Hello, Cloud Gurus, and welcome to this lesson,

where we'll be launching an EC2 instance with an S3 role.

So let's take a look at the objectives for this lesson.

First of all, we're going to create an Identity

and Access Management role with S3 access.

Next, we'll create an EC2 instance,

and we'll attach the role that we just created.

And then finally, we're going to try

and access S3 from our EC2 instance.

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'm going to do is search for IAM.

Select IAM, and we'll head over to Roles.

Now, IAM roles are, of course,

a secure way to grant permission

to entities that you trust.

For instance, an IAM user in another account,

application code running on an EC2 instance,

or an AWS service that needs to perform an action

on resources in your account.

So basically, you use a role

to grant permission to entities within AWS,

and that's exactly what we'll do now.

So select Create Role, and then you need to select

which type of trusted entity we are creating the role for.

And it can either be an AWS service, like EC2 or Lambda;

another AWS account; a web identity;

for instance, Cognito, which we'll be learning about

later on in this course;

another identity provider, for instance,

a SAML 2.0 compliant identity provider,

like Facebook or Google; or even federation

with a corporate directory, like Active Directory.

And you can also create a custom trust policy,

enabling an IAM user from another account

to perform actions in this account.

But we are going to be creating a role

for an EC2 instance.

So just make sure that AWS Service is selected.

Then down here, under common use cases,

select EC2, and then hit Next.

And this is where we add a permissions policy to our role.

And this is going to be for S3 access.

So in the search box, just search for S3.

And we're going to give S3 full access

because we want our EC2 instance

to be able to create buckets, delete them,

add files, and read files as well.

And then if we click on this policy name,

it's going to open the policy up in a new tab.

Make sure that JSON is selected.

And then this is our policy document.

So in the policy statement here,

these are the permissions defined by the policy.

So the effect is allow and then it's allowing these actions.

And with this wildcard,

it's basically allowing all S3 actions.

And this is going to apply to all resources within S3.

So basically, this role is allowed to do anything inside S3

but it doesn't have any other permissions.

So our EC2 instance won't be allowed

to read DynamoDB tables, for example,

or launch another EC2 instance.

It only has permissions for S3.

So now I'm going to close that tab,

come back to my role, and scroll down to the bottom.

Hit Next, we need to give our role a name,

and I'm going to call it MyS3Role.

We can optionally add a tag, and I'm going to add a tag of

Team. And the team name is going to be Developers.

So now hit Create Role.

And that is our role created.

And if you search in the search box for "My",

there is our role.

So now, we need to attach our role to an EC2 instance.

So in the search box at the top, search for EC2.

I'm going to right-click and open in a new tab.

Then select Launch Instance.

I'm going to call it My S3 Instance.

Use the Amazon Linux AMI.

Instance type will be t3\_micro.

Scroll down to key pair.

I'm going to create a new key pair.

Give it a name and Create Key Pair.

Under network settings,

make sure that auto-assign public IP is set to enable.

And then scroll down to advanced details.

And it's here under IAM instance profile

that you need to select the role that you just created.

So make sure that your role has been selected.

And then you can go ahead and launch your instance.

And while it's initializing,

just select the instance ID here.

It might take a few moments to complete,

but as soon as it's finished initializing,

then you will be good to go.

And you can refresh this view using this button.

Once your instance has finished initializing

and it's past its status check, we are ready to log in.

And we're going to log in using EC2 Instance Connect.

And to do that, just select your instance on the left

and then select Connect at the top.

We'll use EC2 Instance Connect, hit Connect.

And it might just take a few seconds to connect.

And this method is not going to be covered in the exam--

it's just a quick and easy way

for you to connect your EC2 instance using SSH

from inside the AWS console.

So there we are, we are on our EC2 instance.

I'm going to clear the screen.

And now, let's test if we can access S3.

So type aws s3 ls.

There's no error message so that's a good sign.

And if you did have any S3 buckets in this account,

then they would show up here.

But let's see if we can create a new S3 bucket.

So type aws s3 mb, for make bucket.

Then s3:// and your bucket name.

And I'm just going to add some random numbers on the end

to make a unique bucket name.

And there we go.

We've successfully created an S3 bucket.

So now if we type aws s3 ls,

it's showing us the bucket name in our account.

So now, let's create a new file.

So we're just going to create a simple text file.

So type echo "This is a new file"

And then redirect that &gt; to a new file.

And the name of our file is file.txt and hit Enter.

There's our file.

And now we can just run aws s3 cp, for copy,

and then the name of our file.

And then I'm just going to copy and paste

the name of my S3 bucket and hit Enter.

So that has uploaded my file.

And then we can check that it's in there

by running aws s3 ls and the name of our bucket.

And hit Enter.

And there we go, it's found our file in our bucket.

So now we've verified that our EC2 instance

is able to read and write to S3.

And if we run aws configure list and hit Enter,

we can see that it's actually provided

an access key and secret access key.

And these are provided by the IAM role

that we associated with our EC2 instance

when we launched it.

And this is what is allowing our EC2 instance to access S3.

So what do you think would happen

if I go in and detach this role?

Well, let's try it and see.

Head back to the console, select your instance--

make sure it's selected on the left-hand side here--

select Actions, Security, and Modify IAM Role.

Here's the role that we attached.

And from the dropdown, select No IAM Role,

Update, and we need to confirm.

So we need to confirm that we want to detach this role.

Hit Detach, and then if we come back

to our Instance Connect console,

and sometimes when you leave it and come back,

you've lost your session.

So if that does happen, then just go back to your instance,

select your instance, and Connect again.

So from Instance Connect,

we'll run aws configure list again and hit Enter.

And there we go.

We can see that we've lost

our access key and secret access key.

So the credentials have gone,

and our permission has been taken away.

So if we run aws s3 ls

and our bucket name and hit Enter,

it's saying it's unable to locate our credentials,

and we need to run AWS Configure.

So that's pretty cool, right?

And we can actually go back in and reattach that role.

So let's come back to our instance.

Select Actions, Security,

we'll modify the IAM role again.

Select your S3 role and Update.

Come back to our Instance Connect session,

and we're going to run AWS configure list again,

and we'll see that our credentials have come back.

And you'll notice that they're actually different this time.

If you look at the last 4 characters of your access key

and secret access key, they're actually different

to the credentials that we had previously.

And that is because these are only temporary credentials.

And when you get credentials that are provided

by the IAM role, they are always temporary.

But if we go ahead and run aws s3 ls and our bucket name

and hit Enter, we've got the same access that we had before.

So what are my exam tips?

Well just remember that roles

are the preferred option from a security perspective.

And if you do see any questions in the exam

where they're asking you about how to enable

your application or your EC2 instance

to access AWS resources,

then the best practice is to use roles.

You should always avoid hard coding security credentials

in your application or on your EC2 instances.

And roles allow you to provide the access that you need

without you having to configure any access keys

and secret access keys--it's all done for you.

Policies are used to control the roles permissions.

So if you remember, we attached the S3

full access policy to our role,

and you can update a policy attached to a role,

and it's going to take immediate effect.

And as we saw earlier, you can attach

and detach roles from running EC2 instances

without having to stop or terminate the instance.

And when you do, it will take immediate effect.

So that is it for this lesson.

If you have any questions, please let me know.

Otherwise, I will see you in the next lesson.

Thank you.