Now **EC2 integrate tightly with Amazon Cloudwatch, which is AWS's monitoring service. So you can click on the monitoring tab for a specific instance and get access to CPU utilization, disk read and writes, network in and out, and other instance specific telemetry.**

What you can't do is get access to data that's inside the instance. So an example of this is **memory utilization. Now there are ways to do this. You'll need to install an agent inside the EC2 instance and connect it to CloudWatch** and I'll explain how to do that as we go through the course. For now, I want you to be aware that you've always got the ability to monitor an EC2 instance from CloudWatch. **Now the default is EC2 delivers this data to CloudWatch with a five minute granularity. You can enable detailed monitoring which comes at an extra cost to improve this to one minute granularity.** So you generally do that for any EC2 instances involved in production platforms but for this demonstration, I won't bother and I'll stick to the standard.

I also want to point out **account limits.** Every AWS account comes with limits and limits control the number of things that you can launch, this size of things that you can launch, and other capabilities that you're allowed to use. This is just a way the AWS can control the likelihood of bill shock or any or the large expense for new users. A lot of these could be increased by logging a support ticket and in the case **of EC2, there is a nice dashboard, which lists the limits for all of the different types of EC2 instances and give you a link where you can request a limit increase.** So that's pretty handy and you will get used to using this as you design production systems because often these limits are set really low on new accounts. Because this is a Linux Academy provided account, these limits tend to be a fair bit higher. On new accounts where you've not used AWS before, you might find a lot of these set to five, which could be pretty restrictive for production usage.   
Okay if I go back to my instances console. So I click on instances. I'll see my instance state is now running, and it's got two out of two status checks. Now these status checks, I'll talk about these lots as we go through the course. **One of the status checks relates to the hardware that you EC2 instance runs on and the other relates to the actual instance itself**. So what you're looking for is two out of two, which indicates that this instance is completely healthy. So now that we have these two out of two status checks and the instance state is running and I'll talk about what the states are in a second, we can go ahead and connect to this instance. Now to connect, we'll need to use this **EC2 PEM file** that we created earlier. To find out how to do that, I'm going to right click and hit connect. Now this will bring up a dialogue on this. Two key parts that I want to bring your attention to. The first is that if you're running Linux or Mac OS, you'll need to change the permissions on any newly downloaded key file and you can do that by running chmod 400 then the name of this PEM key. Once you've done that, you can run this command to go ahead and connect to the instance. So I'm going to do that now I'll switch over to my terminal. First I'll need to adjust the permissions so I'll copy this into my clipboard, move to my terminal, change the permissions, move back, copy the command to connect to the instance, move to my terminal, paste this in, press enter, if it's the first time you'll need to verify the authenticity, answer yes, and that will connect you straight through to the instance. Now, if you are using Windows as your operating system, the connection instructions will be slightly different, and you can find those at the top of this connect dialogue. You can right click on this, open it in new tab, and this takes you to an AWS piece of documentation, which will detail how to use the popular terminal application PuTTY to connect to EC2 instances. For Linux and Mac OS, these are the instructions. Windows adjusts it slightly differently.

Now, **EC2 instances have two main states running and stopped. It also has intermediate ones, such as pending and stopping and, of course, you can also terminate an instance which deletes it.** Now by default, **EC2 usage is billed by the second with a minimum of 60 seconds and this charge occurs whenever the instance is in the running state. If you stop the instance or terminate the instance, you aren't billed this per second cost. Stopping the instance is like shutting it down**. You can do that by right clicking, go to instance state, and then stop. You'll need to confirm it. Click yes stop and the instance will shut down and you'll be disconnected. Now I want to point out that that is different than shutting down from the guest operating system. **If you shut down from the guest operating system all you're doing is stopping the guest operating system.** When you stop the instance, you're stopping the virtual hardware and it's a different thing. **When the instance is in a stopped state, you're not being billed that per second charge but if you shut it down from the operating system and that doesn't actually stop the instance, you still will be getting billed.** That's critical to understand.

Another cost factor of EC2 is the storage that the instance uses. So in this case, I've got an attached EBS volume and if you recall it's an eight gig volume that's got the Amazon Linux 2 OS installed on it. **When the instance is stopped, I'm still being charged for the space consumed on this EBS volume. The only way that I can get away from being charged for that EBS volume is to terminate the instance** and I could go there by going instance state and then terminate that will delete the instance and the attached volume. **You can change that so terminating instances won't delete the storage but generally you want to be in a position where, when you do terminate an instance, it also removes the storage but be aware of that, unless you configured backups or any other snapshot technology, you will be losing the data.** Now you're also able to stop an instance by right clicking, going to instance state, and then stop hibernate. **That hibernate mode is like the feature that's probably available on your laptop or desktop. It essentially saves the memory of the instance to disk and then it stops it and that means you can start it up again and have it be in the same state that it was before you stopped it**. There are limitations, so certain types of instances support it and certain types don't and there are restrictions based on the memory size of the instance and the OS type. So just remember that it's not available across the board. Now that's what I wanted to cover in this lesson, its just the basic fundamentals of EC2 the architecture, the networking, the resilience, the security, and the storage. One of the requirements for the Solutions Architect Associate exam and to be a good solutions architect is knowing when to use EC2 versus other types of compute. Right now, we're only covering EC2, but as we go through the course, you'll encounter other compute services. At this point, when you're looking at a scenario that needs a compute service, **you should preference EC2 if it needs a traditional operating system so Windows or Linux that requirement for that computers consistent. It's something that's running and consuming CPU constantly then that suits EC2. If it's a monolithic application, so traditional application that requires a server to host its components that also suits EC2 and finally, even if you've got an application that is fairly bursty, if it requires to be active even during situations where it's not consuming resources, then you need EC2. An example of this might be an active directory, authentication service, or business network active directory. During off periods the CPU usage might be naught 0.1% or even less, but it does need to be active 24/7 even if that's only one instance out of, say, 100. If you need an active, consistent compute service, then you should pick EC2**. Now I'll contrast this is a move on to the service compute topic on the containers, a microservices topic we'll present some alternative types of compute, which a useful for different types of scenario. For now, though, we're talking about EC2. So this lesson was just the architectural fundamentals. In the next lesson, I'm going to talk about the different types of instance and the different architectures of instance so it's going to be a lot of theory, but it's an essential bit of knowledge to do well in the exam. So go ahead mark this video as complete and when you're ready, join me in the next.