Up and Running with Amazon Web Services

Kevin Hastie

EECS 441 - Winter '17

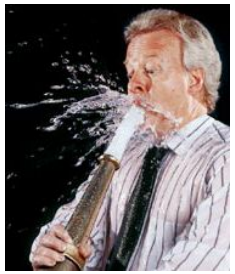
Who Am I?

Kevin Hastie

- Technology Manager @ ProQuest
- 20+ years of Software Engineering experience
- Experience at both small startups like Cappex.com, and industry leaders like Orbitz.com, Cars.com
- https://www.linkedin.com/in/kevinhastie/

Agenda

- Introduction
- AWS Fundamentals
  - Compute
    - EC2
  - Storage
    - S3
    - EBS
  - Security
    - IAM
  - Database
    - RDS (SQL)
    - DynamoDB (NoSQL)
  - Management
    - ELBs/ALBs
- Putting It All Together



Why the cloud?

Advantages and Benefits of AWS Cloud Computing

- Trade capital expense for variable expense.
- Increase speed and agility.
- Benefit from massive economies of scale.
- Stop spending money on running and maintaining data centers.
- Stop guessing capacity.
- Go global in minutes.

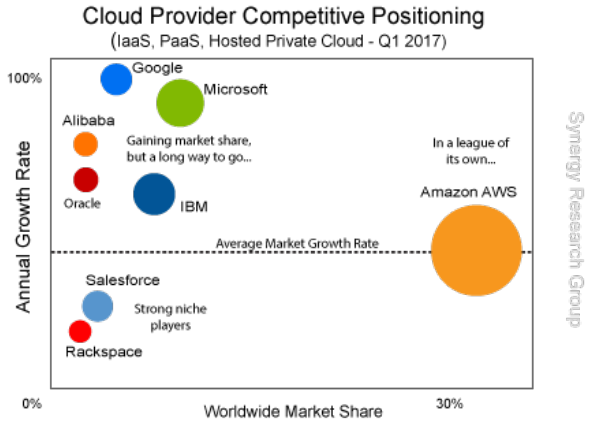
Advantages and Benefits of AWS Cloud Computing

- Trade capital expense for variable expense.
- Increase speed and agility.
- Benefit from massive economies of scale.
- Stop spending money on running and maintaining data centers.
- Stop guessing capacity.
- Go global in minutes.

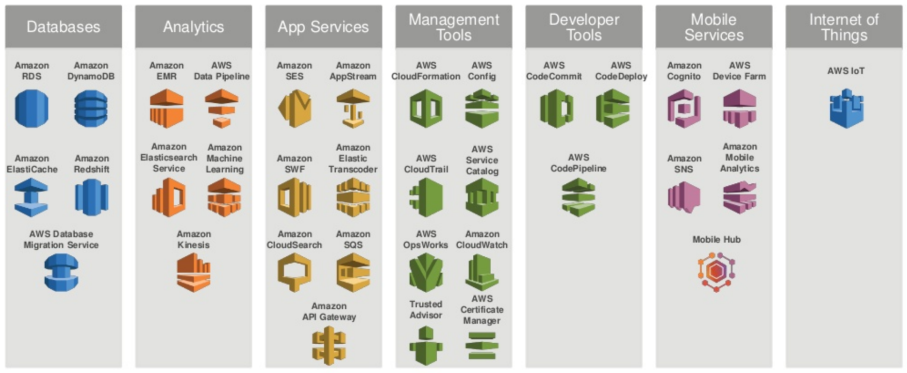
© 2016, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

amazon web services | Training and Certification 10

Why Amazon?



AWS Platform Services



© 2016, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

AWS Global Infrastructure

Clip slide

Regions

- Geographic locations
- Consist of **at least two** Availability Zones

Availability Zones

- Clusters of data centers
- **Isolated from failures** in other Availability Zones

Other Terms

- **Amazon Virtual Private Cloud (Amazon VPC)** lets you provision a logically isolated section of the AWS cloud where you can launch AWS resources in a virtual network that you define
- **Amazon Route 53** is a Domain Name System (DNS) web service
- **AWS Lightsail** - preconfigured setup, including DNS, storage, servers, developer stack (LAMP, LEMP, MEAN, or Node.js), or application etc...
- **AWS Lambda** – run code without thinking about servers, paying for only the compute time you consume

Amazon EC2 Facts

- **Scale capacity** as your computing requirements change
- Pay only for capacity that you actually use
- Choose **Linux** or **Windows**
- Deploy across **AWS Regions** and **Availability Zones** for reliability
- Use **tags** to help manage your Amazon EC2 resources

Launching an Amazon EC2 Instance via the Management Console

1. **Determine the AWS Region** in which you want to launch the Amazon EC2 instance.
2. **Launch** an Amazon EC2 instance from a pre-configured Amazon Machine Image (AMI).
3. **Choose an instance type** based on CPU, memory, storage, and network requirements.
4. **Configure** network, IP address, security groups, storage volume, tags, and key pair.

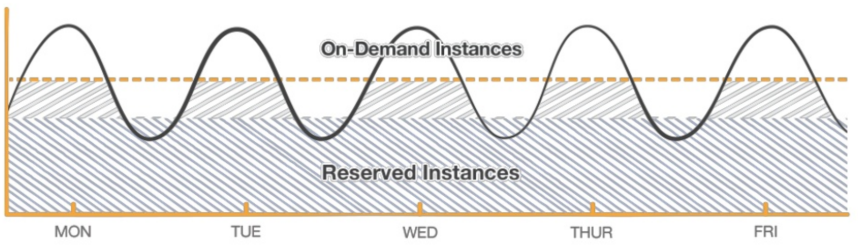
Current Generation Instances

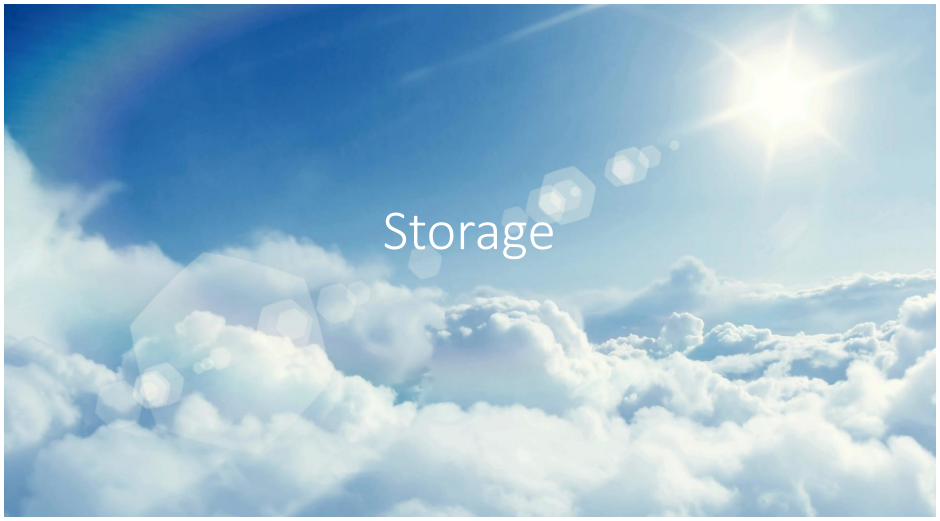
Instance Family	Some Use Cases
General purpose (t2, m4, m3)	• Low-traffic websites and web applications • Small databases and mid-size databases
Compute-optimized (c4, c3)	• High performance front-end fleets • Video-encoding
Memory-optimized (r3)	• High performance databases • Distributed memory caches
Storage-optimized (i2, d2)	• Data warehousing • Log or data-processing applications
GPU instances (g2)	• 3D application streaming • Machine learning

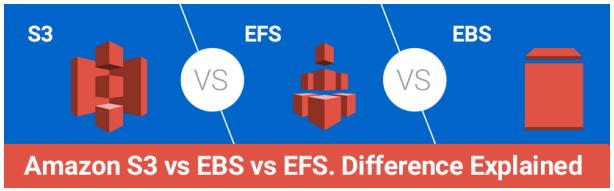
Amazon EC2 Purchasing Options

On-Demand Instances	Reserved Instances	Scheduled Instances	Spot Instances	Dedicated Instances	Dedicated Hosts
Pay by the hour.	Purchase, at a significant discount, instances that are always available 1-year to 3-year terms.	Purchase instances that are always available on the specified recurring schedule, for a one-year term.	Bid on unused instances, which can run as long as they are available and your bid is above the Spot price.	Pay, by the hour, for instances that run on single-tenant hardware.	Pay for a physical host that is fully dedicated to running your instances.

Layer your options







AMAZON S3	AMAZON EBS	AMAZON EFS
Can be publicly accessible	Accessible only via the given EC2 Machine	Accessible via several EC2 machines and AWS services
Web interface	File System interface	Web and file system interface
Object Storage	Block Storage	Object storage
Scalable	Hardly scalable	Scalable
Slower than EBS and EFS	Faster than S3 and EFS	Faster than S3, slower than EBS
Good for storing backups	**Is meant to be EC2 drive**	**Good for shareable applications and workloads**

Amazon Simple Storage Service (S3)

Amazon S3

- Storage for the Internet
- Natively online, HTTP access
- Storage that allows you to store and retrieve **any amount of data**, any time, from anywhere on the web
- **Highly scalable**, reliable, fast and durable

Cache Me Outside...

Amazon Elastic Block Store (EBS)

Amazon EBS

- **Persistent block level storage** volumes offer consistent and low-latency performance.
- Stored data is automatically replicated within its Availability Zone.
- Snapshots are stored durably in Amazon S3.

Amazon EBS and Amazon S3

Clip slide

	Amazon EBS	Amazon S3
Paradigm	Block storage with file system	Object store
Performance	Very fast	Fast
Redundancy	Across multiple servers in an Availability Zone	Across multiple facilities in a Region
Security	EBS Encryption – Data volumes and Snapshots	Encryption
Access from the Internet?	No (1)	Yes (2)
Typical use case	It is a disk drive	Online storage

(1) Accessible from the Internet if mounted to server and set up as FTP, etc.
(2) Only with proper credentials, unless ACLs are world-readable

Amazon EBS vs. Amazon EC2 Instance Store

Amazon EBS

- Data stored on an Amazon EBS volume can persist independently of the life of the instance.
- Storage is **persistent**.

Amazon EC2 Instance Store

- Data stored on a local instance store persists only as long as the instance is alive.
- Storage is **ephemeral**.

Reboot vs. Stop vs. Terminate

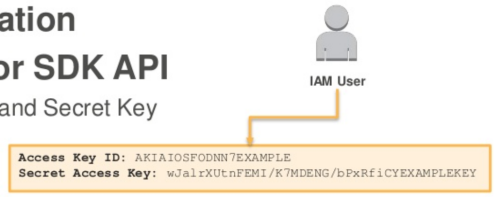
Characteristic	Reboot	Stop/Start (EBS-backed instances only)	Terminate
Host computer	The instance **stays on the same host computer**.	The instance runs on a **new host computer**.	
Public IP address	No change	**New address assigned**	
Elastic IP addresses (EIP)	EIP remains associated with the instance.	EIP remains associated with the instance.	EIP is **disassociated** from the instance.
Instance store volumes	Preserved	**Erased**	**Erased**
EBS volume	Preserved	Preserved	Boot volume is **deleted by default**.
Billing	Instance billing hour doesn't change.	You **stop incurring charges** as soon as state is changed to *stopping*.	You **stop incurring charges** as soon as state is changed to *shutting-down*.

SSL Endpoints

SSL Endpoints	Security Groups	VPC
Secure Transmission	**Instance Firewalls**	**Network Control**
Use secure endpoints to establish secure communication sessions (HTTPS).	Use security groups to configure firewall rules for instances.	Use public and private subnets, NAT, and VPN support in your virtual private cloud to create low-level networking constraints for resource access.

AWS IAM Authentication

- **Authentication**
- **AWS CLI or SDK API**
  - Access Key and Secret Key

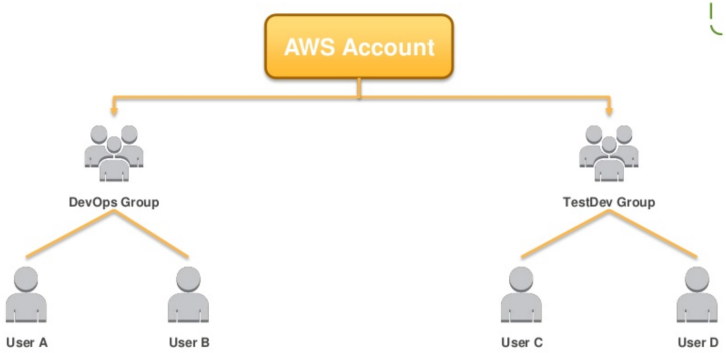


AWS CLI

```
[redacted]:~ [redacted]$ aws configure
AWS Access Key ID [****************O22A]:
AWS Secret Access Key [****************4m8i]:
Default region name [ap-southeast-1]:
Default output format [json]:
```

AWS SDK & API

AWS IAM User Management - Groups



AWS IAM Policy Assignment

AWS IAM Roles

- An IAM role uses a policy.
- An IAM role has no associated credentials.
- IAM users, applications, and services may assume IAM roles.


IAM Roles

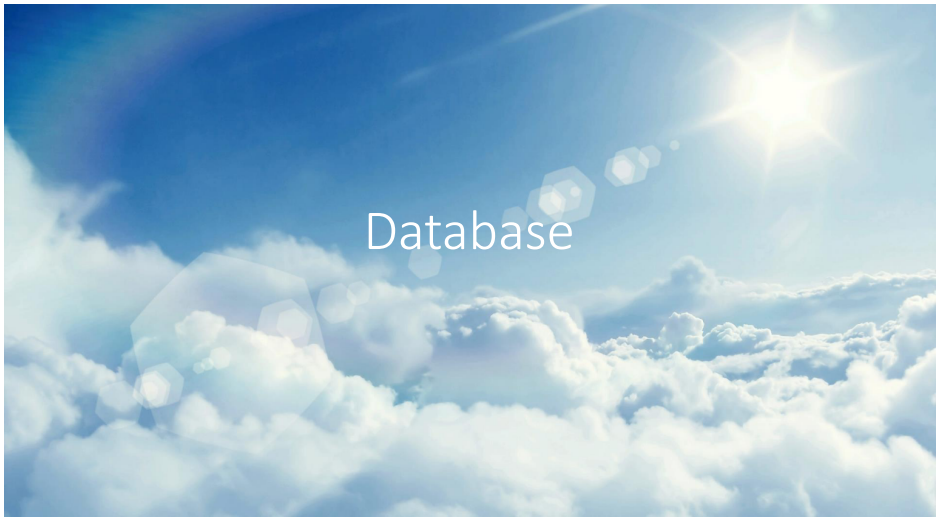
Example: Application Access to AWS Resources

- Python application hosted on an Amazon EC2 Instance needs to interact with Amazon S3.
- AWS credentials are required:
  - ~~Option 1: Store AWS Credentials on the Amazon EC2 instance~~.
  - Option 2: Securely distribute AWS credentials to AWS Services and Applications.

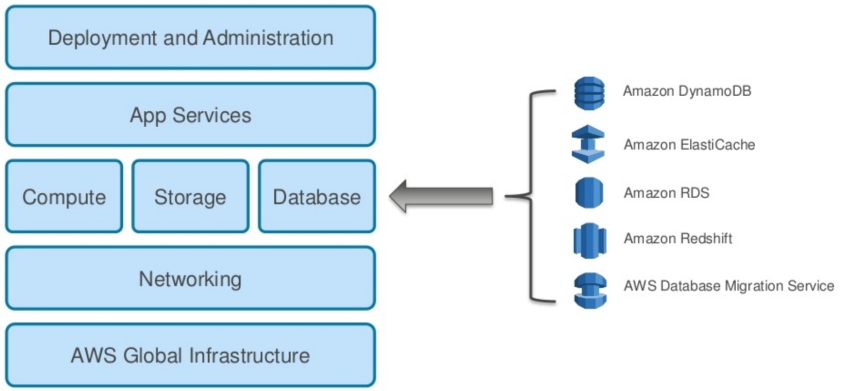

IAM Roles

AWS IAM Best Practices

- **Delete** AWS account (root) access keys.
- Create **individual** IAM users.
- **Use groups** to assign permissions to IAM users.
- Grant **least privilege**.
- Configure a **strong password policy**.
- Enable **MFA** for privileged users.



AWS Managed Database Services



Amazon Relational Database Service (RDS)

Amazon RDS

- Cost-efficient and **resizable capacity**
- Manages time-consuming **database administration** tasks
- Access to the full capabilities of **Amazon Aurora**, **MySQL**, **MariaDB**, **Microsoft SQL Server**, **Oracle**, and **PostgreSQL** databases

Multi-AZ RDS Deployment

- With **Multi-AZ** operation, your database is **synchronously replicated to another Availability Zone** in the same AWS Region.
- **Failover** to the standby **automatically** occurs in case of master database failure.
- Planned maintenance is applied first to standby databases.

Amazon DynamoDB

Amazon DynamoDB

- Allows you to store any amount of data with **no limits.**
- Provides fast, predictable performance using **SSDs.**
- Allows you to easily provision and change the **request capacity** needed for each table.
- Is a **fully managed**, **NoSQL** database service.

Amazon RDS and Amazon DynamoDB

Factors	Relational (Amazon RDS)	NoSQL (Amazon DynamoDB)
Application Type	• Existing database apps • Business process–centric apps	• New web-scale applications • Large number of small writes and reads
Application Characteristics	• **Relational** data models, transactions • **Complex** queries, joins, and updates	• Simple data models, transactions • Range queries, simple updates
Scaling	Application or **DBA–architected** (clustering, partitions, sharding)	**Seamless, on-demand scaling** based on application requirements
QoS	• Performance–depends on data model, indexing, query, and storage optimization • Reliability and availability • Durability	• Performance–**Automatically optimized** by the system • Reliability and availability • Durability

Database Considerations

If You Need	Consider Using
A relational database service with minimal administration	**Amazon RDS** • Choice of Amazon Aurora, MySQL, MariaDB, Microsoft SQL Server, Oracle, or PostgreSQL database engines • Scale compute and storage • Multi-AZ availability
A fast, highly scalable NoSQL database service	**Amazon DynamoDB** • Extremely fast performance • Seamless scalability and reliability • Low cost
A database you can manage on your own	Your choice of **AMIs** on Amazon EC2 and Amazon EBS that provide scale compute and storage, complete control over instances, and more.

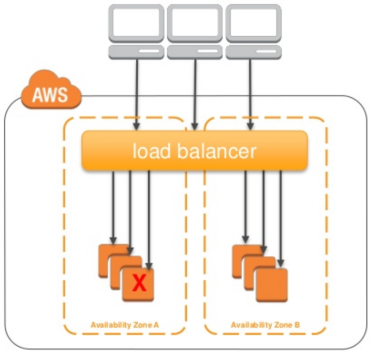
Elastic Load Balancing

Elastic Load Balancing

- **Distributes** traffic across multiple EC2 instances, in multiple Availability Zones
- Supports **health checks** to detect unhealthy Amazon EC2 instances
- Supports the **routing and load balancing** of HTTP, HTTPS, SSL, and TCP traffic to Amazon EC2 instances

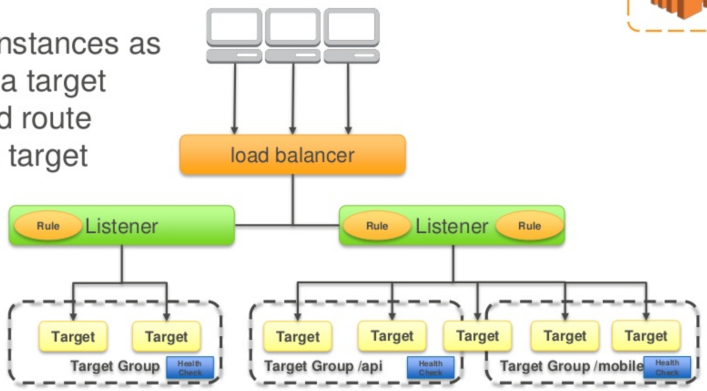
Classic Load Balancer - How It Works

Register instances with your load balancer.



Application Load Balancer – How It Works

Register instances as targets in a target group, and route traffic to a target group.



Load Balancer Comparison

Classic Load Balancer benefits include support for:

- EC2-Classic.
- VPC.
- TCP and SSL listeners.
- Sticky sessions.

ALB benefits include support for:

- Path-based routing.
- Routing requests to multiple services on a single EC2 instance.
- Containerized applications.
- Monitoring the health of each service independently.

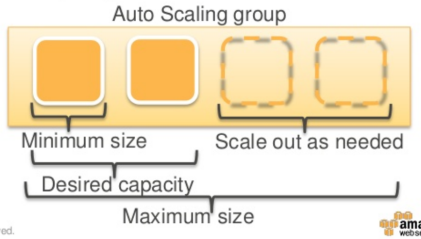
Auto Scaling

Auto Scaling

- **Scale** your Amazon EC2 capacity **automatically**
- Well-suited for applications that experience **variability in usage**
- Available at no additional charge

Auto Scaling Groups

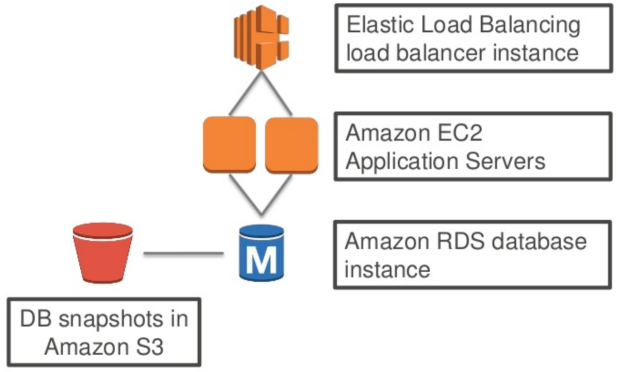
- Contain a collection of EC2 instances that share similar characteristics.
- Instances in an Auto Scaling group are treated as a **logical grouping** for the purpose of instance scaling and management.



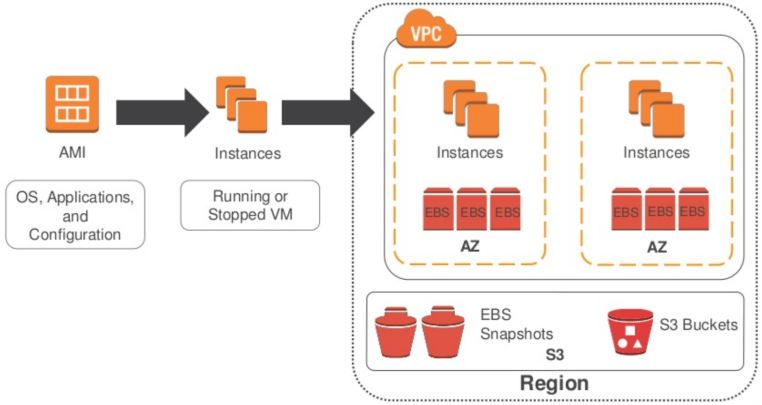
Dynamic Scaling

- You can create a scaling policy that uses **CloudWatch alarms** to determine:
  - When your Auto Scaling group should **scale out**.
  - When your Auto Scaling group should **scale in**.
- You can use alarms to monitor:
  - Any of the metrics that AWS services send to Amazon CloudWatch.
  - Your own **custom metrics**.

A Simple Application Architecture



Amazon EC2 Instances



Example Instructions to Connect to an EC2 Instance

Search for “AWS Free Tier” for the AWS Dashboard

Here’s an example of the instructions you’ll see when you create an EC2 instance:

To access your instance:

1.Open an SSH client. (find out how to connect using PuTTY)
2.Locate your private key file (MyPrivateKeyFoo.pem). The wizard automatically detects the key you used to launch the instance.
3.Your key must not be publicly viewable for SSH to work. Use this command if needed: `chmod 400 MyPrivateKeyFoo.pem`
4.Connect to your instance using its Private IP: `10.241.128.99`

Example:
`ssh -i "MyPrivateKeyFoo.pem" ec2-user@10.241.128.99`

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our connection documentation.

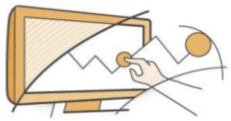
Experiments to try

- Launch a new instance
- Create an s3 bucket
- Create a database
- Create a queue
- Create a lambda
- One of the quickstart guides (app server, static website, Lightsail, etc...)

Resources

Expand Your Cloud Skills with AWS

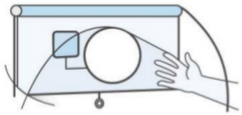
Online videos and labs



Start working with an AWS service in minutes with free online instructional videos and labs

aws.amazon.com/training/self-paced-labs

Instructor-led courses



Learn how to design, deploy, and operate highly available, cost-effective, and secure applications on AWS

aws.amazon.com/training

Certification

Validate your proven technical expertise with the AWS platform and gain recognition for your skills

aws.amazon.com/certification

Self-Paced Labs

- Learn an individual AWS Service topic
- Follow a Learning Quest by AWS Service Area or Use Case
- Practice working with AWS as you prepare for an exam

For more information, see aws.amazon.com/training/self-paced-labs/.

AWS Products & Services Guides

- Compute
  - **Amazon EC2**
  - Amazon EC2 Container Registry
  - **Amazon EC2 Container Service**
  - Amazon Lightsail
  - **Amazon VPC**
  - AWS Batch
  - AWS Elastic Beanstalk
  - **AWS Lambda**
  - **Auto Scaling**
  - **Elastic Load Balancing**
- Storage
  - **Amazon Simple Storage Service (S3)**
  - **Amazon Elastic Block Storage (EBS)**
  - Amazon Elastic File System (EFS)
  - Amazon Glacier
  - AWS Storage Gateway
  - AWS Snowball
  - AWS Snowball Edge
  - AWS Snowmobile
- Database
  - Amazon Aurora
  - **Amazon RDS**
  - **Amazon DynamoDB**
  - Amazon ElastiCache
  - Amazon Redshift
  - AWS Database Migration Service
- Migration
  - AWS Application Discovery Service
  - AWS Database Migration Service
  - AWS Server Migration Service
  - AWS Snowball
  - AWS Snowball Edge
  - AWS Snowmobile
- Networking & Content Delivery
  - **Amazon VPC**
  - Amazon CloudFront
  - **Amazon Route 53**
  - AWS Direct Connect
  - **Elastic Load Balancing**
- Developer Tools
  - AWS CodeCommit
  - AWS CodeBuild
  - AWS CodeDeploy
  - AWS CodePipeline
  - AWS X-Ray
  - AWS Command Line Interface
- Management Tools
  - **Amazon CloudWatch**
  - Amazon EC2 Systems Manager
  - AWS CloudFormation
  - AWS CloudTrail
  - AWS Config
  - AWS OpsWorks
  - AWS Service Catalog
  - AWS Trusted Advisor
  - AWS Personal Health Dashboard
  - **AWS Command Line Interface**
  - **AWS Management Console**
  - AWS Managed Services

AWS Products & Services Guides (con't)

- Security, Identity & Compliance
  - **AWS Identity and Access Management (IAM)**
  - Amazon Inspector
  - AWS Certificate Manager
  - AWS CloudHSM
  - AWS Directory Service
  - Amazon Cloud Directory
  - AWS Key Management Service
  - AWS Organizations
  - AWS Shield
  - AWS WAF
  - AWS Artifact
- Analytics
  - Amazon Athena
  - Amazon EMR
  - Amazon CloudSearch
  - Amazon Elasticsearch Service
  - Amazon Kinesis
  - Amazon Redshift
  - Amazon QuickSight
  - AWS Data Pipeline
  - AWS Glue
- Artificial Intelligence
  - Amazon Lex
  - Amazon Polly
  - Amazon Rekognition
  - Amazon Machine Learning
- Mobile Services
  - **AWS Mobile Hub**
  - Amazon API Gateway
  - Amazon Cognito
  - Amazon Pinpoint
  - AWS Device Farm
  - AWS Mobile SDK
- Application Services
  - AWS Step Functions
  - Amazon API Gateway
  - Amazon Elastic Transcoder
  - Amazon AppStream
- Messaging
  - **Amazon SQS**
  - Amazon Pinpoint
  - Amazon SES
  - Amazon SNS
- Business Productivity
  - Amazon Chime
  - Amazon WorkDocs
  - Amazon WorkMail
- Desktop & App Streaming
  - Amazon WorkSpaces
  - Amazon AppStream 2.0
- Software
  - AWS Marketplace
- Internet of Things
  - AWS IoT Platform
  - AWS Greengrass
  - AWS IoT Button
- Game Development
  - Amazon Lumberyard

Mobile Products/Services

- AWS Mobile Hub
  Build, Test, and Monitor Apps
- Amazon API Gateway
  Build, Deploy, and Manage APIs
- Amazon Cognito
  User Identity and App Data Synchronization
- Amazon SNS – Pub/sub mobile notifications. Send push notifications and SMS to mobile devices.
- Amazon Pinpoint
  Push Notifications for Mobile Apps
- AWS Device Farm
  Test Android, FireOS, and iOS Apps on Real Devices in the Cloud
- AWS Mobile SDK
  Mobile Software Development Kit

Choosing the Right Amazon EC2 Instance

AWS uses Intel® Xeon® processors to provide customers with high performance and value. EC2 instance types are optimized for different use cases, workload requirements and come in multiple sizes.

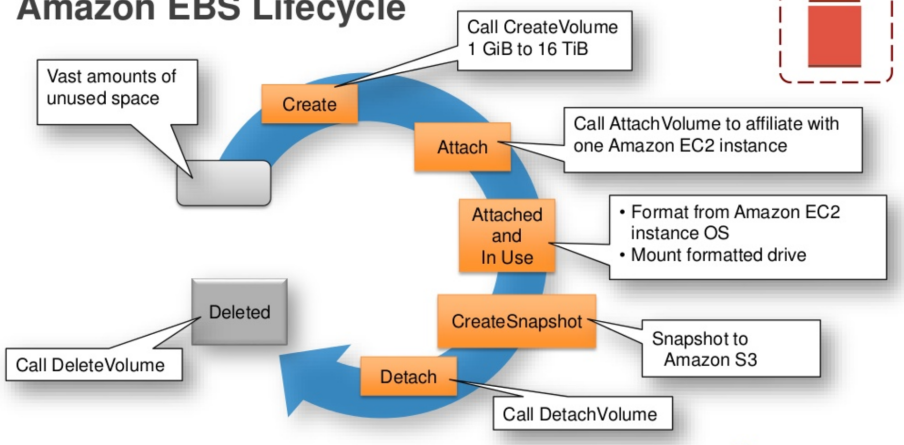
Consider the following when choosing your instances:

- Core count
- Memory size
- Storage size and type
- Network performance
- CPU technologies

Amazon S3 Facts

- Can store an **unlimited number of objects** in a bucket
- Objects can be **up to 5 TB**; no bucket size limit
- Designed for **99.999999999%** durability and **99.99%** availability of objects over a given year
- Can use **HTTP/S** endpoints to store and retrieve any amount of data, at any time, from anywhere on the web
- Is highly scalable, reliable, fast, and inexpensive
- Can use optional server-side **encryption** using AWS or customer-managed provided client-side encryption
- Auditing is provided by access logs
- Provides standards-based **REST** and SOAP interfaces

Amazon EBS Lifecycle



Amazon EBS Facts

- EBS is recommended when data must be **quickly accessible** and requires **long-term persistence**.
- You can launch your EBS volumes as **encrypted** volumes – data stored at rest on the volume, disk I/O, and snapshots created from the volume are all encrypted.
- You can create **point-in-time snapshots** of EBS volumes, which are persisted to Amazon S3.

ELB Classic vs. Application ELB

Application ELB	
Protocols	HTTP, HTTPS
Platforms	EC2-VPC
Sticky sessions (cookies)	load balancer generated
Back-end server authentication	
Back-end server encryption	✓
Idle connection timeout	✓
Connection draining	✓
Cross-zone load balancing †	Always enabled
Path-based routing	✓
Route to multiple ports on a single instance	✓
HTTP/2 support	✓
Websockets support	✓
Load balancer deletion protection	✓

Classic ELB	
Protocols	HTTP, HTTPS, TCP, SSL
Platforms	EC2-Classic, EC2-VPC
Sticky sessions (cookies)	✓
Back-end server authentication	✓
Back-end server encryption	✓
Idle connection timeout	✓
Connection draining	✓
Cross-zone load balancing †	✓
Path-based routing	
Route to multiple ports on a single instance	
HTTP/2 support	
Websockets support	
Load balancer deletion protection	

Auto Scaling Benefits