Hello, Cloud Gurus,

and welcome to this lesson,

where we'll be getting our hands dirty with Route 53.

So what is Route 53?

Well, Route 53 is Amazon's DNS service,

and it allows you to map a domain name that you own

to EC2 instances, load balancers, and S3 buckets.

And in this demo, we are going to launch an EC2 instance.

We'll install httpd.

We'll then create an application load balancer.

And we're going to configure our EC2 instance

as a target for the load balancer.

And then finally, we'll configure a Route 53 alias,

and we'll use this to map a friendly domain name

to our application load balancer,

so that we can access our website

using our own domain name.

And let's say, for example,

that we own a domain name called ilovecloud.com,

then we could create an alias,

which allows us to access the website

using that domain name

instead of having to use

the application load balancer DNS address.

Now we'll be doing everything

using the A Cloud Guru sandbox.

And we've provided a Route 53-hosted zone for you to use.

And a hosted zone is simply a container

for DNS records.

And it's already configured

with a DNS-registered domain name that we own

(well, it's actually owned by the Sandbox account).

So if you'd like to join me

in the AWS console, we'll get started.

So here I am in the console,

and before we build anything,

the first thing I want to show you is Route 53.

So search for Route 53 up here.

And don't worry if you see any error messages like this,

it won't affect anything that we are doing here,

and they just mean that you don't have access

to see everything in this account.

Now first of all,

I want to go straight to hosted zones on the left,

so select that.

And you should see something similar to this

because we've already purchased a domain name

for this account.

And within Route 53, they call this a hosted zone.

And it's basically just a container

for all of your DNS records.

So now that we've identified our hosted zone

and our domain name,

we can go ahead and create our EC2 instance.

So search for EC2.

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

From the EC2 dashboard, select Launch Instance.

We'll call our instance, My web server.

Scroll down and make sure that Amazon Linux is selected.

Instance type is t3.micro.

Under key pair,

we'll proceed without a key pair

because we don't need to log into this instance.

Under network settings,

select Edit.

It's automatically selected our default VPC.

But for subnet,

I want you to select a subnet

in the us-east-1a availability zone.

So make sure that us-east-1a is selected.

Auto-assign public IP should be set to enable.

Then scroll down to firewalls and security groups.

And we're going to change the name

of our security group,

so that we can identify it later on.

And I'm going to call it mywebdmz.

Under the security group rule,

we're going to update this

because we don't actually need SSH access for this instance.

But what we do need is HTTP,

so make sure that's selected.

Protocol is TCP.

Port is port 80.

And source type is Anywhere.

Now scroll down to the bottom

until you get to Advanced Details, and open that.

Then scroll right down to the bottom once again,

until you get to the user data section.

And this is where we're going to add a bootstrap script

to run some commands

the very first time this system boots.

And there's a link to this script

in the resources for this lesson.

So you can just copy everything

and paste it into user data.

Now this script just starts off

by telling the operating system

to use the Bash interpreter.

Next, we'll run yum update -y

to update the operating system,

install httpd because this going to be a web server.

This command just creates a very simple webpage

that's going to be stored in /var/www/

html/index.html.

And then these 2 final commands simply start httpd

and then enable it to start every time the system boots.

So once you've added your user data,

you can go ahead and select Launch Instance.

Then, select the instance ID.

And while it's initializing,

we can go ahead and create the load balancer.

So on the left-hand side,

scroll down until you get to Load Balancing.

Select Load Balancers

and Create Load Balancer.

And if you remember,

there are a few different types of load balancer.

So we've got the Application Load Balancer,

which load balances for HTTP and HTTPS.

We've got the Network Load Balancer,

which works with TCP, UDP, and TLS.

And this is the one to use for ultra-high performance.

So it's the low latency option.

There's also the Gateway Load Balancer,

which is the one to use

if you need to load balance

third-party virtual appliances.

Then down here, right at the bottom,

depending on your Region,

you might see the Classic Load Balancer as well.

However, this is the previous generation,

and they kept it going

just to support existing applications

that are on the EC2 Classic network.

But we are going to create an Application Load Balancer.

So make sure that you select that one and hit Create.

We'll give it a name and I'm going to call it myalb.

It's going to be internet facing,

using IPv4.

If we scroll down to the network section,

it's already selected our VPC.

Under mappings, we need to select

the availability zones

that we want our elastic load balancer to operate in.

And you have to select at least 2 Availability Zones.

So first of all,

we'll select us-east-1a

because that's where we created our instance.

And I'm also going to select us-east-1b.

Scroll down to security groups,

and we're going to select the security group

that we created earlier.

And it's this one, mywebdmz.

So this is going to allow HTTP traffic

on port 80 to reach the load balancer.

And I'm going to delete the default security group

because we're not going to need that one.

Scrolling down to listeners,

our listener is going to be listening for the HTTP protocol

on port 80, and we need to define a target group.

So this is going to be the target instances

that we are going to direct our traffic to.

So select Create Target Group,

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

The target type is going to be instances,

but we can also specify a target type of IP address,

lambda function, and even Application Load Balancer.

And they describe some use cases,

just describing why you would want to do that.

But don't worry, that's out of scope for the exam.

Scrolling down to target group name,

we're going to call it mytg.

Protocol is HTTP and port is 80.

It's already selected our VPC.

We'll stick with the default protocol version of HTTP1.

Under health checks,

the protocol is HTTP.

For the health check path,

we just need to add the name of our index.html file.

And as it's in the default location

of /var/www/index.html,

just the file name will do,

and we don't need to specify the full path.

It's going to find it automatically.

Then hit Next.

So select your instance,

and before you can add the target,

you need to select this button here,

Include As Pending Below.

So click that.

And then after you've clicked that,

you can review the targets,

and your EC2 instance should have appeared here.

Then select Create Target Group.

So that is our target group created.

And if you click on your group,

it should show your instance down here.

So now we can head back to our load balancer screen

on the other tab,

and we need to select our target group.

So select this dropdown.

So now scroll down to the bottom

and Create Load Balancer.

Then select View Load Balancer.

And then down here under listeners,

we can see that by default,

it's forwarding everything to my target group.

So if you select your target group

and select it over here as well,

you can see that target registration is still in progress.

But in a few minutes, it should show

that our instance is healthy.

So I'm just going to refresh.

And if it's all working correctly,

the instance in your target group

should eventually show as healthy.

So now, if we go back to our load balancer,

and you can select it here,

copy the DNS name of your load balancer.

We'll paste it into a new browser tab.

And hit Enter.

We should be able to access our website.

And there we go.

So we've established that we can reach our website

using the DNS address of our Application Load Balancer.

But the problem is

it's not really a very catchy web address, is it?

If somebody asks you the name of your website,

you're going to feel a little bit silly

giving out this kind of address.

And that is where Route 53 comes in.

So we're going to use Route 53

to give us a much nicer DNS name for our website.

So just come back to the consult.

Search for Route 53.

Select Hosted Zones.

And select your hosted zone.

So we're going to create a DNS record.

And essentially, we are creating a mapping

between our Route 53 domain name,

which is this domain name,

and the Application Load Balancer.

So the record type is going to be an A record.

Next, select Alias.

And we're creating an alias,

because when we type this top-level DNS address

into our browser,

we want it to resolve to the elastic load balancer.

And as it says here,

we are creating a record for the root domain,

which is this one.

And we can only do that using an alias record.

Down here, we can select

where we are routing our traffic to

and choose an endpoint.

And you want to select an Application Load Balancer.

The region is going to be us-east-1.

And then you should be able to select your load balancer.

And there it is.

Scrolling down to routing policy,

and we have a choice of different policies.

But we are just going to be using simple routing,

because we are going to be using standard DNS

to resolve to a single target,

which is our load balancer.

And if you select Info on this side,

it's going to give you a rundown

of all the different routing policies

that are available.

But as we're just going to be routing our traffic

to a single load balancer,

we're going to select Simple Routing.

So once you've done that,

select Create Records.

So that is our record created,

and you'll see that it's created a mapping

between this DNS address, our domain name,

and the DNS address of our elastic load balancer.

So now, if we copy this domain name,

open a new tab,

paste in our domain name, and hit Enter,

it should take you to your website,

via the elastic load balancer.

And now, if somebody asks for your website address,

you can just give them this simple domain name,

instead of giving them the DNS address

of your elastic load balancer.

So that is pretty much everything

that you need to know about Route 53 for the exam.

So just to recap,

we created a web server, and we installed HTTPD.

We created an Application Load Balancer

and configured our EC2 instance as a target.

And then we configured a new alias

in our Route 53 hosted zone.

So recreated an A record,

which was also an alias,

to map our domain name

to the DNS address of our Application Load Balancer,

which allowed us to access our website

using the domain name that we own.

And it would've been a domain name

similar to this one.

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.