0059b159ca91e2...</a>
<input checked="" type="checkbox"/>	<a href="#">sg-025e5e60516cdacf1</a>	database-sg	<a href="#">vpc-0059b159ca91e2...</a>
<input type="checkbox"/>	<a href="#">sg-0988dc25ab5c8f6bf</a>	launch-wizard-1	<a href="#">vpc-0059b159ca91e2...</a>

Name	Security group rule...	IP version	Type	Protocol
<input type="checkbox"/>	sgr-056c4eb649bb670...	IPv4	MYSQL/Aurora	TCP



# AWS RDS Security Groups

- Add a new rule
- Select MySQL/Aurora for the rule type
- For the source, click in the input field, and scroll down until you find the “launch-wizard-1” security group
- Click “Save rules”

The screenshot shows the AWS Management Console interface for editing inbound rules for a security group. The breadcrumb navigation indicates the path: EC2 > Security Groups > sg-025e5e60516cdacf1 - database-sg > Edit inbound rules. The main heading is 'Edit inbound rules' with an 'Info' link. Below the heading is a descriptive sentence: 'Inbound rules control the incoming traffic that's allowed to reach the instance.'

The 'Inbound rules' section contains a table with the following columns: Security group rule ID, Type, Protocol, Port range, Source, and Description - optional. There are two rows of rules. The first row has a rule ID of 'sgr-056c4eb649bb67044', Type 'MYSQL/Aurora', Protocol 'TCP', Port range '3306', and Source 'Cu...'. The second row has a rule ID of '-', Type 'MYSQL/Aurora', Protocol 'TCP', Port range '3306', and Source 'Cu...'. An 'Add rule' button is located at the bottom left of the table.

A dropdown menu is open under the Source column of the second rule, showing a search bar and a list of options. The options include '::/48', '::/64', and a section titled 'Security Groups' with three entries: 'database-sg | sg-025e5e60516cdacf1', 'default | sg-03355c8030a1258fe', and 'launch-wizard-1 | sg-0988dc25ab5c8f6bf'. Below this is a section titled 'Prefix lists' with two entries: 'com.amazonaws.us-east-1.dynamodb | pl-02cd2c6b' and 'com.amazonaws.us-east-1.ipv6.route53-healthchecks | pl-05c0959a59362110e'.

The footer of the console shows 'CloudShell', 'Feedback', and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie'.

# ECS → RDS

## Connecting at last

- Connect to your EC2 instance using your method of choice
- We need the mysql client software
- Docker!
- Use the hostname for your RDS instance, and the password you wrote down for the admin user (you did write down the password, right?)

```
sudo docker run -it --rm mysql:latest mysql -h class-db...rds.amazonaws.com -u admin -p
```



# ECS → RD

## Connecting at

- Docker lets us run programs without installing them permanently

```
ec2-user [SSH: csc346ec2]
$ mysql-docker.sh
$ mysql-docker.sh
1  #!/bin/bash
2
3  sudo docker run \
4  -it \
5  --rm \
6  mysql:latest \
7  mysql -h class-db.cb2k68ew0n3c.us-east-1.rds.amazonaws.com -u admin -p
8
```

PORTS SERIAL MONITOR DEBUG CONSOLE TERMINAL PROBLEMS ... bash + -

```
[ec2-user@ip-172-31-21-172 ~]$ sudo docker run -it --rm mysql:latest mysql -h class-db.cb2k68ew0n3c.us-east-1.rds.amazonaws.com -u admin -p
```

SSH: csc346ec2 0 0 0 Ln 8, Col 1 Spaces: 4 UTF-8 LF Shell Script

# ECS → RD

## Connecting at

- Docker lets us run programs without installing them permanently

```
ec2-user [SSH: csc346ec2]
$ mysql-docker.sh
$ mysql-docker.sh
1  #!/bin/bash
2
3  sudo docker run \
4  -it \
5  --rm \
6  mysql:latest \
7  mysql -h class-db.cb2k68ew0n3c.us-east-1.rds.amazonaws.com -u admin -p
8

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 30
Server version: 8.0.35 Source distribution

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```



# ECS → RDS

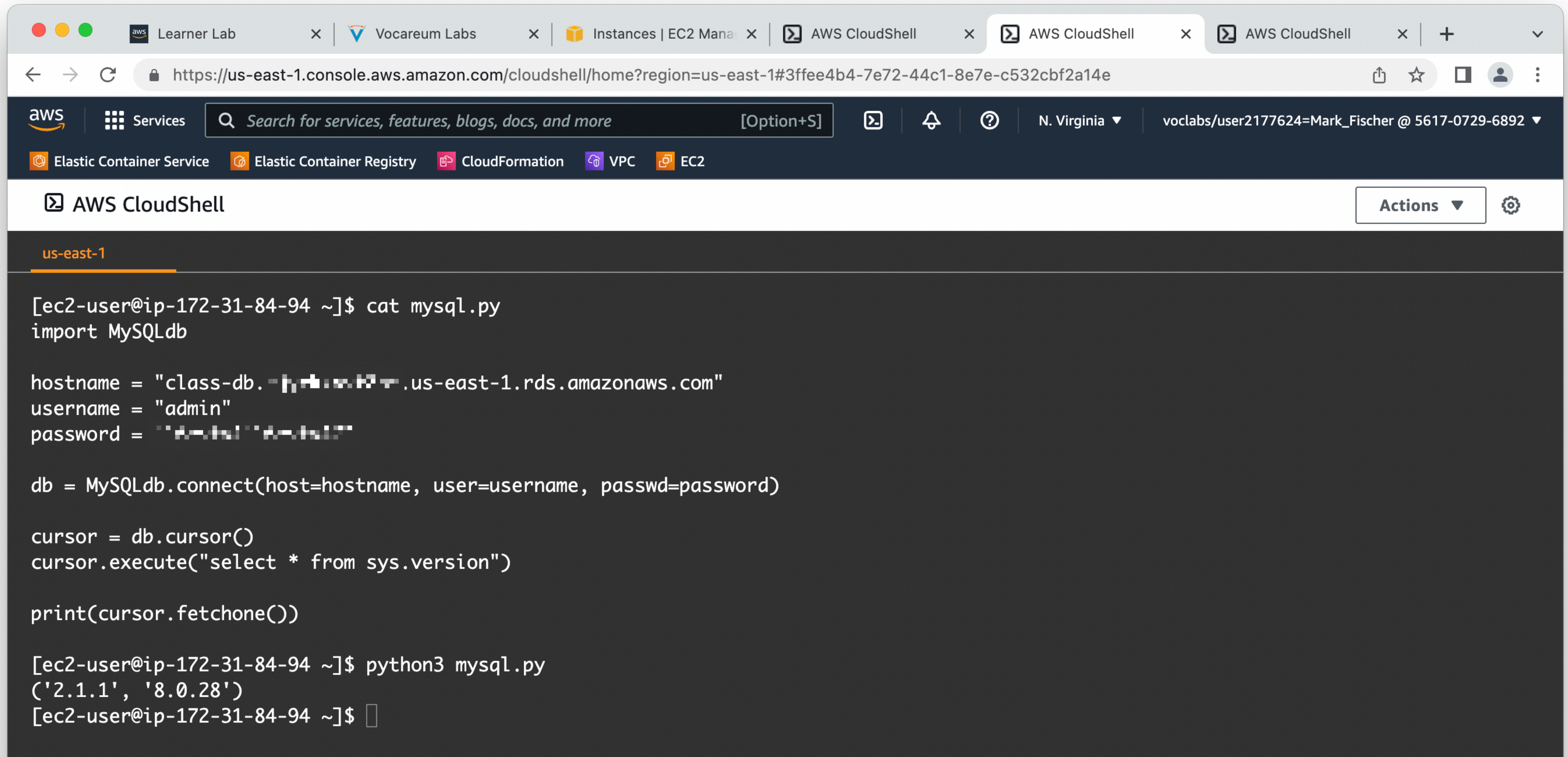
## Connecting from python

```
sudo yum install mariadb105-devel gcc python3.11-devel python3.11-pip  
sudo pip3.11 install mysqlclient
```

- Now we can use the `MySQLdb` module within python on our EC2 instance.

# ECS → RDS

## Connecting from python



The screenshot shows the AWS CloudShell interface. The terminal displays the following Python code and its output:

```
[ec2-user@ip-172-31-84-94 ~]$ cat mysql.py
import MySQLdb

hostname = "class-db-1-1-1-1.us-east-1.rds.amazonaws.com"
username = "admin"
password = "password"

db = MySQLdb.connect(host=hostname, user=username, passwd=password)

cursor = db.cursor()
cursor.execute("select * from sys.version")

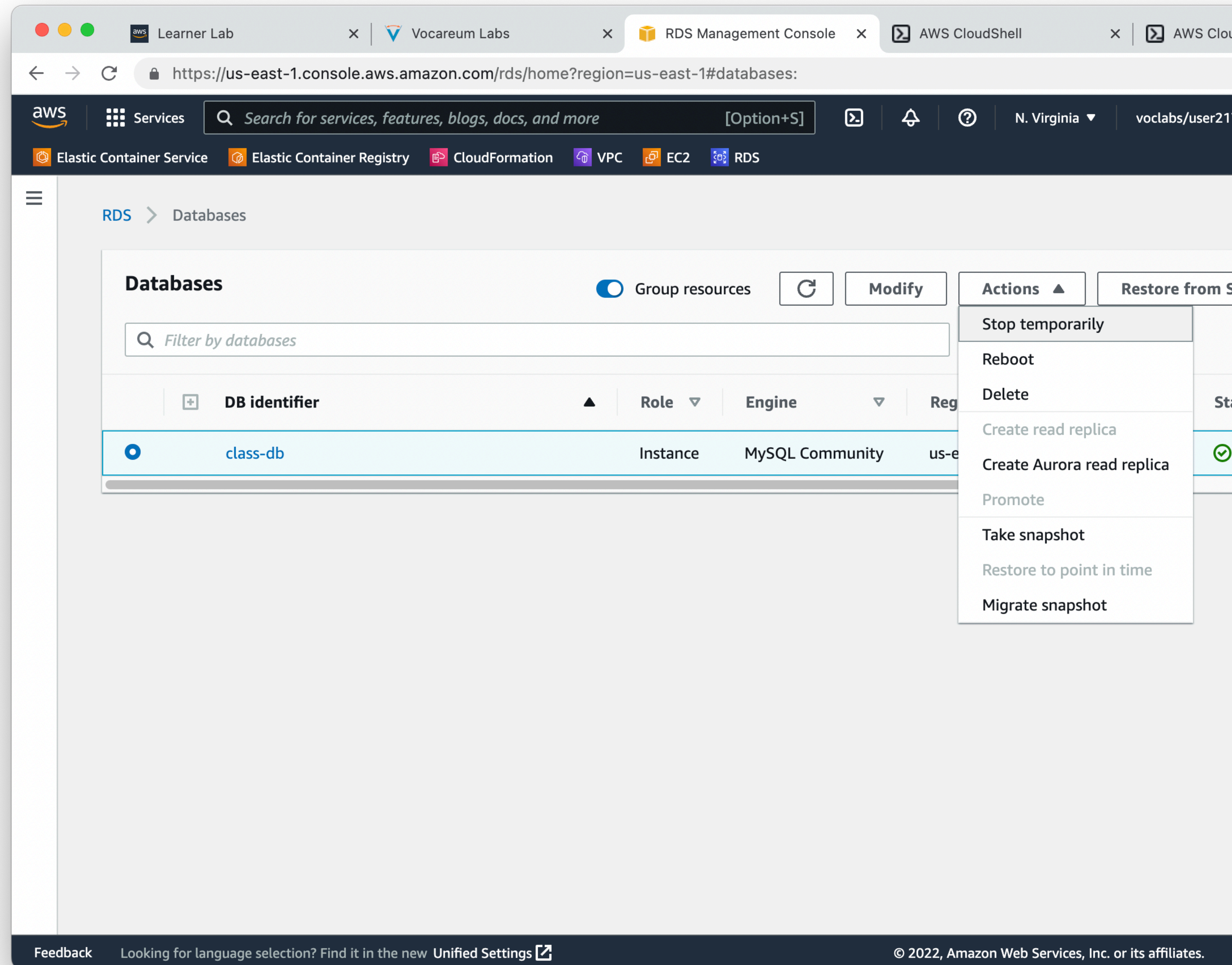
print(cursor.fetchone())

[ec2-user@ip-172-31-84-94 ~]$ python3 mysql.py
('2.1.1', '8.0.28')
```



# AWS RDS Cleaning Up

- RDS instances are NOT automatically stopped when your AWS Academy lab session ends
- You will keep getting charged as long as it is active
- You can temporarily stop an RDS instance though





# AWS RDS

## Cleaning Up

- Can stop an RDS instance for up to 7 days
- After that it will automatically restart so AWS can keep it patched
- Still have to pay for the storage
- If you are done with an RDS instance, terminate it instead

