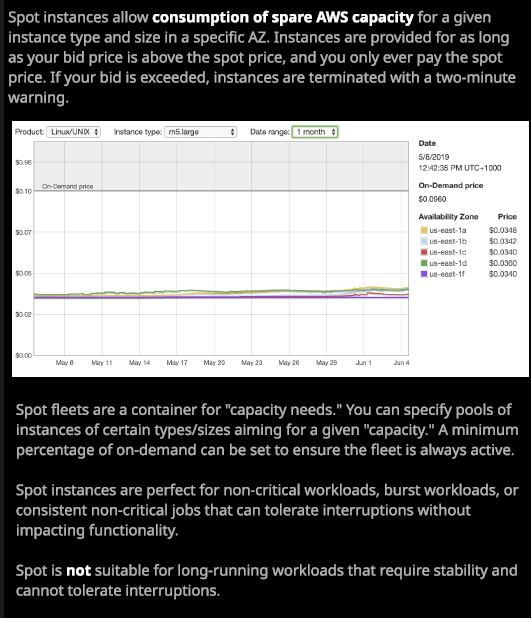
Billing models that are available within EC2.

Now so far in the course I've talked about on demand billing. This is the typical or default EC2 billing model. **With on demand billing, you pay per second charge for EC2 instances with a 60 second minimum when the instance is running it immediately gets charged for 60 seconds of usage and then continues after those 60 seconds on a per second basis. This is based on the type and size of the instance and the region that it's launched in.** So that's on demand billing and it's the default whenever you're launching instances into EC2 and when **you're not aware of what the expected load or the projected load is, then you should default to on demand.**



Now there are other building models available within EC2. We've got **spot billing and a close relation spot fleet.** With on demand there's no relationship or advanced warning between you and AWS. Think about this for a second when you launch an instance using on demand . So the default way with EC2 you just launched that instance, and when you launch it, you pay a cost for that instance. The flip side of that from AWS's perspective is that they're not aware of your usage and because they don't know your usage it's hard for them to plan. So to capacity plan the advanced purchase of physical capacity. Remember, when you're launching EC2 instances at the same time, you probably got hundreds or thousands or hundreds of thousands of other accounts doing the same, and at some AWS need to purchase additional capacity to meet those demands. When you're using on demand, you're not giving AWS any advance warning until that is a fairly difficult process. **When AWS do purchase physical capacity, they've got two problems. The first is they have to do it in advance, so there aren't on going capacity issues.** What you don't want is you launching an EC2 instance and not being able to so AWS have to purchase in advance, and that means that they always have spare capacity. Now throughout a 24 hour period there are also going to be times during that period where demand is high and times where demand is low and so there are going to be times your a normal day where AWS have short term spare capacity and because AWS purchase infrastructure in advance there's also going to be long term spare capacity.

**Now spot pricing and reserved instances are two ways that you and AWS can make a deal or solve each other’s problems.** So, depending on the type of usage that you've got, **you'll generally have an ability to select spot or select reserve pricing and in doing so, you're either solving AWS's capacity problem or you're informing them in advance of your expected usage and in both of those billing models, you're exchanging something. You're giving AWS a benefit, and in return, you're getting better instance pricing.**

So let's have a look at both the spot billing model and the reserved instance billing model. Now the model around spot billing is fairly easy to understand. I mentioned a second ago AWS have spare capacity in a region they might have a lot of spare capacity or not so much spare capacity. That spare capacity might also be different, based on the different types and sizes of instances, It might be different based on different availabilities zones. The idea is this. **You can launch given instance so we might click on launch instance. Pick a particular operating system, a particular size and type of instance.** **Well, when we go to configure instance details, we can request spot instances. So how this works is that for each type and size of instance, in every availability zone AWS has a spot price. A current spot price and that represents the available capacity in the region. If there is a lot of spare capacity then that current spot price will be relatively low. As spare capacity decreases then the spot price will go up.** Now you bid a maximum price if your maximum prices of both the spot price on this graphic, let's say you bid seven cents. **If your maximum price is above the spot price you get the EC2 instance, and you get it for the current spot price. So you never pay more than the current spot price. Now as spare capacity decreases the spot price will go up, and at some point it might go above your maximum price and when this happens, you're given a warning.** **It's two minutes, and then your instance is terminated.** **So spot pricing can give you access to instances at a massively reduced cost but it comes with the risk that the instances will be terminated.**

You bid a maximum price, and you only have to pay the spot price. The spot price, though, can go up and down. Spot pricing could save you as much as 90% of the costs of an EC2 instance but spot pricing can actually go above the normal on demand price, in some cases a lot higher than the on demand price. So you need to make sure that you **pick your maximum price really carefully**. So when would you select to use spot pricing? That's a key question that's relevant to the exam. Well, imagine that you have a scientific application and it runs on one EC2 instance, and it performs some work. Let's say that work isn't time critical. **You could use spot instances to buy instance time when it's really cheap as long as you work load can tolerate the termination of an instance part of the way through a job then you might just have that one running on demand EC2 instance, and you could farm out those jobs to spot instances that could give you a 90% saving**. **You might have a web application, which is stateless. Remember, that means the session state, the user session state, isn't stored on the server so every server is the same. You could launch spot instances during periods of high demand instead of using on demand instances. What you wouldn't use spot instances for are things that can't tolerate failure. Things like file servers, game servers, active directory domain controllers, or applications which always need an active server because if they can't tolerate failure, then there's always a chance that spot instance is going to be terminated, and that will disrupt service.** So when you're thinking about these type of questions in the exam, **you need to think of spot instances as the best product feature to use when your workloads can tolerate failure or when your workload store their state somewhere else in the case of web applications, and if either of those two apply than spot instances can offer a really good way of reducing your EC2 costs.**

Now **spot fleets are an evolution of normal spot pricing.** Spot fleet you can think of it as a container at a high level. You're making a request for a certain capacity an amount of compute and giving AWS a directive on how to fulfill it. You can define pools of certain types and sizes of instances, and these could be different, and you give AWS some flexibility on which type and size of instance to use and get you the best value. You can also direct AWS to make sure that for a given spot fleet, a certain amount of that fleet is based on on demand instances. So if the fleet is always going to be active then using on demand is a great way of fulfilling baseload and then using spot instances to provide that variable load and you'll do that and always achieve the best cost efficiency.

**You'd utilize spot pricing when you need really cheap EC2 capacity, when you can tolerate failure, when your workloads aren't time sensitive, or you might use spot pricing as a buffer to cope with bursts, for example, during sale periods. If you do you spot pricing you need to make sure that your application can tolerate failure. If you use it for web applications the architecture needs to make sure that the state is not on those servers because you don't want to disrupt your user experience.**