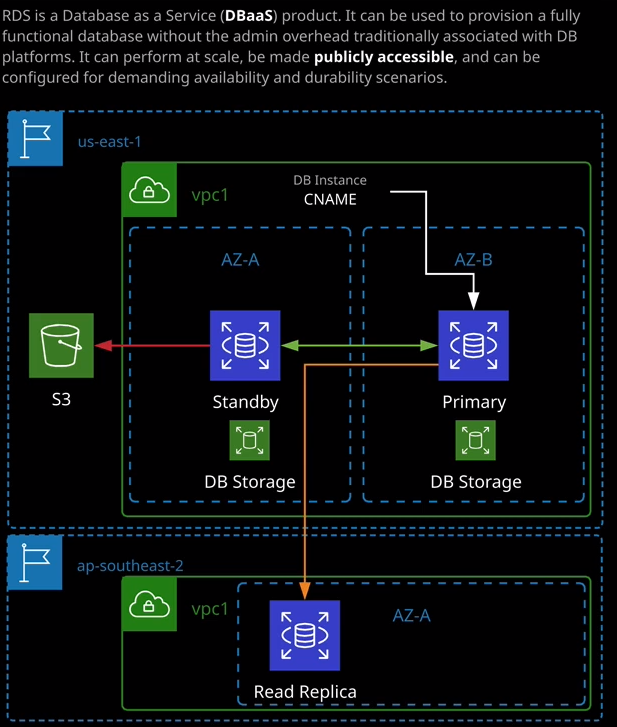
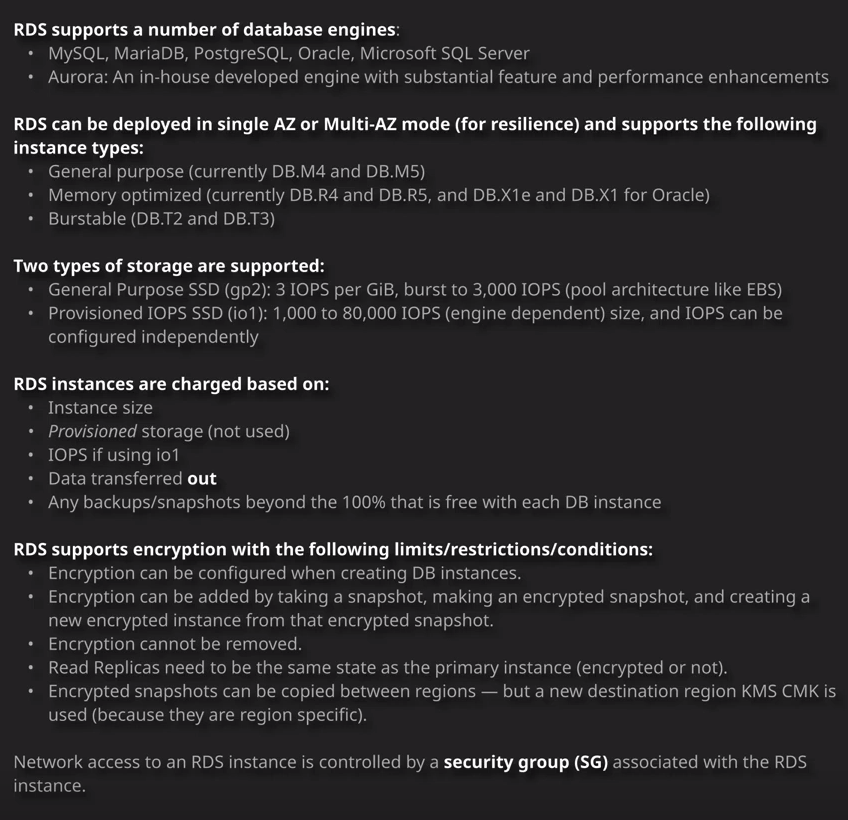
Welcome back and welcome to this topic of the course where I'm going to be talking about the important architectural considerations for the relational database service known as RDS and **RDS is a product available from AWS that allows you to provision a fully functional database without any of the usual admin overhead.** So let's jump in and get started. This first lesson I want to talk about the essential elements of the product so its architecture and basic functionality. Now, this lesson is going to include some practical demonstration so to allow that before I get started, I'm going to move across to the EC2 console and just go ahead and create a key pair. So I'm going to go to key pairs, create key pair, and then I'm going to call this key pair RDS just to keep things simple. Once I got that key pair, I'll move over to CloudFormation I go to create a stack, upload a template, and then I'm going to choose the vpc.yaml file that's available inside the lesson files inside 06 databases, 02 RDS, and then RDS essentials and all this is going to do is create a simple VPC with three tiers one public, one application, and one for databases and it'll utilize two different availability zones. So go ahead and create this and the only thing I'll need to specify is a name I'll pick RDS and the key pair I'll choose the RDS key that I created moments ago. So next, next again, scroll down and acknowledge that this might create IAM resources and I'll click on "Create Stack." So while that's deploying in the background, I want to talk about the architecture of RDS.



**RDS is a database as a service product. It can be used to provision a fully functional database without the admin overhead traditionally associated with DB platforms. It can perform at scale, be made publicly accessible, and it could be configured for demanding availability and durability scenarios. So that is what the product does at a high level. Essentially, you can deploy RDS instances, and they can be configured to be just standard, or they could be highly available, and you can deploy those into an existing VPC, and all of that is done as a service.** Now I'll show you exactly how RDS is deployed momentarily once this CloudFormation template is finished applying, but for the exam, you need to understand a number of key points. **One of those points is that RDS supports a number of different database engines. It supports mySQL. It supports MariaDB. It supports PostgreSQL. It supports Oracle, and it also supports Microsoft's SQL server. In addition, AWS have designed their own database engine, which is called Aurora**, and I'll be talking about that in detail in the next topic of the course. So for this topic, I'm focusing specifically on mySQL, MariaDB, Postgres, Oracle and Microsoft SQL server. I will be demonstrating most of functionality using mySQL. Okay, so I'm going to go ahead and create an RDS instance, so I'm going to move to the RDS area of the AWS console. I'm just going to bring up the exam hints screen while I do that. So I already mentioned that RDS supports a number of these different database engines we'll be focusing on mySQL for this lesson, but it does support all of these other common database engines in addition to Aurora, which is the in-house developed database engine. Now **RDS could be deployed into single AZ or multi AZ mode, and you'd pick between those based on your resilience requirements. So if you're just deploying a development or a test database, then you could potentially deploy it in a single availability zone. If you're looking for a production scale deployment then you need to use multi AZ. So the first thing that I need to do before I deploy a database assuming that I'm not deploying it into the default VPC, is specify which availability zones are going to be used or more specifically, which subnets. So I need to create a subnet group.**



So I'll do that first I'll go to subnet groups and create a subnet group. I'm going to call this RDS lesson. I'll use the same text for the description. I'm going to pick the VPC that the CloudFormation template has just created and then for the subnets in this subnet group, I want to define which subnets are available for any RDS instances associated with this subnet group. Now, the VPC that I've created contains two subnets that are suitable the database subnet in both available in zone A and B. So I'm going to go ahead and pick availability zone 1a and then I'm presented with three different subnets to use. Now if you've got a good memory from previous lessons, you remember that 10.0.1 is the public 10.0.11 is the application, and 10.0.21 is the database subnet. I'm going to go ahead and pick 10.0.21 then I'll click on "Add Subnet" and that adds it to the list of possible subnets for this subnet group. I'm going to change the availability zone to 1b, expand the subnet dropdown, and again, we'll see this list of subnets and again, if you remember from previous lessons 10.0.2 was the public 10.0.12 was the application, and 10.0.22 was the database subnet. So I'm going to go ahead and select that one and again click on 'Add Subnet." So now we've got the database subnet selected inside this VPC1 that's in availability zone 1a on one that's in availability zone 1b. So I've got everything I need. I can go ahead and create this subnet group. Now that I've got that subnet group, I can go ahead and create a database instance so I'm going to go ahead and click on "Create Database." Now, you have got the option to easy create, which handles a lot of this configuration for you but as you're looking to learn how this works for the exam. I'm going to leave this switched off and do everything manually. So step one is that I need to select the database engine type to use. Now, Aurora is the default because that's the one that I would normally recommend for all production scenarios if using relational databases. I'm going to cover that in the next topic of the course. So I won't be focusing on that in this lesson. So I'm going to go ahead and select mySQL. Now for a specific engine type, you're also able to select an edition and a version. Now, the options that you have here depend on the specific engine that you select. So if I select MariaDB, then I've got a different set of version numbers than if I select mySQL and a difference set again, than if I pick Oracle. With commercial database such as Oracle, you've got different licensing options available in this addition section as well as versions. If I select Microsoft SQL server again, I've got the same sort of structure. I can define different license levels and different versions but for the open source database options such as mySQL or MariaDB, we've only got the version to select from, and that's fine. Now, in terms of the version number, it follows a usual version convention. So you've got the major version number, the minor version number, and then the revision. I would always recommend, unless you've got a specific reason not to, to pick the most recent version number from this list. Now, if you scroll down further in this list, you'll see some even more recent versions of mySQL. So I'm going to pick mySQL 8.0.15. Now scrolling down further still were able to select from a particular template to use so these templates just offer a collection of commonly used settings. Production will define highly available database instance. If you select dev and test that provision is one the operates in a single availability zone and if you select free tier, you make sure that you've got one that falls under the free tier that's available inside AWS. Now, I don't necessarily recommend that you're following along with this in your own environment. You can use the Linux Academy Cloud Playground and lab environments to do all of this in accounts that we provide for free but if you do follow along, make sure that you're either picking the free tier or you're okay with paying the costs for RDS and I'll talk more about the costs in a minute but for now, I'm going to go ahead and pick dev and test because I don't want to use at this stage multi AZ I want to deploy it in a single availability zone. Now, you'll need to select a number of other options and detailed some of these on these exam hints and key facts. But essentially, we need to start by picking a database identifier and I'm going to call this RDS lesson to keep things simple. Now that **we've got the credentials now, these are credentials that are built into the database engine itself. So these the credentials any applications using this database will use when it's logging into the database. Now, if this is going to be a production database, you need to make sure that these are secure credentials. If it's a dev or test installation it can perhaps be a little bit more open to using simple usernames and passwords but generally it is best practice to make sure that you are using something secure. And so even for development or staging implementations I do recommend that you something fairly secure fairly random. You do have the option of letting RDS auto generate your password to use**, and I'm going to select this option. So I'm going to let RDS autogenerate the password. Next, we've got the database instance size. We've got various different sizes that you can actually use. These mirror normal EC2 instance classes and so we've got various different M4 and M5. We've got R4 R5. All of the various EC2 instance types and sizes that you're used to you'll have RDS equivalents or subset of those. Now you've got standard classes which includes the M series. If I click on the dropdown, you'll see all of those that are available. Memory optimize which includes the R and the X and then burstable, which includes the T type instances and to keep things simple for this demonstration, I'm going to go ahead and pick a T3.medium for my database instance. Again, if you are doing this in an environment that you're paying for, please make sure that you select one that you're willing to carry the cost for. I would strongly recommend that you do this using a Linux Academy provided environment and pick one of the smaller classes of instances. So I'm going to go ahead and pick the T3.medium. You'll also need to specify the storage type that you're going to use and the options that you've got is general purpose, SSD, and provisioned IOPS. Remember from the EBS lessons earlier in the course general purpose or GP2 is the same type of storage that EBS uses for EC2 instances. It provides three IOPS gigbites of storage and it can burst up to 3000 IOPS and it uses this pool architecture just like EBS. For 99% of cases you will use general purpose storage. If you do need really high performance and you need high performance for smaller storage sizes, then you need to pick provisioned. But for this demo, I'm going to pick provisioned and pick the minimum amount of storage, which is 20 gb. Now historically, the size of the storage is something that you couldn't change at a later point, so you needed to be really careful in picking this storage. The other point is that you are billed off the actual allocated size. So if I select 20 gig here I'm billed for 20 gig if I select 200 gig or up to 16 TB, which is the maximum and then that is what I am billed for, it's not about the use storage. It's about the allocated storage. So pay special attention to this if you deploying it in production use cases. Now, we've also got storage auto scaling, which is a relatively new feature for RDS. It's been out for a fair amount of time now, but it wasn't originally part of the RDS product, and that will allow the storage to increase from this initial allocated amount all the way up to this maximum threshold as you require it or as your database requires it. Now for this particular lesson. I'm going to untick this because I don't want this size changing but if you're looking at using this for production, you might want to consider it. Next we've got availability and durability. So if you refer back to the architecture diagram that I showed it to start this lesson all RDS instances start with at least a primary instance and you can also specify a standby. So if you define this is a multi AZ RDS instance, and it creates a standby instance, I'm going to talk about that in a dedicated lesson. So for now, I'm not going to create a standby instance. I'll need to pick the specific VPC to use. So I'll need to make sure in this particular lesson, I'm selecting VPC1. This dropdown includes additional connectivity information, so it's here where you select the subnet group to use. You can specify whether you want the database instance to be publicly accessible. So if you deploy into a public set of subnets, you can pick whether you want it to have public IPs, I don't. You can specify a VPC security group to use so RDS instances are secured for a networking perspective using security groups. So I'm going to go ahead and create a new security group and I'm going to call the rds\_sg for RDS security group. It's in this security group that I'll need to configure any rules to allow accesses to my RDS instances. So this is especially important that you have this configuration locked down and you understand it. You need to specify if you've got an availability zone preference. If you specify no preference than RDS automatic select which availability zone the primary on the standby instances go in if applicable. In this case, I'm creating one in a single AZ. So I get to pick what availability zone I want the primary instance to be in. I could select a specific availability zone, or I could leave it at no preference and let RDS pick and that's what I'm going to do. There's some additional configuration that you can use. You can specify an initial database to create. RDS is capable of handling multiple databases, but you are able to specify an initial one to use and I'm going to go ahead and do that. I'm going to call it RDS lesson db just to keep things simple. This parameter and option groups that you don't really need to be aware about for the exam but I will include a link in the lesson description with more information. **RDS is capable of using IAM authentication.** So instead of using the built-in authentication information you can log on using IAM users and roles. But again, that's beyond the scope of what you need for the associate exam. I'll include a link in the lesson description with more information, but you don't need to be aware of that. Now RDS is capable of handling backups automatically, or you can do it manually. I'm going to have a specific lesson that's coming home in this topic, of course, talking about backups and then lastly, RDS is capable of performing upgrades during a maintenance window. So you can specify a maintenance window, or you can let RDS decide on it, and then it can perform minor version upgrades during that window. Again, not something you need to be aware of in detail for the exam. So at this point, I'm going to go ahead and create the database. Now, I know I've skip through a lot of this really quickly, but I want you to be aware of the architectural concepts, but you don't need to be aware of a lot of it in detail for the exam. I want to focus on what really matters. So passing you the example as well, giving you the foundational knowledge that you'll need to be able to use RDS in production. So I think we've covered most of that in what I've talked about so far in this lesson. For any databases creating, and if you've elected to pick the auto generate password, you need to click on "View Credential Details" to get access to those credentials. This is the only time that you built of you with this particular set of automatically generated credentials. There's nothing preventing you from changing your username and password at a later date but if you do want to get access to it and I will do because I'll be using this later in this lesson, I'll need to click on "View Credential Details" and this will give me an overview of the automatically generated master username a master password the RDS has created for me as part of this instance. So I'm going to load both these down and then pause the video on wait for this RDS instance to finish creating. Now, this lesson is getting a little bit on the long side. So I wanted to split it into a number of different parts and give you the opportunity to take a small break. So this is the end of part one. Go ahead and mark this lesson as complete and when you're ready, you can join me in part two.