Hello, Cloud Gurus, and welcome to this lesson,

which is an "RDS Demo."

And we'll begin by creating an RDS instance

using the AWS console.

Next, we'll launch an EC2 instance,

and we'll install the MySQL database client.

And then finally, we're going to connect to our RDS instance

from our EC2 instance

using the MySQL database client.

So if you'd like to join me in the AWS console,

we'll get started.

So here I am in the console,

and the first thing I'll do is search for RDS.

Select RDS.

And then scroll down and select Create Database.

Make sure that Standard Create is selected.

And then here are all the different database engine options.

And we're going to select MySQL.

Scroll down,

and the default version is fine.

Under templates, we've got a choice of production,

development test, or free tier,

and I'm going to select Free Tier.

Under availability and durability,

this is where you can configure multi-AZ,

and in fact if we'd selected Production,

you can see that with the production template,

they actually give you a multi-AZ deployment by default.

So this will create a standby database

in a different availability zone.

So if something goes wrong with your primary instance

or the primary availability zone,

then your RDS instance is going to fail over automatically

to the standby instance.

So you will not lose your database.

But we don't need that for this demo.

So make sure you go back up and select the Free Tier

because it really takes a long time

to create that second instance.

And for the purposes of this demo, we don't need it.

Moving down to settings, this is where we're going

to configure our database identifier,

and this is just a name for our database.

And the name for this database is going to be acloudguru.

And I'm going to copy that name to my clipboard

because we'll use it later on.

Under master username, that's going to be acloudguru as

well. Now it's possible to autogenerate a secure password,

but we're going to define our own.

So down here, just type in a password of your choice,

and confirm your password,

but just make sure that it's a password

that you are going to remember

because we're going to use it later on.

Under instance configuration,

we'll stick with the t3.micro.

Under storage, this is where you can allocate

the storage for your database.

The minimum is 20 gig,

and by default, they enable storage autoscaling.

So this allows the storage to increase

after a specified threshold is exceeded.

Down here, you can configure a maximum storage threshold,

and I'm just going to change that to 100.

Scrolling down to connectivity,

select Don't Connect To An EC2 Compute Resource

because we're going to do that later.

And this is where you can define

which VPC this RDS instance is going to be created in.

And we'll just stick with the default VPC,

and it's reminding us that after the database is created,

you won't be able to change its VPC.

Down here, we can select the subnet group,

and we'll just stick with the default.

By default, there's no public access to the database,

and that's fine.

So it's not going to assign a public IP address

to your database.

Then under VPC security group,

we're going to create a new security group.

We'll give it a name,

and I'm just going to call it rds-sg.

Down here, we can select which availability zone

the database is going to be created in,

but I'm going to stick with No Preference.

And then, if you select Additional Configuration,

this is where you'll find the database port.

So this is a port that we can connect to the database on.

And by default for MySQL, it's 3306,

but if you want to change it, this is where you can do that.

Then under database authentication,

this is where you can define how we're going to authenticate

to our database when we are connecting.

So make sure that password authentication is selected, and

we're just going to use the password that we input earlier,

but there's also password with IAM authentication

and password with Kerberos authentication as well.

But don't worry, both of those

are out of scope for the exam.

Under monitoring, you can enable enhanced monitoring,

and when you do, you can set the granularity.

And when you can configure enhanced monitoring,

it will deliver the metrics to CloudWatch logs as well,

and you will also be charged for enhanced monitoring.

So I'm going to deselect that.

Then under additional configuration,

we can give our database an initial name.

It's going to be acloudguru.

The parameter group is used for providing parameters

and optional functionality,

so you can pass in configuration settings here.

But don't worry about that,

it is beyond the scope of the exam.

And then we also have this option group,

which allows you to configure support

for things like SQL Server encryption

or MySQL memcached support.

But again, this is all out of scope for the exam as well.

Onto backups.

And by default, it enables automatic backups

with a backup retention period of 7 days,

and you can change that to anything between 0 and 35 days.

And then down here, you can also select a backup window.

So you can set the backup to happen

at a time that is appropriate for your application.

And I'm just going to select No Preference.

And then by default,

it's going to copy tags to your snapshots,

and that's just going to make it easier

for you to organize them and find them when you need them.

Encryption is enabled by default.

And then down here, you can select the encryption key

that you want to use.

Under log exports, you can select the logs

that you want to send to CloudWatch logs.

And then in the maintenance section,

this is where you can configure a maintenance window.

So Amazon is going to apply minor upgrades

to your database software,

and you can define when you want those to happen.

For instance, you might want them to only happen

at 2:00 AM on a Saturday night.

Then down here, you've got deletion protection as well.

And this protects the database

from being deleted accidentally.

However, do beware that if you enable this option,

then you will not be able to delete the database

until you come in and disable the option.

So just make sure that's disabled for now.

And then right at the end,

it's going to give you the estimated monthly costs.

And if you're within the free tier,

then you shouldn't see any costs.

So now, let's go ahead and create our database.

So there we go.

That is launching now,

and it does take a few minutes to complete.

So while we're waiting,

let's go ahead and create our EC2 instance.

So up in the search box, I'm going to search for EC2

and open it up in a new tab.

Select Launch Instance.

We'll call it My Application Server.

Select the Amazon Linux AMI.

The instance type is going to be t3.micro.

We're going to create a new key pair.

Under network settings, select Edit.

Make sure that auto-assigned public IP is set to enable.

And then under the security group,

I'm going to change the name of my security group

so that I can identify it later.

And I'm just going to call it ec2-sg.

And then you just want to scroll down

to advanced details and select that.

And then scroll all the way down to the end

until you find the user data section.

So we're going to add a little bootstrap script

that you will find in the resources section for this course.

So here's my script.

And first of all, we're going to tell the operating system

to use the Bash interpreter.

Next, we'll run yum update -y

to update the operating system.

And then finally, we'll install the MySQL client.

And it's this MySQL client that we can use

to connect to our database as soon as it's ready.

So if you're happy with that, just select Launch Instance.

So that's successfully initiated.

I'm just going to select my instance.

And it might just take a minute or 2 to launch

because we are doing the yum update

and the MySQL installation.

So just be patient, and it should be ready soon.

So now that my instance is ready,

the next thing we're going to do is log in

and make sure that everything installed correctly.

So select your instance,

select Connect,

and we're going to connect using EC2 Instance Connect.

So hit Connect.

And there we are on our EC2 instance.

So now, let's check that the MySQL client

was installed correctly.

And to do that, just type mysql --version and hit Enter.

And there we go.

Our MySQL client has successfully installed.

So now the last thing we need to do

to get everything working

is we need to configure the RDS security group

to accept incoming connections from our EC2 instance.

So in the search box,

we can actually search for security groups.

And there we go.

I'm going to open it in a new tab.

So here's our security groups,

and this is the one that we need to update, rds-sg.

So select that.

Then select Inbound Rules.

And we need to edit our inbound rules.

We're going to add a rule,

and the rule is going to be for MySQL.

The port is 3306,

and the source is going to be the security group

of our EC2 instance.

So in the search box, just type sg,

and there is our EC2 security group.

So select that one,

and Save Rules.

So now that we've updated our security group,

we should now be able to connect from our EC2 instance.

So come back to your session on your EC2 instance.

And then this is the command that we're going to run.

So we're using the MySQL client.

The username is acloudguru.

p means that we're going to provide the password,

and the client will ask us for the password

once we've hit Enter on this command.

h is what we use to provide the endpoint name

of our database.

So we need to find the endpoint of the RDS database.

So come back to your RDS management console,

select the database.

And then here, under connectivity and security,

here is our endpoint.

So just copy that,

come back to your EC2 instance,

paste in the RDS endpoint,

and then finally, we need to provide the name

of the database, which is acloudguru,

and hit Enter.

Now it's asking us for the database password.

So hopefully you remembered it.

So enter the password and hit Enter.

And if it's all worked, this is what you should see.

So we have successfully connected to the database,

but if we don't have anything stored in this database,

we can't actually run any queries.

But what we can do is use this status command,

and this will give us the status of the database.

So here's the current database we are connected to,

the current user, and the IP address,

the database version,

here's our endpoint,

the port that we're connected to,

and how long the database has been running.

And then another command that we can run from here

is show databases,

and we need to add a ; on the end of the command,

so don't forget that.

And this will give us information

about the database that we are connected to.

So there is our database name.

So there we go.

We've connected to our database

and run a couple of commands.

And when you are done with using the MySQL client,

you can just type exit to quit the database connection.

So in summary, we've built our own RDS database,

we launched an EC2 instance,

and installed the MySQL client on our instance.

We updated the RDS security group

to allow connections from our instance,

and then we were able to connect to the database

from our instance using the MySQL database client.

So that is it for this lesson.

Well done, Cloud Gurus.

And if you have any questions, please let me know.

Otherwise, I will see you in the next lesson.

Thank you.