FS Mobility Academy

# An introduction to Cloud Computing and Amazon Web Services

Luigi Libero Lucio Starace

luigi.starace@gmail.com

March 27, 2019

University of Naples, Federico II

Ferrovie dello Stato Italiane | Università degli Studi di Napoli Federico II

# Cloud Computing

# Cloud Computing

> Cloud computing is the on-demand delivery of computing resources through a cloud services platform via the internet with pay-as-you-go pricing.

# Cloud Computing

> Cloud computing is the **on-demand delivery** of computing resources through a cloud services platform via the internet with pay-as-you-go pricing.

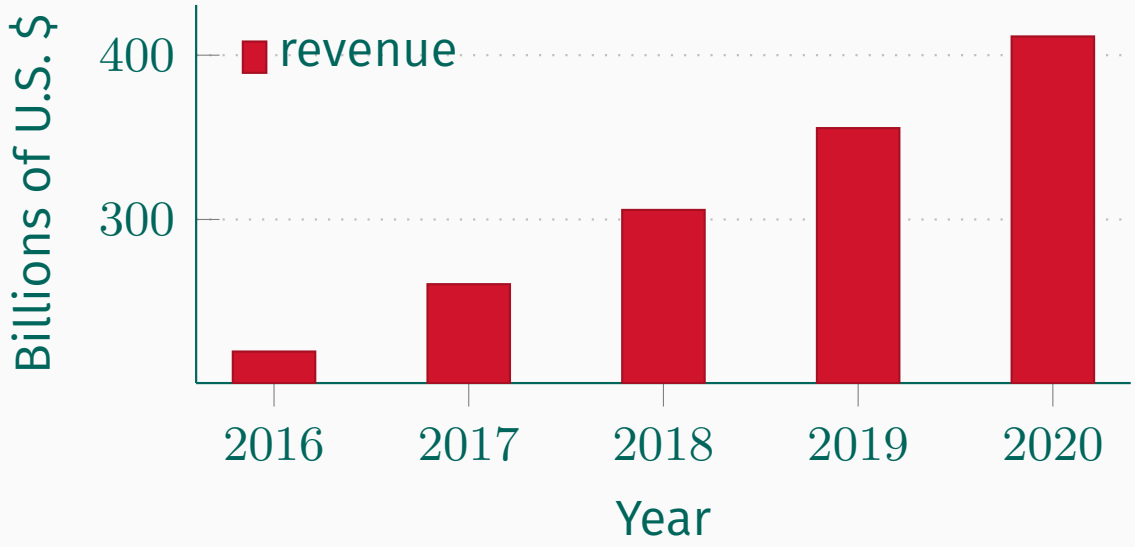
# Cloud Computing

> Cloud computing is the **on-demand delivery** of computing resources through a cloud services platform via the internet with **pay-as-you-go** pricing.

# Some stats

Worldwide Public Cloud Services Revenue Forecast (Billions of U.S. Dollars) [Gar17]

| 2016 | 2017 | 2018 | 2019 | 2020 |
|---|---|---|---|---|
| 219,6 | 260,6 | 305,8 | 355,6 | 411,4 |



# Why might one use Cloud Computing?

Suppose you have a great business idea: the **DATA** system (**D**ata **A**nalytics for **T**ransport **A**gencies).

What steps do you take to start making money?

# Initiating a business: the traditional approach

1. Estimate supply and demand;

# Initiating a business: the traditional approach

1. Estimate supply and demand;
2. Estimate infrastructural needs;

# Initiating a business: the traditional approach

1. Estimate supply and demand;
2. Estimate infrastructural needs;
3. Purchase and deploy infrastructure;

# Initiating a business: the traditional approach

1. Estimate supply and demand;
2. Estimate infrastructural needs;
3. Purchase and deploy infrastructure;
4. Install and test your system;

# Initiating a business: the traditional approach

1. Estimate supply and demand;
2. Estimate infrastructural needs;
3. Purchase and deploy infrastructure;
4. Install and test your system;
5. Offer your services to clients.

# Initiating a business: issues with the traditional approach

- Infrastructure is very expensive:

# Initiating a business: issues with the traditional approach

- Infrastructure is very expensive:
  - Hardware costs; Dell

# Initiating a business: issues with the traditional approach

- Infrastructure is very expensive:
  - Hardware costs; Dell
  - Real estate costs;

# Initiating a business: issues with the traditional approach

- Infrastructure is very expensive:
  - Hardware costs; Dell
  - Real estate costs;
  - Cooling system; Power bill;

# Initiating a business: issues with the traditional approach

- Infrastructure is very expensive:
  - Hardware costs; Dell
  - Real estate costs;
  - Cooling system; Power bill;
  - A few systems engineers;

# Initiating a business: issues with the traditional approach

- Infrastructure is very expensive:
  - Hardware costs; Dell
  - Real estate costs;
  - Cooling system; Power bill;
  - A few systems engineers;
  - Redundancy for high reliability.

# Initiating a business: issues with the traditional approach

- Infrastructure is very expensive:
  - Hardware costs; Dell
  - Real estate costs;
  - Cooling system; Power bill;
  - A few systems engineers;
  - Redundancy for high reliability.
- It takes time and deploy the infrastructure;

# Initiating a business: issues with the traditional approach

- Infrastructure is very expensive:
  - Hardware costs; Dell
  - Real estate costs;
  - Cooling system; Power bill;
  - A few systems engineers;
  - Redundancy for high reliability.
- It takes time and deploy the infrastructure;
- What if the estimations were wrong?

# Initiating a business: using Cloud Computing

1. Choose one or more cloud services providers;

# Initiating a business: using Cloud Computing

1. Choose one or more cloud services providers;
2. Deploy your systems on the cloud;

# Initiating a business: using Cloud Computing

1. Choose one or more cloud services providers;
2. Deploy your systems on the cloud;
3. Offer your services to clients;

# Initiating a business: using Cloud Computing

1. Choose one or more cloud services providers;
2. Deploy your systems on the cloud;
3. Offer your services to clients;
4. Pay for what you use.

# Why might one use Cloud Computing?

**Traditional process**

- × High investment risk;

**With Cloud Computing**

# Why might one use Cloud Computing?

**Traditional process**

- × High investment risk;

**With Cloud Computing**

- ✓ Reduced risk;

# Why might one use Cloud Computing?

**Traditional process**

- × High investment risk;
- × Long time-to-market;

**With Cloud Computing**

- ✓ Reduced risk;

# Why might one use Cloud Computing?

**Traditional process**

- × High investment risk;
- × Long time-to-market;

**With Cloud Computing**

- ✓ Reduced risk;
- ✓ Shorter time-to-market;

# Why might one use Cloud Computing?

**Traditional process**

- × High investment risk;
- × Long time-to-market;
- ✓ Manages own data;

**With Cloud Computing**

- ✓ Reduced risk;
- ✓ Shorter time-to-market;

# Why might one use Cloud Computing?

**Traditional process**

- × High investment risk;
- × Long time-to-market;
- ✓ Manages own data;

**With Cloud Computing**

- ✓ Reduced risk;
- ✓ Shorter time-to-market;
- × Trust the vendor?

# Why might one use Cloud Computing?

**Traditional process**

- × High investment risk;
- × Long time-to-market;
- ✓ Manages own data;
- ✓ Completely in control;

**With Cloud Computing**

- ✓ Reduced risk;
- ✓ Shorter time-to-market;
- × Trust the vendor?

# Why might one use Cloud Computing?

**Traditional process**

- × High investment risk;
- × Long time-to-market;
- ✓ Manages own data;
- ✓ Completely in control;

**With Cloud Computing**

- ✓ Reduced risk;
- ✓ Shorter time-to-market;
- × Trust the vendor?
- × Dependant from a specific vendor?

# Cloud Computing: Service Models

- Software as a Service (SaaS)

- **Software as a Service (SaaS)**
  The service vendor provides the user with a completed product that is run and managed by the service provider.

# Cloud Computing: Service Models

- **Software as a Service (SaaS)**
  The service vendor provides the user with a completed product that is run and managed by the service provider.
- **Platform as a Service (PaaS)**

# Cloud Computing: Service Models

- **Software as a Service (SaaS)**
  The service vendor provides the user with a completed product that is run and managed by the service provider.
- **Platform as a Service (PaaS)**
  The service vendor provides the user with a set of API which can be used to build, test and deploy applications.

# Cloud Computing: Service Models

- **Software as a Service (SaaS)**
  The service vendor provides the user with a completed product that is run and managed by the service provider.
- **Platform as a Service (PaaS)**
  The service vendor provides the user with a set of API which can be used to build, test and deploy applications.
- **Infrastructure as a Service (IaaS)**

# Cloud Computing: Service Models

- **Software as a Service (SaaS)**
  The service vendor provides the user with a completed product that is run and managed by the service provider.
- **Platform as a Service (PaaS)**
  The service vendor provides the user with a set of API which can be used to build, test and deploy applications.
- **Infrastructure as a Service (IaaS)**
  The service vendor provides users access to computing resources such as servers, storage and networking.

# Service Models: a visual comparison

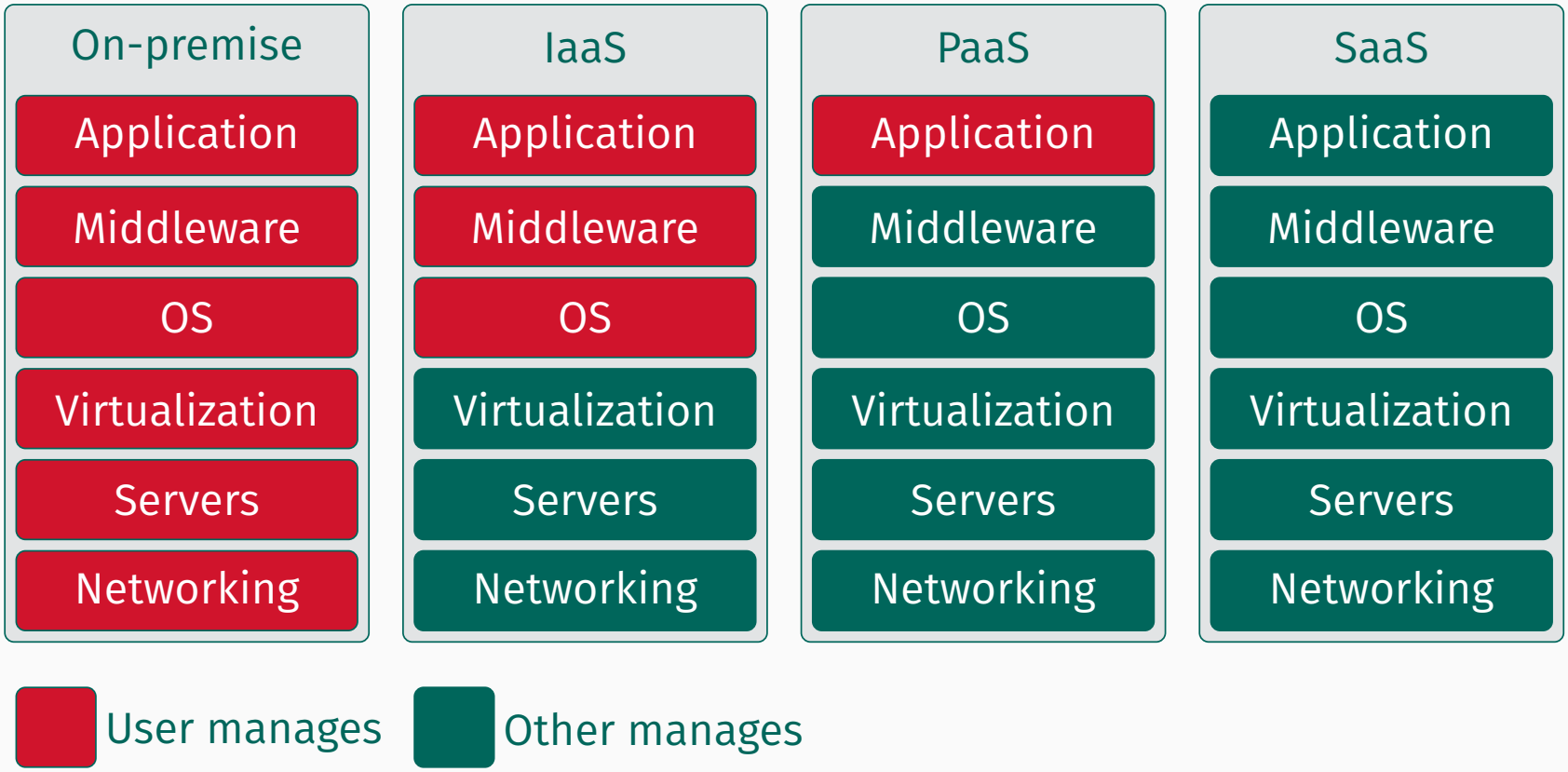



**Figure 1:** A service models comparison

# The bigwigs

- Google

Google Cloud

# The bigwigs

- Google
- IBM

# The bigwigs

- Google
- IBM
- Microsoft

# The bigwigs

- Google
- IBM
- Microsoft
- Alibaba

# The bigwigs

- Google
- IBM
- Microsoft
- Alibaba
- Oracle

# The bigwigs

- Google
- IBM
- Microsoft
- Alibaba
- Oracle
- Amazon

# Market Share

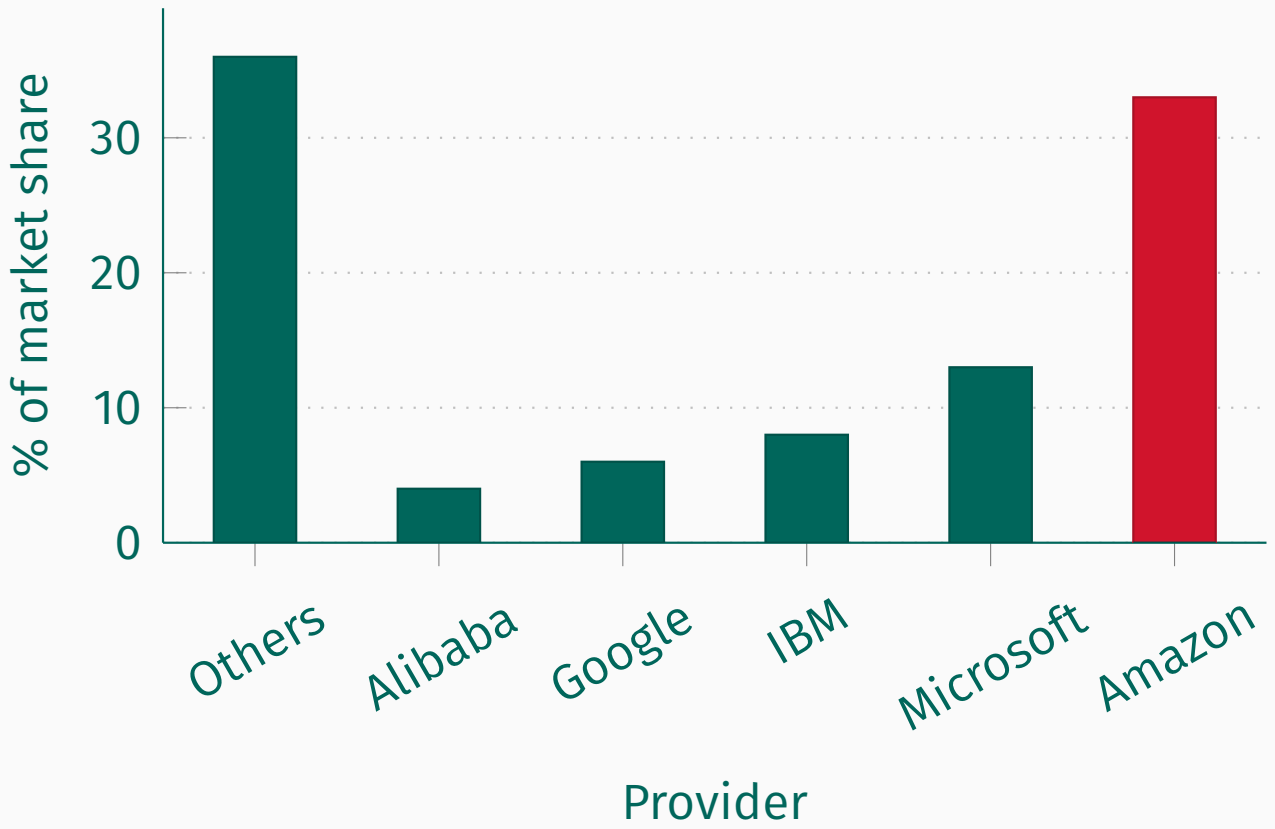



**Figure 2:** Market share in Q4 2017 (IaaS, PaaS, Hosted Private Cloud) [Syn18]

# Amazon Web Services (AWS)

# Amazon Web Services


Amazon Web Services is a collection of cloud-based services.

# Amazon Web Services


Amazon Web Services is a collection of cloud-based services.

**A very big one.**

# Amazon Web Services


Amazon Web Services is a collection of cloud-based services.

**A VERY big one.**

# Amazon Web Services (AWS)

## Computing

# Amazon Elastic Compute Cloud (EC2)

- (Virtual) Servers on demand

Azure: Virtual Machines web
Google Cloud: Compute Engine web

# Amazon Elastic Compute Cloud (EC2)

- (Virtual) Servers on demand
- Different types of instances to suit computing needs

Azure: Virtual Machines web
Google Cloud: Compute Engine web

# Amazon Elastic Compute Cloud (EC2)

- (Virtual) Servers on demand
- Different types of instances to suit computing needs
- Per-second (or per-hour) billing

Azure: Virtual Machines web
Google Cloud: Compute Engine web

# Amazon Elastic Compute Cloud (EC2)

- (Virtual) Servers on demand
- Different types of instances to suit computing needs
- Per-second (or per-hour) billing
- Data transfer **not** included!

Azure: Virtual Machines web
Google Cloud: Compute Engine web

# Amazon Elastic Compute Cloud (EC2)

- (Virtual) Servers on demand
- Different types of instances to suit computing needs
- Per-second (or per-hour) billing
- Data transfer **not** included!
- Persistent storage **not** included!

Azure: Virtual Machines web
Google Cloud: Compute Engine web

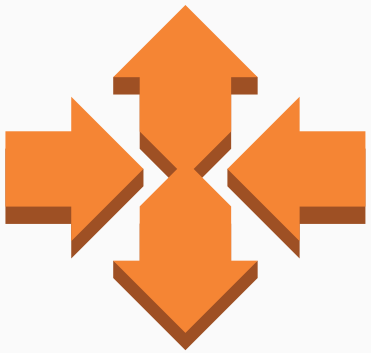
# Amazon Elastic Compute Cloud (EC2)

- (Virtual) Servers on demand
- Different types of instances to suit computing needs
- Per-second (or per-hour) billing
- Data transfer **not** included!
- Persistent storage **not** included!
- Scaling **not** included!

Azure: Virtual Machines web
Google Cloud: Compute Engine web

# Amazon EC2 Auto Scaling

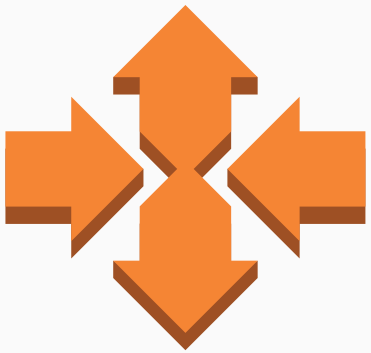
- *Scaling is the ability to increase or decrease the compute capacity of your application*



Azure: Virutal Machine Scale Sets web
Google Cloud: Load Balancing web

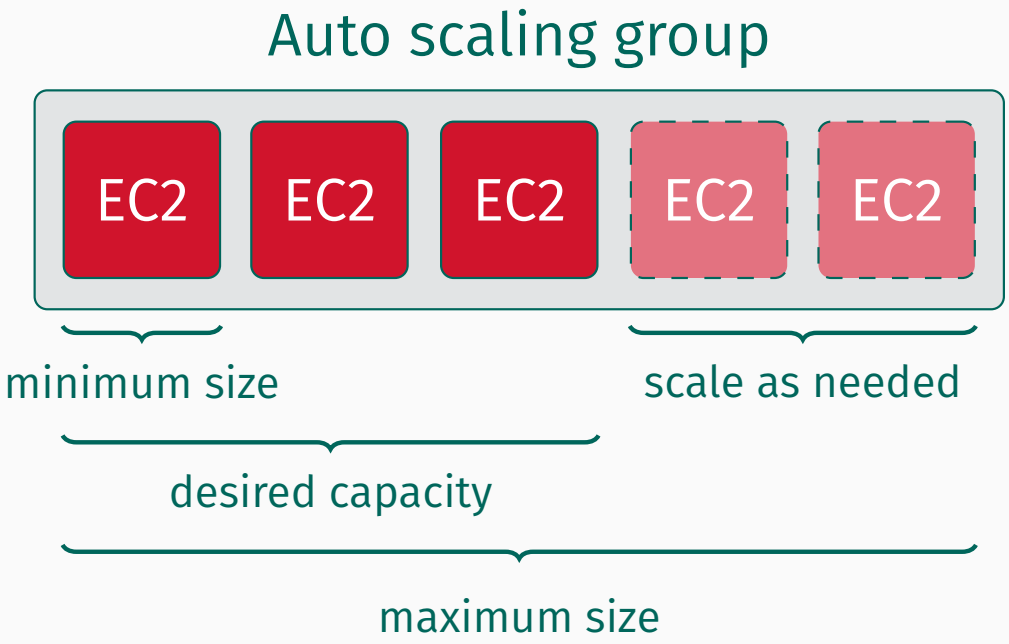
# Amazon EC2 Auto Scaling

- *Scaling is the ability to increase or decrease the compute capacity of your application*
- Scale your application manually, on a scheduled basis or on demand



Azure: Virutal Machine Scale Sets web
Google Cloud: Load Balancing web

# Amazon EC2 Auto Scaling: details



# Amazon Elastic Load Balancing (ELB)

- Distributes incoming traffic across multiple EC2 instances

Azure: Load Balancer web

# Amazon Elastic Load Balancing (ELB)

- Distributes incoming traffic across multiple EC2 instances
- Pay-per-use billing

Azure: Load Balancer web

# Amazon Elastic Load Balancing (ELB)

- Distributes incoming traffic across multiple EC2 instances
- Pay-per-use billing
  - Execution time

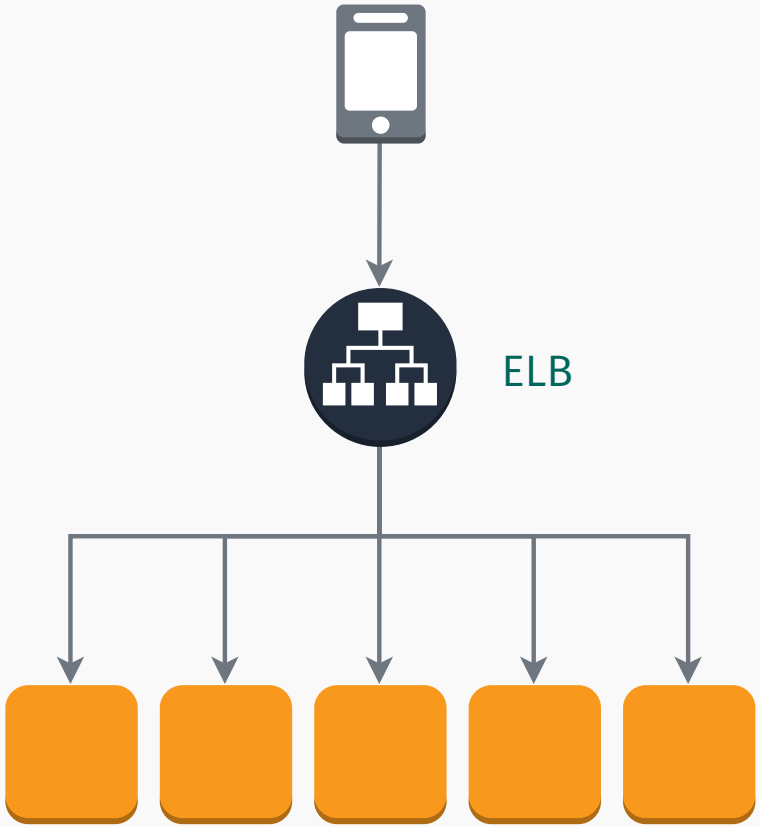
Azure: Load Balancer web

# Amazon Elastic Load Balancing (ELB)

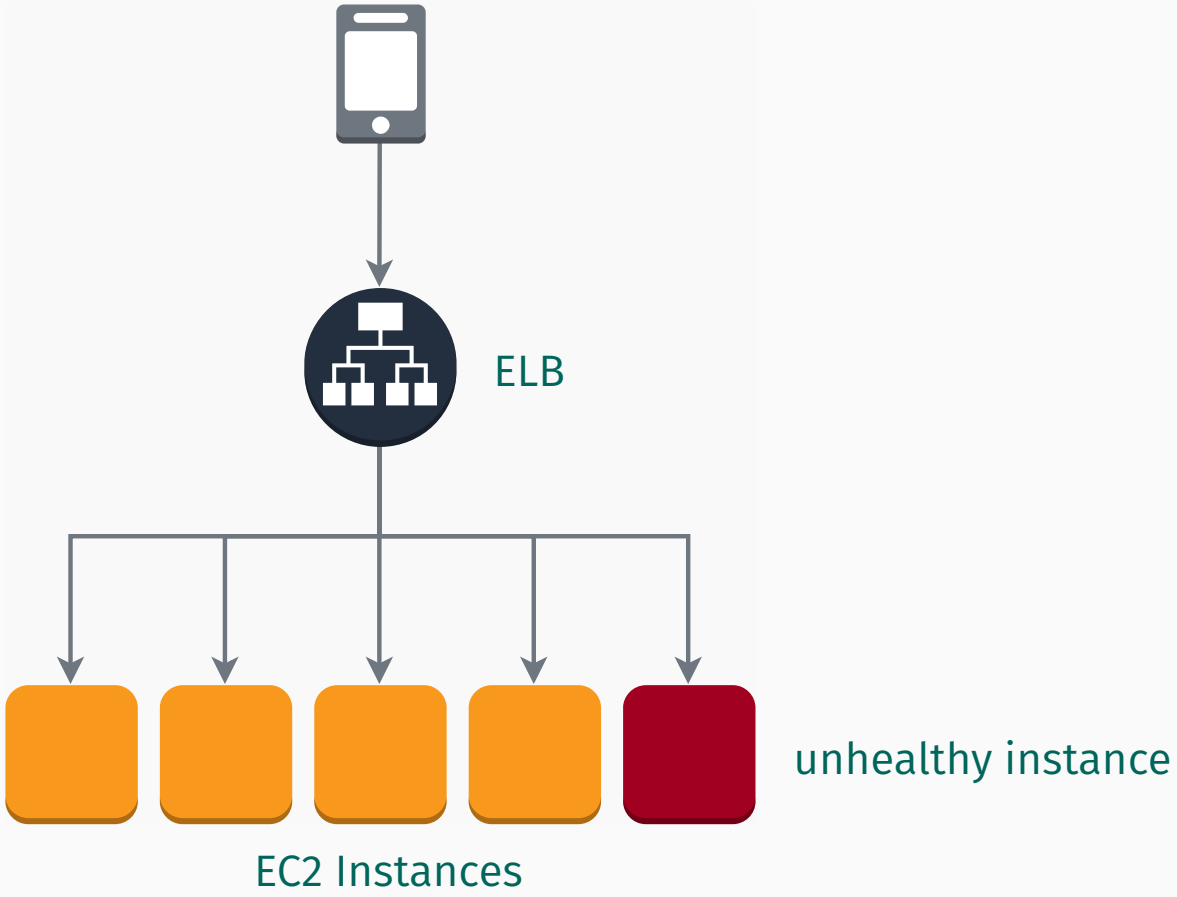
- Distributes incoming traffic across multiple EC2 instances
- Pay-per-use billing
  - Execution time
  - Number of requests / traffic

Azure: Load Balancer web

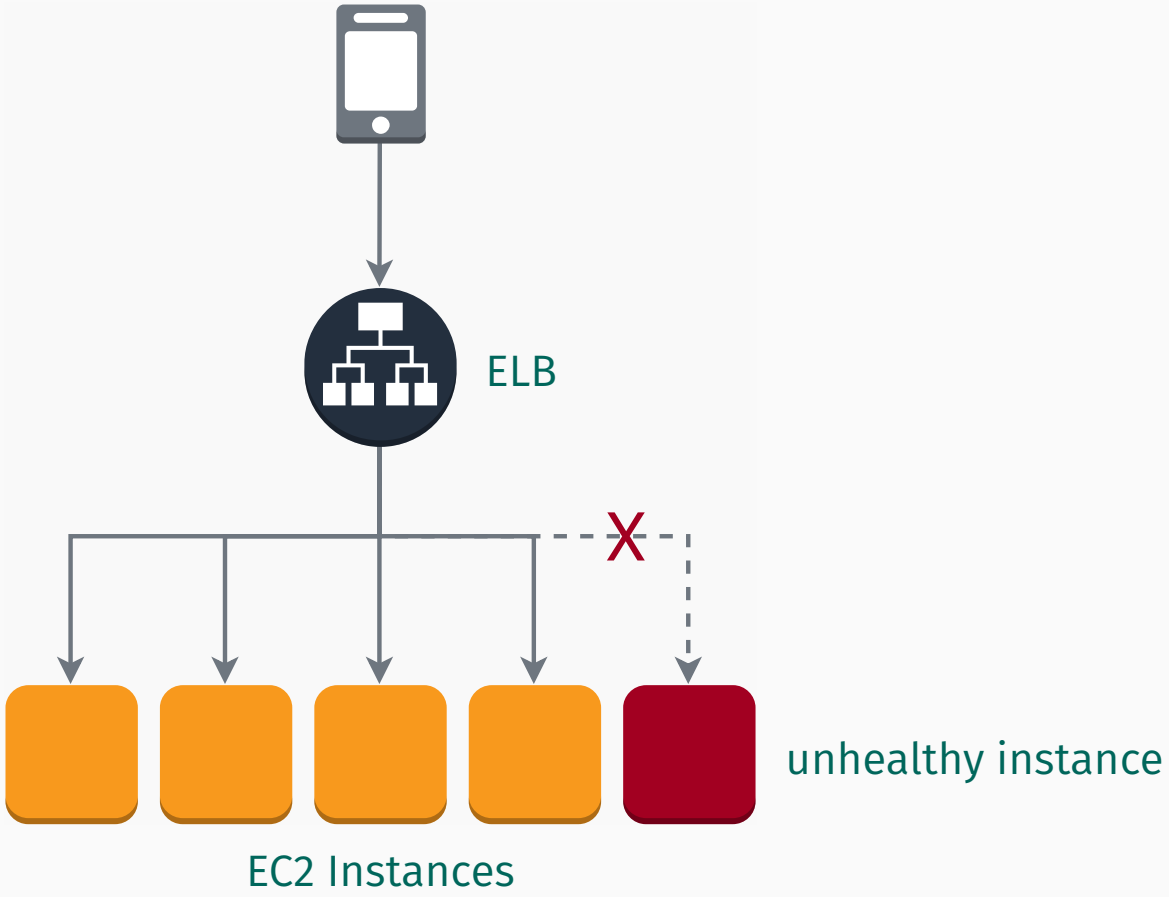
# Amazon Elastic Load Balancing (ELB)



# Amazon Elastic Load Balancing (ELB)



# Amazon Elastic Load Balancing (ELB)



# Amazon Web Services (AWS)

## Cloud storage

# Cloud storage products

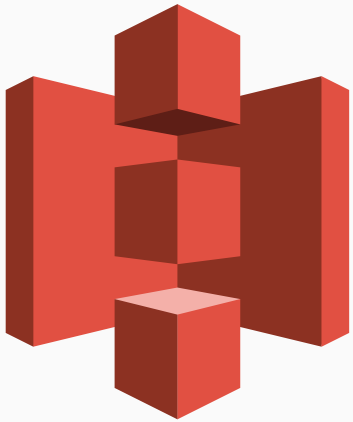
- Elastic Block Storage (EBS)
  - Persistent local storage for EC2 instances.

# Cloud storage products

- Elastic Block Storage (EBS)
  - Persistent local storage for EC2 instances.
- Elastic File System (EFS)
  - File system interface to share data between EC2 instances.

# Cloud storage products

- Elastic Block Storage (EBS)
  - Persistent local storage for EC2 instances.
- Elastic File System (EFS)
  - File system interface to share data between EC2 instances.
- Simple Storage Service (S3)

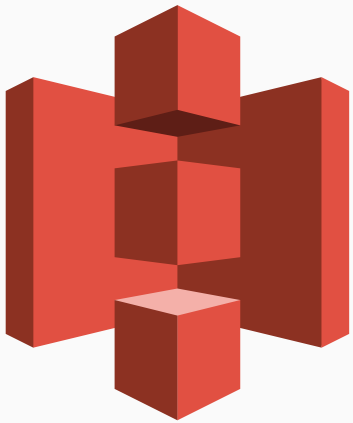


# Cloud storage products

- Elastic Block Storage (EBS)
  - Persistent local storage for EC2 instances.
- Elastic File System (EFS)
  - File system interface to share data between EC2 instances.
- Simple Storage Service (S3)
- Glacier
  - Durable and cheap long-term storage.

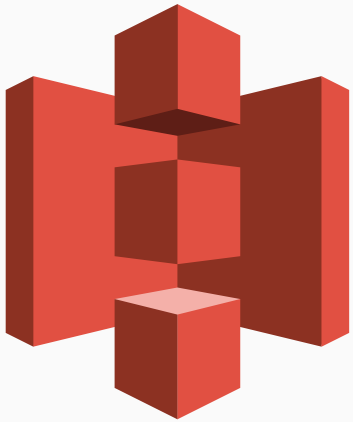
# Amazon Simple Storage Service (S3)

- *store and retrieve any amount of data from anywhere*



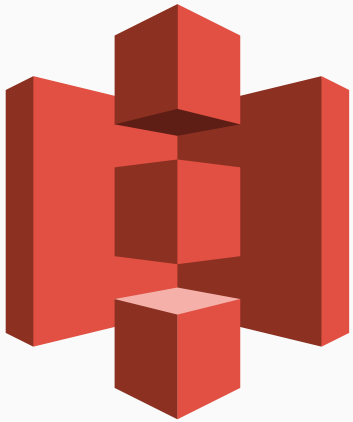
# Amazon Simple Storage Service (S3)

- *store and retrieve any amount of data from anywhere*
- 99.999999999% durability (nine nines!)



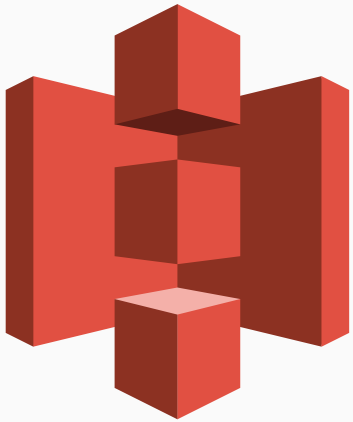
# Amazon Simple Storage Service (S3)

- *store and retrieve any amount of data from anywhere*
- 99.999999999% durability (nine nines!)
- Data is distributed across a *minimum* of three availability zones



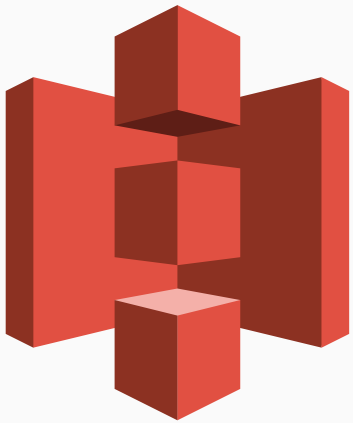
# Amazon Simple Storage Service (S3)

- *store and retrieve any amount of data from anywhere*
- 99.999999999% durability (nine nines!)
- Data is distributed across a *minimum* of three availability zones
- A logical unit of storage is a *bucket*



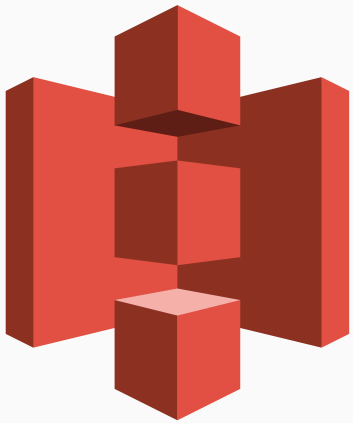
# Amazon Simple Storage Service (S3)

- *store and retrieve any amount of data from anywhere*
- 99.999999999% durability (nine nines!)
- Data is distributed across a *minimum* of three availability zones
- A logical unit of storage is a *bucket*
- Multiple storage classes



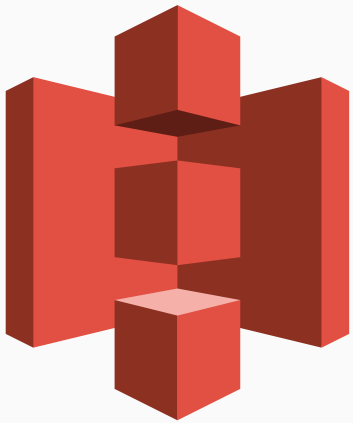
# Amazon Simple Storage Service (S3)

- *store and retrieve any amount of data from anywhere*
- 99.999999999% durability (nine nines!)
- Data is distributed across a *minimum* of three availability zones
- A logical unit of storage is a *bucket*
- Multiple storage classes
  - Standard



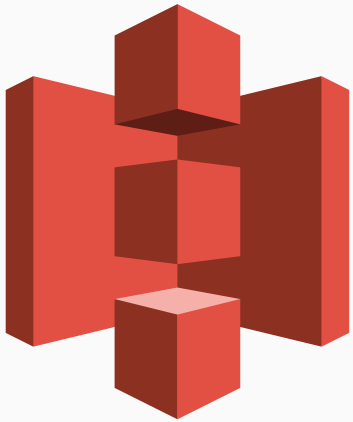
# Amazon Simple Storage Service (S3)

- *store and retrieve any amount of data from anywhere*
- 99.999999999% durability (nine nines!)
- Data is distributed across a *minimum* of three availability zones
- A logical unit of storage is a *bucket*
- Multiple storage classes
  - Standard
  - Infrequent Access



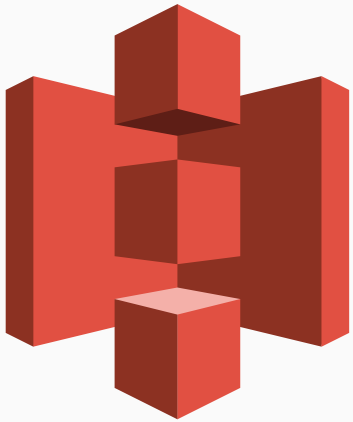
# Amazon Simple Storage Service (S3)

- *store and retrieve any amount of data from anywhere*
- 99.999999999% durability (nine nines!)
- Data is distributed across a *minimum* of three availability zones
- A logical unit of storage is a *bucket*
- Multiple storage classes
  - Standard
  - Infrequent Access
  - One zone-Infrequent Access



# Amazon Simple Storage Service (S3)

- *store and retrieve any amount of data from anywhere*
- 99.999999999% durability (nine nines!)
- Data is distributed across a *minimum* of three availability zones
- A logical unit of storage is a *bucket*
- Multiple storage classes
  - Standard
  - Infrequent Access
  - One zone-Infrequent Access
  - Amazon Glacier



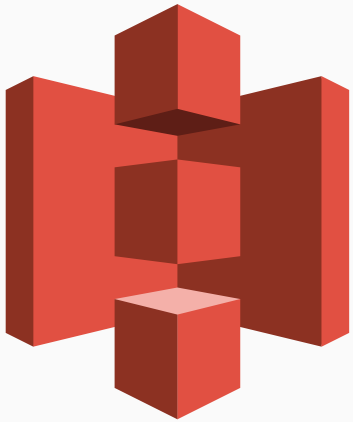
Multiple storage classes

| Storage class | Storage | Retrieval (per 1K req.) |
|---|---|---|
| Standard | $0.022 per GB | $0.0004 |
| Infrequent access | $0.0125 per GB | $0.001 |
| IA single zone | $0.01 per GB | $0.001 |

**Table 1:** S3 pricing (Ireland)

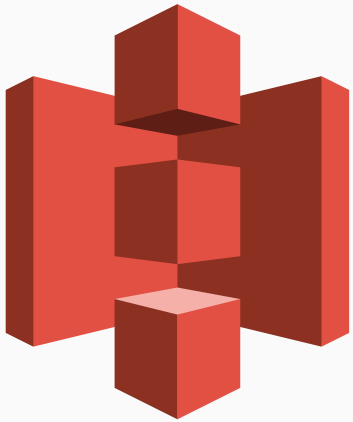
# Amazon Simple Storage Service (S3) - more

- Well-integrated with other services



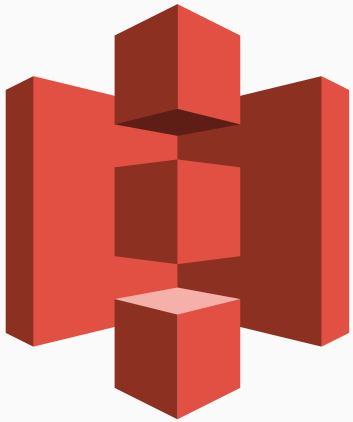
# Amazon Simple Storage Service (S3) - more

- Well-integrated with other services
  - Machine Learning



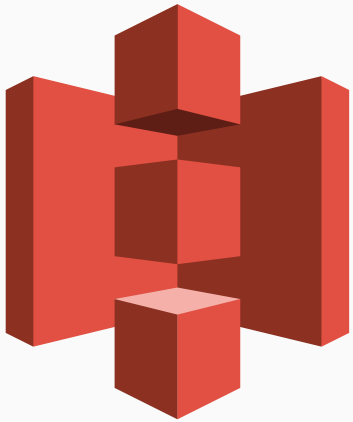
# Amazon Simple Storage Service (S3) - more

- Well-integrated with other services
  - Machine Learning
  - Big Data Analysis



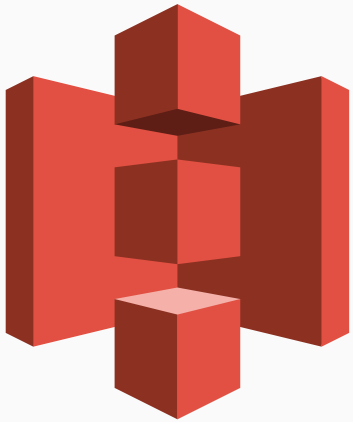
# Amazon Simple Storage Service (S3) - more

- Well-integrated with other services
  - Machine Learning
  - Big Data Analysis
- REST API



# Amazon Simple Storage Service (S3) - more

- Well-integrated with other services
  - Machine Learning
  - Big Data Analysis
- REST API
- Can be used to host static websites



# Amazon Web Services (AWS)

## Database services

# Relational Database Service (RDS)

- Set up, operate a relational database in the cloud.

# Relational Database Service (RDS)

- Set up, operate a relational database in the cloud.
- Takes care of backups, patching.

# Relational Database Service (RDS)

- Set up, operate a relational database in the cloud.
- Takes care of backups, patching.
- Supports:

# Relational Database Service (RDS)

- Set up, operate a relational database in the cloud.
- Takes care of backups, patching.
- Supports:
  - MySQL, PostgreSQL, MariaDB



# Relational Database Service (RDS)

- Set up, operate a relational database in the cloud.
- Takes care of backups, patching.
- Supports:
  - MySQL, PostgreSQL, MariaDB
  - Oracle, MS SQL Server


# Relational Database Service (RDS)

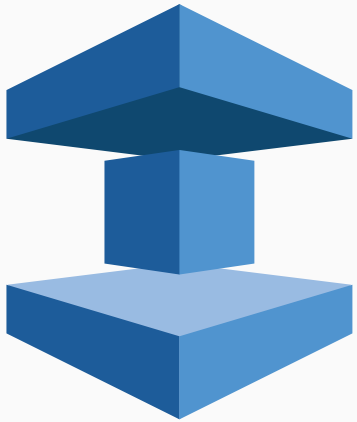
- Set up, operate a relational database in the cloud.
- Takes care of backups, patching.
- Supports:
  - MySQL, PostgreSQL, MariaDB
  - Oracle, MS SQL Server
  - Amazon Aurora

# Non Relational Database Services

- DynamoDB
  - *Fast and flexible NoSQL database service for any scale.*

# Non Relational Database Services

- DynamoDB
  - *Fast and flexible NoSQL database service for any scale.*
- ElastiCache
  - In memory data store.
  - Supports memcached, Redis



# Non Relational Database Services

- DynamoDB
  - *Fast and flexible NoSQL database service for any scale.*
- ElastiCache
  - In memory data store.
  - Supports memcached, Redis
- Neptune
  - Graph database service
  - Supports RDF, SPARQL, ...

# Amazon Web Services (AWS)

## Data Analysis

# Amazon Athena

- Run SQL-like queries on S3-stored data in seconds;

Amazon Athena

# Amazon Athena

- Run SQL-like queries on S3-stored data in seconds;
- Completely managed;

Amazon Athena

# Amazon Athena

- Run SQL-like queries on S3-stored data in seconds;
- Completely managed;
- You are charged for the number of bytes scanned per query, rounded up to the nearest megabyte, with a 10MB minimum per query. Scanning 1TB costs 5$.

Amazon Athena

# Amazon EMR (Elastic MapReduce)

- Easily Run and Scale Big Data Frameworks such as Apache Spark and Hadoop;

Amazon EMR

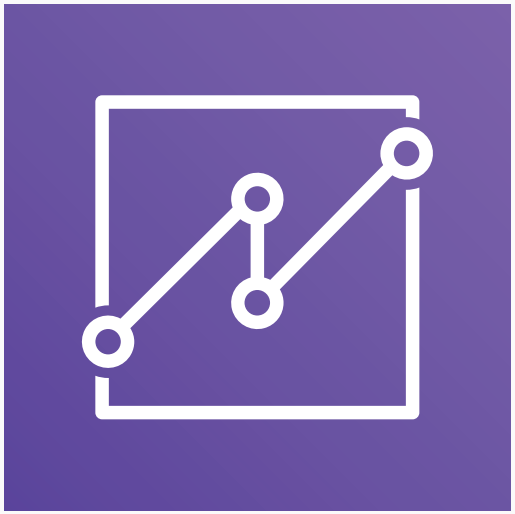
# Amazon EMR (Elastic MapReduce)

- Easily Run and Scale Big Data Frameworks such as Apache Spark and Hadoop;
- You pay a per-instance rate for every minute used;

Amazon EMR

# Amazon QuickSight

- Business Intelligence service that makes it easy to deliver insights to everyone;

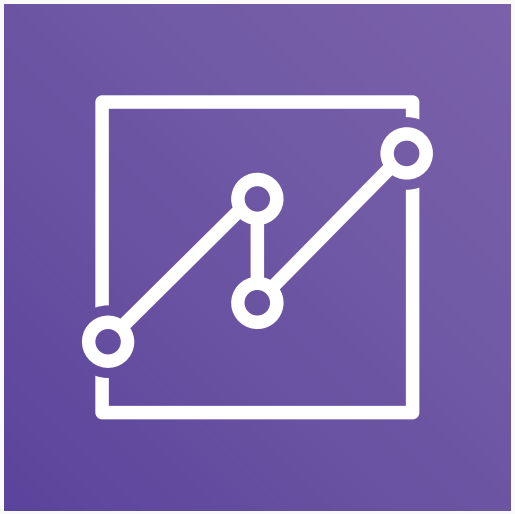


Amazon QuickSight

QuickSight overview (Youtube)

# Amazon QuickSight

- Business Intelligence service that makes it easy to deliver insights to everyone;
- Load data from anywhere;

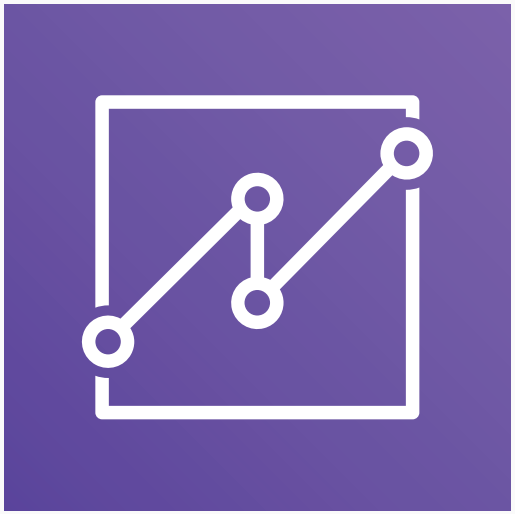


Amazon QuickSight

QuickSight overview (Youtube)

# Amazon QuickSight

- Business Intelligence service that makes it easy to deliver insights to everyone;
- Load data from anywhere;
- You pay a per-session rate (from 0,30$ per session up to 5$ a month);



Amazon QuickSight

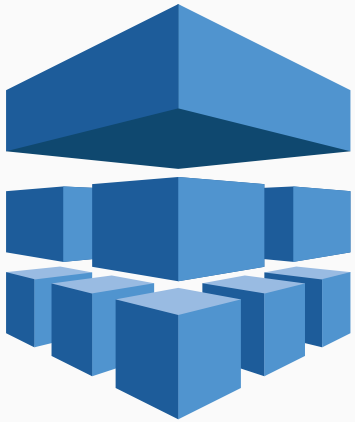
QuickSight overview (Youtube)

# Amazon Web Services (AWS)

## Machine Learning

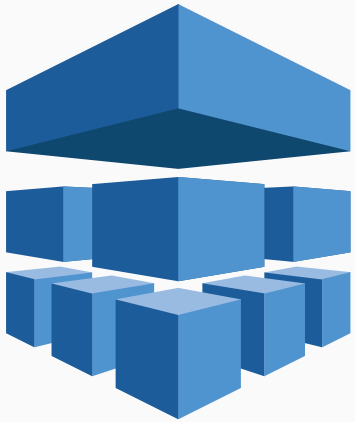
# Machine Learning: platform services

- Amazon SageMaker



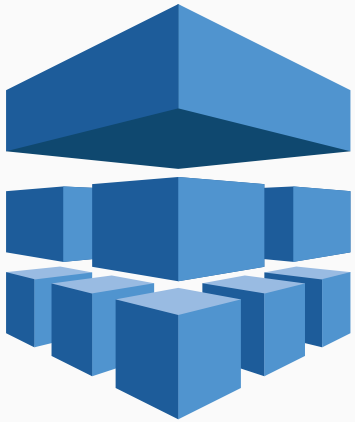
# Machine Learning: platform services

- Amazon SageMaker
  - Preconfigured for Tensorflow, MXNet...



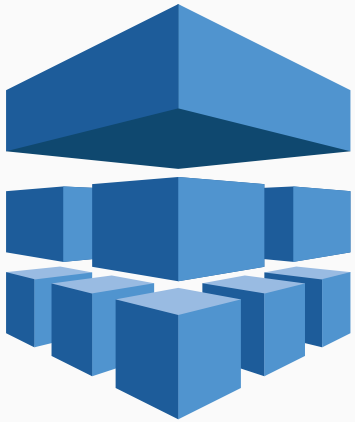
# Machine Learning: platform services

- Amazon SageMaker
  - Preconfigured for Tensorflow, MXNet...
  - Build, Train and Deploy phases



# Machine Learning: platform services

- Amazon SageMaker
  - Preconfigured for Tensorflow, MXNet...
  - Build, Train and Deploy phases
  - Pay based on build time, train time and hosting time

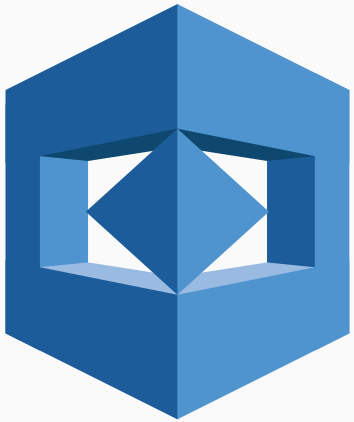


# Machine Learning: application services

- Comprehend (for NLP) Dashboard

# Machine Learning: application services

- Comprehend (for NLP) Dashboard
- Rekognition (Visual Analysis) Dashboard



# Machine Learning: application services

- Comprehend (for NLP) Dashboard
- Rekognition (Visual Analysis) Dashboard
- Translate

# Machine Learning: application services

- Comprehend (for NLP) Dashboard
- Rekognition (Visual Analysis) Dashboard
- Translate
- Polly (text-to-speech)

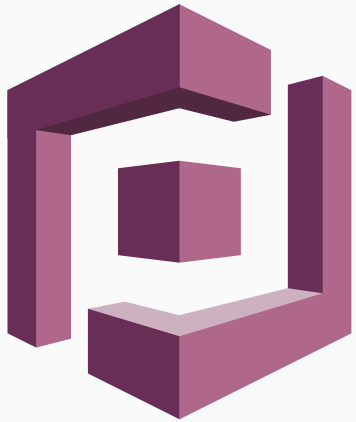
# Machine Learning: application services

- Comprehend (for NLP) Dashboard
- Rekognition (Visual Analysis) Dashboard
- Translate
- Polly (text-to-speech)
- Transcribe (speech-to-text)

# Amazon Web Services (AWS)

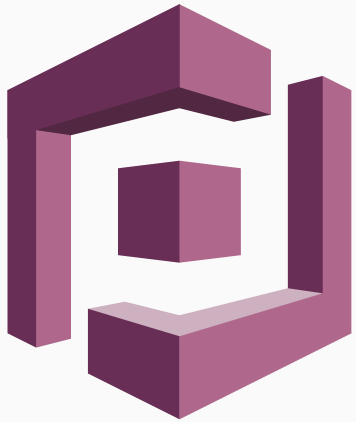
## Miscellanea

- Cognito



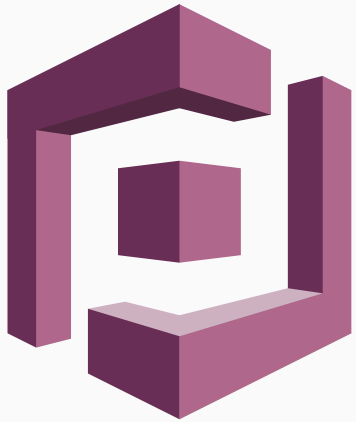
# Miscellanea

- Cognito
  - Sign-up and authentication



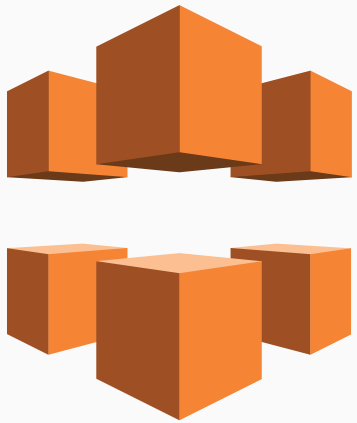
# Miscellanea

- Cognito
  - Sign-up and authentication
  - Federated identities



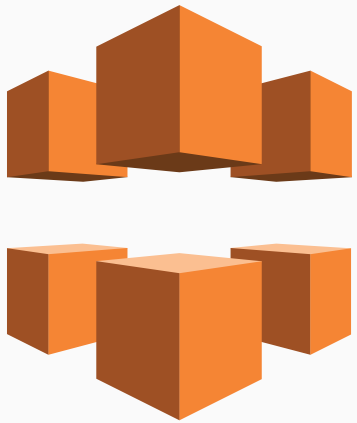
# Miscellanea

- Cognito
  - Sign-up and authentication
  - Federated identities
- CloudFront



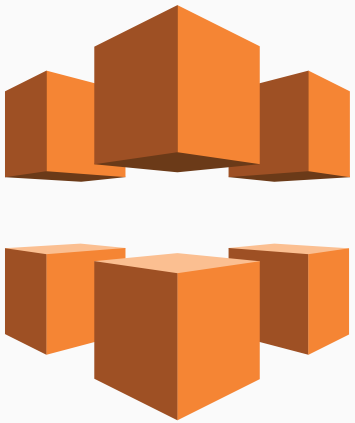
# Miscellanea

- Cognito
  - Sign-up and authentication
  - Federated identities
- CloudFront
  - Content Delivery Network



# Miscellanea

- Cognito
  - Sign-up and authentication
  - Federated identities
- CloudFront
  - Content Delivery Network
  - 116 Points of Presence in 56 cities across 24 countries



# Practice time!

# Scenario

Your **DATA** (Data Analytics for Transport Agencies) service is ready for deployment on the Cloud (using AWS), and your first client is eager to start using it!

# Scenario: details

The system you want to deploy consists in the following components:

- internet-connected boxes, equipped with the necessary sensors, which are installed in every vehicle the client wants to monitor and periodically send data to your application;
- a Relational Database (PostgreSQL);
- a web service which receives data from the vehicles, performs ETL operations, archives raw data and inserts new records in the Relational Database;
- a web application which allows your client to perform analytical queries on the relational database;

# Scenario: details

Your first client wants to manage a fleet of 100 vehicles. Each vehicle is active for about 10 hours a day and, when active, sends 1KB of data every minute to your application.

# Tasks

**Task 1** Think of a Cloud infrastructure for **DATA** using AWS. Which services would you use? Is there any weak spot in your architecture?

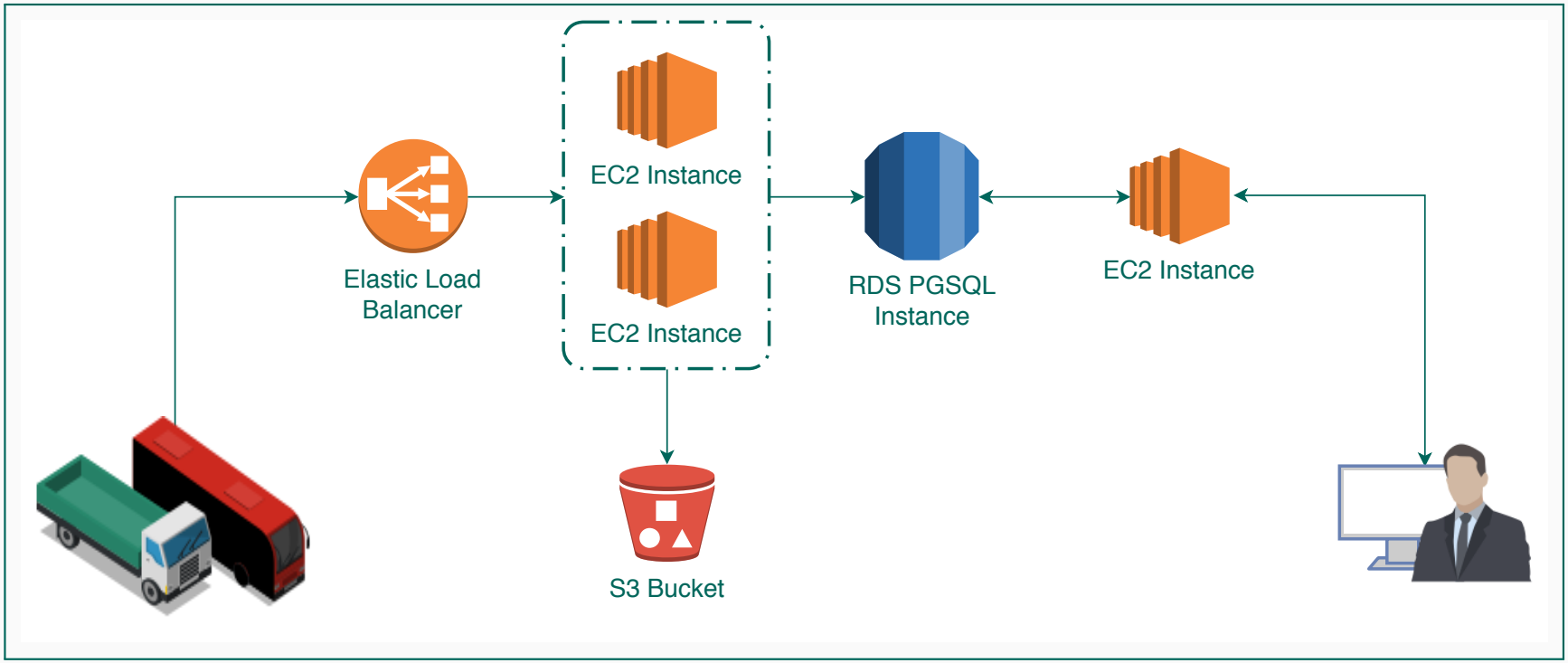
**Task 2** Use AWS Simple Monthly Calculator web to estimate the monthly bill for your application.

# Get the slides

You can download these slides from:

- `http://tiny.cc/cloud-aws-slides`
- `http://bit.do/cloud-aws-slides`

# Proposed Architecture



# Proposed Architecture: costs

One vehicle sends in a day 1KB$\times$ 60$\times$ 10 = 600KB of data, therefore 100 vehicles send about 60MB of data every day. In one month we collect about 1,8GB of data. In the proposed solution the web service runs on two `t3.micro` (2 vCPU, 1GB memory) EC2 instances, while the web applications runs in a single `t3.small` (2 vCPU, 2GB memory) instance. The database runs on a `db.m5.large` (2 vCPU, 8GB memory) RDS PostgreSQL instance with 240GB SSD storage. A Load Balancer distributes incoming data among the two web service instances, and the raw data is saved in S3 buckets.

The estimated monthly bill is

# Proposed Architecture: costs

One vehicle sends in a day 1KB$\times$ 60$\times$ 10 = 600KB of data, therefore 100 vehicles send about 60MB of data every day. In one month we collect about 1,8GB of data. In the proposed solution the web service runs on two `t3.micro` (2 vCPU, 1GB memory) EC2 instances, while the web applications runs in a single `t3.small` (2 vCPU, 2GB memory) instance. The database runs on a `db.m5.large` (2 vCPU, 8GB memory) RDS PostgreSQL instance with 240GB SSD storage. A Load Balancer distributes incoming data among the two web service instances, and the raw data is saved in S3 buckets.

The estimated monthly bill is .

# Proposed Architecture: costs

One vehicle sends in a day 1KB$\times$ 60$\times$ 10 = 600KB of data, therefore 100 vehicles send about 60MB of data every day. In one month we collect about 1,8GB of data. In the proposed solution the web service runs on two `t3.micro` (2 vCPU, 1GB memory) EC2 instances, while the web applications runs in a single `t3.small` (2 vCPU, 2GB memory) instance. The database runs on a `db.m5.large` (2 vCPU, 8GB memory) RDS PostgreSQL instance with 240GB SSD storage. A Load Balancer distributes incoming data among the two web service instances, and the raw data is saved in S3 buckets.

The estimated monthly bill is . .

# Proposed Architecture: costs

One vehicle sends in a day 1KB$\times$ 60$\times$ 10 = 600KB of data, therefore 100 vehicles send about 60MB of data every day. In one month we collect about 1,8GB of data. In the proposed solution the web service runs on two `t3.micro` (2 vCPU, 1GB memory) EC2 instances, while the web applications runs in a single `t3.small` (2 vCPU, 2GB memory) instance. The database runs on a `db.m5.large` (2 vCPU, 8GB memory) RDS PostgreSQL instance with 240GB SSD storage. A Load Balancer distributes incoming data among the two web service instances, and the raw data is saved in S3 buckets.

The estimated monthly bill is . . .

# Proposed Architecture: costs

One vehicle sends in a day 1KB$\times$ 60$\times$ 10 = 600KB of data, therefore 100 vehicles send about 60MB of data every day. In one month we collect about 1,8GB of data. In the proposed solution the web service runs on two `t3.micro` (2 vCPU, 1GB memory) EC2 instances, while the web applications runs in a single `t3.small` (2 vCPU, 2GB memory) instance. The database runs on a `db.m5.large` (2 vCPU, 8GB memory) RDS PostgreSQL instance with 240GB SSD storage. A Load Balancer distributes incoming data among the two web service instances, and the raw data is saved in S3 buckets.

The estimated monthly bill is 246,32 $ web

# References I


[Gar17] Gartner. *Gartner Forecasts Worldwide Public Cloud Services Revenue to Reach $260 Billion in 2017*. Oct. 12, 2017. URL: https://www.gartner.com/newsroom/id/3815165 (visited on 03/30/2018).

[Syn18] Synergy Research Group. *Cloud Growth Rate Increases; Amazon, Microsoft & Google all Gain Market Share*. Feb. 2, 2018. URL: https://www.srgresearch.com/articles/cloud-growth-rate-increases-amazon-microsoft-google-all-gain-market-share (visited on 03/30/2018).