Right Sizing

Provisioning Instances to Match Workloads

January 2020

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided "as is" without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents

Abstract

This is the seventh in a series of whitepapers designed to support your cloud journey. This paper seeks to empower you to maximize value from your investments, improve forecasting accuracy and cost predictability, create a culture of ownership and cost transparency, and continuously measure your optimization status.

This paper discusses how to provision instances to match your workload performance and capacity requirements to optimize costs.

Introduction

Right sizing is the process of matching instance types and sizes to your workload performance and capacity requirements at the lowest possible cost. It's also the process of looking at deployed instances and identifying opportunities to eliminate or downsize without compromising capacity or other requirements, which results in lower costs.

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. They lift and shift their environments and expect to right size later. Speed and performance are often prioritized over cost, which results in oversized instances and a lot of wasted spend on unused resources.

Right Size Before Migrating

One reason for the waste is the mindset to overprovision that many IT professionals bring with them when they build their cloud infrastructure. Historically, IT departments have had to provision for peak demand. However, cloud environments minimize costs because capacity is provisioned based on average usage rather than peak usage.

When you learn how to right size, you can save up to 70% percent on your monthly bill. The key to right sizing is to understand precisely your organization's usage needs and patterns and know how to take advantage of the elasticity of the AWS Cloud to respond to those needs.

By right sizing before a migration, you can significantly reduce your infrastructure costs. If you skip right sizing to save time, your migration speed might be faster, but you will end up with higher cloud infrastructure spend for a potentially long time.

Right Sizing is an Ongoing Process

To achieve cost optimization, right sizing must become an ongoing process within your organization. It's important to right size when you first consider moving to the cloud and calculate total cost of ownership, but it's equally

aws

important to right size periodically once you're in the cloud to ensure ongoing cost-performance optimization.

Why is it necessary to right size continually? Even if you right size workloads initially, performance and capacity requirements can change over time, which can result in underused or idle resources. Additionally, new projects and workloads require additional cloud resources, and overprovisioning is the likely outcome if there is no process in place to support right sizing and other cost-optimization efforts.

You should right size your workloads at least once a month to control costs. You can make right sizing a smooth process by:

- Having each team set up a right-sizing schedule and then report the savings to management.

- Monitoring costs closely using AWS cost and reporting tools, such as Cost Explorer, budgets, and detailed billing reports in the Billing and Cost Management console.

- Enforcing tagging for all instances so that you can quickly identify attributes such as the instance owner, application, and environment (development/testing or production).

- Understanding how to right size.

We first describe the types of instances that AWS offers and then discuss key considerations for right sizing your instances.

Overview of Amazon EC2 and Amazon RDS Instance Families

Picking an Amazon Elastic Compute Cloud (Amazon EC2) instance for a given workload means finding the instance family that most closely matches the CPU and memory needs of your workload. Amazon EC2 provides a wide selection of instances, which gives you lots of flexibility to right size your compute resources to match capacity needs at the lowest cost. There are five families of EC2 instances with different options for CPU, memory, and network resources:

aws

- **General purpose** (includes T2, M3, and M4 instance types) –
  T2 instances are a very low-cost option that provide a small amount of
  CPU resources that can be increased in short bursts when additional
  cycles are available. They are well suited for lower throughput
  applications such as administrative applications or low-traffic websites.

  M3 and M4 instances provide a balance of CPU, memory, and network
  resources and are ideal for running small and midsize databases, more
  memory-intensive data processing tasks, caching fleets, and backend
  servers.

- **Compute optimized** (includes the C3 and C4 instance types) – Have a
  higher ratio of virtual CPUs to memory than the other families and the
  lowest cost per virtual CPU of all the EC2 instance types. Consider
  compute-optimized instances first if you are running CPU-bound, scale-
  out applications such as frontend fleets for high-traffic websites, on-
  demand batch processing, distributed analytics, web servers, video
  encoding, and high-performance science and engineering applications.

- **Memory optimized** (includes the X1, R3, and R4 instance types) –
  Designed for memory-intensive applications, these instances have the
  lowest cost per GiB of RAM of all EC2 instance types. Use these
  instances if your application is memory-bound.

- **Storage optimized** (includes the I3 and D2 instance types) – Optimized
  to deliver tens of thousands of low-latency, random input/output (I/O)
  operations per second (IOPS) to applications. Storage-optimized
  instances are best for large deployments of NoSQL databases.

  I3 instances are designed for I/O-intensive workloads and equipped with
  super-efficient NVMe SSD storage. These instances can deliver up to
  3.3 million IOPS in 4 KB blocks and up to 16 GB/second of sequential
  disk throughput.

  D2 or dense storage instances are designed for workloads that require
  high sequential read and write access to very large data sets, such as
  Hadoop distributed computing, massively parallel processing data
  warehousing, and log-processing applications.

- **Accelerated computing** (includes the P2, G3, and F1 instance types) – Provide access to hardware-based compute accelerators such as graphics processing units (GPUs) or field programmable gate arrays (FPGAs). Accelerated-computing instances enable more parallelism for higher throughput on compute-intensive workloads.

Amazon Relational Database Service (Amazon RDS) database instances are similar to Amazon EC2 instances in that there are different families to suit different workloads. These database instance families are optimized for memory, performance, or I/O:

- **Standard performance** (includes the M3 and M4 instance types) – Designed for general-purpose database workloads that don't run many in-memory functions. This family has the most options for provisioning increased IOPS.

- **Burstable performance** (includes T2 instance types) – For workloads that require burstable performance capacity.

- **Memory optimized** (includes the R3 and R4 instance types) – Optimized for in-memory functions and big data analysis.

Identifying Opportunities to Right Size

The first step in right sizing is to monitor and analyze your current use of services to gain insight into instance performance and usage patterns. To gather sufficient data, observe performance over at least a two-week period (ideally, over a one-month period) to capture the workload and business peak. The most common metrics that define instance performance are vCPU utilization, memory utilization, network utilization, and ephemeral disk use. In rare cases where instances are selected for reasons other than these metrics, it is important for the technical owner to review the right-sizing effort.

Tools for Right Sizing

You can use the following tools to evaluate costs and monitor and analyze instance usage for right sizing:

aws

- Amazon CloudWatch – Lets you observe CPU utilization, network throughput, and disk I/O, and match the observed peak metrics to a new and cheaper instance type. You can also regularly monitor Amazon EC2 Usage Reports, which are updated several times a day and provide in-depth usage data for all your EC2 instances. Typically, this is feasible only for small environments given the time and effort required.

- AWS Cost Explorer – This free tool lets you dive deeper into your cost and usage data to identify trends, pinpoint cost drivers, and detect anomalies. It includes Amazon EC2 Usage Reports, which let you analyze the cost and usage of your EC2 instances over the last 13 months.

- AWS Trusted Advisor – Lets you inspect your AWS environment to identify idle and underutilized resources and provides real- time insight into service usage to help you improve system performance and reliability, increase security, and look for opportunities to save money.

- Third-party monitoring tools, such as CloudHealth, Cloudability, and CloudCheckr, are also an option to automatically identify opportunities and suggest alternate instances. These tools have years of development effort and customer feedback points built into them. They also provide additional cost management and optimization functionality.

Tips for Developing Your Own Right-Sizing Tools

You can also develop your own tools for monitoring and analyzing performance. The following guidelines can help if you are considering this option:

- Focus on instances that have run for at least half the time you're looking at.

- Focus on instances with lower reserved instance coverage.

- Exclude resources that have been switched off (reducing search effort).

- Avoid conversions to older-generation instances, where possible.

- Apply a savings threshold below which right sizing is not worth considering.

- Make sure the following conditions are met before you switch to a new instance:

- o The vCPU of the new instance is equal to that of the old instance *or* the application's observed vCPU is less than 80% of the vCPU capacity of the new instance.

- o The memory of the new instance is equal to that of the old instance *or* the application's observed memory peak is less than 80% of the memory capacity of the new instance.

  Note: You can capture memory utilization metrics by using monitoring scripts that report these metrics to Amazon CloudWatch. For more information, see [Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances](#).

- o The network throughput of the new instance is equal to that of the old instance *or* the application's network peak is less than the network capacity of the new instance.

  Note: Maximum NetworkIn and NetworkOut values are measured in bytes-per-minute. Use the following formula to convert these metrics to megabits per second:

  Maximum NetworkIn (or NetworkOut) x 8 (bytes to bits) /1024/1024/ 60 = Number of Mbps

- o If the ephemeral storage disk I/O is less than 3,000, you can use [Amazon Elastic Block Store](#) (Amazon EBS) storage. If not, use instance families that have ephemeral storage.

  For more information, see [Amazon EBS Volume Types](#).

Tips for Right Sizing

This section offers tips to help you right size your EC2 instances and RDS DB instances.

Right Size Using Performance Data

Analyze performance data to right size your EC2 instances. Identify idle instances and ones that are underutilized. Key metrics to look for are CPU usage and memory usage. Identify instances with a maximum CPU usage and memory usage of less than 40% over a four-week period. These are the instances that you will want to right size to reduce costs.

aws

For compute-optimized instances, keep the following in mind:

- Focus on very recent instance data (old data may not be actionable).

- Focus on instances that have run for at least half the time you're looking at.

- Ignore burstable instance families (T2 instance types) because these families are designed to typically run at low CPU percentages for significant periods of time.

For storage-optimized instances (I2 and D2 instance types), where the key feature is high data IOPS, focus on IOPS to see whether instances are overprovisioned. Keep the following in mind for storage-optimized instances:

- Different size instances have different IOPS ratings, so tailor your reports to each instance type. Start with your most commonly used storage-optimized instance type.

- Peak NetworkIn and NetworkOut values are measured in bytes per minute. Use the following formula to convert these metrics to megabits per second:

  Maximum NetworkIn (or NetworkOut) x 8 (bytes to bits) /1024/1024/ 60 = Number of Mbps

- Take note of how I/O and CPU percentage metrics change during the day and whether there are peaks that need to be accommodated.

Right size against memory if you find that maximum memory utilization over a four-week period is less than 40%. AWS provides sample scripts for monitoring memory and disk space utilization on your EC2 instances running Linux. You can configure the scripts to report the metrics to Amazon CloudWatch.

When analyzing performance data for Amazon RDS DB instances, focus on the following metrics to determine whether actual usage is lower than instance capacity:

- Average CPU utilization

- Maximum CPU utilization

- Minimum available RAM

- Average number of bytes read from disk per second

- Average number of bytes written to disk per second

Right Size Based on Usage Needs

As you monitor current performance, identify the following usage needs and patterns so that you can take advantage of potential right-sizing options:

- **Steady state** – The load remains at a relatively constant level over time, and you can accurately forecast the likely compute load. For this usage pattern, you might consider Reserved Instances, which can provide significant savings.

- **Variable, but predictable** – The load changes, but on a predictable schedule. Auto Scaling is well suited for applications that have stable demand patterns with hourly, daily, or weekly variability in usage. You can use this feature to scale Amazon EC2 capacity up or down when you experience spiky traffic or predictable fluctuations in traffic.

- **Dev/test/production** – Development, testing, and production environments are typically used only during business hours and can be turned off during evenings, weekends, and holidays. (You'll need to rely on tagging to identify dev/test/production instances.)

- **Temporary** – For temporary workloads that have flexible start times and can be interrupted, you can consider placing a bid for an Amazon EC2 Spot Instance instead of using an On-Demand Instance.

Right Size by Turning Off Idle Instances

The easiest way to reduce operational costs is to turn off instances that are no longer being used. If you find instances that have been idle for more than two weeks, it's safe to stop or even terminate them. Before terminating an instance that's been idle for two weeks or less, consider:

- Who owns the instance?

- What is the potential impact of terminating the instance?

- How hard will it be to re-create the instance if you need to restore it?

Stopping an EC2 instance leaves any attached EBS volumes operational. You will continue to be charged for these volumes until you delete them. If you need the instance again, you can easily turn it back on. Terminating an instance,

aws

however, automatically deletes attached EBS volumes and requires effort to re-provision should the instance be needed again. If you decide to delete an EBS volume, consider storing a snapshot of the volume so that it can be restored later if needed.

Another simple way to reduce costs is to stop instances used in development and production during hours when these instances are not in use and then start them again when their capacity is needed. Assuming a 50-hour work week, you can save 70% by automatically stopping dev/test/production instances during non-business hours. Many tools are available to automate scheduling, including Amazon EC2 Scheduler, AWS Lambda, and AWS Data Pipeline, as well as third-party tools such as CloudHealth and Skeddly.

Right Size by Selecting the Right Instance Family

You can right size an instance by migrating to a different model within the same instance family or by migrating to another instance family. When migrating within the same instance family, you only need to consider vCPU, memory, network throughput, and ephemeral storage. A good, general rule for EC2 instances is that if your maximum CPU and memory usage is less than 40% over a four-week period, you can safely cut the machine in half. For example, if you were using a c4.8xlarge EC2, you could move to a c4.4xlarge, which would save $190 every 10 days.

When migrating to a different instance family, make sure the current instance type and the new instance type are compatible in terms of virtualization type, network, and platform:

- **Virtualization type** – The instances must have the same Linux AMI virtualization type (PV AMI versus HVM) and platform (EC2-Classic versus EC2-VPC). For more information, see Linux AMI Virtualization Types.

- **Network** – Some instances are not supported in EC2-Classic and must be launched in a virtual private cloud (VPC). For more information, see Instance Types Available Only in a VPC.

aws

- **Platform** – If your current instance type supports 32-bit AMIs, make sure to select a new instance type that also supports 32-bit AMIs (not all EC2 instance types do). To check the platform of your instance, go to the Instances screen in the Amazon EC2 console and choose **Show/Hide Columns, Architecture**.

When you resize an EC2 instance, the resized instance usually has the same number of instance store volumes that you specified when you launched the original instance. You cannot attach instance store volumes to an instance after you've launched it, so if you want to add instance store volumes, you will need to migrate to a new instance type that contains the higher number of volumes.

Right Size Your Database Instances

You can scale your database instances by adjusting memory or compute power up or down as performance and capacity requirements change. The following are some things to consider when scaling a database instance:

- Storage and instance type are decoupled. When you scale your database instance up or down, your storage size remains the same and is not affected by the change.

- You can separately modify your Amazon RDS DB instance to increase the allocated storage space or improve the performance by changing the storage type (such as General Purpose SSD to Provisioned IOPS SSD).

- Before you scale, make sure you have the correct licensing in place for commercial engines (SQL Server, Oracle), especially if you Bring Your Own License (BYOL).

- Determine when you want to apply the change. You have an option to apply it immediately or during the maintenance window specified for the instance.

Conclusion

Right sizing is the most effective way to control cloud costs. It involves continually analyzing instance performance and usage needs and patterns— and then turning off idle instances and right sizing instances that are either overprovisioned or poorly matched to the workload. Because your resource needs are always changing, right sizing must become an ongoing process to

aws

continually achieve cost optimization. You can make right sizing a smooth process by establishing a right-sizing schedule for each team, enforcing tagging for all instances, and taking full advantage of the powerful tools that AWS and others provide to simplify resource monitoring and analysis.

Contributors

Contributors to this document include:

- Amilcar Alfaro, Sr. Product Marketing Manager, AWS

- Erin Carlson, Marketing Manager, AWS

- Keith Jarrett, WW BD Lead – Cost Optimization, AWS Business Development

Document Revisions

Date	Description
January 2020	Minor revisions
March 2018	First publication