

AWS instance pricing

Amazon Web Services provide on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered, pay-as-you-go basis.

With AWS you pay only for the individual services you need, for as long as you use them, and without requiring long-term contracts or complex licensing. You only pay for the services you consume, and once you stop using them, there are no additional costs or termination fees.

Pricing is per instance-hour consumed for each instance, from the time an instance is launched until it is terminated or stopped.

(a) What are the parameters that will impact the price?

There are many parameters that can impact the price when creating AWS instances, such as:

- The region in which you choose to deploy your instance. There are many available AWS regions you can choose from to deploy your instance, and they are spread across multiple countries around the world.
- The number of vCPU ,memory size, storage size, and network performance, are all parameters you can configure when launching an instance, that impacts the price in a way such that if you increase these parameter's size, you will increase the price accordingly.
- The type of instance you are using, for example if you're launching an RDS, different database engines vary in price, for example MySQL costs less than Aurora engine.
- The operating system you choose to run your ec2 instance on (for example Linux, windows, ubuntu...)
- If you're deploying an RDS, the price varies if you use single-AZ or multi-AZ (availability zones).
- If you are using reserved instances, on-demand, or spot instances.
- The number of instances that you use.
- Change in prices, AWS offers pay-as-you-go pricing for most services and prices may change over time.
- There could be taxes that may be applied to your purchase of the services.

Note: there are many more parameters that impact the price.

(b) How can you reduce the price?

- Choose a cheaper region, as long as it still meets the requirements of your applications and workload.

- Lower the number of vCPUs and memory size if your workload doesn't require that much.
- Choose a cheaper instance type which meets the requirements of your workload, for example, if MySQL RDS is enough and meets the requirements, choose it instead of a more expensive option like Aurora engine.
- You can use reserved instances as opposed to on-demand if you know you are going to use the instance for a long period of time, since reserved instances offer a much cheaper price. Also, that way if the prices change and become more expensive, as they may change, by using the reserved instance you ensure to keep paying for the cheaper price for as long as the reserved period (1 year or 3 years).
- You can use spot instances, which are cheaper than on-demand by 90%. If your application is flexible and fault-tolerant. Such as a stateless web server.

(c) What can you look at to optimize the cost?

- You can split the workload across multiple regions. If there are components that are not sensitive to network latency, you can choose a cheaper region for this workload, that may be far from your geographical location, and thus pay less money for the same workload.
- You can use the AWS Budgets tool, which lets you set custom cost and usage budgets that alert you when your budget thresholds are exceeded or forecasted to exceed.
- You can enforce tagging, meaning whenever someone creates an instance they need to tag it, so that we know who created and who's using every instance, so that at the end of the month when we review the total price, we can ask the specific person or team who created a specific instance if they are still using it.
- We can create a schedule for our instance, for example if we know that they're going to be used only between the hours of 6 am to 6 pm, we can schedule for those instances to stop working between the hours of 7 pm to 5 am, they don't need to be working 24 hours, and that way we can save money and optimize the overall cost.
- You can use AWS Auto Scaling, which monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost, meaning you only pay for what you use and not for more resources that might be idle without you knowing or noticing.
- You can use the Amazon CloudWatch tool, which is a service that monitors applications, responds to performance changes, optimizes resource use, and provides insights into operational health. By collecting data across AWS resources, CloudWatch gives visibility into system-wide performance and allows users to set alarms, automatically react to changes, and gain a unified view of operational health.

- You can use the AWS Compute optimizer tool, which analyzes the current configuration of your AWS Compute resources, and their utilization metrics from Amazon CloudWatch over a period of the last 14 days. It identifies the resources that are optimal and not optimal, then offers recommendations for your resources to reduce their cost, improve their performance, or a combination of both.
- You can use the AWS Price calculator tool to tune the parameters to get the best price before launching your instance.
- You can also use third party tools to help optimize your cost, these tools provide detailed insights into AWS spending and offer additional functionalities like security and compliance monitoring, which are not as comprehensive in AWS's native offerings. Some of those tools are:
 - o 'CloudHealth by VMware': known for its robust cost optimization and governance features.
 - o 'CloudCheckr': Offers extensive cost and security management capabilities.
 - o 'Datadog': Provides real-time monitoring and analytics, extending beyond cost management.

Note: there are many more tools that AWS offers which we can use to optimize the cost, as well as third-party tools.

- Most of the information written here I got from the AWS website which offers an extensive explanation about all of its products, tools, and services, as well as some articles I found on the internet.
- Here are some links:
 - o <https://www.concurrencylabs.com/blog/choose-your-aws-region-wisely/>
 - o <https://aws.amazon.com/blogs/architecture/what-to-consider-when-selecting-a-region-for-your-workloads/>
 - o <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-reserved-instances.html>
 - o <https://aws.amazon.com/cloudwatch/>
 - o https://aws.amazon.com/pricing/?aws-products-pricing.sort-by=item.additionalFields.productNameLowercase&aws-products-pricing.sort-order=asc&awsf.Free%20Tier%20Type=*all&awsf.tech-category=*all
 - o <https://calculator.aws/#/>
 - o <https://redresscompliance.com/third-party-aws-cost-monitoring-tools/#:~:text=The%20market%20has%20various%20third,VMware%2C%20CloudCheckr%2C%20and%20Datadog.>