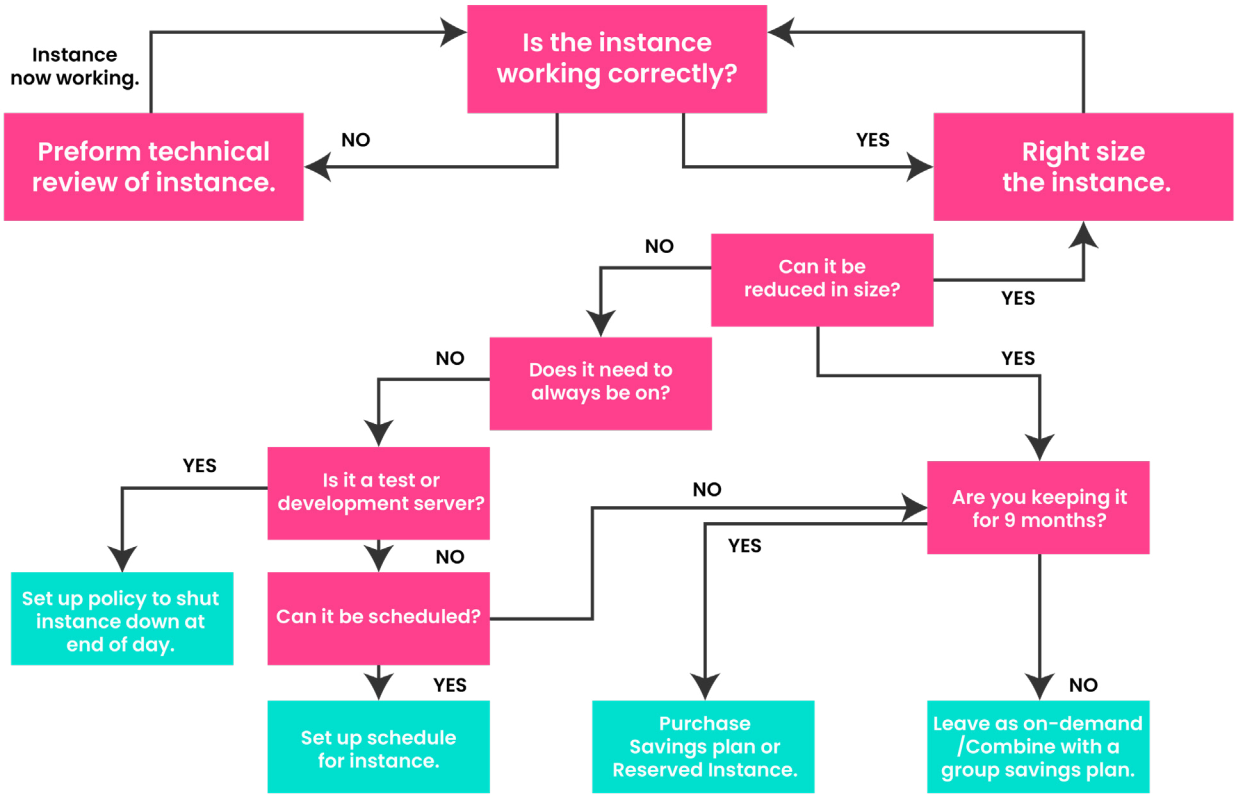
digital space
powered by aws

Journey to Optimising Elastic Cloud Compute (EC2)

Introduction

I recently had a conversation with a colleague about the best way to explain to customers how to optimise Elastic Cloud Compute (let us call it EC2 from here on out) instances within AWS. Not one to let a good discussion go to waste, I thought it would be a clever idea to share my thoughts on the process.

After the call, I began creating a document to explain the general lifecycle of a typical EC2 instance. However, let us be honest - a decision tree or spider diagram is much more engaging.



In my view, this is the most basic approach to optimisation, but it can be a bit confusing, so let us delve into some specifics below.

Start with the Technology

Before you start focusing on "how cheap can I make this instance," it is essential to approach EC2 instance optimisation from a technical standpoint. If you are like me and come from a non-technical background, this is where I rely on the technology team to confirm that the instance is fit for purpose. You can use CPU and RAM data to get a good indication that you have enough resources for your workload, but do not forget to ask the people using the application or system running on the instance before making any changes. If your company does not have metrics to measure performance before moving to the cloud, this could be a suitable place to start.

Are You Using It?

Once you are confident that the EC2 instance is effective when in use, the next step is to consider, "Do I always need this on?" In a data centre, where you typically invest significant CapEx in infrastructure, maximising usage can be the most efficient way to operate. However, in the cloud, where you only pay when the server is in use, you can be more imaginative with your server uptimes.

Let us take an example: Suppose you are running a business-critical application on an EC2 instance. Alongside this, you have a development and test server of the same specification. Obviously, the critical application cannot be turned off at will, but what about the test and dev instances?

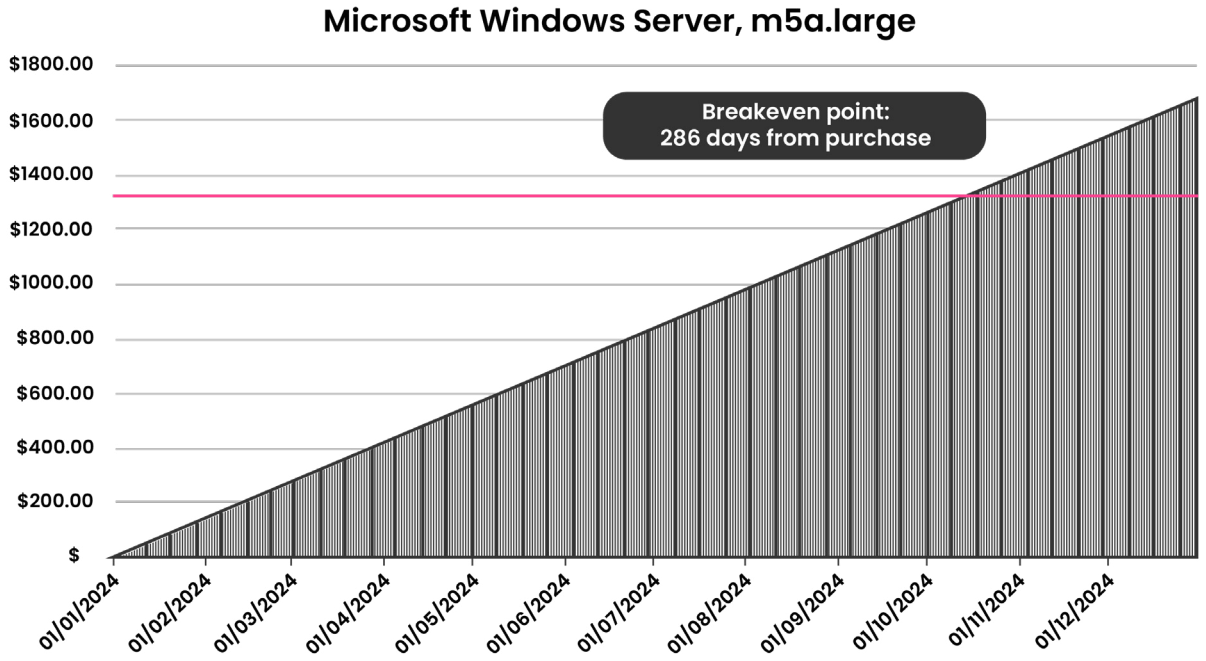
Did you know that a year has 8,760 hours? (Well, 2024 has 8,784, but let us stick with the norm.) If you leave your dev server running all the time, you would be charged for all 8,760 hours of the year. But if you turn it off on weekends, this drops to 6,240 hours. What about outside of work hours? If you consider a typical work week of 9-5:30, Monday to Friday, the 8,760 hours drop to 2,210 hours! So, if you were running an m5a.large on Windows in London ($0.192 per hour), the yearly cost would drop from $1,681.92 to $424.32 - a saving of $1,257.60 per year, or 75%. And that is not even considering bank holidays or annual leave.

This is a straightforward example, but it illustrates the kind of benefits you can achieve by determining whether an instance needs to be on all the time. You do not even have to remember to turn the instance on when needed; you can set up scheduling groups that automatically turn the instances on and off as required.

Needs to Be On

If you have determined that an instance needs to be on all the time - like the business-critical application mentioned earlier - you might think you can simply purchase a commitment and forget about it. However, I urge you to think carefully before doing so. You might have noticed the question "Are you keeping this for 9 months?" in the above diagram. This is where it is important to consider your options. The usual breakeven point for a 1-Year All Upfront (AUF) Reserved Instance or Savings Plan is around 20% for a Windows instance. You save a bit more for Linux instances (40%) due to the licensing commitment, and a bit less for SQL instances (10%), again because of the licensing costs. (I will be writing another blog on optimisation and licensing assessment in the future, so keep an eye out for that.) But 30% is a good benchmark. 80% of the year takes you to around month 10, which conveniently aligns with the question, "Will you still be using this in Q3?"

Here is the breakeven analysis for the m5a.large instance mentioned earlier: If you made the Reserved Instance purchase on January 1, 2024, you would reach breakeven on your $1,321.00 commitment by October 13, 2024, showing a saving of $360.92 (21% saving).



You might also consider that the application or service running on the EC2 instance could be modernised during the 9-month window to increase efficiency or reliability, or, in the case of Windows instances, eliminate the need for licensing altogether. At Digital Space, we recently conducted a modernisation exercise where we suggested moving customers away from Citrix on EC2 instances and onto AppStream 2.0.

How Can Digital Space Help?

Digital Space is an AWS Managed Service Provider and an AWS Well-Architected Partner. As a member of the FinOps Foundation and head of the FinOps practice within Digital Space, I spend my time understanding bills, forecasting costs, and helping customers optimise their spend.

Get my dedicated FinOps advice alongside your AWS infrastructure at no extra cost to you.

Book a FinOps Review with me today here.

Antony Mitchell, FinOps Specialist, Digital Space

Request a free Cloud FinOps Review with us today - FinOps Review