

Welcome back, and in this lesson I'm going to cover security groups in a little bit more detail. **Security groups are essentially software firewalls that surround AWS products. They're presented in the interface as surrounding an EC2 instance. What they actually do, though, is they're attached to a network interface and by default, every EC2 instance is created with this default ENI or elastic network interface and so the UI presents this as security groups been associated with the instance**, and at this level, that's fine but just be aware that it's actually the network interfaces that secured by the security group, not the instance. **Now each ENI or elastic network interface can have up to five security groups associated with it.** Remember this for the exam because you might get a question which tests your knowledge of this number. **So it's five security groups maximum for every elastic network interface.** Now at a high level, **all security groups belong to one VPC**. So when I create a security group I am required to select the VPC. Now, in addition, **every security group has a unique identifier**: **the security group ID and this could be used to reference this group from anywhere. Now a security group actually consists of two main components or two main parts. We've got a set of inbound rules which affect traffic coming into whatever the security group is attached to and a set of outbound rules which impact traffic leaving the thing that the security group is attached to. So** if I go ahead and name the security group and for now, I'm just going to call it SG test, I'll use that same text for both the name and the description and I'll hit Create. Now if I look at the group that I've just created go into inbound rules. You see that it has no inbound rules, but it does have this single outbound rule. An important concept to understand for the exam and this is really critical so please do make an effort to understand it is that **every security group both the inbound and a outbound rule set has a hidden rule, and that hidden rule is what's known as a default or implicit deny.** **So this default or implicit deny only takes effect if none of the security groups that are being evaluated have a rule which matches so if none of the security groups have any rules which match there's this default rule, and it's a deny rule that take effect.** So let's have a look at how this works in practice. I'm going to go to instances and I'm going to launch an instance I'll pick the default AMI. I'll pick a small instance type so T3.micro. I'm going to skip past most of these options and just leave the defaults. but I am going to paste some user data into this advanced detail section and the user data is located inside the folder for this lesson in the course GitHub repository. I'll make sure it's linked in the lesson description. So I'll paste that user data in and then I'll skip past the rest of the options and leave as default. I won't make any changes to the storage. I won't add any tags on then for security group, I'm going to go ahead and create a new security group. I'm going to call it Security Group Demo. I'll use the same text for both the group name and the description and I'm going to leave this for now as default. I'll go to review and launch. I'll go to launch. I'll create a new key pair. I'll call it Security Group Demo and download it, launch the instance, and then view instances. Now don't mind any of these other instances if you ever log in to on AWS account and see other instances saying terminated or shutting down, especially if you're using any Linux Academy provided accounts, that's fine. I'm only concerned with the instance I've just created, which is this one that's in the running state. So if I select this and right click and hit Connect I'll want to get the command to fix up the permissions on the SSH key pair, so I'll copy that into my clipboard. I'll move across from my terminal, paste this in to fix those permissions. I'll copy the command to connect to the instance into my clipboard, go back to my terminal, paste this in, accept the authenticity, and that'll connect me to the instance. Now, the reason that I've been able to connect to this instance if I move back over to the console, I've got the instance selected, and so I'm able to see any security groups that are associated with this instance and this is a security group demo one that I created when making the instance. So I'm going to select that security group and just evaluate what rules the security group has. I'll make sure that I've got the inbound rule set selected, and we can see that it's got a single rule that was added when I created the instance. I'll click on edit because I just want to evaluate the structure of an individual rule. So every rule is either an inbound rule or an outbound rule. This one's an inbound rule so it matches a certain type of traffic. It's looking specifically for SSH traffic. This uses the TCP protocol, and it uses Port 22. So this rule matches any traffic that uses SSH, TCP Port 22 and it's from this custom IP address. Now 0.0.0.0/o matches any IP address. So this rule matches any SHH traffic coming from any IP address. This is added by default when you create an EC2 instance, if it's a Linux instance, it's SSH. If it's a Windows instance, it's RDP or remote desktop protocol, and this is added so that you can connect to the instance using the protocol that's appropriate so SSH for Linux or RDP for Windows. Now we could tune this a little bit instead of it being 0.0.0.0/0 which is any IP address. We could change this dropdown to my IP. Now, this would only match for my particular eyepiece only SSH connections coming from my IP and it's always best practice to limit these rules wherever possible. So because this is a demo, I'm only going to allow connections from my IP address. So I'll change that and hit save. Now keep in mind and this is important whenever you're looking at a security group, you've got to approach it with these two different rule sets in mind so inbound and outbound. So in this particular case, because this is the only security group that's associated with my instance, this is the only rule set that's going to be evaluated and what happens when AWS is evaluating a security group is it checks if any of these rules matches the traffic and if it does, it allows that traffic. So in this case, this rule will match me trying to connect to this instance from my IP address and so it will allow that. Now remember how I said there was this hidden default or implicit deny well, if there are no rules that match so if somebody else attempts to connect to my EC2 instance, this rule won't match. It would have previously matched because it had this 0.0.0.0/0 catchall but because I've changed it to match only my IP address now anyone else attempting to connect will hit this hidden or implicit deny, and they won't be allowed to connect to my instance. Now, the other thing I want to draw your attention to is if I go to instances and a select this instance, copy the I P address into my clipboard and then attempt to open that in a new tab I'll get this spinning icon. It doesn't allow HTTP or port 80 traffic. There is a web server that's ruling on this EC2 instance because of that user data that pasted in that builds a web server. That's what that user data did, but because I don't have the relevant security group permissions or the security group rule to connect to it using HTTP I'm not able to do that. So on this instance's security group, because web traffic so HTTP or TCP port 80 because that's not matched by this rule, I'm hitting that default implicit deny. Now, to rectify that, I could click on edit and add an additional rule. So I'm going to do that. I'm going to go to add rule. This time, I'm going to change it to HTTP so that's TCP port 80. Now, you are able to define a port range if you wanted to. So if none of these default protocol settings actually works, you could define a custom one and then define your own port range. So if your application uses a group of ports, then you're able to select a port range, maybe 80 to 100 or 80 to 8,000. In this case, it's fine because it matches a single port. Again, I'm going to change it from custom to my IP and that will limit the connections to this instance to only my IP address. Once I've done that you can save and this time I'll be able to load the website that's on this instance. So I'm going to go back to instances again, I'll copy the IP address and then browse to it in a new tab. Now, because I'm not hitting that default implicit deny rule the traffic has been allowed through from my IP address to this website, and it's loading the Cat Hall of Fame. So that's essential that you understand that about security groups. **By default, there is this hidden deny rule and that applies to both the inbound and the outbound. This default or implicit deny rule affects traffic if no other rule in any security groups associated with that instance take effect. So if I was to have a security group that was completely empty associated with this instance and I had this security group then the default or implicit deny in the empty security group wouldn't take effect because I've got rules that take effect in this one. So all of the security groups that are associated with, in this case, an EC2 instance, they're all evaluated at the same time. All the rules or the inbound and outbound rules are all merged together, and they're evaluated at the same time, and only if none of the rules apply does this default or implicit deny take effect.**

Now you might have noticed one thing, and that's that wasn't able to specify allow or deny. This is critical. This is probably one of the most important things to understand about security groups. **You cannot add a deny rule. You have this implicit or hidden deny, and you're allowed to allow traffic through but you are not allowed to deny traffic explicitly. Now the inability to add a deny rule becomes especially important for large enterprise.** Let me give you an example. Let's say I built this website, this Cat Hall of Fame and I wanted to give access to everyone inside Linux Academy. What I would do is I'd try to get the external IP address range for the Linux Academy Network, and I would add it into this source address that would allow everyone from Linux Academy to connect to Cat Hall of Fame but what if there was one bad actor inside Linux Academy, somebody with an infected laptop or somebody who was trying to cause damage to my Cat Hall of Fame website? **With security groups, I couldn't add an explicit deny. I couldn't block that one single person. So in situations where you add entire IP address ranges you can't add an explicit deny for a subset of that range.** It's quite crucial limitation that you need to be aware of the exam. **So it means that wherever possible, you need to add allow rules for very tight groups of IPs. If you want to be able to block specific IPs you can't explicitly deny them. You have just to not allow them and if you allow an entire range then by definition, you allow everyone in that range.** So when you're thinking about security groups, you need to be very careful when you're opening up these allow IP address ranges. In this particular case, I'm the only person using my IP address because I've allowed my IP address as a single IP address and that's what this /32 means. It means that if I wanted to block myself for whatever reason then I could just remove this rule. Likewise, if I allowed somebody else at Linux Academy a single person with a single IP to access my website, and if they became infected with malware, I could block them by removing the rule but if I'd allow the entire Linux Academy IP address range, I then couldn't go back and add an explicit deny that need to remove the whole range and then add everyone individually. So it's actually something you need to be aware of for the exam is **the inability to add explicit denies.**

The one other thing that I want to talk about when it comes to security groups is the fact that they are what's known a **stateful. You might not realize this, but if you use your web browser and connect to a website, the communication between your web browser and that website is actually formed over two different streams of traffic. You've got the originating stream, which is the data from your web browser to the web server asking for that web page, and you've got the returned data known as return traffic, which is coming back from that website back to your web browser.** Now, security groups are able to see that the return traffic for a particular request is part of that original request. It's important to know, because **when you allow, for example, HTTP or SSH to access your server in an inbound way. You're also allowing the outbound return traffic. You don't need to specify it, and in fact you can't not allow it. It's known as stateful for anything that you allow in you're also allowing the return traffic for that out. Now on the outbound rules for security groups by default, there's a rule which allows all outbound traffic so I could connect into my instance**, for example, I'll move to my terminal and let's say that I wanted to do a YUM update, which is the way that I would update the software on this instance, so I'll go ahead and run that command. I don't have any packages to update, but it has gone ahead and made that request. What's actually happened is my web servers used outbound traffic to communicate with some software update services so that's what this rule matches. It allows any outbound traffic, so that's the light green arrow to the software update service and then because of security group is stateful automatically the return traffic for that is allowed, so I don't need to worry when it comes to security groups of are adding individual rules, inbound and outbound for every stream of traffic, they're known as stateful. Not every security product in AWS is stateful for example, **Network Access Control Lists, which we'll talk about in the networking section of the course are not stateful** and that means that you do need to add two rules for every traffic, the outbound and the inbound but for security groups, you don't need to worry about that, they're stateful and they understand that if you allow originating traffic, you're also going to allow the return communication for that stream of traffic. So key points to remember is that **you're actually associating a security group with the VPC. So every security group belongs to a particular VPC. Security groups have an ID and they consist of a collection of inbound and outbound rules. Every security group on a particular entity, let's say this EC2 instance, in this case they're all evaluated at the same time, they're rules are collected together and evaluated as one big set if none of the rules apply, so if traffic is not matched against any of the rules then the default implicit deny takes effect.**

One final capability of security groups, which I'm not going to cover it in detail in this lesson because we'll expand upon this later in the course, is that **as well as IP addresses, you are able to reference other logical entities inside AWS. So I could reference another security group. That means that rather than having to refer to another AWS EC2 instance by its IP address, I could refer to another security group and that would mean that anything associated with that security group so other instances would be referenced when I typed the security group in. I could also have this security group reference itself.** Now, this is a fairly complicated thing to imagine, but I'm going to give you an example. I'll just return this rule to how it was. So I'll put my IP address back in. So in this case, this security group allows HTTP connections from my IP address and SSH from my IP address but let's add another rule and this time I'm going to allow all traffic. So all traffic, this is quite a dangerous rule to add, and you would never do that and allow the public internet to access it using all traffic. That would be a huge security vulnerability but what I might do is add all traffic from itself. So let's say that I selected security group demo. What this is doing is it's allowing all traffic from the security group through this security group. Now that's a powerful feature. What it means is that **for any servers that are associated with this security group, they can communicate with each other using any type of traffic.** I'm not necessarily saying I would suggest you use this in production, but it's another capability of security groups that are quite interesting. I won't be going into detail about how you can use this in practice in this lesson, I'll be revisiting that in the networking section of the course, but I wanted to introduce at this stage that a security group is capable of referencing other logical AWS resources so other security groups and itself. Now at this point, that is everything I wanted to cover in this lesson. I've gone through quite a lot of theory, as well as demonstrating how this works with a Cat Hall of Fame. We will be revisiting this in much more depth in the networking section of the course but security groups are quite relevant to EC2 and so I wanted to introduce the concept at this stage. So go ahead, mark this video as complete and when you're ready, join me in the next.