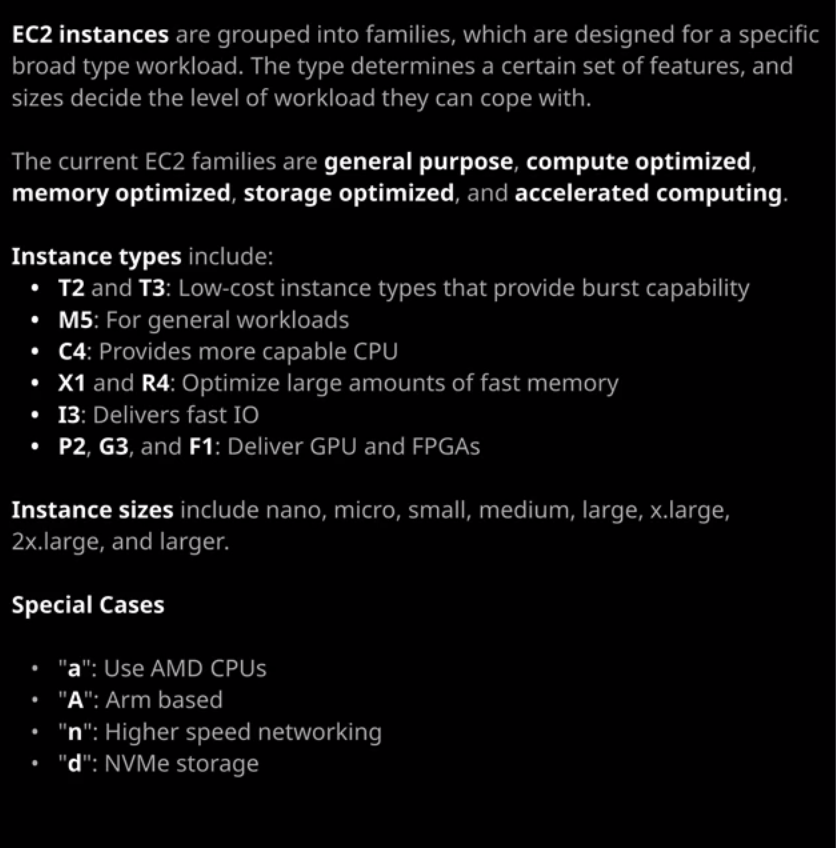
Welcome back, and in this lesson, I want to go through in a little bit more detail the different configurations available for EC2 instances. Now AWS provide a lot of different instances that you can launch. Most of us at one time or another have felt intimidated when opening this screen for the first time. There are a lot of things to consider in this list and to be a good solutions architect, you need to pick the correct one for a given scenario. You need to know when and where to choose a certain number of CPUs or memory, what type of instance storage to pick, what type of storage or network optimization to choose, as well as any IPv6 support, and other instance capabilities. There's a lot of information to consider and in this lesson, I want to step you through the main things to consider when selecting a certain type of instance. Now I'll start by introducing some terminology we'll use throughout the course. I'm the big fan if using the right words for the right things because it makes communications clear. AWS use a lot of terminology in their documentation, and you need to understand exactly what they mean so you can use the same communication both in the exam and with your colleagues or clients.



Now AWS group all of the available instances into what they call families. The families are the top level way that you can select the right instance for the right job and the families are **general purpose, compute optimized, memory optimized, storage optimized, and then accelerated computing**. Picking the correct instance family is like picking a type of car, bus, truck, or bike. You need to pick the right one for a given situation. Now there are lots of different types of trucks, but generally you'll start off by knowing that you need a truck. Once you know that you need a truck, you can select the type of truck. There are some bigger cars, which might do a lot of the work that a smaller truck can do and there's some smaller trucks that some people might select instead of a bigger car. So there's always going to be some overlap between the different families in the same way that there's some overlap between cars, busses, trucks, or bikes depending on your exact scenario, selecting the family of instance to use starts by knowing and understanding your scenario. What does your application require? What does the architecture that you're implementing or designing? What does it require from its EC2 instances? Now I always start off by default with the general purpose family.

The **general purpose family** provides instances which provide a good balance of resources, a good amount of CPU, a good amount memory, a good amount of storage capability, and no special features or capabilities. Now you might select compute optimized instances when you've got a compute intensive work load.  
**Compute optimized instances** provide the best CPUs available. So the most modern instruction sets, the highest frequency, the largest number of cores, and the highest performance. With the compute optimized family, if you have a given monthly spend for an instance and you choose one of the compute optimized family of instances, then you know that you're going to get a priority on CPU versus other resources. and also the compute optimized family is going to allow you to go to these higher CPU counts. So if you want access to the highest CPU count machines and you want each of the CPUs to deliver the maximum possible performance, you'll start with compute optimized.   
Likewise, **memory optimized** is what you'll select when you have large memory uses, like database server or in memory caching. With memory optimized, you'll get good quality CPUs, but you'll also get a priority on memory. So the fastest memory available and the highest amount of memory of any EC2 instances available. So memory optimized is what you'll select when you're really concerned about memory usage and then we've got storage optimized instances and you'll see a theme developing.   
**Storage optimized instances** are what you'll select when you want access to superfast storage. Picking storage optimized means that you'll get access to the fastest instance store or local physical storage available inside AWS. So this is ideal for things like noSQL databases, data warehousing, Elasticsearch, or analytic workloads where they need superfast local storage. So these instances tend to provide really good networking, really good storage, and a good balance of CPU or memory, and they also provide additional capabilities, so advanced storage technologies such as NVMe. So if you need anything that relies on storage, storage optimized family is the way to go.   
**Accelerated computing** is a special one with this family, you can access specific hardware capability. Now this comes in the form of FPGAs which are field programmable gate arrays with FPGAs you essentially get the ability to program hardware. So you can get hardware that functions really well for one specific type of task and by utilizing FPGAs you get to tune the hardware specifically for a given situation. You've also got GPU capability on some of the accelerated computing instance types, and they provide CPUs or graphical processing units and these are really good for certain types of 3D modeling, machine or deep learning, high performance computing. There's lots of really advanced scientific analysis use cases that really fit accelerated computing.

Now this is just a high level. What I would suggest did you go through this course is to constantly come to this page, which I'll make sure is in the lesson description on refresh your memory about the types of architecture that provided by each of the different families of EC2 instances. You won't be expected to know this in detail for the exam but you may face some questions where a certain type of EC2 instance is mentioned, and it might give you just that little bit of advantage that you need if you understand what that family of instance provides. So if it's a C type of instance, you'll know by the end of this course that's a compute optimized instance and so you'll know it delivers really good quality compute. So the families are the high level categories. Each family includes a number of different instance types. So when you see something listed, for example, as a T3.medium, it's useful if you can decode what this actually means. The first letter, the T, that is the type of instance. So this is a T type instance. We've got others. For example, M. This is an M type instance. The number which directly follows that is the generation. So an M4 is the previous generation. M5 is the newest generation and the T type of instance that's evolved from T1 to T2 to T3. The letter immediately after that in some cases, so, for example, a small a that indicates that it's an AMD CPU in most cases, instances use Intel CPUs. If it's got a small a after the T3 or the M5 that indicates that it's using AMD CPUs and you might have a preference so your application might mandate a certain type of CPU and if that's the case, you know that you can select an AMD version of the instance. Everything else should be the same or similar, so M5 should be fairly similar to M5a. The difference being, of course, the CPU and perhaps the motherboard because different CPUs require different system boards, so you might get a slight difference but in general, the instance capability should be almost identical, just with a different CPU.

Now the type controls the exact capabilities provided by the instance. I want to give you an example. **The M5 instance is the general workhorse in AWS. It's in the general purpose family, and it provides a good balance between CPU, memory, and capabilities. It's not expensive. It's not cheap. It's not special, but it's not slow. It's your general workhorse instance. It** includes a great balance for the price. Now it should be your default for everything. If you don't know if you're in doubt, start off with an M type instance. **The reason to default to an M is that it gives you 100% of the resources 24/7 365. If you want to use an application on this type of instance, where you're going to be using 100 CPU 24/7 365 that's fine. It's designed for that but then compare that to a T3. The T3 is a similar type of instance to the M5 but with T3 there's an assumption that you won't be using its full capabilities.** Let me give you an example of what I mean. If we look at the T3 instance documentation and scroll down this page, there's what's known as a **baseline performance**. What this means is that this expectation if you pick a T3.medium, you're going to be using 20% of each of the CPUs. Now if you're using lower than that, you gain CPU credits. If you consume higher than that, say you go to 100%. You allowed to do that but you consume credits. Now if the CPU credit balance goes to zero, then depending on what you select, you could be throttled. I created an instance earlier in this cloud sandbox environment. So if I open that up and go to monitoring and scroll down initially when I created this instance, I was consuming credits but after the instance has been built and it's just sat there idling, I'm not consuming any, and I'm actually gaining CPU credits. I could leave this instance in this state for a while, and it would continue to gain CPU credit balance because I'm not consuming any because I'm using less than this baseline. Using a T3 it's a lot cheaper than an M5, but there is this expectation of not using above a certain CPU level. You're allowed to do it. You're allowed burst up to whatever you need, but you'll consume the credits and when you run out the credits, either you'll be throttled or you'll be allowed to use on limited mode and limited mode just charges you extra for going outside of the CPU credit balance. So there's a lot of things to decide upon when you're looking at EC2 instance types. The type is the thing that determines the exact capabilities. So you might look at compute optimized instances, for example, at the C5, and that will give you a certain type of CPU. You'll be able to select from advanced CPU related technologies. You get turbo boost; you might be able to control CPU states. There's a lot of other capabilities you might get with a compute optimized.

Memory optimized will give you a lot of memory, a lot of high speed memory, and you'll be able select instances that have got the largest amount of memory. So, for example, the **X1 instance type gives you up to 1,952 gb of memory. It's one of the largest instance types available**. Now you don't need to know these from memory for the exam, but as you go through the course, I want you to use this page, go through it on a daily or weekly basis. Just become comfortable with what you get in each of the overarching instance types. So know, for example, that memory optimized is the R type of instance, or the X type. Know that accelerated computing includes Ps, Gs, and Fs. Know that storage optimized is Is and Ds and know, for example, that the I series is good for high random IO because it includes a lot of SSD local storage. know that the D type is good for high throughput because it includes a lot of traditional high capacity hard drives. Nobody knows all of this from memory. Not even me. I've been doing AWS for 10 years now. I've worked in a lot of large projects, and I still don't memorize every single one of these facts and if I could memorize it AWS are releasing features so quickly that it becomes impossible to understand everything. So I've got this page bookmarked, and I come back to it fairly often just to refresh my memory. The last type of the instance configuration you need to be aware of is the size. So I've talked about how the letter represents the type, the number represents the generation, and you might also have a lower case letters after the number and that represents an AMD CPU in the case of an A. Now, after the period, you're going to have something else. It might be Nano. It might be micro, small, medium, large, extra large, two extra large, and so on. This indicates the size, so for a given type of instance you're going to get a certain set of resources, a type of CPU, a type of memory, and an overall amount. The size dictates the exact quantity of those resources, so you'll note that as you go from T3.nano to T3.2extralarge. You increase in CPU. You increase in the CPU credits per hour, you increase in memory and then, in the case of the other types of instances. So, for example, M5 you also increase in network or storage performance as you increase in size. So be aware that the difference between, for example, an M5.large and an M5.extralarge is generally just a doubling of certain resources. Again you don't need to know in detail for the exam, but do be aware that as you increase in the size categories, you generally increase corresponding resources. Now you're also going to see some special cases to these instance names you'll see other letters so I've mentioned how a lower case a indicates AMD CPUs. You'll also see an uppercase A. So remember when I was creating an instance? If I go to launch instance and then this time instead of selecting X86 I might want to select Arm. If I do that I will only have access to the A type of instance. So A indicates that it's R based a starting letter of a indicates that uses an arm architecture. There's also all the letters that you'll see when decoding these different instance types. You might notice an N, which is highest speed networking and then you've got **lower case d, which offers this advanced form of storage is his NVMe storage, it's super fast. It's low latency, and it's great for applications that really need that high level of performance.** Now, don't worry, I'm not expecting you to know all of this at this point. This is the fundamentals topic in an associate course. I'm just taking the strategy as I teach this topic to introduce everything, even if it's just at a low level. I want you to be aware of these things so that when I start talking about them more and introducing additional features and capabilities, you understand the foundation. So it's important that even if you don't remember this and give you some exposure, so I'll make sure that this is linked in the lesson description and as I said at the start, I want you to bookmark this and get used to spending a little bit of time every day, looking at the different families, the different types, the different sizes, and the different capabilities and features. If you do that a little bit every day, I promise you the exam and this course will be much easier. Now I'm also going to include a link in the lesson description to a YouTube video of a re:Invent talk and the topic of the talk is Nitro, which is AWS's new virtualization platform. A lot of the more advanced and high performance instances use Nitro, and I want you to have some exposure to exactly what that means, so it's not required for the exam but I want you to watch this video at some point while you are taking the course, even if it feels at this stage to be too technically advanced just try and watch it through to the end because it will introduce a lot of really cool features of EC2 that even if you just get a little bit of exposure to will help you in the course and in the exam. So that's everything I wanted to cover in this lesson. I know it's been a lot of theory, and we're going to start getting now to a lot of lessons where I'm showing you how to do things practically but I did need to get across this theory on instance types in this lesson. So go ahead, mark this lesson as complete, and when you're ready, join me in the next.