So it's active and if I copy the DNS name into my clipboard and move across my browser, I should find that I can open that DNS name and access the three cat web web servers using this load balancer, so I'll just test it out quickly. Go to FireFox. I'll paste this URL in, press enter, hit refresh a couple of times, and you'll see that it's rotating between those existing cat web servers. So that's working as intended is the same functionality as we had with the classic load balancer.

Now, the way the application load balancer is configured as I've hinted to earlier in this lesson is slightly different. If a go to listeners configuration, essentially, everything is configured using rules. So in this particular case, we've actually got a default rule that's in place and what this default rule does is forward any traffic which hits this load balancer through to the cats target group. So that's all it does. So that's the reason why this is currently operating in a similar way to a classic load balancer. **We have a single default rule. That rule is applied whenever any traffic enters the listener, and it's directed through to this cats target group**. Now to illustrate the differences I wanted to go ahead and do is go to instances and I'm going to create three brand new instances, and these are going to be dog web one, dog web two, and dog web three. So the same exact configuration as the cat web servers only they're going, of course, serve pictures of dog. Now for this lesson, so inside this lesson's folder I have provided another three bootstrap scripts which automatically configure the web servers. So I'm going to use those to configure three additional servers. So I'm going to go to launch instance, select Amazon Linux 2, we do know most of this by now, so I'm going to skip through really quickly going to go to configure instance details. Make sure that I pick 1a, 1b, and 1c respectively, for each of these three servers and then for each one of them, I'm going to scroll down go into advanced details and paste in the relevant bootstrap script. So in this case, I need the first one. So bootstrap one. So I got that in my clipboard paste that in. It does exactly the same as the previous bootstrap scripts only this time it's using dog pictures. Don't forget to add storage accept the defaults. I'll be adding a tag to each of these. The key will be consistent. It will be name and then for the value, I'll pick dog web one. Go to security group. I'll create a new security group. I'll call it dog web. I use the same for the description. I'll add a new rule HTTP and I allow connectivity from the application load balancer security group and just in case you need to do any diagnostics I'll also add another rule again selecting HTTP. This time I'll change it to my IP address. So if I do need to diagnose this at all, I've got the option to do so. So review and launch launch it. I'll use the LB keypad I've used before and launch that instance and then I'm just going to follow exactly the same process and create dog web two and dog web three and there we go I've started off the creation of these three brand new web servers dog web one, two, and three. Next what I want to do is to create a second target groups. We've already got a target group called cats, which contains those original three cat web servers. Now I want to create a corresponding target group called dogs. So I'm going to go ahead and create this so I'll name the target group dogs. It will also use the instance as the target type. The protocol will be HTTP. The port will be 80. I'll have to select the VPC that they're in. For the health check settings, I'm going to make sure this also has HTTP as a health check and then index.php is the path. Then for advanced self check settings, I'll follow the same process. So I'm going to change this to healthy and unhealthy threshold, for the time out four seconds and for the interval five seconds. Once I've done that I'll create. Then I'll select the dogs target group, click on the target's tab. I'll go ahead and add targets as registered targets inside this target group. Now again, if you're going to be following along with this in production. You'll probably be integrating this with an auto scaling group to automate this process, but I want to demonstrate the architecture so it's easier to do it manually. So click on edit. I'm going to order these by name. I'm going to select dog web 1, dog web 2, dog web 3. Go ahead and click add to registered and then click on save. So now that they're registered, the status will show unused and that's because this target group is not used by any load balancers and so there'll be no health checks so logically, what I need to do is to go to load balancers, go to the application load balancer, and start utilizing this target group as part of this application load balancer. Now, at this point, I want to demonstrate some of the really cool functionality of the application load balancer product and to do that, I'm going to go to the listeners tab and I mentioned earlier how application load balancers use a concept called rules or content rules. Now, the way that this works is because with a classic load balancer it worked layer three and four of the OSI seven layer model and so it didn't understand any of the details about the HTTP or HTTPS protocol. So, while it understood enough to use a health check and to check whether a certain web page or a certain path was actually working as expected, it couldn't make any decisions around load balancing based on any aspect of HTTP and that's an area where the application load balancer could do a lot better. Now to illustrate that, we're looking at the main listener for this application load balancer, and I'm going to click on view and edit rules. Now all we have at the moment is a default rule. Essentially, this rule takes effect for any traffic that arrives at a website that this load balancer is managing. So let's say, for argument's sake, this load balancer is for pets.com. What we have at the moment is for all traffic that arrives at pets.com. It will be directed in our case at the cats target group. Now, this is the functionality that we were limited to with a classic load balancer. Classic load balancer could serve for one DNS name for one application, but we might be in a situation for example, where we want a load balancer to do one thing if we browse to amazingcats.com and another thing if we browse to amazingdogs.com and for classic load balancers, we would need two individual load balancers to support that functionality. One for cats, one for dogs. If you're wanting to support SSL so HTTPS and each load balancer it would have one SSL certificate. One of them will be for amazingcats.com, the other for amazingdogs.com. Now that's one area where application load balancers are much better because of the rule functionality. What we can do is define a different content rule that performs different actions based on a number of criteria. So let's look at that. So I'm going to click on the plus, and I'm going to insert a new rule. All we've got at the moment is this default rule that takes effect if nothing else matches. So what I want to do is add a condition. You got a number of different types of condition we can use. We got source IP addresses. So if you want one customer with one IP address to be directed towards one target group and then another customer or group of customers to be directed towards another target group. Well, you've got that functionality. You could also define one which is host header. So like with the previous example I was just talking about. If you want amazingcats.com to be directed at one target group, you'd enter a rule which match amazingcats.com. If you wanted amazingdogs.com to go to another target group. You would have another all that matched amazingdogs.com and those would be host header rules. What's more common is you might have a single website, for example, pets.com and want different load balancing decisions based on paths on that website. So let's have a look at that. So I'm going to add a condition, and it's going to be a path condition, and I'm going to set the value to be /cat/\*. So I want any connection that hits this load balancer and has /cats and then anything to be directed at the cats target group. So I'm going to do that. I'm going to forward that through to the cats target group. Don't go ahead and save that. In addition, I want to add another rule. So we're going to add a condition it's going to be another path based rule this time /dog/\* So I want any connections which you're using the path of /dog and then anything to be forwarded through to the dogs target group. So I'm going to set that and click on save.

**Now this is one of the advantages of the application load balancers supporting layer seven of the seven layer OSI model because it understands the HTTTP protocol it can make decisions based on certain** **attributes of that protocol and the path is one attribute. The host header is another attribute. The source IP address is another attribute. So by supporting layer seven, you get to perform a lot more complex routing decisions and one benefit of that is being able to create this rule base that I've just created.**

**Another benefit is to be able to support lots of individual sites using individual SSL certificates and that's one of the reasons why, in almost all cases, it's cheaper to use application load balancers because at the very least they can support a large collection of websites using individual DNS names, individual domains, paths**. What's all the configurable element? **They can all be supported with a single application load balancer compared with potentially tens or even more classic load balancers that will be needed to support that historically.** Okay, so now we've added these rules. Let's just test it. So if I go to description and just get the DNS name for this load balancer and the open that in a new tab without any paths. Now, because we've got the default rule in place it goes to the cats target group, and that's what displayed on screen now. Now we'll quickly move across to my Firefox browser. Remember, this doesn't suffer from the image caching issue and if I refresh this a couple of times, I can see there is moving between each of the instances in the cat web group of instances. So the cats target group. Now what I can do is go /dog and then forward slash and press enter now because I've got this path pattern matching rule. So because it matches /dog/anything now it's directing these connections through to the dog target group. So this is a picture of Sparky who's the dog of one of my colleagues at Linux Academy so Mike Chamber's dog. If I hit refresh a couple of times, and it's going to cycle through the different dog web servers. So this is a picture of Sparky with the chicken. This is a picture of Sparky on a table, and then we're back again at Sparky sitting on the deck. Now, if I don't have this path, then it's going to use that default rule. So it's going to switch to the default configuration. So under listeners go to rules. If I don't have any path, it's always going to fail back to this default action, which is going to forward through to the cats target group but if I have any specific matching then it's going to do, as the rule dictates, which is to forward it to a specific target group, so this is a really powerful feature of application load balancers. It allows you to implement some really complex architectures. **So application load balancers they're much preferred to using classic load balancers. The functionality is much improved, you've got the ability to support various different elements of the layer seven protocol, they can support EC2 instances, ECS, EKS. They can support Lambda functions. They support web sockets and HTTP 2. So enhancements to the original HTTP protocol.** So if you face any exam questions, **where you do need web sockets, where you need HTTP 2, where you need to be able to point any connections at Lambda ECS or EKS. Then you know you need to use application load balancers.** Now at an associate level, you don't really need to know anything in any more detail but I did just want to give you this theoretical and practical introduction in hope that it will give you a bit more confidence when it comes to answering exam questions. So with that being said, that is everything that I wanted to cover about application load balancers. Go ahead and mark this video as complete, and when you're ready, you can join me in the next where I'm going to be talking about network load balancers.