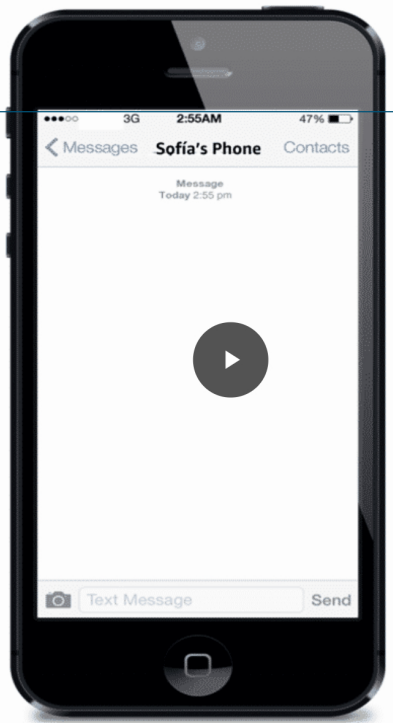
Lesson 5 of 12

Selecting the Correct Instance Type

Sofía has been busy ensuring that each member of her technical team is getting ready for their part in the company's migration to the AWS Cloud. She hasn't seen John in a few hours, so she reaches out through text messaging to his phone.

Choose the play button to read the text message discussion between Sofía and John. This video has no audio associated with it. For screen reader users, the text conversation is listed in the Transcript section below the video. To slow the speed of the video, select the play button and then change the speed (.75x to slow the speed or 1.25x to increase the speed) by changing the number to the right of the play bar.



Transcript +

Use cases

Amazon EC2 offers a vast array of instance types and oftentimes, you will find that your workload could run on several instance types and sizes, and it would run perfectly fine. The goal in choosing the correct instance type for your workload is to measure your performance and cost needs. Any application would run great on a 32xlarge instance, but is this instance really the right size if you were running a couple of ecommerce websites? Your performance would be great, but would the cost be giving you the value that you deserve?

Right-sizing is the process of matching the resource types and sizes to your workload requirements. It's also the process of looking at deployed instances and identifying opportunities to eliminate or downsize without compromising capacity or other requirements. Imagine you are grocery shopping. If you buy too much food, it will go to waste. If you buy too little food, you will go hungry. You want the right amount to suit your needs, no more, no less.

Right-sizing is a key mechanism for optimizing AWS costs, but it is often ignored by organizations when they first move to the AWS Cloud. Many organizations rebuild their identical on-premises servers in the cloud and plan to go back and right-size later. Oftentimes, speed and performance are prioritized over cost, which results in oversized, underused instances and a lot of money spent

on unused resources.

Experiment with instance types

Picking an EC2 instance for a given workload means finding the instance family that most closely matches the CPU and memory needs of your workload. However, sometimes an instance is chosen for cost over performance or the reverse, and in those cases, changing instance types offers you the flexibility to change your mind.

John finds an AWS use case that matches the workload that he's managing. He shows Sofía the use case and how Geodata changed instance types to save cost and still get the performance they needed.

Geodata provides geographical information systems (GIS) software and technical and professional services across Norway. The company specializes in map data analytics and visualization solutions based on Esri software.

To learn more, select the appropriate tab.

CHALLENGE | SOLUTION | BENEFITS

Continually evaluate new EC2 instances to identify innovative, cost-effective ways to process its workloads.

Using newer processor generations

Sofía really likes this use case. It's a great example of how using the newer generations of AWS processors can improve performance while lowering costs. She explains that the A1 instance uses a first-generation Graviton processor. Amazon EC2 offers you the option to run different processor architectures in your instances. Currently you can run Intel, AMD, and ARM (Graviton) architectures. Always assess which instance type and which processor type is the right option for your price-performance ratio. To see if Graviton is a good choice for your workload, see AWS Graviton Processor.

Geodata Use Case
To read the use case, choose the Use Case button.

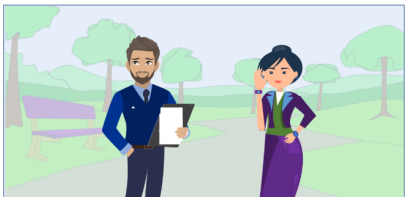
USE CASE

Changing instance types

After you have decided on an instance, you are not required to use just that one instance type forever. As your needs change, you might find that your instance is overused (the instance type is too small) or underused (the instance type is too large). If this is the case, you can resize your instance by changing its instance type. For example, if your t2.micro instance is too small for its workload, you can increase its size by changing it to a bigger T2 instance type, such as t2.large. Or you can change it to another instance type, such as m5.large. You might also want to change from a previous generation to a current generation instance type to take advantage of some features, such as support for IPv6.

Some instance types can be changed while the instance is running. Other instance changes, especially if you are switching to a different processor type, require you to stop the instance. You also have the option of launching a completely new instance type and migrating your application to the new instance. There are always options available to help you get to or move to the instance for your application. Therefore, choose the instance you think is right for your workload, and then adjust if you determine that another instance would be more appropriate. You are not locked into your decision.

Reviewing John's workflows



John and Sofía review the requirements for the log servers and the ML workflows that John just finished baselining. There are several different instance types that would meet the performance needs. John came from a data center and has experience with sizing and procuring hardware on site. Because of his past experience, John wants to size the instance larger so

that he doesn't need to worry about adding instances or having his workloads slowed down on peak performance days.

Sofía explains the AWS pricing model on their way back to the building, but John is struggling to understand. He needs a better explanation of Amazon EC2 pricing and economies of scale.