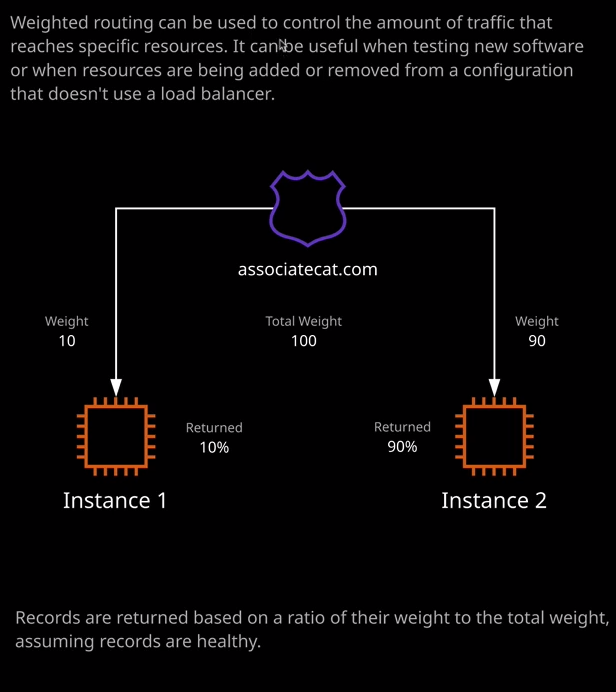
weighted routing.

So this is another routing policy that's available inside Route 53. In the previous lesson, I talked about fail over routing, which is a simple architecture **allowing me to define primary resources that are associated with a health check and then have secondary resources which can be returned if that primary resource or resources report as unhealthy. So fail over routing is generally used to implement a form of high availability and provide secondary resources but the next type of routing, which is weighted routing that's used to implement some form of leveling across different resources so weighted routing can be used to control the amount of traffic that reaches specific resources**.



So in previous lessons I have demonstrated how I've created three different EC2 instances, and each of those is running a different variation of this public cat website. With weighted routing, Route 53 provides functionality which allows you to direct traffic at specific resources with a different weight. Now, the way that that works is we create a number of independent records inside Route 53 and we assign each of those individual records a weighting. What actually happens is when a resolution request hits Route 53, it looks at the total weight for that particular record. Now, let me show you how that works because it's often easier to demonstrate this. So I'm going to go to the Route 53 management console, and I'm going to go ahead and delete both of these records sets that I created in the previous lesson when I talked about fail over routing. So I'll delete that record set and I'll delete the original one, which was the primary record as well. So I'll confirm that and then I'll go ahead and create a new record set. So I'm going to call this records at www and get the IP address of web1. So I'll select web1 and copy its public IP address. I'll go back to the Route 53 console and enter that in the value field with a one minute TTL. Then I'll drop down the routing policy and I'll select weighted. Now when you create a record using the weighted routing policy, you'll need to specify two different attributes. The first is the weighting and I'll explain in detail how this works in the second. The second thing you need to specify is a set ID. Now this is a unique string that's unique amongst the group of records with the same name. So what I'm going to do is I'm going to call this web1. So this will be unique amongst these three records I'm going to create. So once I've set that, I'll need to give this a weighting and I'm going to select a value of seven. I'll explain exactly what this means when I've created all three of the records. So that looks good. I'll hit create. Next I'm going to create another record. I'm going to use the same name, so www and I'm going to go back to the instances part of the EC2 console and select web2 because I'll need the public IP address of this instance as well. I'll go back to the Route 53 console enter this as the value, select this one minute TTL, and change the routing policy to weighted. In this case, I'm going to set the set ID to web2 because it needs to be unique. It can't be the same as this existing record set and in this case for this weighting, I'm going to select two. So that looks good and again I'll go ahead and click on Create. Finally, I'll do this for the third time. I'll create a new record set. I'll call it www again, set this one minute TTL, and I'll get the IP address for the web3 instance. So I'll copy that into my clipboard, go back set this as the value, again, I'll set the weighted routing policy. This time I'll select one as the weight and I'll call it web3 for the set ID. So what I've done here is create three individual records, one with the weight of seven, one with weight of two, and one with the weight of one. Now the way that Route 53 works is it performs an addition of all of the individual weights for any of these records that have the same name. So in this case, we've got seven, two, and one which is a total of 10 and then whenever any requests come into Route 53 into the hosted zone for this specific record, we take the total weight and the individual weighting of these individual records and that determines the percentage of the time that these individual records are returned to our customer. So in this example, I've got two instances, one with the weight of 90 and one with the weight of 10. That represents a total weight of 100 and what this means is that instance 2 will be returned 90% of the time because 90 out of a total weight of 100 represents 90%. Instance one will be returned a total of 10% of the time. In the example that I've created inside AWS, I've got three values seven, two, and one. This also represents a value of 10 and it means that web1 will be returned 70% of the time. Web2 will be returned 20% of the time. Web3 will only be returned for 10% of the time. **So weighted routing allows you to create multiple records with the same name and then when any resolution requests come into Route 53 for that particular record, the rate that that individual record is returned is based on its individual weight and the total weight. So you can always work out how often a record is going to be returned as long as you know all of the individual weights and then you can take one of those individual weights, compare it to the total weight, and work out a percentage. Now weighted routing policies are generally used when you want to test new features of software. It's important to understand that weighted routing doesn't actually ensure a level usage of resources. The only thing it's controlling is how often this resource is returned. It doesn't control the load that's placed on this resource,** so you might have three different customers browsing to your web application. One of the customers might get one of these EC2 instances returned, but that customer might represent 90% of the usage of your application. That means that if you use weighted routing alone, you can end up with some pretty uneven usage of your resources, **so you would not use weighted routing for effective leveling of incoming load over resources, that's actually an anti pattern. It doesn't work as you'd expect. What weighted routing is generally used for is when you may be testing new features of your application, then you can have a small portion of DNS request directed at maybe a second platform or a second EC2 instance, that allows you to perform initial testing and you gradually move more and more of the customer base over to this new feature set want you're confident that it works. So generally weighted routing is used for testing new product features or when you're adding new platforms to your application, want to migrate users over to it. It's not used in place of a load balancer** and when we talk about load balancing later in the course, you'll see how different it is using a real load balancer versus weighted routing. Now, for the exam you need to know is the architecture. You need to know the patterns and anti patterns of weighted routing, which is what I've just discussed. Remember, **the real key point to take away from this lesson is you would not use weighted routing for load balancing. Weighted routing is generally only used when you need to test out new software features and with weighted routing, you can create multiple record sets with same name the only thing that needs to be unique is this set ID. You need to make sure that you specify a weight for each of the individual records and then that's how Route 53 gets this total weight and then using these individual weights that determines how often these individual resources are returned.** Now, that's all I wanted to cover it for this lesson. It's just been a simple introduction to the architecture of weighted routing. That's all you really need to use for the exam. So go ahead, mark this lesson as complete, and when you're ready, you can join me in the next.