

iamneo



# Amazon EC2



[www.iamneo.ai](http://www.iamneo.ai)

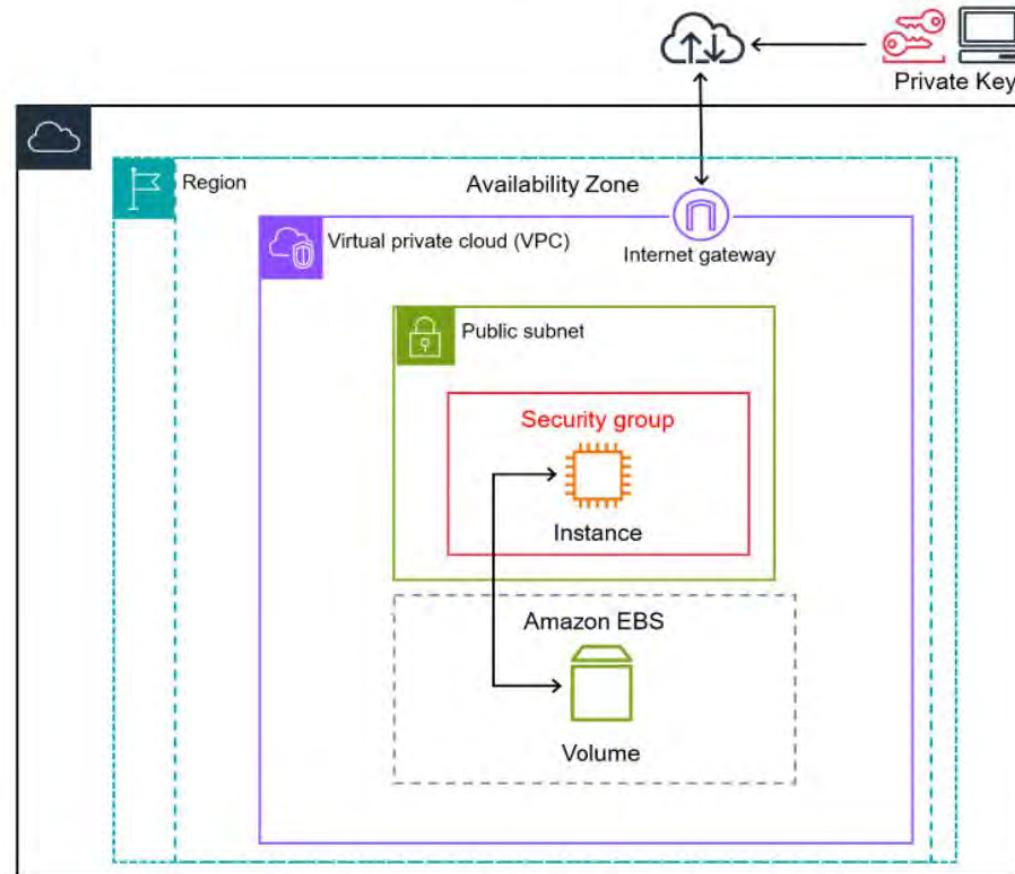
# Introduction to Amazon EC2



Amazon EC2

- Amazon Elastic Compute Cloud (Amazon EC2) is a web service that
- provides secure, resizable compute capacity in the cloud.
- Access reliable, scalable infrastructure on demand. Scale capacity within minutes with SLA commitment of 99.99% availability.  
Provide secure compute for your applications.

# Introduction to Amazon EC2



# Introduction to Amazon EC2

- 1 A computing powerhouse
- 2 Flexible and reliable
- 3 Affordable and cost-effective

# What can you do with Amazon EC2?

```
def db():
    if os.path.isfile(FILE_URL):
        db.create_all()

    if __name__ == "__main__":
        books = db.session.query(Book).all()
        render_template("index.html", books=books)

    @app.route("/edit", methods=["GET", "POST"])
    def edit(book_id):
        book_to_update = Book.query.get(book_id)
        book_to_update.rating = request.form["rating"]
        db.session.commit()
        return redirect(url_for("index"))

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```

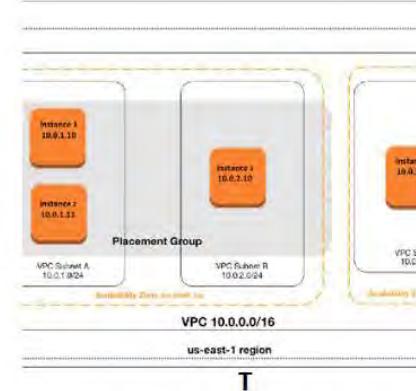
**Application Development**

Use EC2 as an environment to develop, test, and deploy applications, from simple web apps to complex enterprise solutions.



**Big Data**

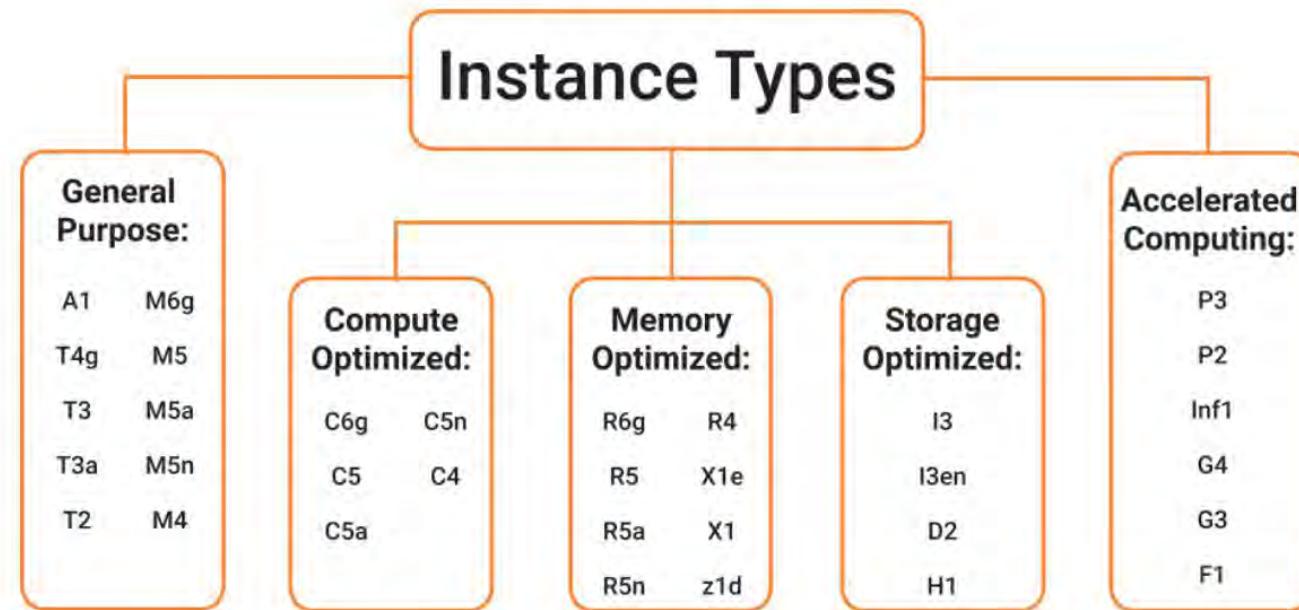
Run big data applications and workloads, including Hadoop and Spark, on EC2's powerful clusters of instances.



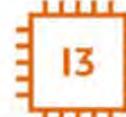
**Team Collaboration**

EC2 can be used for team collaboration and project management. Share instances and allow various group-level permissions to drive collaboration and productivity.

# Instance Types



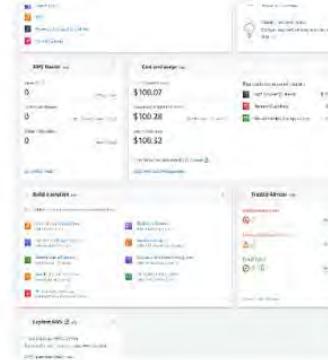
# Instance Types

General Purpose	Compute Optimised	Memory Optimised	Accelerated Computing	Storage Optimised
 ARM based core and custom silicon	 Compute - CPU intensive apps and DBs	 RAM - Memory intensive apps and DB's	 Processing optimised - Machine Learning	 High Disk Throughput - Big data clusters
 Tiny - Web servers and small DBs		 Xtreme RAM - For SAP/Spark	 Graphics Intensive - Video and streaming	 IOPS - NoSQL DBs
 Main - App servers and general purpose		 High Compute and High Memory - Gaming	 Field Programmable - Hardware acceleration	 Dense Storage - Data Warehousing

# Launching and Configuring EC2 Instances



# Introduction to AWS Management Console



## Sign Up or Log In

Create and log in to your AWS Management Console account. Follow the instructions to access the console.

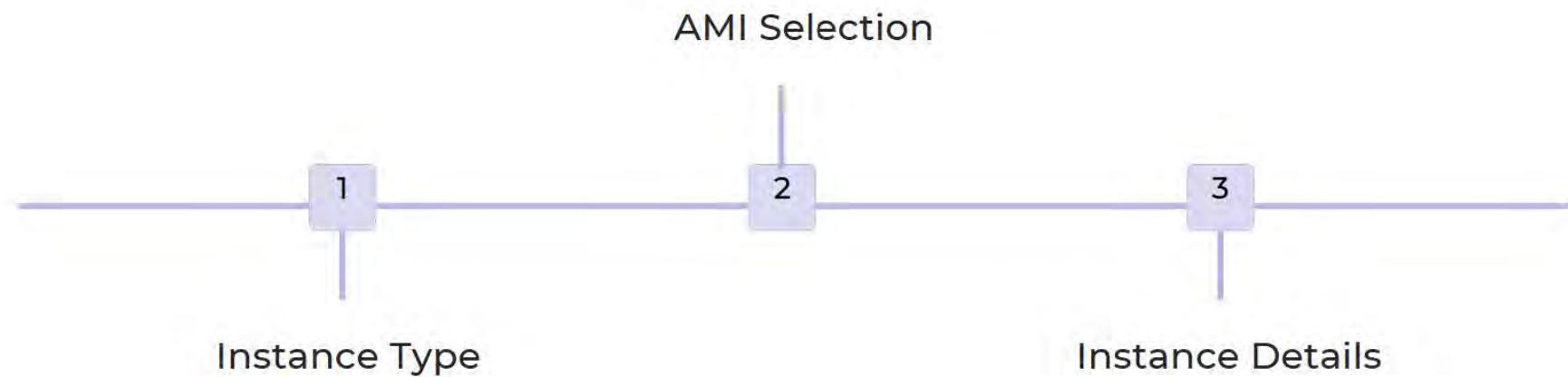
## AWS Dashboard

The dashboard is the home page of your AWS Management Console. Explore the various services AWS offers.

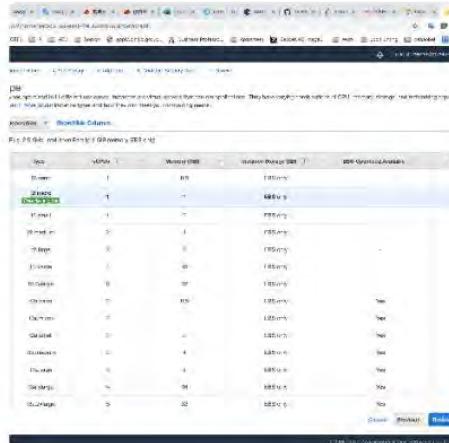
## Configuring Settings

Quickly access, configure, and customize your settings and preferences according to your needs.

# Launching an EC2 Instance



# Configuring Instance Details



## Instance Name

Name your instance and specify the purpose so you can easily identify it later on.

CPU	Architecture	Memory (MB)	Storage (GB)	Storage type	Network performance
I386, 800_64	327	-	-	-	Very Low
I386, 800_64	312	-	-	-	Low to Moderate
I386, 800_64	1924	-	-	-	Low to Moderate
I386, 800_64	2348	-	-	-	Low to Moderate
I386, 800_64	4996	-	-	-	Low to Moderate
x86_64	8192	-	-	-	Low to Moderate
x86_64	14354	-	-	-	Moderate
x86_64	32708	-	-	-	Moderate
x86_64	312	-	-	-	Up to 2 Gbps
x86_64	1924	-	-	-	Up to 5 Gbps
x86_64	5048	-	-	-	Up to 8 Gbps

## Instance Type

Specify the details for the instance type.

# Configuring Instance Details

Port range	Protocol	Source
22	TCP	[REDACTED]/1
443	TCP	[REDACTED]/1
80	TCP	[REDACTED]/1
0 - 65535	TCP	0.0.0.0/0

Port range	Protocol	Destination
All	All	0.0.0.0/0

## Security Groups

Choose the security groups you want your instance to be associated with.



## Key Pairs

Choose the key pair to log in to your instance.

# Choosing Storage Options

## EBS Volume

Elastic Block Store (EBS) lets you store data separately from the instance. It can be easily attached or detached to the instance.

## Instance Store Volume

Instance Store Volumes work similarly to EBS, but the data is tied to the instance's lifecycle.

## Snapshot

Creating a snapshot ensures that you have a backup of your instance. Snapshots can be created on demand or scheduled.

# Setting Up Security Groups

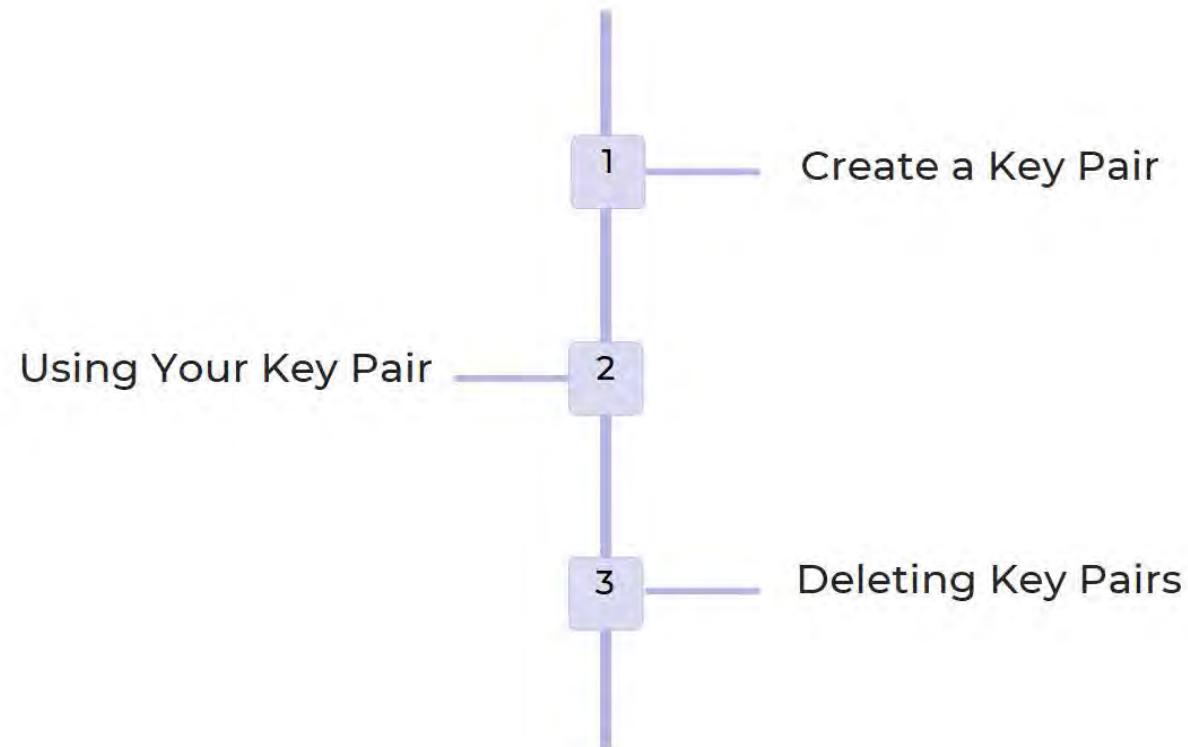
## Inbound Rules

- Control access to the instance
- Specify ports and protocols
- Allow unrestricted access

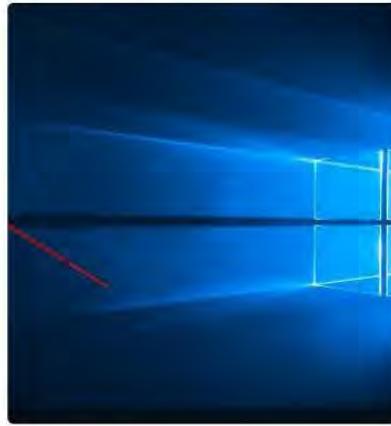
## Outbound Rules

- Control instance access to the internet
- Specify ports and protocols
- Allow unrestricted access

# Creating and Using Key Pairs



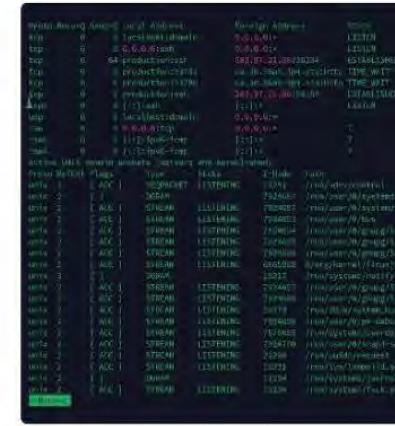
# Connecting to Instances using SSH or RDP



Windows -RDP



Mac -SSH



Linux -SSH

# Understanding EC2 Instance States and Lifecycle

## Running Instances

EC2 instances launched and running in a specific Availability Zone.

## Stopped Instances

EC2 instances that have been stopped and can be re-launched when needed.

## Terminated Instances

EC2 instances that have been terminated, and their data cannot be recovered.

# Managing EC2 Instances using AWS Management Console

Launch an Instance

Reboot and Terminate

Scaling and Load Balancing

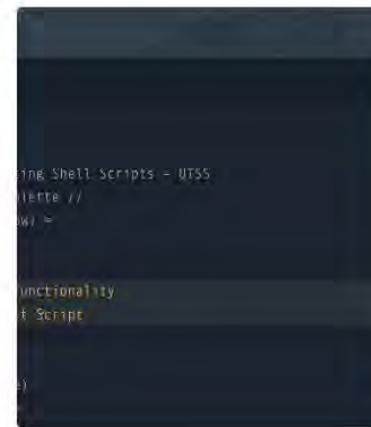
Monitoring and Troubleshooting

# Managing EC2 Instances using CLI



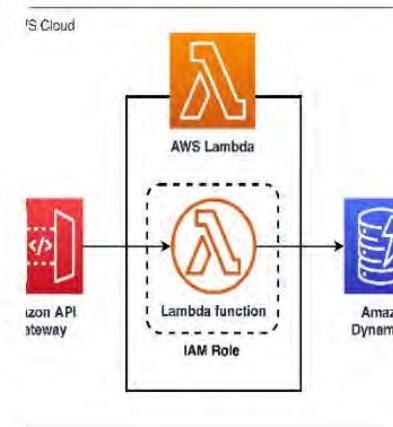
```
aws ec2 describe-instances --region us-east-1
{
    "Reservations": [
        {
            "Instances": [
                {
                    "InstanceId": "i-000000000000000000",
                    "ImageId": "ami-000000000000000000",
                    "InstanceType": "t2.micro",
                    "State": {
                        "Name": "running"
                    },
                    "PublicIpAddress": "54.122.122.122",
                    "PrivateIpAddress": "10.0.3.10",
                    "NetworkInterfaces": [
                        {
                            "MacAddress": "e8:0c:2d:34:56:78",
                            "PrivateDnsName": "ip-10-0-3-10.ec2.internal",
                            "PrivateIpAddress": "10.0.3.10",
                            "SubnetId": "subnet-000000000000000000",
                            "Status": "in-use"
                        }
                    ],
                    "OwnerId": "123456789012",
                    "RootDeviceType": "Amazon EBS",
                    "RootDeviceName": "/dev/sda1",
                    "StateTransitionReason": null,
                    "Tags": [
                        {
                            "Key": "Name",
                            "Value": "MyTestInstance"
                        }
                    ],
                    "Tenancy": "default",
                    "VirtualizationType": "hvm"
                }
            ],
            "Filters": [
                {
                    "Name": "tag:Name",
                    "Values": [
                        "MyTestInstance"
                    ]
                }
            ],
            "MaxResults": 50,
            "NextToken": null
        }
    ],
    "ResponseMetadata": {
        "RequestId": "00000000-0000-0000-0000-000000000000",
        "HTTPStatusCode": 200,
        "HTTPHeaders": {
            "Content-Type": "application/json",
            "Content-Length": "1234",
            "Date": "Tue, 01 Jan 2019 12:00:00 GMT"
        },
        "RetryAttempts": 0
    }
}
```

Command Line Interface



```
#!/bin/bash
# Script to manage EC2 instances
# Usage: ./ec2_manager.sh [option] [args]
# Options:
#   -l: List all EC2 instances
#   -r: Run a command on all instances
#   -s: Stop all instances
#   -u: Start all instances
#   -d: Delete all instances
#   -h: Help
# Example usage: ./ec2_manager.sh -l
```

Automation using Shell Scripts



AWS Lambda Functions

# Managing EC2 Instances using SDKs

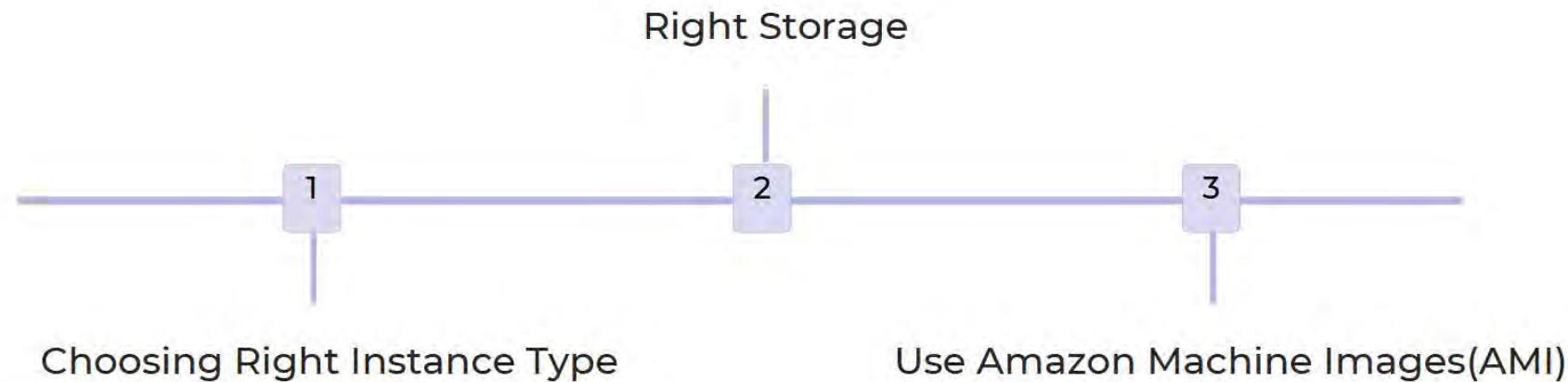
## Amazon SDKs

Use AWS SDKs to manage EC2 instances programmatically from your preferred programming languages such as Node.js, Java, Python, etc.

## AWS CloudFormation

Create your instances along with all required dependencies, security, network, and storage using AWS CloudformationStacks!

# Best Practices for Optimizing EC2 Instances for Cost



# Best Practices for Optimizing EC2 Instances for Cost

## Use AWS Compute Optimizer

Utilize available AWS Compute Optimizer tools to efficiently choose the most cost-effective instance type for your workload

## Reserved Instances

Save you up to 75% on EC2 instances and provide capacity reservation when you need it most.

## Spot Instances

Utilize the unused capacity of Amazon EC2 instances at highly reduced prices and perform cost-efficient batch processing or run other workloads with flexible start and end times.