

# Cloud Computing Architecture

AWS Compute Services: EC2

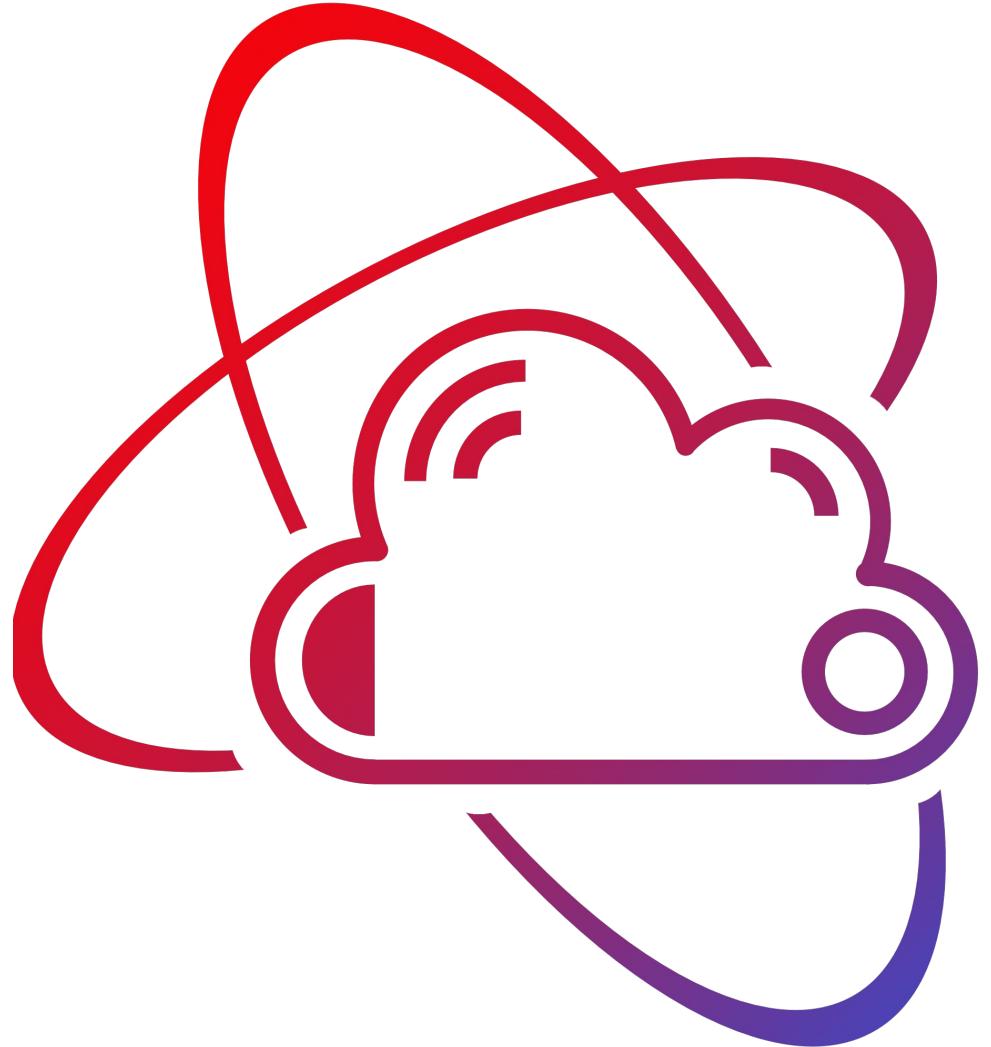


Image licensed under creative commons

- 
- 
- 
- 
- 
- 
- 
- 
- 
- 

## Acknowledgement of Country

We respectfully acknowledge the Wurundjeri People of the Kulin Nation, who are the Traditional Owners of the land on which Swinburne's Australian campuses are located in Melbourne's east and outer-east, and pay our respect to their Elders past, present and emerging.

We are honoured to recognise our connection to Wurundjeri Country, history, culture, and spirituality through these locations, and strive to ensure that we operate in a manner that respects and honours the Elders and Ancestors of these lands.

We also respectfully acknowledge Swinburne's Aboriginal and Torres Strait Islander staff, students, alumni, partners and visitors.

We also acknowledge and respect the Traditional Owners of lands across Australia, their Elders, Ancestors, cultures, and heritage, and recognise the continuing sovereignties of all Aboriginal and Torres Strait Islander Nations.



# AWS Compute Services: EC2

In this Presentation:

- Virtualized Computation
- AWS Compute Services Overview
- Amazon EC2
- EC2 Cost Optimization



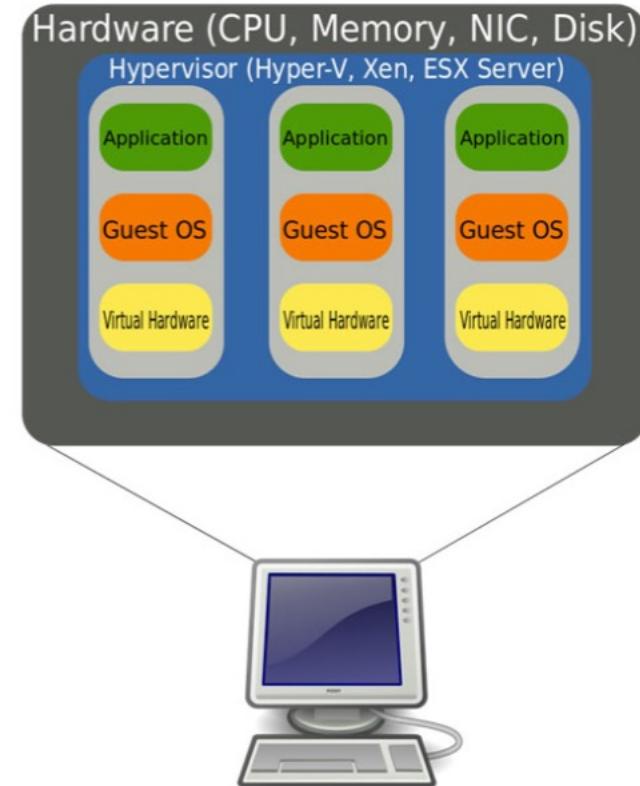
Image from: <https://medium.com/javarevisited/7-best-aws-ec2-amazon-elastic-compute-cloud-online-courses-for-beginners-in-2021-f7a1a55ea719>

# Virtualized Computation

# What do we mean by Virtualized Computation?

### Definitions:

- Virtualization is a technology that allows multiple virtual machines (VMs) to run on a single physical machine.
- Each VM has its own operating system and can run different applications, making virtualization a powerful tool for consolidating hardware and increasing resource utilization.
- Virtualization is widely used in data centres and cloud computing, as well as for remote work, online gaming, and other applications. It allows multiple workloads to share the same physical resources and improves the flexibility, scalability and manageability of IT infrastructures.



# History of Virtualization

- 1960s: IBM develops the concept of virtualization with the IBM System/360 mainframe
- 1970s: IBM releases VM/370, the first commercial implementation of virtualization
- 1980s: VMware is founded and develops a virtualization platform for x86-based computers
- 2000s: Virtualization becomes increasingly popular in data centers and cloud computing
- 2010s: Containers, a more lightweight form of virtualization, gain popularity for application deployment and management
- 2020s: Virtualization continues to be widely used in cloud computing, and is also used for remote work, online gaming, and other applications.

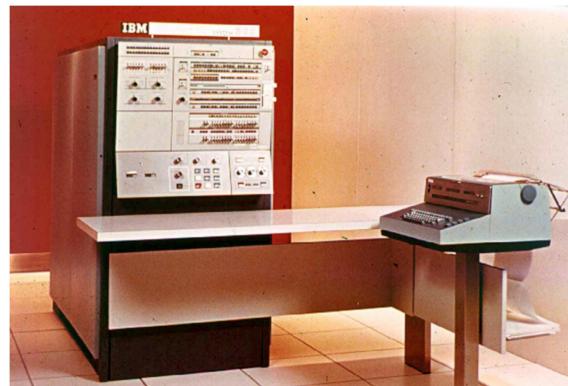


Image from:

<https://about.sourcegraph.com/blog/the-ibm-system-360-the-first-modular-general-purpose-computer>



Image from:

<https://www.ubackup.com/enterprise-backup/vmware-powercli-commands.html>

# What are 'Containers' in Virtualization?

- Containers are a form of operating system virtualization that allows applications to run in isolated environments on a single host.
- Containers use less resources than traditional virtual machines (VMs) and can start and stop faster, making them more efficient for application deployment and scaling.
- Containers rely on the host operating system's kernel, while VMs run on a dedicated hypervisor, which provides a full virtualized hardware environment to each VM. This makes containers more lightweight and portable than VMs.

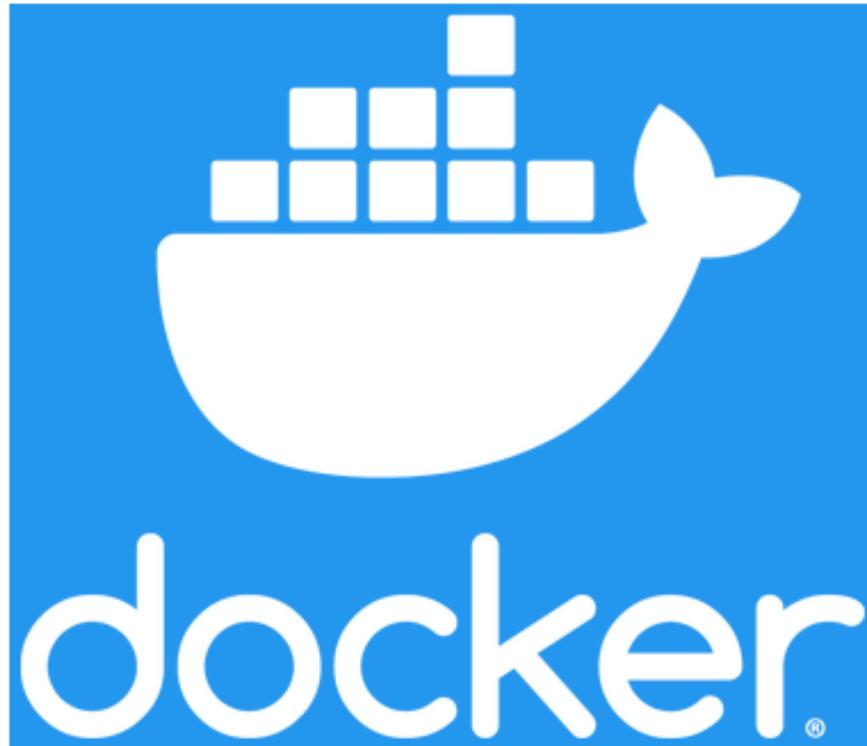


Image from:  
<https://www.docker.com/company/newsroom/media-resources/>

# Serverless Virtual Computation

- Virtualization and serverless computation are complementary technologies that can be used together to optimize resource usage and reduce costs.
- Serverless computing allows developers to run their code without the need to provision or manage servers, which eliminates the need for virtual machines (VMs) for running the code.



Image from: <https://allcode.com/what-is-aws-lambda/>

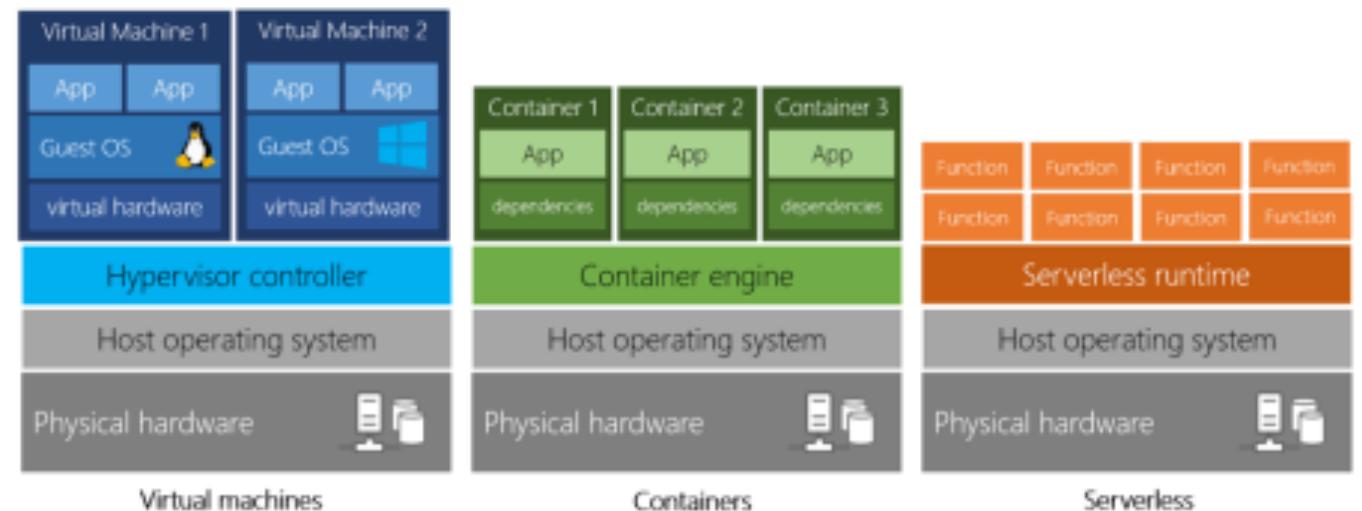


Image from: <https://allaboutdynamic.com/2020/05/30/azure-fundamentals-virtual-machines-vs-containers-vs-serverless-computing/>

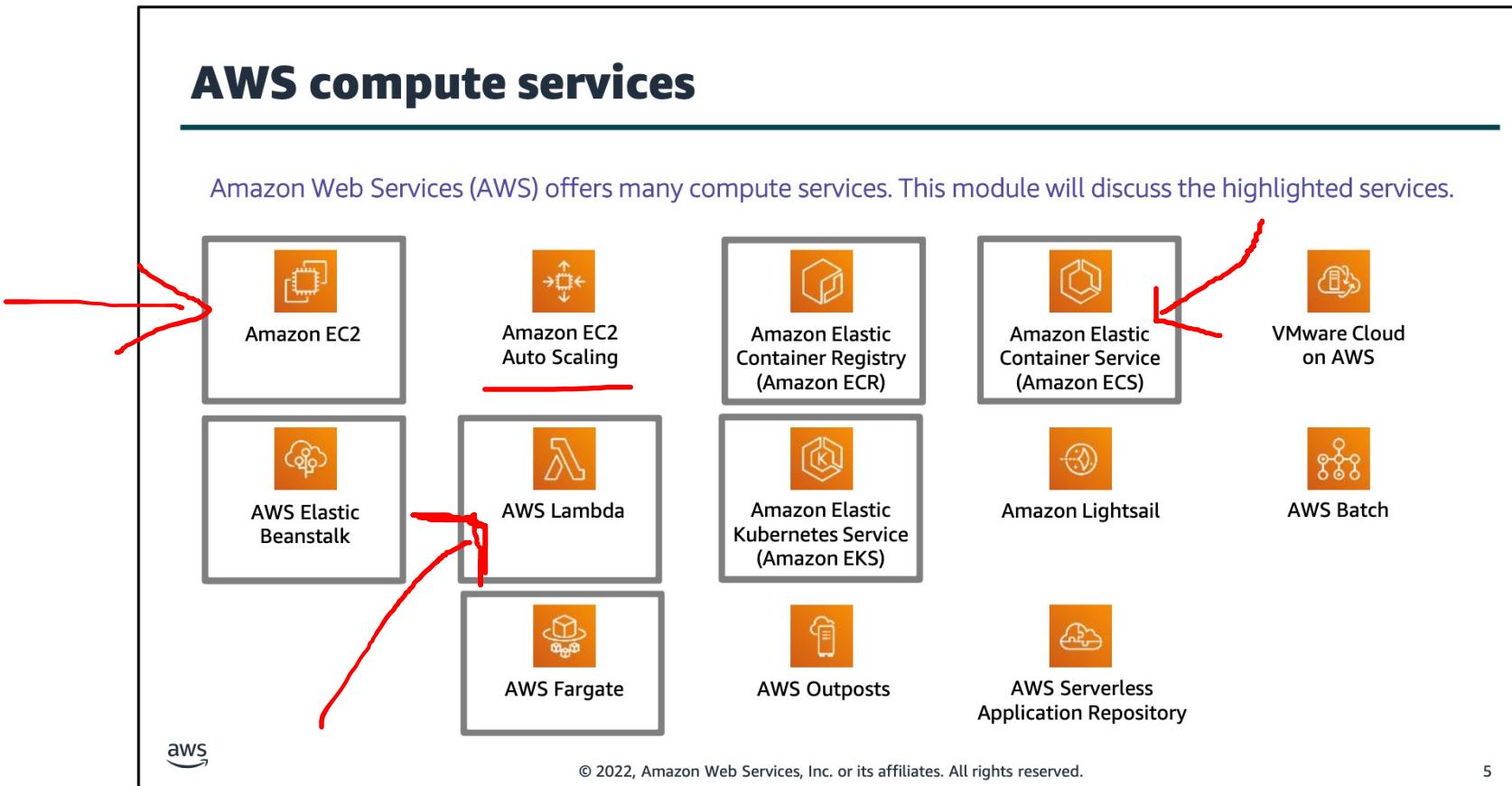
# Serverless Virtual Computation

- Virtualization can be used to run the underlying infrastructure for serverless services, such as containers or functions, thus allowing for more efficient usage of resources.
- Virtualization can also be used to run custom runtime environments, allowing developers to use languages and frameworks that are not natively supported by the serverless provider.
- Serverless providers such as AWS Lambda and Google Cloud Functions use virtualization to provide isolation between different functions and to scale instances as needed.



# AWS Compute Services Overview

# AWS Compute Services Overview



# AWS Compute Services Overview



# AWS Compute Services Overview

## **Choosing the optimal compute service**

- The optimal compute service or services that you use will depend on your use case
- Some aspects to consider –
  - What is your application design?
  - What are your usage patterns?
  - Which configuration settings will you want to manage?
- Selecting the wrong compute solution for an architecture can lead to lower performance efficiency
- A good starting place—Understand the available compute options



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

7

# AWS Compute Services Overview



Image from: <https://clouductivity.com/amazon-web-services/aws-lambda-vs-ecs/>

# Amazon EC2

# AWS Compute Services Overview

## Amazon EC2 overview

- **Amazon Elastic Compute Cloud (Amazon EC2)**
  - Provides virtual machines—referred to as **EC2 instances**—in the cloud.
  - Gives you *full control* over the guest operating system (Windows or Linux) on each instance.
  - You can launch instances of any size into an Availability Zone anywhere in the world.
    - Launch instances from **Amazon Machine Images (AMIs)**.
    - Launch instances with a few clicks or a line of code, and they are ready in minutes.
  - You can control traffic to and from instances.



Amazon  
EC2



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

10

# Amazon EC2

## Amazon EC2 storage options

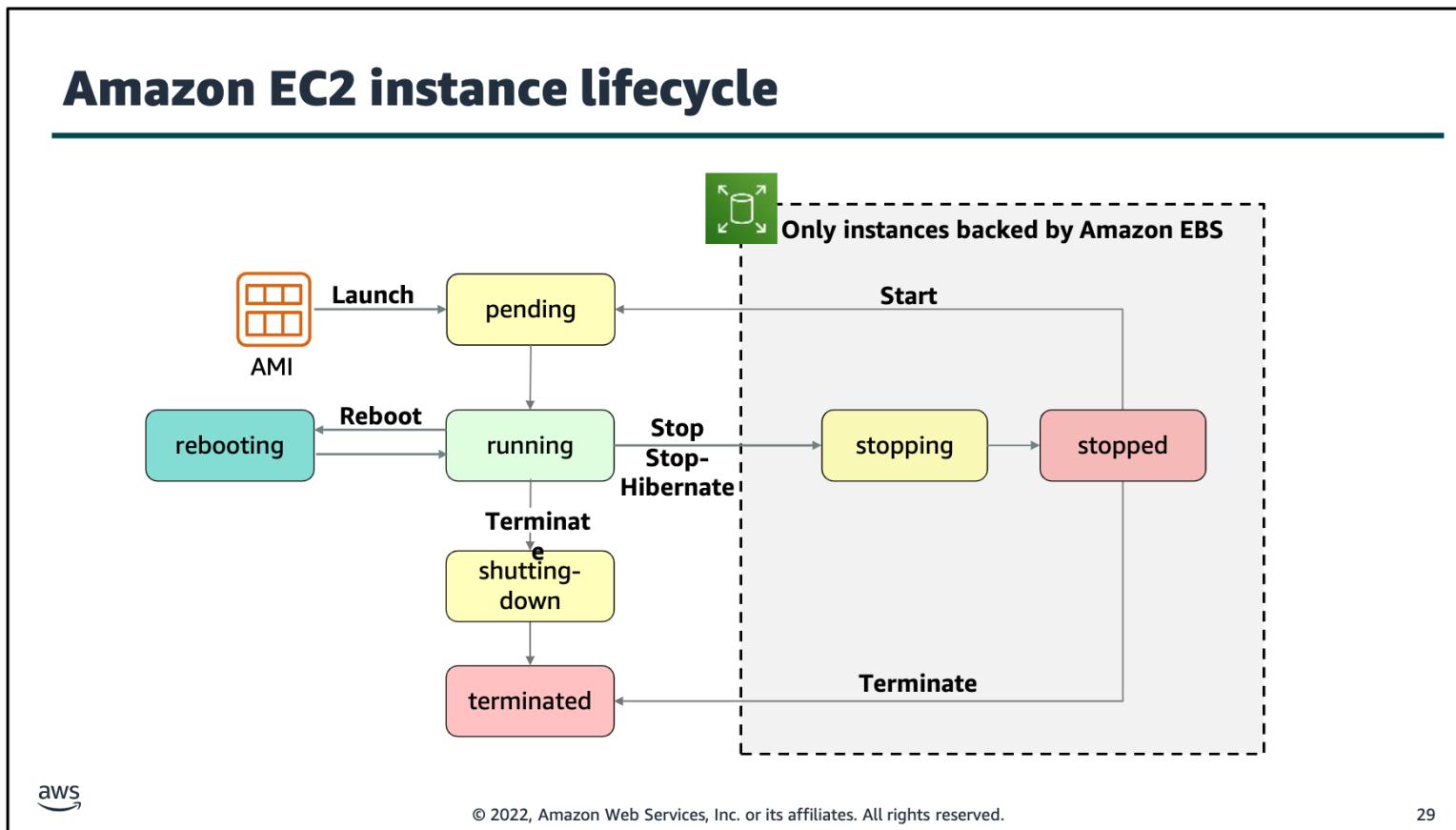
- **Amazon Elastic Block Store (Amazon EBS) –**
  - Durable, block-level storage volumes.
  - You can stop the instance and start it again, and the data will still be there.
- **Amazon EC2 Instance Store –**
  - Ephemeral storage is provided on disks that are attached to the host computer where the EC2 instance is running.
  - If the instance stops, data stored here is deleted.
- Other options for storage (not for the root volume) –
  - Mount an **Amazon Elastic File System (Amazon EFS)** file system.
  - Connect to **Amazon Simple Storage Service (Amazon S3)**.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

22

# Amazon EC2



# EC2 Cost Optimization

# EC2 Cost Optimization

The screenshot shows the AWS EC2 On-Demand Pricing page. At the top, there's a navigation bar with links like Products, Solutions, Pricing, Documentation, Learn, Partner Network, AWS Marketplace, Customer Enablement, Events, Explore More, and a search bar. To the right of the search bar are Contact Us, Support, English, My Account, and a Sign In to the Console button.

The main heading is "Amazon EC2 On-Demand Pricing". Below it are two buttons: "Get started for free" (orange) and "Request a pricing quote" (blue). The page title is "On-Demand Plans for Amazon EC2".

On the left, there's a sidebar with a "PAGE CONTENT" section containing links to various AWS services: On-Demand Pricing, Data Transfer, Data Transfer within the same AWS Region, EBS-Optimized Instances, Elastic IP Addresses, Carrier IP Addresses, Elastic Load Balancing, On-Demand Capacity Reservations, T2/T3/T4g Unlimited Mode Pricing, Amazon CloudWatch, Amazon Elastic Block Store, Amazon EC2 Auto Scaling, AWS GovCloud Region, and AMD SEV-SNP.

The main content area has sections for "Select a location type and region" (Location Type: AWS Region, Region: Asia Pacific (Sydney)) and "Select an operating system, instance type, and vCPU to view rates" (Operating system: Linux). Below these are dropdowns for Instance type (All) and vCPU (All).

A table titled "Viewing 567 of 567 available instances" lists the following data:

Instance name	On-Demand hourly rate	vCPU	Memory	Storage	Network performance
t4g.nano	\$0.0053	2	0.5 GiB	EBS Only	Up to 5 Gigabit
t4g.micro	\$0.0106	2	1 GiB	EBS Only	Up to 5 Gigabit
t4g.small	\$0.0212	2	2 GiB	EBS Only	Up to 5 Gigabit
t4g.medium	\$0.0424	2	4 GiB	EBS Only	Up to 5 Gigabit
t4g.large	\$0.0848	2	8 GiB	EBS Only	Up to 5 Gigabit

At the bottom right, there's a link: "Image from: <https://aws.amazon.com/ec2/pricing/on-demand/>".

# EC2 Cost Optimization

## Amazon EC2 pricing models

### On-Demand Instances

- Pay by the hour
- No long-term commitments.
- Eligible for the [AWS Free Tier](#).

### Dedicated Hosts

- A physical server with EC2 instance capacity fully dedicated to your use.

### Dedicated Instances

- Instances that run in a VPC on hardware that is dedicated to a single customer.

### Reserved Instances

- Full, partial, or no upfront payment for instance you reserve.
- Discount on hourly charge for that instance.
- 1-year or 3-year term.

### Scheduled Reserved Instances

- Purchase a capacity reservation that is always available on a recurring schedule you specify.
- 1-year term.

### Spot Instances

- Instances run as long as they are available and your bid is above the Spot Instance price.
- They can be interrupted by AWS with a 2-minute notification.
- Interruption options include terminated, stopped or hibernated.
- Prices can be significantly less expensive compared to On-Demand Instances
- Good choice when you have flexibility in when your applications can run.

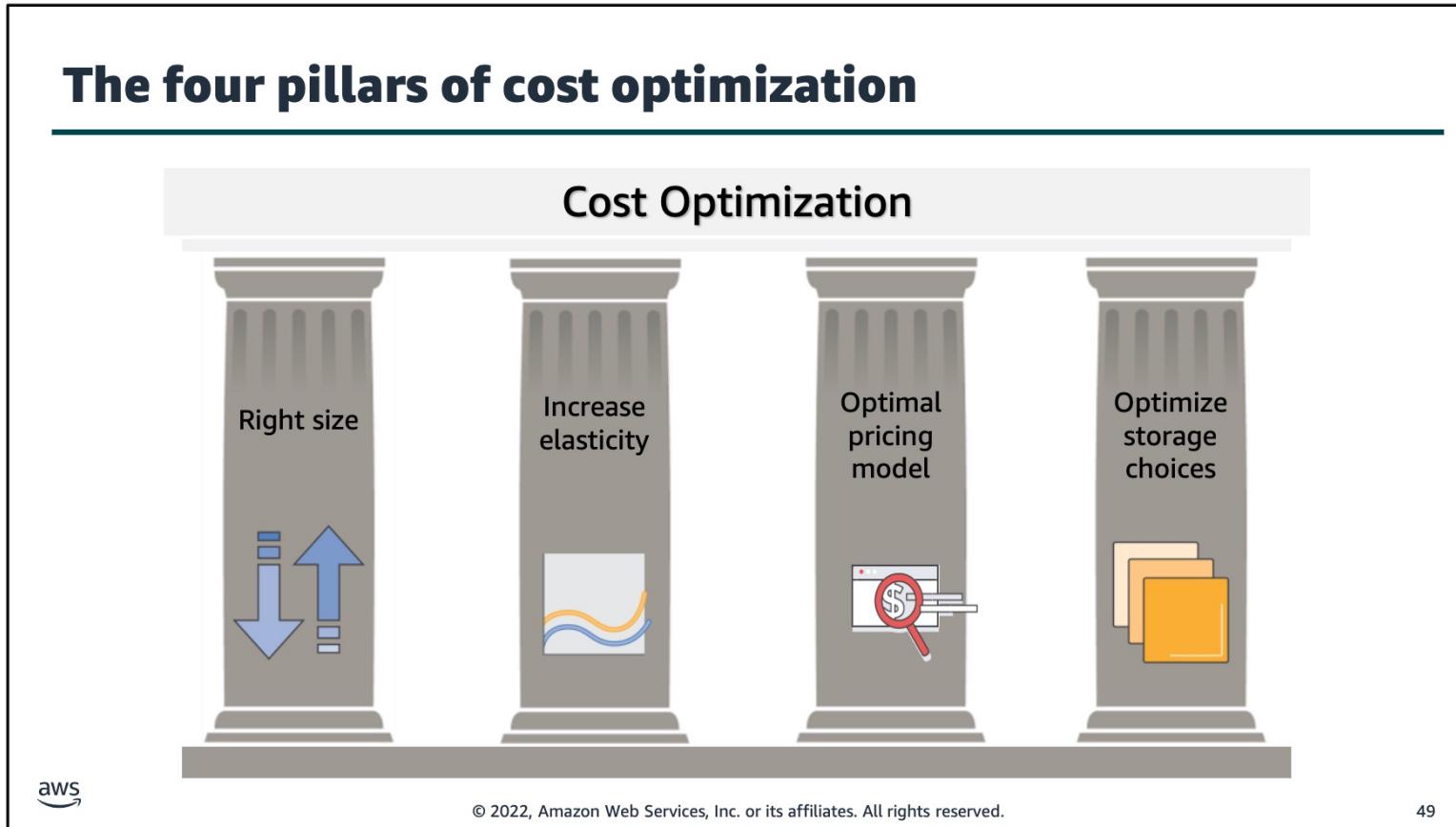


**Per second billing** available for On-Demand Instances, Reserved Instances, and Spot Instances that run Amazon Linux or Ubuntu.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

46

# EC2 Cost Optimization



Want to learn more?

Check out:

[AWS Academy ACF Module 6 – Compute](#)

[AWS Documentation](#)