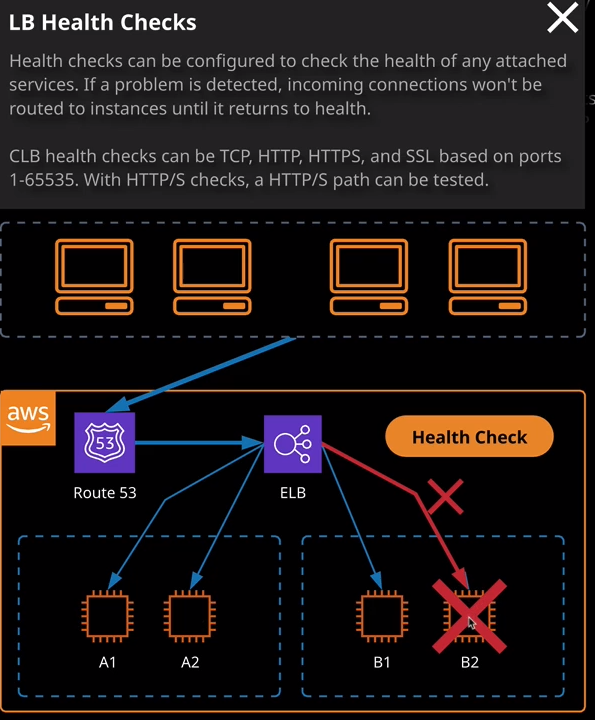
Now the security group of a load balancer is important to understand. The security group defines two things. I'm going to go ahead and actually create a brand new security group for this, and I'm going to call it CLB\_SG for classic load balancer security group and I'll put the same for the description. **Now the security group that's associated with a load balancer defines the type of traffic that is allowed to connect to load balancer.** So of course, we need port 80 and if we wanted to do secure web we would need 443 in addition, but also because security groups a capable of being referenced from each other it does mean that whatever security group is associated with this load balancer, we could in theory, restrict our backend instances to only allow connections from the load balancer. Remember, it's the job of the load balancer to accept connections from the customers and then distribute those connections to the backend instances so we might want to prevent customers using our web servers directly, and that's the way that you can do it. **You can apply this security group and then only allow connections to your instances using HTTP or HTTPS but from the security group of the load balancer**. Now, at this point, I'm going to leave this. I only need port 80. So I'm going to move on to configure security settings. Don't need to configure anything there. So next configure health check.

Now, health checks are a really important part of load balancers. **Essentially, if you configure a health check then the load balancer is constantly performing individual health checks against any attached instances. So an instance has a state. It can either be healthy or unhealthy for any healthy instances and these are instances that pass this health check then the load balancer will send connections to this instance. If it's in an unhealthy state then the load balancer won't send connections** and as you'll learn later in this topic, **if you do integrate a load balancer with an auto scaling group and you tell the auto scaling group to use the load balancer health check, it means that if an instance fails this health check, it will be terminated and recreated**. So this is really important to understand.

So architecturally on the diagram on the right, **I've got this load balancer. It's sending connections to each of these four EC2 instances, so let's say that one of them fails. If it does fail, it'll be marked as unhealthy then the load balancer will direct all new connections coming into the load balancer to instances a1, a2, and b1. Until b2 becomes healthy again it won't receive any more connections. So this is a really powerful feature.**



Now, the way that this works is **you can define a number of different types of health check you can use HTTP which will actually check for a successful HTTP code**. So that's important to understand. Remember when I talked about a **classic load balancer only supporting layers three and four. It isn't a real layer seven device, but it is capable of understanding some HTTP and HTTPS features**. Well, this is one of those features, so you can have **it talk to the backend instances using HTTPD and just verifying that it can achieve a successful connection**. So that's what I'm going to do for this demonstration. So I'm going to choose HTTPD port 80 but instead of index.html, I've configured these three web servers to use a PHP file. **It's the PHP file that actually generates this IP address**. So I need to make sure that it's this that's being checked. So I'm going to go to index.php as the ping path, so it's going to attempt to open index.php on the backend instances, and it will do this periodically. Now, in the advance detail section, it's here where we configure some of the options that allow us to influence this. **So the response time out essentially defines the amount of time that the load balancer will wait for a response, either positive or negative, and if a response is not received, it will be deemed to be unhealthy. The interval is how often the health check occurs. Valid values are anywhere between five seconds and 300 seconds.** So, in this case, every 30 seconds, the health of any associated instances will be checked.

Now, **these next two values define how the healthy or unhealthy state of an instance is determined. So the unhealthy threshold, the number of times the instance needs to fail a health check before it's deemed to be unhealthy.** Now, by setting a value of two, it means that you can have a number of failed checks before it's viewed as unhealthy. Now, this gives you a little bit of buffer in case you have some short term issues and don't want it to be immediately taken out of action if it fails a single health check. **The reverse of this is the healthy threshold determines how many checks need to come back as passed or healthy before the instance itself is marked as healthy so in this case, this is set to 10**. Now to keep this demonstration fairly responsive, what I'm going to do is change the interval to five. **Now, this is the minimum possible value for the interval. So this means every five seconds it's going to attempt to do a health check of these EC2 instances.** I'm going to set the response time out to be three seconds, So I want to be fairly aggressive I want to check every five seconds, and I only want to give my instances three seconds to respond. For the unhealthy and the healthy threshold I want to set this to two. So it will take two failed checks to move it to be unhealthy and two successful checks to move it to being healthy. So that looks good. I'm happy with all those values.

So next I'm going to go to add EC2 instances. So it's here where **we can enable cross zone load balancing.** Remember, that's the feature **where individual nodes can load balance both to the same availability zone and other availability zones.** That's a value that used to be disabled by default, but now is enabled by default. So I'm going to leave that selected. Now its also here we're manually adding EC2 instances so they're associated with the load balancer. Now, if you'll see later in this topic, if you do integrate a load balance it with an auto scaling group and this process is handled automatically but for this demonstration, I'm going to go ahead and select each of these three instances explicitly because I want all of them to be associated with the load balancer. Once I've done that, I'm going to go to next add tags. Not going to bother about any tags. So I click on review and create check. Everything looks good, which it does, and then hit create. **Creating a load balancer is something that does takes on time**. It's not instant. So because there's physical infrastructure that gets provisioned behind the scenes. So we need to provisioned these individual load balancer and nodes in each of the subjects that we've selected, generally can take anywhere from 5 to 50 minutes to provision this load balancer. Sometimes it's quick, but sometimes it can take a short amount of time.