

Case study on Amazon EC2

Abhinav Kame
(22CO308)

What is Amazon Elastic Compute Cloud (EC2)?

Amazon Elastic Compute Cloud (EC2) is a web-based service that enables businesses to run application programs in the Amazon Web Services (AWS) public cloud. With Amazon EC2, companies can access scalable and secure computing capacity on demand to run many kinds of workloads in AWS. Amazon EC2 is a cloud-based platform that provides on-demand, scalable computing capacity for many kinds of enterprise applications and workloads. Through Amazon EC2, teams can access reliable, scalable infrastructure and secure computing capabilities in the AWS cloud to develop applications without having to invest in expensive on-premises hardware. It's easy to add or reduce capacity based on changing business requirements and control AWS costs based on capacity used. The AWS cloud consists of more than 200 individual services, and as shown here, AWS is one of the largest cloud providers. The Amazon EC2 platform includes more than 750 instances (virtual servers) of different types to support different types of enterprise workloads. Users can also choose from the latest processor, storage, networking, operating system and purchase model to meet their requirements. These components are packaged into pre-configured templates for every Amazon EC2 instance.

Key use cases for Amazon EC2

Amazon EC2 is a good computing platform for running cloud-native and enterprise applications. It can help organizations looking to launch virtual servers, configure security and networking, and manage storage in the cloud. Its ability to provide on-demand, scalable infrastructure and capacity, plus a wide range of instance types, also makes it highly suitable for all these use cases:

- Web servers.
- Code repositories.
- Batch processing.
- Media transcoding.
- High-performance web servers.

- High-performance computing (HPC).
- Scientific modeling.
- Dedicated gaming servers.
- Processing of large memory data sets.
- Ad server engines.
- Input/output (I/O)-intensive applications.
- Computational fluid dynamics.
- Weather forecasting.
- Molecular dynamics.

Amazon EC2 is also optimized for machine learning inference, deep learning and generative artificial intelligence applications. Users can also use Amazon EC2 to build on-demand macOS workloads for Apple devices and platforms.

Amazon EC2 instance types

An Amazon EC2 instance is a virtual server offering an appropriate mix of resources used to run applications on the AWS infrastructure without having to purchase any hardware. These instances, which differ in terms of the central processing unit (CPU), memory, storage and networking capacity offered, include the following:

- **General purpose.** These instances can be used for many different workloads, especially workloads that depend upon a balance of compute, memory and networking resources.
- **Compute-optimized.** These instances are meant for compute-bound or compute intensive applications, such as batch processing, media transcoding and HPC.
- **Memory-optimized.** This set of instances delivers fast performance for workloads that process large memory data sets.
- **Accelerated computing.** These instances use hardware accelerators to perform certain functions more efficiently compared to CPU software.
- **Storage-optimized.** These instances deliver a high amount of low-latency, random I/O operations per second (IOPS) for workloads that require high sequential read/write access to large local data sets.

- **HPC-optimized.** These instances offer the best price performance for running HPC workloads at scale on AWS, such as complex simulations and deep learning workloads

Regardless of which instance users select, Amazon EC2 enables them to scale their resources to match the requirements of their target workloads. It's also possible to choose between Fixed Performance instance families and Burstable Performance instance families, with the latter providing a baseline level of CPU performance, plus the ability to burst above the baseline. Amazon EC2's Elastic Block Store-optimized instances deliver dedicated throughput between Amazon EC2 and Amazon EBS, which minimizes contention between Amazon EBS I/O and other traffic from the EC2 instance and provides optimal performance for EBS volumes. For an additional fee, some Amazon EC2 instance types can run as an EBS-optimized instance.

How Amazon EC2 works

To begin using Amazon EC2, users sign up for a root account on AWS. They can then use AWS Management Console to manage Amazon EC2.

Next, they identify the instance type that's most suitable for their workload. They then create a virtual machine (VM) by selecting Launch Instance in the Amazon EC2 dashboard. Once the instance is launched, it's easy to connect to it. The easiest way to connect to the instance, if it has a public IPv4 address, is with EC2 Instance Connect, a browser-based client, via this simple four-step process:

1. In the Amazon EC2 console navigation pane, choose **Instances**.
2. Select the EC2 instance -- where the connection is required -- and choose **Connect**.
3. Select **EC2 Instance Connect**.
4. Verify the username, and choose **Connect**.

If a new terminal window opens, it means the user is connected to that instance.

It's also possible to connect to an Amazon EC2 instance via the following:

- Use a Secure Socket Shell key and then connect to the instance from a preferred SSH client while using the EC2 Instance Connect application programming interface.
- Configure AWS Command Line Interface, and use the instance ID and an SSH client to connect to the instance via Amazon EC2 Instance Connect.

Important features of Amazon EC2

Some of the important features of Amazon EC2 are the following:

- Multiple storage options. Users can choose between multiple storage options, including block-level storage, instance storage and object storage for their specific requirements. Additionally, Amazon EBS provides persistent storage volumes and three volume types -- General Purpose, Provisioned IOPS and Magnetic -- depending on the workload type and need.
- **Enhanced Networking.** This provides high packet-per-second performance, low network jitter and low latency compared to traditional implementations.
- **Access to Intel's features.** Amazon EC2 instances that feature an Intel processor lets users access many processor features, including Intel Advanced Encryption Standard New Instructions, Intel Advanced Vector Extensions, Intel Deep Learning Boost and Intel Turbo Boost Technology.
- **Support for cluster networking.** Some Amazon EC2 instances support cluster networking when launched into a common cluster placement group for low-latency networking between the instances in the cluster.

- **Auto Scaling.** Users can automatically scale Amazon EC2 capacity up or down by adding or removing instances according to their own conditions, using EC2's dynamic and predictive scaling policies.
- **Optimized CPU configurations.** With this feature, users can better control their Amazon EC2 instances by specifying a custom number of virtual CPUs when launching new instances. They can also disable Intel Hyper-Threading Technology for workloads that perform well enough with single-threaded CPUs

EC2 Pricing Model

Like all other AWS services, Amazon EC2 is available on a pay-as-you-use basis. Users are billed for per-second use and pay only for what they use.

The cost of instances depends on their type: On-Demand, Reserved or Spot.

On-Demand Instances enable organizations to access computing capacity as needed and pay for it by the hour or second. There are no long-term commitments.

Reserved Instances (RIs) provide a price discount of up to 72% (discounted hourly rate) in exchange for one- or three-year contract commitments. Organizations can use these instances to reserve capacity in a specific AZ. Three types of RIs are available: Standard, Convertible and Scheduled. Standard and Convertible RIs are suitable for steady-state usage, and Scheduled RIs are suitable for matching capacity reservation to a predictable recurring schedule.

Amazon EC2 Spot Instances are available at up to a 90% discount compared to On-Demand Instances. Spot Instances can be combined with RIs and On-Demand Instances to optimize cost and performance. On their own, Spot Instances are suitable for running several workload types: containerized, big data, machine learning, HPC and continuous integration/continuous delivery pipelines.

Apart from these instance types, Amazon EC2 customers can also adopt a flexible EC2 Instance Savings Plan to save up to 72% compared to On-Demand Instances. The plan

automatically applies to eligible AWS usage but requires a one- or three-year hourly spend commitment.

On-Demand Instance Prices

Linux	RHEL	SLES	Windows	Windows with SQL Standard	Windows with SQL Web
Windows with SQL Enterprise					
Region: US East (N. Virginia) *					
	vCPU	ECU	Memory (GiB)	Instance Storage (GB)	Linux/UNIX Usage
General Purpose - Current Generation					
t2.nano	1	Variable	0.5	EBS Only	\$0.0065 per Hour
t2.micro	1	Variable	1	EBS Only	\$0.013 per Hour
t2.small	1	Variable	2	EBS Only	\$0.026 per Hour
t2.medium	2	Variable	4	EBS Only	\$0.052 per Hour
t2.large	2	Variable	8	EBS Only	\$0.104 per Hour
m4.large	2	6.5	8	EBS Only	\$0.12 per Hour
m4.xlarge	4	13	16	EBS Only	\$0.239 per Hour
m4.2xlarge	8	26	32	EBS Only	\$0.479 per Hour
m4.4xlarge	16	53.5	64	EBS Only	\$0.958 per Hour
m4.10xlarge	40	124.5	160	EBS Only	\$2.394 per Hour

Reserved Instance Example

t2.medium

1-YEAR TERM					
Payment Option	Upfront	Monthly*	Effective Hourly**	Savings over On-Demand	On-Demand Hourly
No Upfront	\$0	\$26.28	\$0.036	31%	\$0.052 per Hour
Partial Upfront	\$204	\$8.76	\$0.0353	32%	
All Upfront	\$302	\$0	\$0.0345	34%	
3-YEAR TERM					
Payment Option	Upfront	Monthly*	Effective Hourly**	Savings over On-Demand	On-Demand Hourly
Partial Upfront	\$436	\$5.84	\$0.0246	53%	\$0.052 per Hour
All Upfront	\$607	\$0	\$0.0231	56%	

Virtual private clouds (VPCs)

Virtual networks you can create that are logically isolated from the rest of the AWS Cloud. You can optionally connect these virtual networks to your own network.

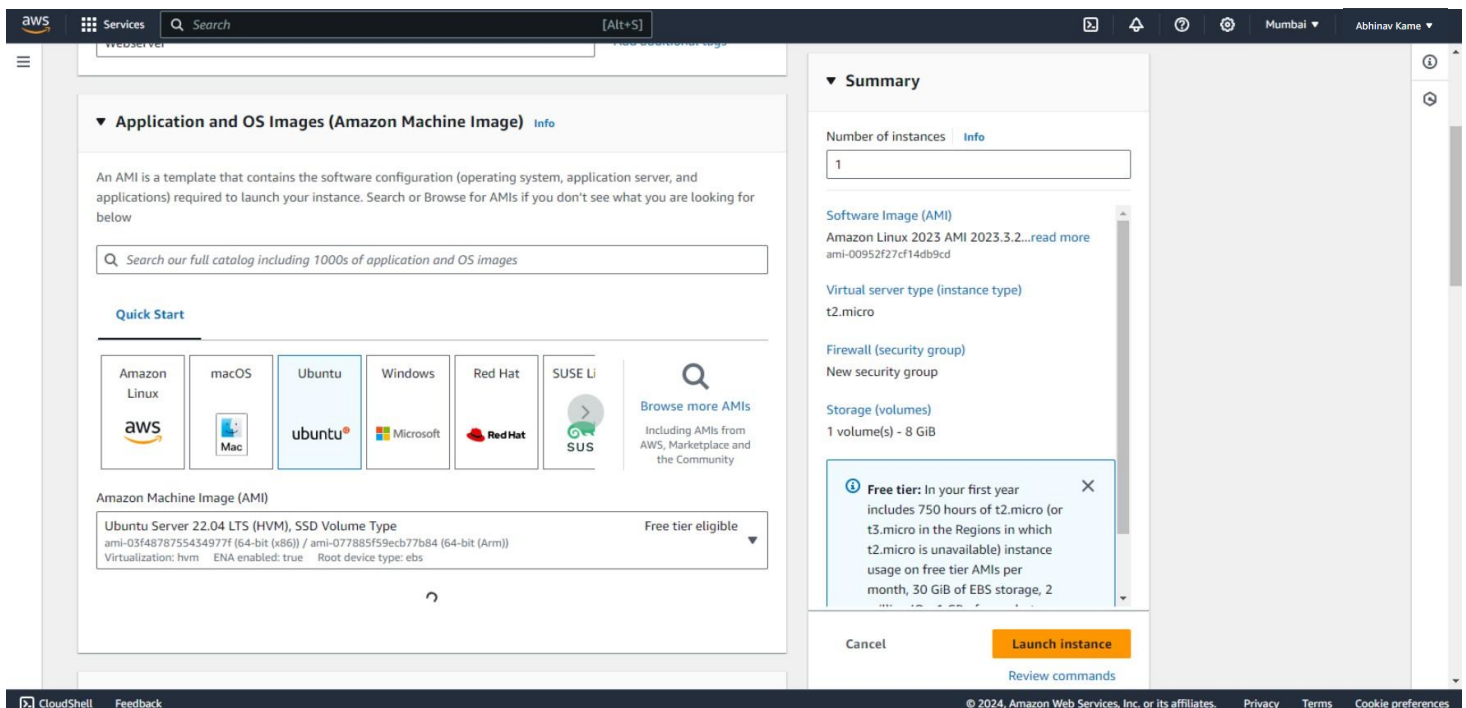
The screenshot shows the AWS Management Console interface for launching an EC2 instance. The main section is titled "Launch an instance" and includes a brief description of Amazon EC2. Below this, there are several configuration sections:

- Name and tags:** A text input field labeled "Name" with the value "Webserver" entered. There is a link to "Add additional tags".
- Application and OS Images (Amazon Machine Image):** A section with a search bar and a "Quick Start" button. Below the search bar, there are buttons for "Amazon Linux", "macOS", "Ubuntu", "Windows", "Red Hat", and "SUSE LI". A "Browse more AMIs" link is also present.
- Summary:** A sidebar on the right containing a "Number of instances" input field set to "1". It lists the following configuration:
 - Software Image (AMI): Amazon Linux 2023.3.2...read more
 - Virtual server type (instance type): t2.micro
 - Firewall (security group): New security group
 - Storage (volumes): 1 volume(s) - 8 GiB

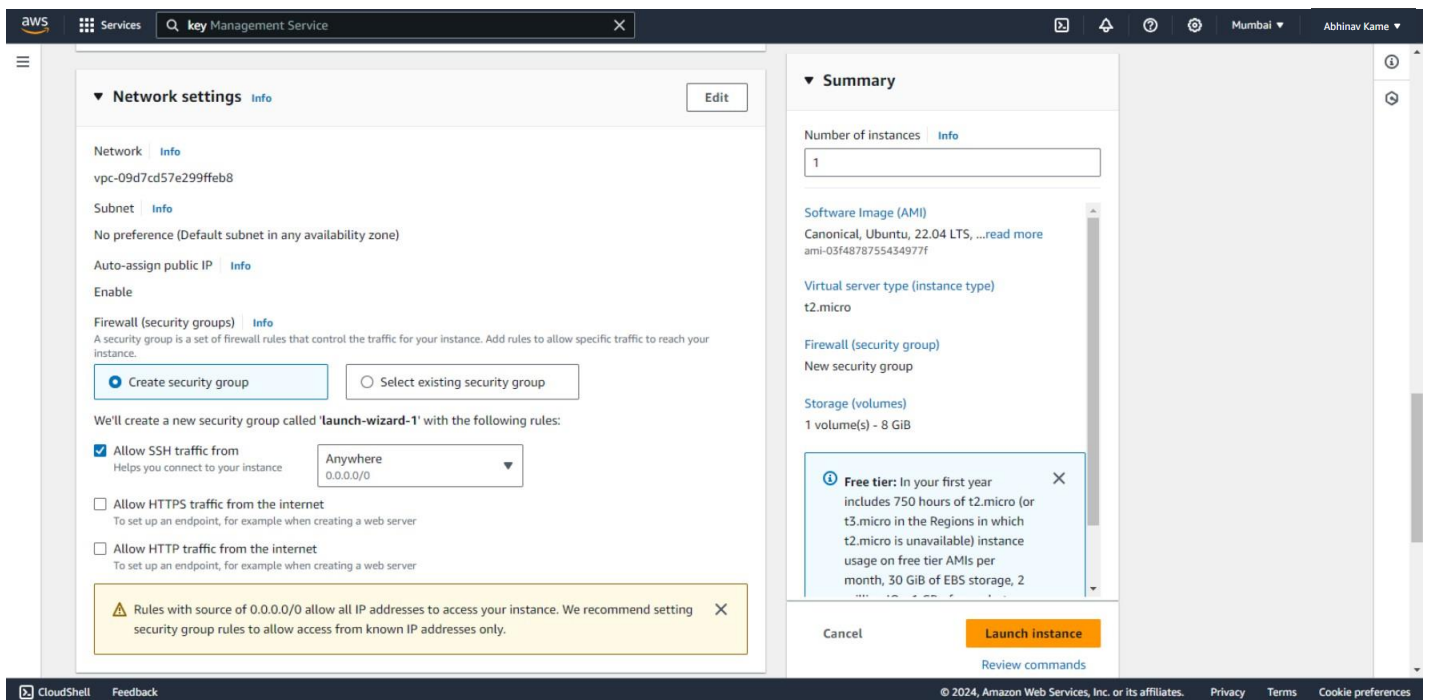
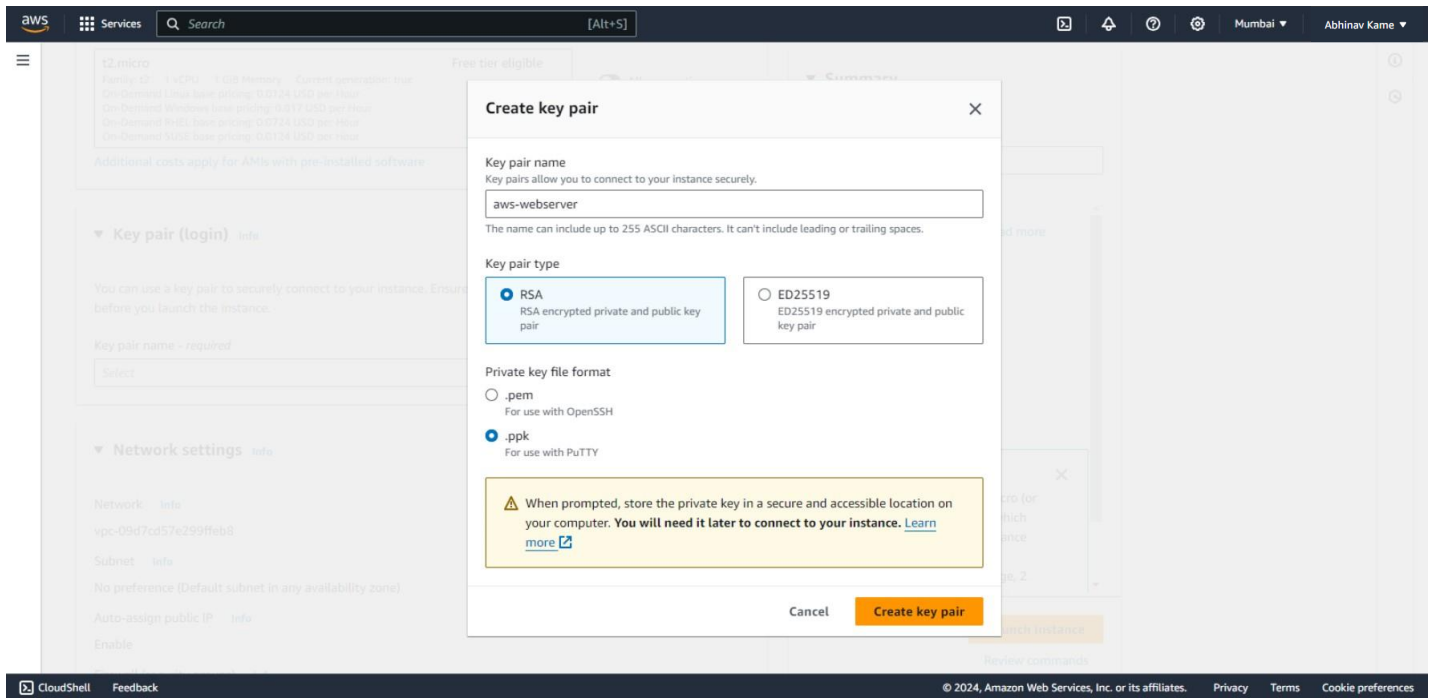
At the bottom of the Summary sidebar, there is a "Free tier" notification: "Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2...". Below this, there are "Cancel" and "Launch instance" buttons, along with a "Review commands" link.

To launch an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the EC2 console dashboard, in the Launch instance box, choose Launch instance.
3. Under Name and tags, for Name, enter a descriptive name for your instance.
4. Under Application and OS Images (Amazon Machine Image), do the following:
 - a. Choose Quick Start, and then choose Amazon Linux. This is the operating system (OS) for your instance.
 - b. From Amazon Machine Image (AMI), select an HVM version of Amazon Linux 2. Notice that these AMIs are marked Free Tier eligible. An Amazon Machine Image (AMI) is a basic configuration that serves as a template for your instance.

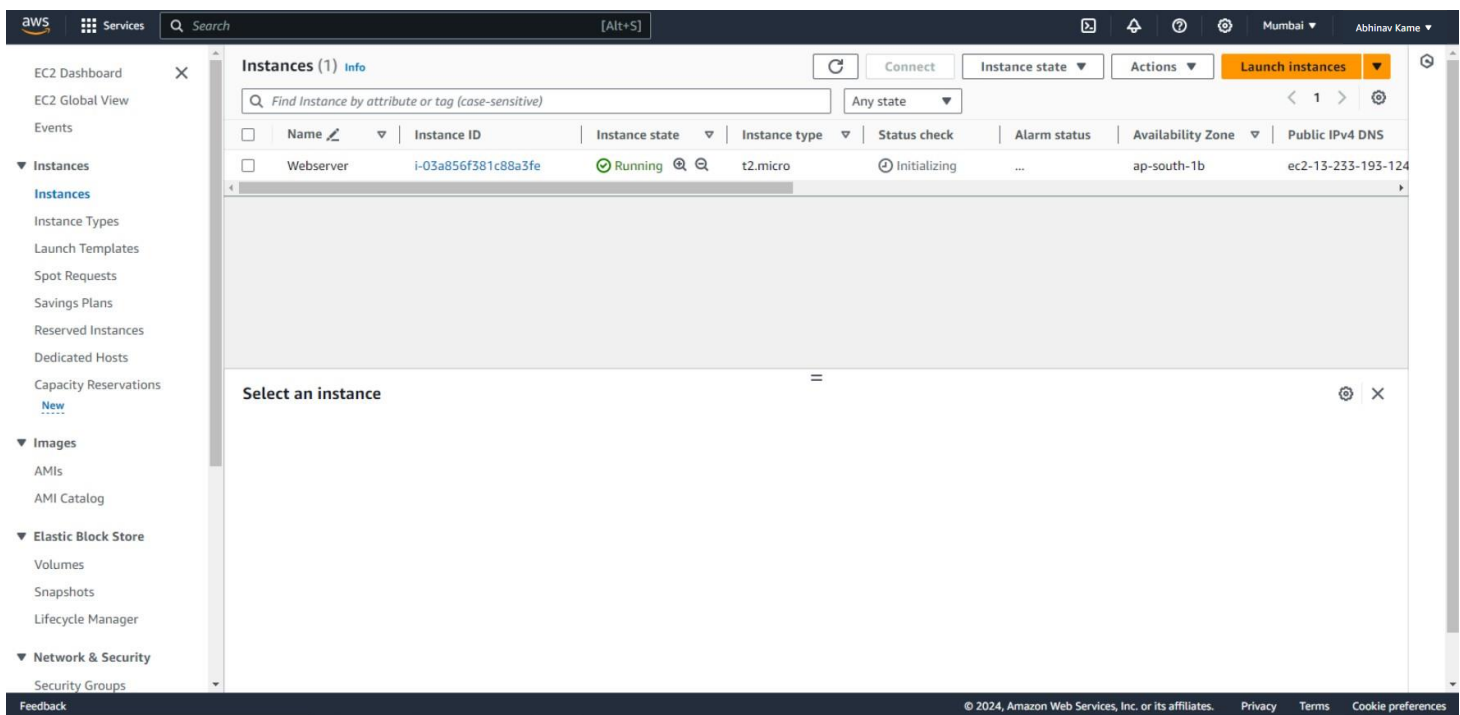


5. Under Instance type, from the Instance type list, you can select the hardware configuration for your instance. Choose the t2.micro instance type, which is selected by default. The t2.micro instance type is eligible for the Free Tier. In Regions where t2.micro is unavailable, you can use a t3.micro instance under the Free Tier.
6. Under Key pair (login), for Key pair name, choose the key pair that you created when getting set up.



1. Next to Network settings, choose Edit. For Security group name, you'll see that the wizard created and selected a security group for you. You can use this security group, or alternatively you can select the security group that you created when getting set up using the following steps:

- a. Choose Select existing security group.
 - b. From Common security groups, choose your security group from the list of existing security groups.
2. Keep the default selections for the other configuration settings for your instance.
 3. Review a summary of your instance configuration in the Summary panel, and when you're ready, choose Launch instance.



1. A confirmation page lets you know that your instance is launching. Choose View all instances to close the confirmation page and return to the console.
2. On the Instances screen, you can view the status of the launch. It takes a short time for an instance to launch. When you launch an instance, its initial state is pending. After the instance starts, its state changes to running and it receives a public DNS name. If the

Public IPv4 DNS column is hidden, choose the settings icon () in the top-right corner, toggle on Public IPv4 DNS, and choose Confirm.

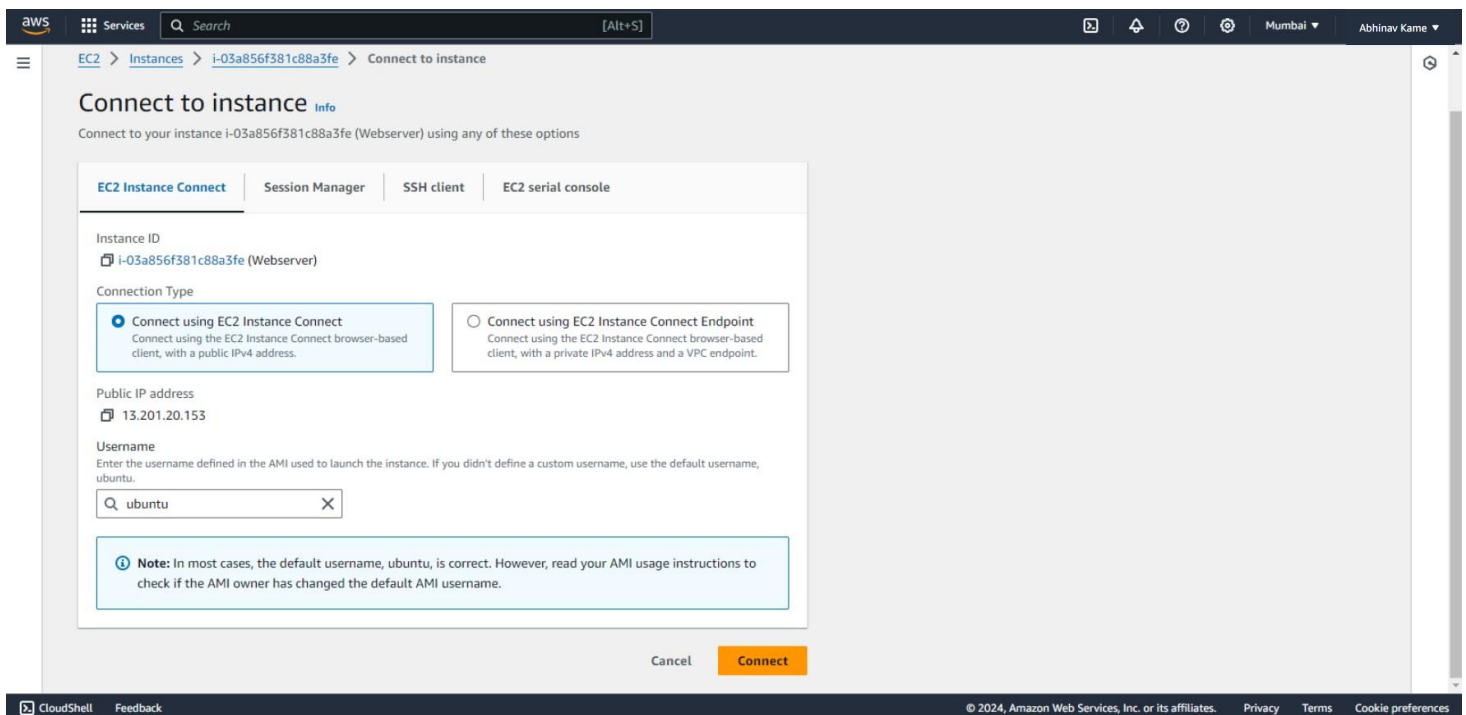
3. It can take a few minutes for the instance to be ready for you to connect to it. Check that your instance has passed its status checks; you can view this information in the Status check column.

Creating a webserver

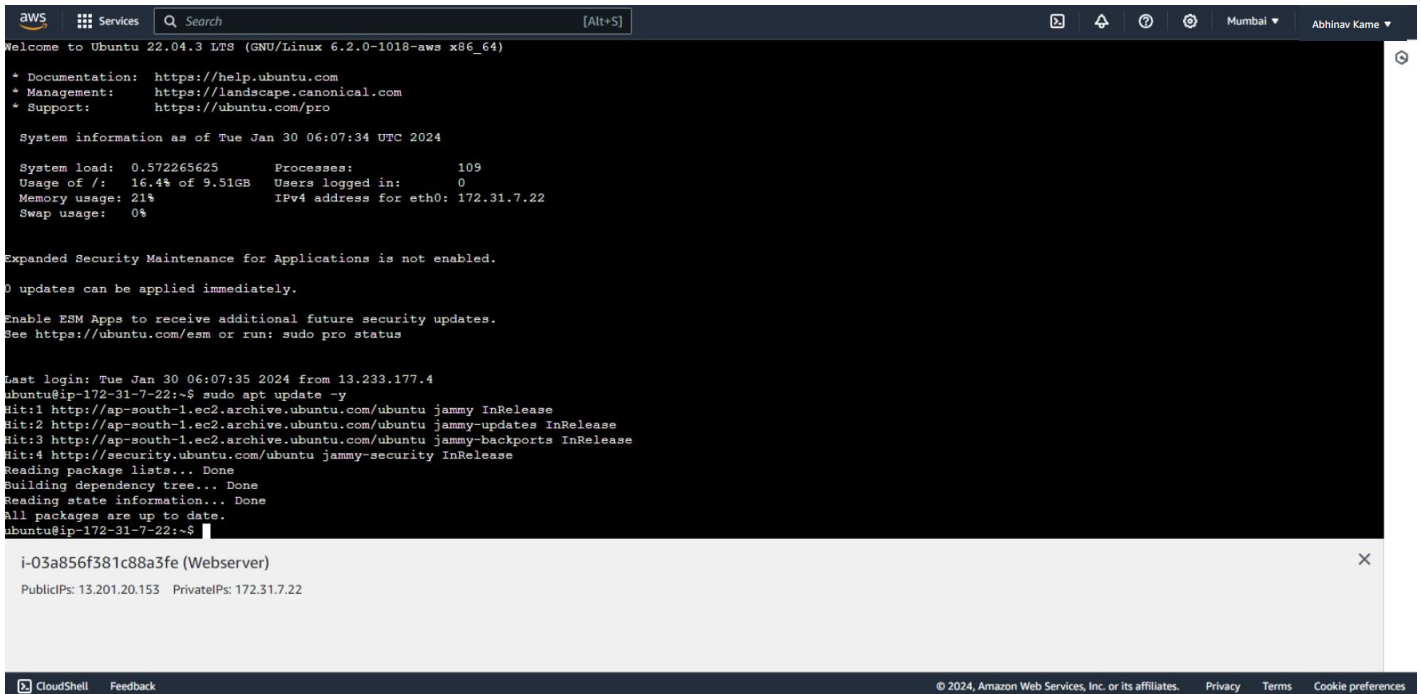
Connect to your EC2 instance and install the web server.

To connect to your EC2 instance and install the Apache web server

1. Connect to the EC2 instance that you created earlier



2. Get the latest bug fixes and security updates by updating the software on your EC2 instance. To do this, use the following command: `sudo apt update -y`



The screenshot shows the AWS CloudShell interface with a terminal window. The terminal output includes the Ubuntu version (22.04.3 LTS), system information (load, processes, memory usage, etc.), and the execution of the command `sudo apt update -y`. The output of the update command shows that several packages are being updated, including `ubuntu-jammy`, `ubuntu-jammy-updates`, `ubuntu-jammy-backports`, and `ubuntu-jammy-security`. The terminal also displays the IP addresses for the instance (Public IP: 13.201.20.153, Private IP: 172.31.7.22).

```
aws Services Search [Alt+S]
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1018-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Tue Jan 30 06:07:34 UTC 2024

System load:  0.572265625   Processes:            109
Usage of /:   16.4% of 9.51GB Users logged in:          0
Memory usage: 21%          IPv4 address for eth0: 172.31.7.22
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

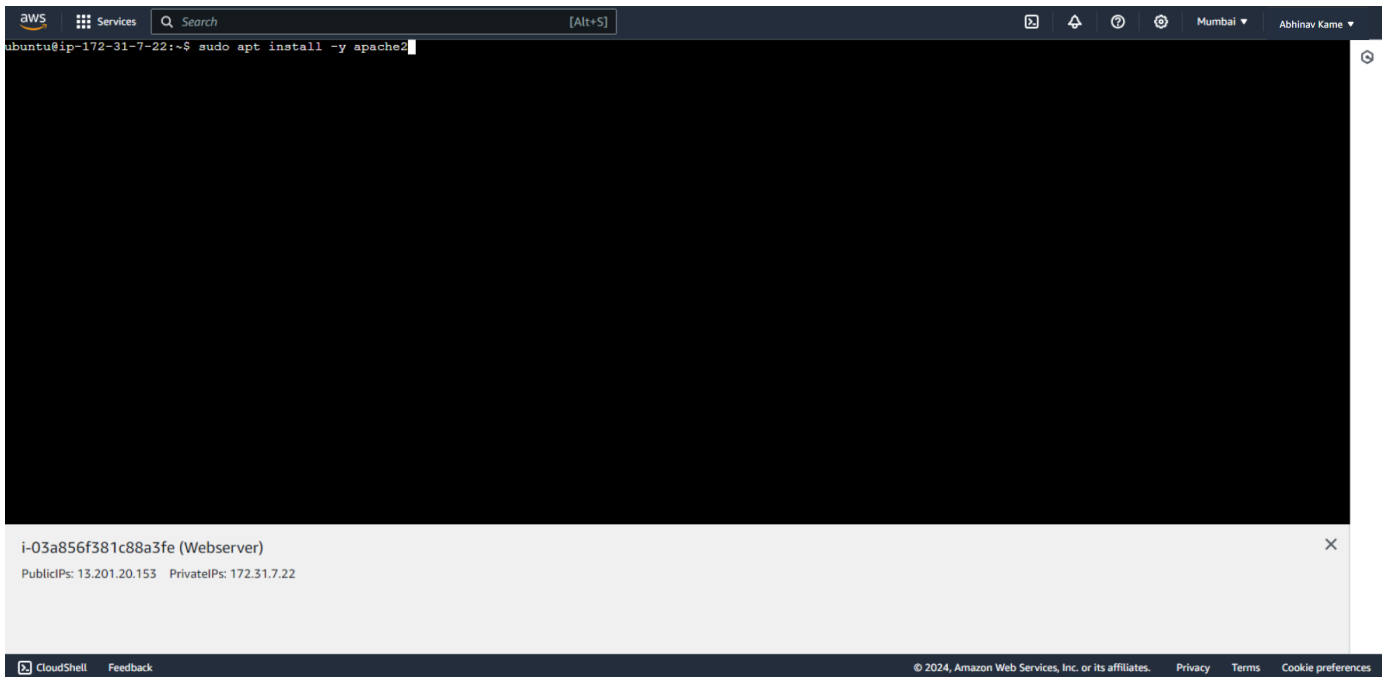
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Tue Jan 30 06:07:35 2024 from 13.233.177.4
ubuntu@ip-172-31-7-22:~$ sudo apt update -y
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
ubuntu@ip-172-31-7-22:~$

i-03a856f381c88a3fe (Webserver)
PublicIPs: 13.201.20.153 PrivateIPs: 172.31.7.22

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

3. After the updates complete, install the Apache web server, PHP, and MariaDB or PostgreSQL software using the following commands. This command installs multiple software packages and related dependencies at the same time: `sudo apt install -y apache2`



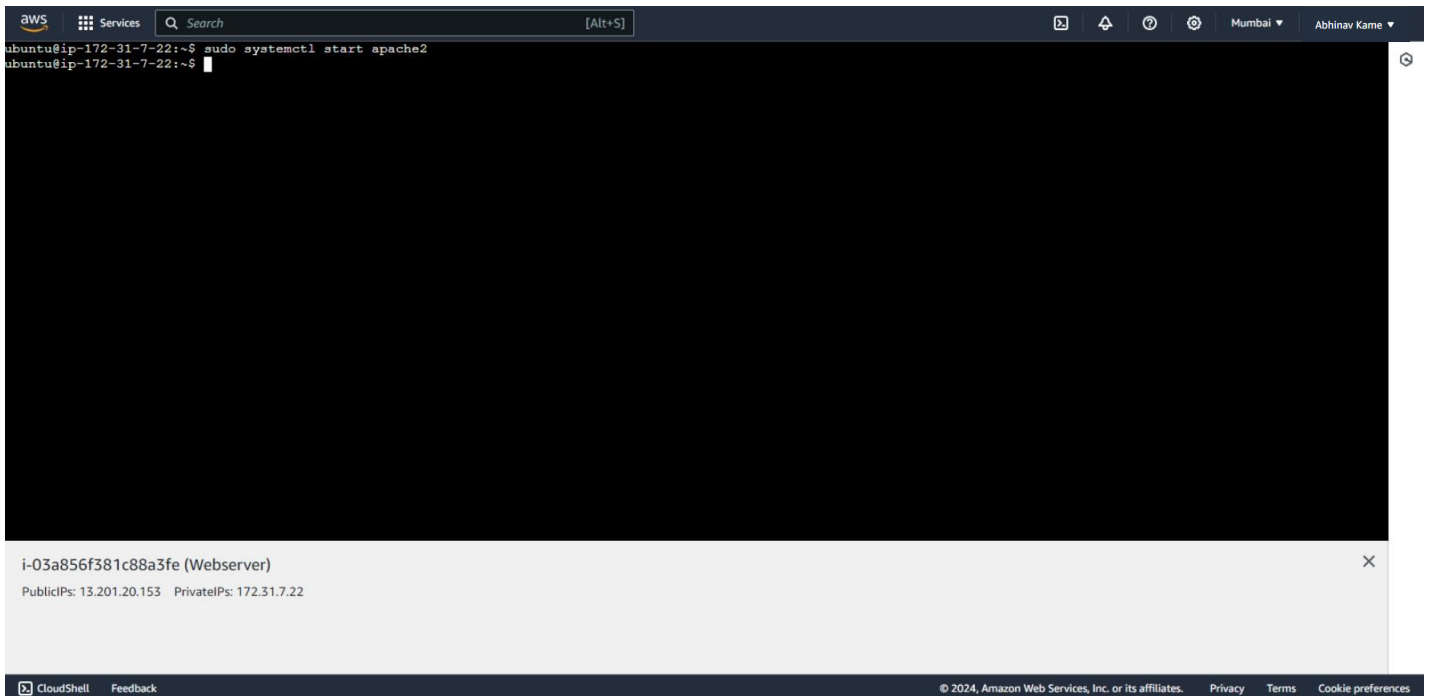
The screenshot shows the AWS CloudShell interface with a terminal window. The terminal output shows the command `sudo apt install -y apache2` being entered. The terminal also displays the IP addresses for the instance (Public IP: 13.201.20.153, Private IP: 172.31.7.22).

```
aws Services Search [Alt+S]
ubuntu@ip-172-31-7-22:~$ sudo apt install -y apache2

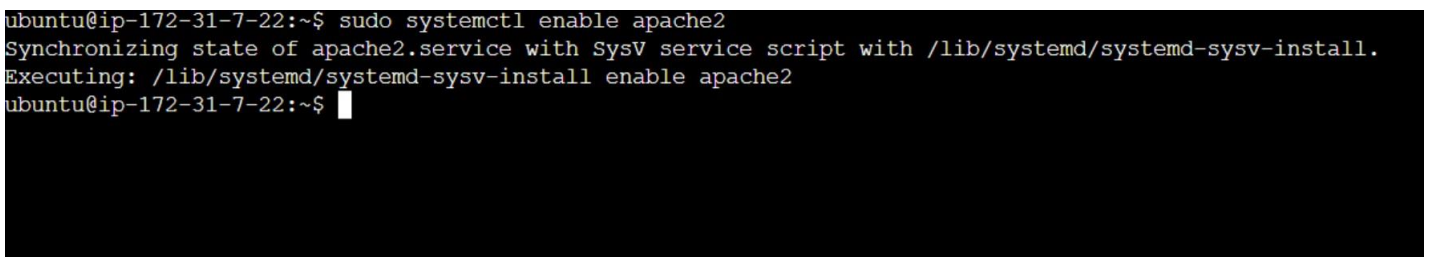
i-03a856f381c88a3fe (Webserver)
PublicIPs: 13.201.20.153 PrivateIPs: 172.31.7.22

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

4. Start the web server with the command: `sudo systemctl start apache2`



5. Configure the web server to start with each system boot using the systemctl command:
`sudo systemctl enable apache2`



6. Enable the apache2 so that it starts everytime our instance boots up with command:
`sudo systemctl enable apache2`

7. Check if apache2 is running: `sudo systemctl status apache2`



8. Enable HTTP and HTTPS traffic in the inbound rules from the security group

Inbound rules [Info](#)

Security group rule ID	Type	Protocol	Port range	Source	Description - optional	
sgr-00696350d6e28913f	SSH	TCP	22	Custom	0.0.0.0	Delete
-	HTTP	TCP	80	Anywhere-I...	0.0.0.0	Delete
-	HTTPS	TCP	443	Anywhere-I...	0.0.0.0	Delete

[Add rule](#)

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Preview changes](#) [Save rules](#)

9. Go to the public ip address of your instance to view the hosted apache2 index page to see if the server is working correctly.



Amazon EC2 benefits

It's easy to get started with Amazon EC2 and access on-demand computing capacity as required. After the instance has been launched, users can connect to it. They can use EC2 Instance Connect, for example, and employ it as easily as a computer that's sitting in front of them. Amazon EC2 offers multiple instance types and offers a wide range of computing, memory and networking resources, making it a suitable solution for many kinds of enterprise workloads. The instances are secured using numerous controls to support all applications. One such control is a security group -- a virtual firewall that controls incoming and outgoing traffic. In addition, both private and public keys are used to authorize users and prevent unauthorized access. Amazon EC2 supports dynamic cloud computing with elastic IP addresses. These addresses are static IPv4 addresses, with each address associated with the user's account rather than a particular instance. Also, users can control the address until they explicitly release it or choose not to do so. Users can create Virtual Private Clouds to launch Amazon EC2 resources in a virtual network. This network is user-defined and logically isolated from the rest of the AWS cloud. It uses AWS' scalable infrastructure and can be connected to the user's own network. Another benefit of Amazon EC2 is that it offers a free tier. With this tier, users can

familiarize themselves with some of its features and pay only when they are ready for the full offering. The free version includes 750 hours of Linux and Windows t2.micro instances per month for a year.

Challenges with Amazon EC2

- **Resource utilization.** Developers must manage the number of instances to avoid costly large, long-running instances.
- **Security.** Developers must ensure that public-facing instances are running securely.
- **Deploying at scale.** Running a multitude of instances can result in cluttered environments that are difficult to manage.
- **Management of Amazon Machine Image (AMI) lifecycle.** Developers often begin by using default AMIs. As computing needs change, custom configurations will likely be required.
- **Ongoing maintenance.** Amazon EC2 instances are VMs that run in Amazon's cloud. However, they ultimately run on physical hardware, which can fail. AWS alerts developers when an instance must be moved due to hardware maintenance. This requires ongoing monitoring.

Surprisingly, you can't scale up that large.

Q: How many instances can I run in Amazon EC2?

You are limited to running up to 20 On-Demand instances, purchasing 20 Reserved Instances, and requesting Spot Instances per your [dynamic Spot limit](#) per region. New AWS accounts may start with limits that are lower than the limits described here. Certain instance types are further limited per region as follows:

Instance Type	On-Demand Limit	Reserved Limit	Spot Limit
m4.4xlarge	10	20	Dynamic Spot Limit
m4.10xlarge	5	20	Dynamic Spot Limit
c4.4xlarge	10	20	Dynamic Spot Limit
c4.8xlarge	5	20	Dynamic Spot Limit
cg1.4xlarge	2	20	Dynamic Spot Limit
hi1.4xlarge	2	20	Dynamic Spot Limit
hs1.8xlarge	2	20	Not offered
cr1.8xlarge	2	20	Dynamic Spot Limit

Amazon EC2 history

Amazon EC2 was the idea of engineer Chris Pinkham, who conceived it as a way to scale Amazon's internal infrastructure and build an "infrastructure service for the world." Pinkham and fellow engineer Benjamin Black presented their ideas to Amazon CEO Jeff Bezos, who then requested more details about virtual servers in the cloud. Amazon EC2 was then developed by a team in Cape Town, South Africa. Pinkham provided the initial architecture guidance, gathered a development team and worked on the project along with Black and Willem van Biljon. In 2006, Amazon announced a limited public beta test of Amazon EC2, and in 2007, the company added two new instance types: Large and Extra-Large. Amazon announced the addition of static IP addresses, AZs and user-selectable kernels in spring 2008, followed by the release of EBS in August 2008. Amazon EC2 went into full production on Oct. 23, 2008. Amazon also released a servicelevel agreement for Amazon EC2 that day, along with Microsoft Windows and SQL Server in beta form on Amazon EC2. Amazon added AWS Management Console, load balancing, autoscaling and cloud monitoring services in 2009. Today, Amazon EC2 is among the most popular of AWS' products and remains one of the foundational elements of Amazon's cloud computing service.

Literature Review: Amazon EC2

Amazon Elastic Compute Cloud (EC2) is a cornerstone service within Amazon Web Services (AWS), offering on-demand virtual servers. Here's a review of key themes explored in research papers regarding EC2:

Scalability and Elasticity:

- A study by [1] examines how companies like Blackboard leverage EC2's on-demand instances for rapid scaling of compute resources. This pay-as-you-go model allows businesses to handle fluctuating workloads efficiently and cost-effectively.
- Research by [2] explores how auto-scaling groups within EC2 enable automatic scaling of resources based on predefined rules. This ensures

optimal resource utilization and avoids overprovisioning, leading to cost savings.

Cost Optimization:

- A paper by [3] investigates how Spot Instances, a feature within EC2 that offers unused EC2 capacity at a discount, can significantly reduce cloud computing costs. However, it highlights the need for careful planning due to the interruptible nature of Spot Instances.
- Research by [4] delves into security considerations when using public Amazon Machine Images (AMIs) within EC2. It emphasizes the importance of security audits to identify potential vulnerabilities within pre-configured AMIs to optimize cost-effectiveness without compromising security.

Security:

- The aforementioned study by [4] highlights security concerns associated with utilizing public AMIs in EC2 environments.
- Research by [5] explores various security measures within EC2, including security groups, IAM roles, and VPCs. It emphasizes the importance of implementing a layered security approach to protect data and resources within EC2 instances.

Performance and Benchmarks:

- A study by [6] benchmarks the performance of different instance types within EC2 for specific workloads. This helps users select the most appropriate instance type based on their application requirements and optimize performance.
- Research by [7] explores containerization technologies like Docker in conjunction with EC2 for improved resource utilization and application deployment efficiency. Containerized applications can be packaged and deployed across different EC2 instances seamlessly.

Integration with other AWS Services:

- The case study on Blackboard mentioned earlier [1] showcases how EC2 integrates with other AWS services like AWS Batch for managing large-scale batch computing jobs. This highlights the comprehensive nature of the AWS cloud platform.
- Research by [8] explores how serverless architectures can be combined with EC2 for optimal resource utilization. Serverless functions can handle stateless workloads, while EC2 instances can be used for stateful applications or tasks requiring more control.

These are just a few examples of the vast amount of research conducted on Amazon EC2. The platform's scalability, cost-effectiveness, security features, performance capabilities, and integration with other AWS services make it a popular choice for cloud computing deployments.

Additional Considerations for Further Research:

- The environmental impact of cloud computing, including the energy consumption of EC2 instances, is an emerging area of research.
- The evolving landscape of containerization technologies and their impact on EC2 deployments is another interesting area for further exploration.
- Security best practices for managing sensitive data within EC2 instances in a multi-cloud or hybrid cloud environment warrant further investigation.

Conclusion: A Powerful Cloud Engine

In conclusion, Amazon EC2 offers a robust and versatile platform for businesses seeking to build and deploy applications on the cloud. Its scalability, cost-effectiveness, security features, performance capabilities, and integration with other AWS services make it a compelling choice for a wide range of cloud computing needs. Businesses can leverage EC2's strengths to achieve their specific goals, while remaining mindful of considerations like environmental impact and security best practices in multi-cloud or hybrid cloud environments. By understanding both the potential and limitations of EC2, businesses can make informed decisions to leverage this powerful cloud computing engine for their success.