

Okay, welcome back and in this next lesson, I want to talk about a number of ways that RDS can provide resilience. So RDS has got two ways that it could do resilience. The first is something called multi AZ mode, and the second is using read replicas. So in this lesson, I'll talk about multi AZ and in the following lesson, I'll talk about read replicas. Now read replicas have another function, they allow the system to scale to really high read levels, so it offers an ability to scale a database out to support greater amounts of read. So in the next lesson, I'll focus specifically on that aspect. In this lesson, I want to focus on multi AZ capability.

Now to get started, I'm just going to quickly tidy things up. So I'm going to go ahead and delete this database that I did a restore on in the last lesson. I'm going to go to delete it. I'll not create a final snapshot. I'll not retain the automated backups and I'll just type, delete me and then click on "Delete." And I'll just delete the database to make it easier to keep track and of course, what I also need to do is to copy the database endpoint address into my clipboard, move across to my terminal and just return WordPress to its initial configuration. So using the initial database instance RDS lesson. So I'm going to save this file out and then move back to the console and just do a refresh and make sure that it still connects and everything looks good. So that's great. So now we've got this single database instance, and we can continue with this lesson. Now when I provisioned the RDS instance in the first lesson of this topic, I elected not to have a standby instance. It's a single availability zone instance when you provision a multi availability zone database instance then RDS creates a secondary instance of that database the standby node or standby instance and RDS is performing automatic synchronous replication between the primary and standby instance. It's important for the exam that you remember that term. **It's replicating synchronously or it's using synchronous replications. If you see any mention of that in any exam text, then you know that it's talking about multi AZ.**

Now, with all RDS instances apart from Aurora, every instance has its own infrastructure, so its own database server and its own local storage. That storage is provided by EBS or the elastic block store. So the primary instance has its own storage, and the standby instance has its own storage. So the replication is occurring between the instances at the instance level but each of these have their own isolated storage. Now **each of these instances runs in a separate availability zone and an availability zone is its own physically distinct independent infrastructure, and it's engineered to be highly reliable. In the case of an infrastructure failure, RDS performs an automatic fail over from the primary instance to the standby instance, this cname changes from pointing at the primary to pointing at the stand by and apart from a brief interruption to your application, the process is completely seamless.** You won't know that you've changed from a primary to the standby instance. So this is the process the cname normally points at the primary. The primary is synchronously replicating to the standby and then when a failure occurs, that cname is moved to point at the new primary, which is the old standby. The hardware error or software error is resolved, and then the process could be changed again and this process of fail over can happen if there are isolated hardware faults but also, **if you need to make certain changes to the RDS instance, the change can be first made on the standby, the cname could be altered to make that the new primary, and then the change can be made on the new standby and so that whole process is designed to minimize any outages.**

Now, right now, this RDS lesson instance that I created earlier in this topic of the course is not running in a multi AZ mode, so it's only running in the U.S. East 1. So I'm going to correct that. I'm going to go to modify and then I'm going to look for this multi AZ deployment option, and I'm going to change it to "Yes," I'm going to change it to be a multi AZ deployment, and then I'm going to scroll down and I'll hit "Continue" and as before because we're changing it from being not a multi AZ to being to a multi AZ. We need to specify when we want this change to occur again, we're presented with the two options. We can either do it during the next maintenance window or we can do the process immediately. Now, this change will cause a significant outage. Essentially the primary instance, the solo instance at the moment there's going to be extensive configuration changes that occur and so there will be an outage to convert from a non multi AZ deployment to a multi AZ deployment. But for this demonstration, I'm happy for the outage. I'm going to go ahead and click "Modify Database Instance." Now, there are some really important things that I need you to understand for the exam. Exceptionally important facts so I want to go through these and really stress why they matter**. So if you're using anything but Aurora so if it's an RDS instance that is not Aurora than this cname will always point at the primary instance unless a fail over happens, in which case it will move to the standby which will become the new primary.** **Now, you don't have any access to the standby instance in a normal multi AZ RDS deployment. That's really important to understand. You can't write to that standby. You can't read from that standby. It does not offer you any additional capability or functionality beyond the additional resilience. So that's really important to understand. There are no hacks, there no workarounds, essentially you're paying for an additional RDS instance that's sat there being the replication target from the primary instance. So it's getting all the data replicated synchronously from the primary to that standby. You're paying for that, but you aren't able to use it. You can only use it during fail over events. Now it does offer some additional functionality beyond just the resilience.** I mentioned this briefly earlier in this lesson.

**If you do perform any size changes or if there's any maintenance on your RDS instance, then generally what occurs is the maintenance occurs on the standby node or standby instance first, that's promoted to be the primary instance, and then the other side of the maintenance occurs on the new standby. So the whole point of this really is to minimize the effect, any outages if there are any maintenance or other changes, so failure can occur in a number of situations.** **Generally, if there's a hardware failure then the fail over will be automatic from the primary to the standby. If there's any loss of network connectivity, if this computer based failure, storage based failure, or any other serious failure on the primary then the cname will be adjusted to the standby and fail over will occur.**

Now, from an RDS perspective and I keep stressing that we're removing Aurora from this because that handles things very differently but **from an RDS perspective, that is going to be a brief outage. That's for two reasons. The first reason is that we're changing this DNS name, the cname, from pointing at the primary to the standby and anything involving DNS can have some level of lag or caching or other elements that can slow down that change. So when thinking about RDS fail over, it can be a brief outage. Think two digit seconds, or maybe at worst, a minute or two just for this cname process to change from the primary to the standby.** So just keep that in mind. **It is not a truly fault tolerant system. There is going to be some user interruptions and system interruption as that cname moves from primary to standby.**

**Certain other operations which occur, such as changing the database size, they will also occur on the standby first and assuming that complete successfully, then the cname will be altered and the same change will occur on the old primary. So everything that's done with RDS is designed to minimize the impact so minimize any outages**. Now, with RDS and again, I'm excluding Aurora from this, **you're limited to two instances the single primary and the single standby. You can't deploy an additional one in a third availability zone. So RDS high availability from this perspective is just two availability zones.**

Now, if you recall back to the start of the course when I talked about **disaster recovery so RPO and RTO values using a multi AZ deployment does provide you with better RTO. So that's recovery time objectives because it is a lot quicker to fail over from the primary node to the standby node than it is to restore a snapshot. Replication of the data between primary and standby is synchronous. It is copied in real time from the primary node to the standby node as it's written. So the primary node and sometimes you'll hear this refer to as a master node. It has its own dedicated storage, as does the standby node. So this replication, just to stress this, is occurring synchronously between the primary and the standby. It's important to understand that there is no performance impact. This is built in as part of the product and then backups are taken from the stand by ensuring there's no performance impact. Maintenance is performed on the standby first, there's a lot of advantages to using multi AZ beyond that resilience element**, and that is pretty much everything that I wanted to cover it in this lesson. It's not a hugely complicated process, and it's really easy to provision or modify an RDS instance to utilize multi AZ. For the exam, all you really need to be aware of is exactly how the architecture works, but I will make sure that there is a Linux Academy hands-on lab in this part of the course that gives you some practical exposure to using multi AZ databases. But at this point that's everything I wanted to cover so go ahead, mark this lesson as complete and when you're ready, join me in the next where I'll be talking about RDS read replicas.