Cloud Computing

Containers and Virtual Machines

Charlotte Laclau

Télécom Saint-Etienne

Outline of today's course

1. Quizz : do you remember something ?
2. Course Part 1 : Containers vs. Virtual Machines
3. Course Part 2 : Data security, Infrastucture Management and Service Level Agreement
4. Lab 1 follow-up

Do you remember something ?

What is Cloud Computing?

1. A software package
2. A spreadsheet
3. A way to store data remotely
4. A technique for managing computer applications in companies
5. An OLAP cube

Do you remember something ?

What is Cloud Computing?

1. A software package
2. A spreadsheet
3. A way to store data remotely
4. A technique for managing computer applications in companies
5. An OLAP cube

Do you remember something ?

What are the real advantages of Cloud Computing?

1. Strong security
2. Outsourcing of the IT department
3. Cost reduction
4. Easy access
5. Infinite capacity

Do you remember something ?

What are the real advantages of Cloud Computing?

1. Strong security
2. Outsourcing of the IT department
3. Cost reduction
4. Easy access
5. Infinite capacity

Do you remember something ?

What are the disadvantages of Cloud Computing?

1. A limited offer
2. Security breaches
3. Strong security
4. Dependence on its supplier
5. The price of subscriptions
6. The cost of managing the cloud’s IT infrastructure

Do you remember something ?

What are the disadvantages of Cloud Computing?

1. A limited offer
2. Security breaches
3. Strong security
4. Dependence on its supplier
5. The price of subscriptions
6. The cost of managing the cloud's IT infrastructure

Do you remember something ?

What is a public cloud?

1. A cloud formation that can be seen across the globe
2. A cloud service that can only be accessed from a publicly shared computer
3. A multi-tenant cloud environment accessed over the internet
4. A cloud environment owned, operated and controlled by a public company

Do you remember something ?

What is a public cloud?

1. A cloud formation that can be seen across the globe
2. A cloud service that can only be accessed from a publicly shared computer
3. A multi-tenant cloud environment accessed over the internet
4. A cloud environment owned, operated and controlled by a public company

Do you remember something ?

Which of the following statements could be used to describe a private cloud deployment?

1. A cloud environment maintained within an enterprise's own data center
2. A private environment within a third-party or public cloud provider's architecture
3. A cloud service inaccessible to anyone but the cultural elite
4. Both #1 and #2

Do you remember something ?

Which of the following statements could be used to describe a private cloud deployment?

1. A cloud environment maintained within an enterprise's own data center
2. A private environment within a third-party or public cloud provider's architecture
3. A cloud service inaccessible to anyone but the cultural elite
4. Both #1 and #2

Do you remember something ?

True or false: Open source tools can be used to manage private and public clouds.

1. FALSE
2. TRUE

Do you remember something ?

True or false: Open source tools can be used to manage private and public clouds.

1. FALSE
2. TRUE

Do you remember something ?

Amazon Elastic Compute Cloud (EC2) is primarily considered which type of cloud computing model?

1. Software as a service (SaaS)
2. Platform as a service (PaaS)
3. Infrastructure as a service (IaaS)
4. None of the above

Do you remember something ?

Amazon Elastic Compute Cloud (EC2) is primarily considered which type of cloud computing model?

1. Software as a service (SaaS)
2. Platform as a service (PaaS)
3. Infrastructure as a service (IaaS)
4. None of the above

Do you remember something ?

True or false: A multi-cloud strategy is the same as hybrid cloud.

1. TRUE
2. FALSE

Do you remember something ?

True or false: A multi-cloud strategy is the same as hybrid cloud.

1. TRUE
2. FALSE

Do you remember something ?

True or false: Enterprises can physically ship their data to some public cloud providers to upload that data into the cloud.

1. FALSE
2. TRUE

Do you remember something ?

True or false: Enterprises can physically ship their data to some public cloud providers to upload that data into the cloud.

1. FALSE
2. TRUE

Do you remember something ?

The Cloud Computing will end the existence of the direction of the computer systems in company if :

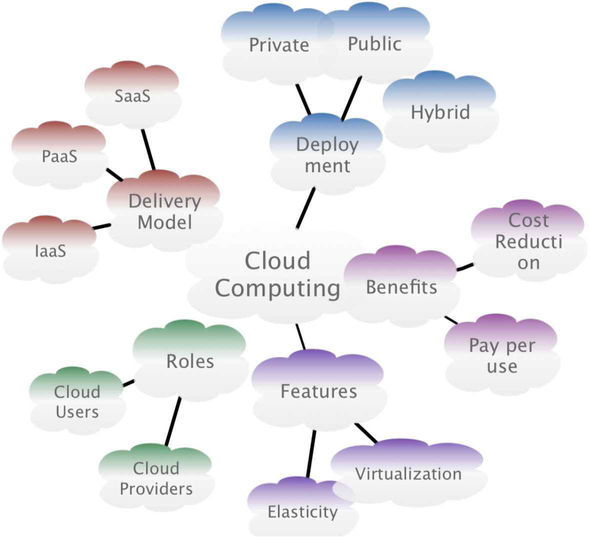
1. The Cloud computing corporation go public
2. The costs and the quality of the services are competitive
3. Confidentiality of the data is assured

Do you remember something ?

The Cloud Computing will end the existence of the direction of the computer systems in company if :

1. The Cloud computing corporation go public
2. The costs and the quality of the service are competitive
3. Confidentiality of the data is assured

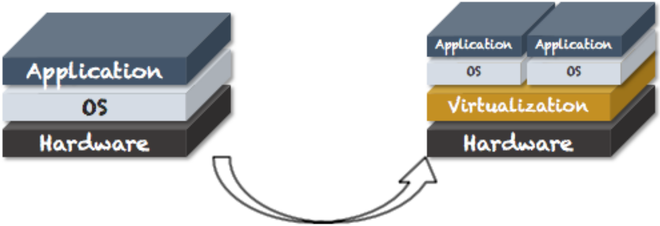
Overview


Private
Public
Hybrid
Deploy ment
SaaS
PaaS
Delivery Model
IaaS
Cloud Computing
Benefits
Cost Reducti on
Pay per use
Roles
Cloud Users
Cloud Providers
Features
Virtualization
Elasticity

Virtualization

Definition: running on the same physical machine, multiple systems as if they were running on separate physical machines

- a technology that transforms hardware into software
- allows to run multiple OS as **virtual machines**



Virtual Machines (VM's)

- **Virtual Machine** : an emulation of a real computer that executes programs like a real computer.

Virtual Machines (VM's)

- **Virtual Machine** : an emulation of a real computer that executes programs like a real computer.
- VM's run on top of a physical machine using a **hypervisor**

Virtual Machines (VM's)

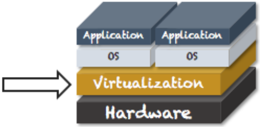
- **Virtual Machine** : an emulation of a real computer that executes programs like a real computer.
- VM's run on top of a physical machine using a **hypervisor**
- **Hypervisor** : a tool which control and distributes the hardware resources to each OS according to their needs

Virtual Machines (VM's)

- **Virtual Machine** : an emulation of a real computer that executes programs like a real computer.
- VM's run on top of a physical machine using a **hypervisor**
- **Hypervisor** : a tool which control and distributes the hardware resources to each OS according to their needs

Roles of a hypervisor

- provides control of the processor and the resources of the host machine
- allocate to each virtual machine the resources it needs
- make sure that VM's do not interfere with each other



Host Metal Hypervisor (type 2)

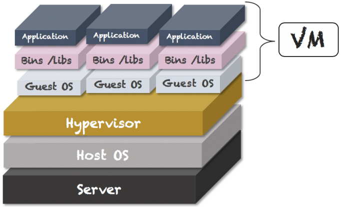
- A hosted virtualization hypervisor runs on the operating system of the host machine

Host Metal Hypervisor (type 2)

- A hosted virtualization hypervisor runs on the operating system of the host machine
- Example :
  - A computer running OSX can have a VM (e.g. VirtualBox or VMware Workstation 8) installed on top of that OS.
  - The VM doesn't have direct access to hardware, so it has to go through the host operating system (in our case, the Mac's OSX).

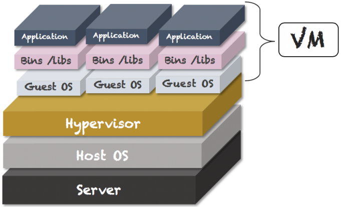
Host Metal Hypervisor (type 2)

- A hosted virtualization hypervisor runs on the operating system of the host machine
- Example :
  - A computer running OSX can have a VM (e.g. VirtualBox or VMware Workstation 8) installed on top of that OS.
  - The VM doesn't have direct access to hardware, so it has to go through the host operating system (in our case, the Mac's OSX).



Host Metal Hypervisor (type 2)

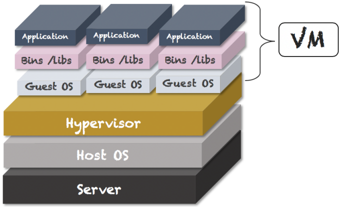
- A hosted virtualization hypervisor runs on the operating system of the host machine
- Example :
  - A computer running OSX can have a VM (e.g. VirtualBox or VMware Workstation 8) installed on top of that OS.
  - The VM doesn't have direct access to hardware, so it has to go through the host operating system (in our case, the Mac's OSX).



+ more hardware compatibility.

Host Metal Hypervisor (type 2)

- A hosted virtualization hypervisor runs on the operating system of the host machine
- Example :
  - A computer running OSX can have a VM (e.g. VirtualBox or VMware Workstation 8) installed on top of that OS.
  - The VM doesn't have direct access to hardware, so it has to go through the host operating system (in our case, the Mac's OSX).



+ more hardware compatibility.

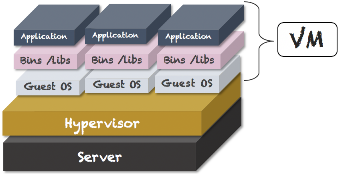
- lower performance of the VM

Bare Metal Hypervisor (type 1)

- A bare metal hypervisor is directly executed on the hardware.
  - Interfaces directly with the underlying hardware
  - Does not need a host OS to run on
  - The first thing installed on a host machine's server as the OS is the hypervisor

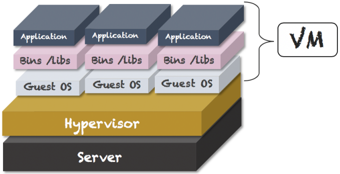
Bare Metal Hypervisor (type 1)

- A bare metal hypervisor is directly executed on the hardware.
    - Interfaces directly with the underlying hardware
    - Does not need a host OS to run on
    - The first thing installed on a host machine's server as the OS is the hypervisor



Bare Metal Hypervisor (type 1)

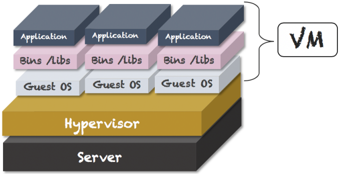
- A bare metal hypervisor is directly executed on the hardware.
  - Interfaces directly with the underlying hardware
  - Does not need a host OS to run on
  - The first thing installed on a host machine's server as the OS is the hypervisor



+ better performance, scalability, and stability

Bare Metal Hypervisor (type 1)

- A bare metal hypervisor is directly executed on the hardware.
  - Interfaces directly with the underlying hardware
  - Does not need a host OS to run on
  - The first thing installed on a host machine's server as the OS is the hypervisor

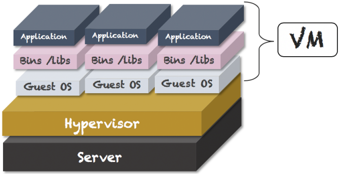


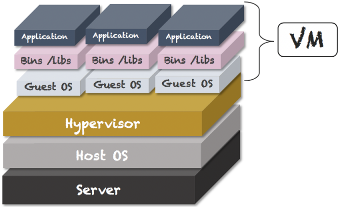
+ better performance, scalability, and stability

- hardware compatibility is limited

Virtual Machines and Hypervisor

Why do we need this additional hypervisor layer in between the VM and the host machine?





Examples of Bare-metal Virtualization Hypervisors (1)

VMware ESX and ESXi

- Belongs to Vmware
- Mature & Stable tool
- Free edition available with limited features
- 5 commercial editions
- Includes its own kernel

Examples of Bare-metal Virtualization Hypervisors (1)

VMware ESX and ESXi

- Belongs to Vmware
- Mature & Stable tool
- Free edition available with limited features
- 5 commercial editions
- Includes its own kernel

Microsoft Hyper-V

- Belongs to Microsoft
- Good for Small-Medium Businesses
- Good for running windows
- Free edition available with limited features
- 4 commercial editions

Examples of Bare-metal Virtualization Hypervisors (2)

Citrix XenServer

- Started as an open source project
- Mature & Stable tool
- Free edition available with limited features
- 2 commercial editions
- One of the cheapest

Examples of Bare-metal Virtualization Hypervisors (2)

Citrix XenServer

- Started as an open source project
- Mature & Stable tool
- Free edition available with limited features
- 2 commercial editions
- One of the cheapest

Oracle VM

- Belongs to Oracle Corporation
- Homegrown hypervisor tech. based on Xen
- Simple, no-frills hypervisor
- Lacks advanced features

Examples of Hosted Virtualization Hypervisors (1)

VMware Workstation/Fusion/Player

- Good for running multiple different OS or versions of one OS on one desktop
- Good for labs and demonstration objectives
- Good Option for running Windows and Linux on Macs

Examples of Hosted Virtualization Hypervisors (1)

VMware Workstation/Fusion/Player

- Good for running multiple different OS or versions of one OS on one desktop
- Good for labs and demonstration objectives
- Good Option for running Windows and Linux on Macs

Oracle VM VirtualBox

- Belongs to Oracle Corporation
- Open source hypervisor
- Similar to VMware and Microsoft Hyper-V
- Mature and Stable
- Suitable for Small-Medium Business & Enterprises

Examples of Hosted Virtualization Hypervisors (2)

KVM

- **Open Source**
- Belongs to Red Hat, Inc.
- Contains features of Type 1 & Type 2
- Compatible with numerous guest operating systems work with KVM

Examples of Hosted Virtualization Hypervisors (2)

KVM

- **Open Source**
- Belongs to Red Hat, Inc.
- Contains features of Type 1 & Type 2
- Compatible with numerous guest operating systems work with KVM

Parallels Desktop

- Belongs to Parallels
- Good for running Windows on OS
- 3 commercial editions

How should I choose a hypervisor ?

- Understand your needs, i.e., flexibility, scalability, reliable support etc.
- Understand the features, i.e., live migration, storage migration, dynamic memory, automated workflows, etc.
- Investigate the ecosystem
  - Possible to evaluate every virtualization hypervisor for free
- Compare costs
- **Remember** : VMs are portable and easily convertible

Containers

- A container provides operating-system-level virtualization

Containers

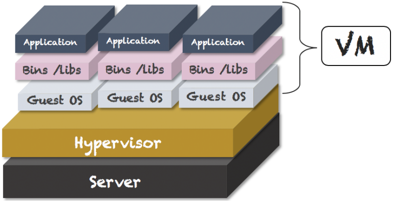
- A container provides operating-system-level virtualization
- Share similarities with a VM
  - they have private space for processing
  - can execute commands as root
  - have a private network interface and IP address
  - can mount file systems

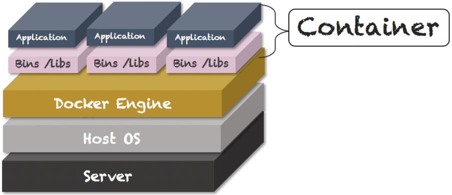
Containers

- A container provides operating-system-level virtualization

- Share similarities with a VM
  - they have private space for processing
  - can execute commands as root
  - have a private network interface and IP address
  - can mount file systems

- **Main difference** : containers **share** the host system's kernel with other containers





What is Docker ?

- An open-source project based on Linux containers.

What is Docker ?

- An open-source project based on Linux containers.
- Uses Linux Kernel features to create containers on top of an OS.

What is Docker ?

- An open-source project based on Linux containers.
- Uses Linux Kernel features to create containers on top of an OS.
- A gentle introduction

Why using Docker ?

Why using Docker ?

- **Ease of use**
  - can quickly build and test portable applications
  - allows anyone to package an application on their laptop
  - *build once, run anywhere*

Why using Docker ?

- **Ease of use**
  - can quickly build and test portable applications
  - allows anyone to package an application on their laptop
  - *build once, run anywhere*
- **Speed :** lightweight and fast
  - Take up fewer resources
  - No need to boot up a full virtual OS every time

Why using Docker ?

- **Ease of use**
  - can quickly build and test portable applications
  - allows anyone to package an application on their laptop
  - *build once, run anywhere*
- **Speed** : lightweight and fast
  - Take up fewer resources
  - No need to boot up a full virtual OS every time
- **Docker Hub** : an increasingly rich ecosystem
  - an *App store* for Docker images
  - public images created by the community

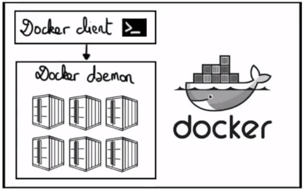
Why using Docker ?

- **Ease of use**
  - can quickly build and test portable applications
  - allows anyone to package an application on their laptop
  - *build once, run anywhere*
- **Speed** : lightweight and fast
  - Take up fewer resources
  - No need to boot up a full virtual OS every time
- **Docker Hub** : an increasingly rich ecosystem
  - an *App store* for Docker images
  - public images created by the community
- **Modularity and Scalability** :
  - Easy to link containers together to create an app
  - Easy to scale or update components independently in the future

Fundamental Concepts of Docker

Docker engine is the layer on which Docker runs.

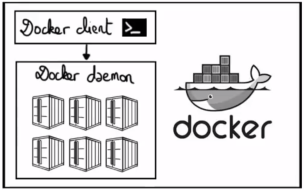
- A **Docker Daemon** that runs in the host computer.
- A **Docker Client** that then communicates with the Docker Daemon to execute commands.
- Can run on the same machine or on two different ones



Fundamental Concepts of Docker

Docker engine is the layer on which Docker runs.

- A **Docker Daemon** that runs in the host computer.
- A **Docker Client** that then communicates with the Docker Daemon to execute commands.
- Can run on the same machine or on two different ones

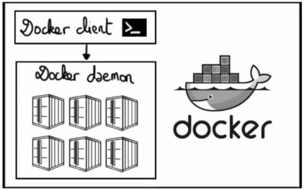


Dockerfile : where you write the instructions to build a Docker image.

Fundamental Concepts of Docker

Docker engine is the layer on which Docker runs.

- A **Docker Daemon** that runs in the host computer.
- A **Docker Client** that then communicates with the Docker Daemon to execute commands.
- Can run on the same machine or on two different ones



Dockerfile : where you write the instructions to build a Docker image.

Docker Image : read-only templates built from a set of instructions written in your Dockerfile.

What is Kubernetes?

- Kubernetes and Docker both help you run applications within linux containers

What is Kubernetes?

- Kubernetes and Docker both help you run applications within linux containers
- What if I want to run multiple containers across multiple machines ?
  - Need to start the right containers at the right time,
  - Figure out how they can talk to each other
  - Handle storage considerations
  - Deal with failed containers or hardware.

What is Kubernetes?

- Kubernetes and Docker both help you run applications within linux containers
- What if I want to run multiple containers across multiple machines ?
  - Need to start the right containers at the right time,
  - Figure out how they can talk to each other
  - Handle storage considerations
  - Deal with failed containers or hardware.
- **Kubernetes is an open source container orchestration platform**
  - Running containers across many different machines
  - Scaling up or down by adding or removing containers when demand changes
  - Keeping storage consistent with multiple instances of an application
  - Distributing load between the containers
  - Launching new containers on different machines if something fails

Summing-up

Benefits of virtualization

- For a company
  - Removal of special hardware and utility requirements
  - Effective management of resources
  - Better accessibility
  - Reduced risk of data loss
- For Data Centers
  - Maximization of server capabilities
  - Smaller footprint

Summing-up

Benefits of virtualization

- For a company
  - Removal of special hardware and utility requirements
  - Effective management of resources
  - Better accessibility
  - Reduced risk of data loss
- For Data Centers
  - Maximization of server capabilities
  - Smaller footprint

Connection to Cloud Computing

Virtualization in **cloud computing** allows you to run multiple applications and OS's on the same server, thereby providing for **efficient resource utilization** and reducing costs.

Deploying VM's and Containers with AWS

- VM's ? You already know how to do it !

Deploying VM's and Containers with AWS

- VM's ? You already know how to do it !
- Containers ? Can be broken down into 3 categories

Deploying VM's and Containers with AWS

- VM's ? You already know how to do it !
- Containers ? Can be broken down into 3 categories
  - Registry : AWS Elastic Container Registry (ECR)

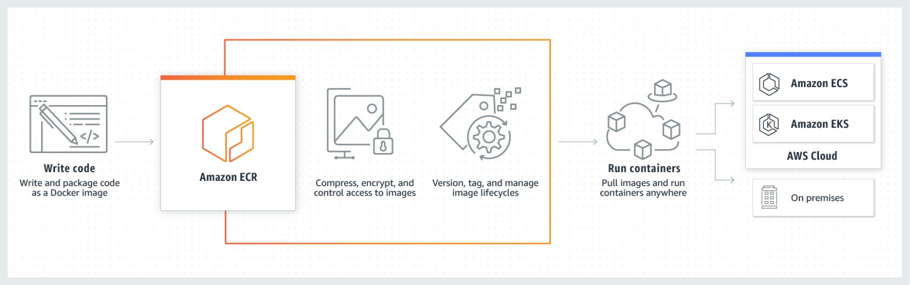
Deploying VM's and Containers with AWS

- VM's ? You already know how to do it !
- Containers ? Can be broken down into 3 categories
  - Registry : AWS Elastic Container Registry (ECR)
  - Orchestration: Amazon Elastic Container Service and Amazon Elastic Kubernetes Service

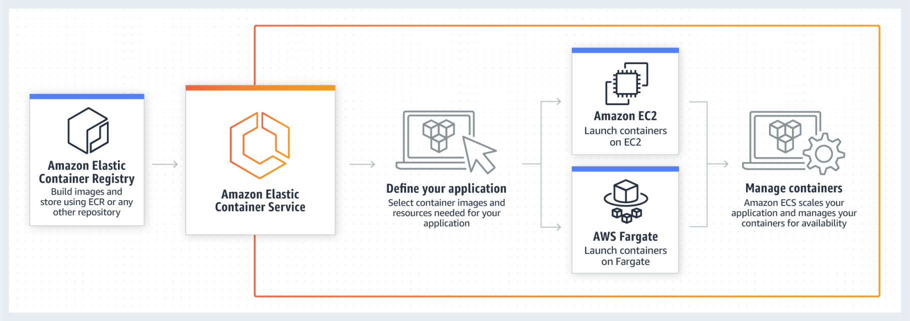
Deploying VM's and Containers with AWS

- VM's ? You already know how to do it !
- Containers ? Can be broken down into 3 categories
  - Registry : AWS Elastic Container Registry (ECR)
  - Orchestration: Amazon Elastic Container Service and Amazon Elastic Kubernetes Service
  - Compute: AWS Fargate and Amazon EC2

Registry: ECR


Write code
Write and package code as a Docker image
Amazon ECR
Compress, encrypt, and control access to images
Version, tag, and manage image lifecycles
Run containers
Pull images and run containers anywhere
Amazon ECS
Amazon EKS
AWS Cloud
On premises

Orchestration: ECS



Amazon Elastic Container Registry
Build images and store using ECR or any other repository

Amazon Elastic Container Service

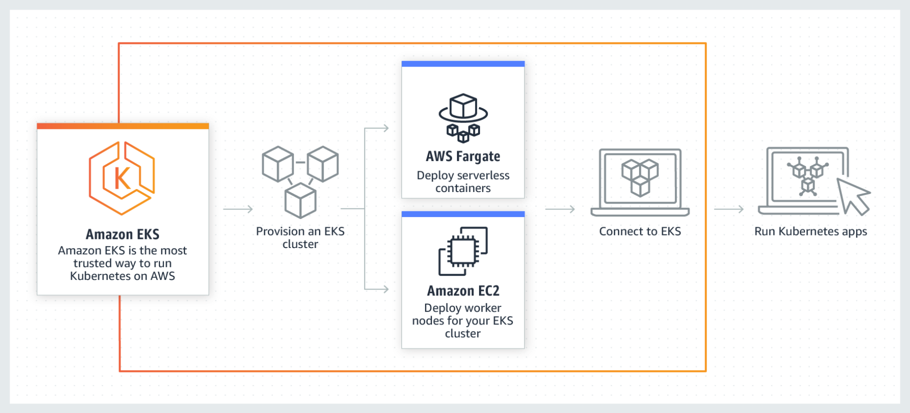
Define your application
Select container images and resources needed for your application

Amazon EC2
Launch containers on EC2

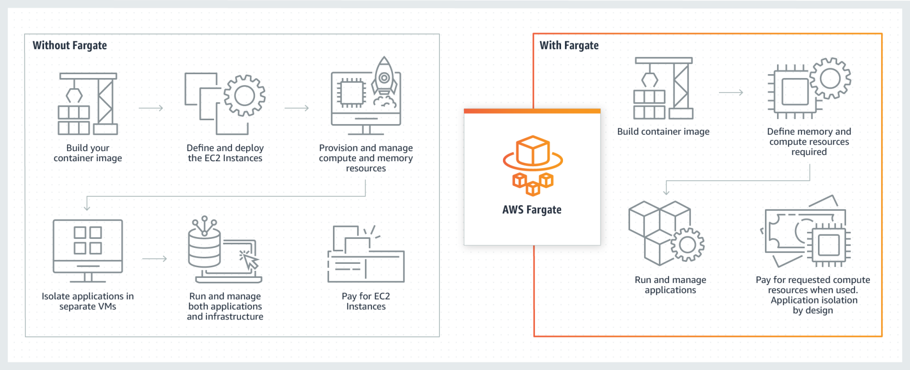
AWS Fargate
Launch containers on Fargate

Manage containers
Amazon ECS scales your application and manages your containers for availability

Orchestration: EKS


Amazon EKS
Amazon EKS is the most trusted way to run Kubernetes on AWS
Provision an EKS cluster
AWS Fargate
Deploy serverless containers
Amazon EC2
Deploy worker nodes for your EKS cluster
Connect to EKS
Run Kubernetes apps

Compute: EC2 / Fargate

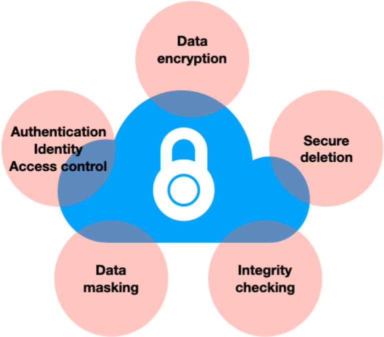

Without Fargate
Build your container image
Define and deploy the EC2 Instances
Provision and manage compute and memory resources
Isolate applications in separate VMs
Run and manage both applications and infrastructure
Pay for EC2 Instances
AWS Fargate
With Fargate
Build container image
Define memory and compute resources required
Run and manage applications
Pay for requested compute resources when used. Application isolation by design

Part 2

Data Security, Infrastructure Management and Service Level Agreement

Data Security in the cloud

- Similar to protecting data within a traditional data center



- Responsibility is shared between the provider and the client
- Cloud providers are providing you with the necessary tools to handle these questions

Privacy Protection

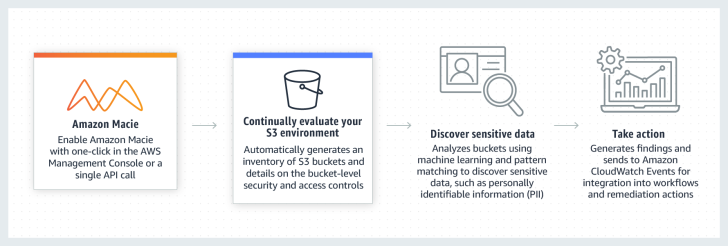
- Data should be protected from unauthorized access
  - Data encryption
  - Control who sees and can access what

Privacy Protection

- Data should be protected from unauthorized access
  - Data encryption
  - Control who sees and can access what
- Identify and classify your data

Privacy Protection

- Data should be protected from unauthorized access
  - Data encryption
  - Control who sees and can access what
- Identify and classify your data
- Amazon Macie ! → inventory of Amazon S3 buckets



Amazon Macie
Enable Amazon Macie with one-click in the AWS Management Console or a single API call

Continually evaluate your S3 environment
Automatically generates an inventory of S3 buckets and details on the bucket-level security and access controls

Discover sensitive data
Analyzes buckets using machine learning and pattern matching to discover sensitive data, such as personally identifiable information (PII)

Take action
Generates findings and sends to Amazon CloudWatch Events for integration into workflows and remediation actions

Data Integrity

- Protect data from unauthorized modification or deletion
- Challenging in the cloud and especially in a multicloud environment
- Authorization becomes crucial
  - Double authentication
  - Logging

Data Integrity

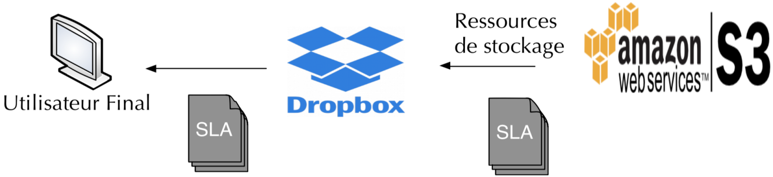
- Protect data from unauthorized modification or deletion
- Challenging in the cloud and especially in a multicloud environment
- Authorization becomes crucial
  - Double authentication
  - Logging
- AWS Key Management Service and EncryptionContext
  - Additional authenticated data (AAD)
  - Audit trail
  - Authorization context

Data Availability: Service Level Agreement (SLA)

- A Service Level Agreement is a document that defines the quality of service required between a provider and a client
- **Definition** : an explicit statement of expectations and obligations that exist in a business relationship between two organizations: the service provider and the customer
- There are several languages of SLA
  - WS-Agreement by OGF (Open Grid Forum)
  - WSLA by IBM
- An SLA is a document of a legal nature
  - if the SLA is not satisfied, there will be penalties to pay

SLA in the Cloud

- Often several SLAs because there are several actors involved
- Ideally, the SLA between a cloud provider and users must be negotiated.
- Unfortunately, often suppliers offer a single contract for everyone



Service Level Objective (SLO)

- A SLA is composed of a set of SLOs
- A SLO specifies the limits of the acceptable
  - availability$>$ 95%
  - response time t $<$5 millisecond
- A SLA Violation is Made When a SLO Is Not Respected
  - The penalty is often proportional to the violation
- Often SLOs are about non-functional requirements

Cloud Non-Functional Requirements

- **Availability:** an essential requirement, the service must be always present even in case of breakdown -
  - e.g. 99.99% during work days, 99.9% for nights/weekends)
- **Reliability: disaster Recovery expectations**
  - System operation must be consistent
  - Data must be stored in redundancy (e.g. RAID systems)
  - Multiple VMs must provide the same tasks
- **Performance**
  - Maximum response times
- **Location of the data**: consistent with local legislation
- **Portability of the data** (e.g. ability to move data to a different provider)

How to manage infrastructure state effectively ?

Visibility

- Do you know the state of your infrastructure ? (what, how, who)
- Can you detect misconfigurations or incident with your service?

How to manage infrastructure state effectively ?

Visibility

- Do you know the state of your infrastructure ? (what, how, who)
- Can you detect misconfigurations or incident with your service?

Automation

- Can you reconfigure your infrastructure to reproduce past configurations or scale up existing ones without a lot of extra manual work?
- Can you respond to incidents easily or automatically?

How to manage infrastructure state effectively ?

Visibility

- Do you know the state of your infrastructure ? (what, how, who)
- Can you detect misconfigurations or incident with your service?

Automation

- Can you reconfigure your infrastructure to reproduce past configurations or scale up existing ones without a lot of extra manual work?
- Can you respond to incidents easily or automatically?

Flexibility

- Can you improve your configurations and scale up easily ?
- Can you add more complexity using the same tools?
- Do you share, review, and improve your configurations within your team?

AWS Configuration Management

AWS Console

- Web interface to control most of the functionalities
- When should we use the AWS console ?
  - Read-only usage
  - Workable for small systems and teams
  - Rare operations
- ! AWS console is not compatible with automation, reproducibility and team communication (documentation and standardization)

AWS Configuration Management

AWS Console

- Web interface to control most of the functionalities
- When should we use the AWS console ?
  - Read-only usage
  - Workable for small systems and teams
  - Rare operations
- ! AWS console is not compatible with automation, reproducibility and team communication (documentation and standardization)

AWS Command-Line tools (CLI)

- Most basic way to save and automate AWS operations
- ++ well-maintained and covers a large proportion of all AWS services
- Write simple Bash scripts that invoke aws with specific arguments
- Interactive (not scripted) use : aws-shell

AWS Configuration Management

APIs and SDKs

- **SDKs** for using AWS APIs are available in most major languages, with Go, iOS, Java, JavaScript, Python, Ruby, and PHP [a]
- ! **Don't use APIs directly** -> use the SDKs for your preferred language to access APIs.

[a] (see https://github.com/donnemartin/awesome-aws)

Amazon SDK for Python : Boto3

- Allows to automate operations
- Contains a variety of APIs that operate at either a high level or a low level
- When should I use boto ? -> If you find yourself writing bash script with more than 2 CLI.

Visibility

General Visibility

- Tagging Essential practice to better understand the resource usage
- Through automation or convention, you can add tags
  - For the developer that “owns” that resource
  - For the product that resource supports
  - To label lifecycles, such as temporary resources
  - To distinguish resources with special security or compliance needs
  - To distinguish production-critical infrastructure

! Tags are case sensitive

Visibility

General Visibility

- Tagging Essential practice to better understand the resource usage
- Through automation or convention, you can add tags
  - For the developer that "owns" that resource
  - For the product that resource supports
  - To label lifecycles, such as temporary resources
  - To distinguish resources with special security or compliance needs
  - To distinguish production-critical infrastructure

! Tags are case sensitive

AWS Services for visibility

- Access management : IAM
- Monitoring resources : **Cloud Watch**
- Monitoring team : Cloud trail

CloudWatch

Monitors resources and **application**s, captures **logs**, and **sends events**.

- Standard service for **keeping tabs on AWS resources** :
  - **EC2** : percentage of allocated EC2 compute units, number CPU credits spent/earned by the instance etc.
  - **Bucket storage** : amount of data (in bytes), number of objects, number of HTTP requests made to bucket etc.

CloudWatch

Monitors resources and **application**s, captures **logs**, and **sends events**.

- Standard service for **keeping tabs on AWS resources** :
  - **EC2** : percentage of allocated EC2 compute units, number CPU credits spent/earned by the instance etc.
  - **Bucket storage** : amount of data (in bytes), number of objects, number of HTTP requests made to bucket etc.
- Allows to create time based graphs, alarms, and dashboards
  - Alarms to **trigger notifications** from any given metric.
  - Trigger SNS notifications, Auto Scaling actions, or EC2 actions.
  - Publish and share graphs of metrics by creating customizable dashboard views.

CloudWatch

Monitors resources and **application**s, captures **logs**, and **sends events**.

- Standard service for **keeping tabs on AWS resources** :
  - **EC2** : percentage of allocated EC2 compute units, number CPU credits spent/earned by the instance etc.
  - **Bucket storage** : amount of data (in bytes), number of objects, number of HTTP requests made to bucket etc.
- Allows to create time based graphs, alarms, and dashboards
  - Alarms to **trigger notifications** from any given metric.
  - Trigger SNS notifications, Auto Scaling actions, or EC2 actions.
  - Publish and share graphs of metrics by creating customizable dashboard views.

Limitations

- Limited to one metric for an alarm.
- Alarm's notification do not have any contextual detail;
- Data about metrics is kept in CloudWatch for 15 months.

CloudWatch

CloudWatch Events

- Mechanism to **automate actions** in various services on AWS.
- Can create **event rules** from instance states, AWS APIs, Auto Scaling, Run commands, deployments or time-based schedules.
- **Triggered events** can invoke Lambda functions, send messages, or perform instance actions.

CloudWatch

CloudWatch Events

- Mechanism to **automate actions** in various services on AWS.
- Can create **event rules** from instance states, AWS APIs, Auto Scaling, Run commands, deployments or time-based schedules.
- **Triggered events** can invoke Lambda functions, send messages, or perform instance actions.

CloudWatch Logs

- Streaming log storage system
- A log agent installed on your servers will process logs over time and send them to CloudWatch Logs.
- Can export logged data to S3 or stream results to other AWS services.
- CloudWatch Logs can be encrypted using keys managed through KMS.

AWS Billing and Cost Management

- AWS offers a **free tier of service**, that allows very limited usage of resources at no cost.
- **Pay your AWS bill**, **monitor** your usage and **budget** your costs. Provides features that you can use to
  - Estimate and plan your AWS costs
  - Receive alerts if your costs exceed a threshold that you set
  - Assess your biggest investments in AWS resources
- Set **billing alerts** to be notified of unexpected costs

Cost Explorer

- Tool that enables you to view and analyze your costs and usage.
- View data for up to the last 13 months
- Forecast for the next 3 months

– For large accounts, the AWS billing console can **time out** or be **too slow** to use.

Other tools for cost management

Tools

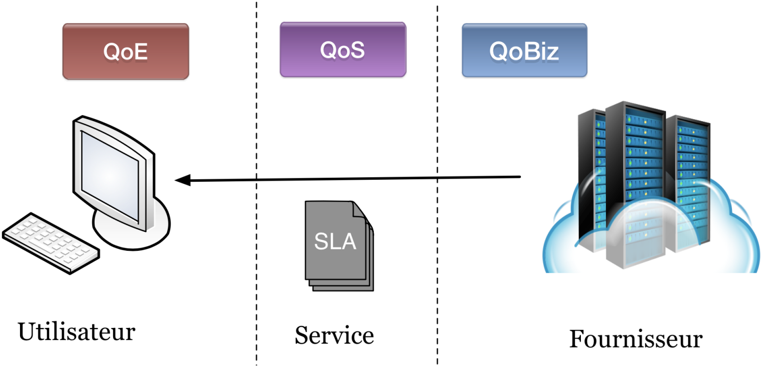
- Teevity Ice (Netflix) - Docker Ice
- Security Monkey and Cloud Custodian.
- AWS Simple Monthly Calculator to get an estimate of usage charges for AWS based on your actual usage of AWS services.

Third-Party Services

- Cloudability
- CloudHealth Technologies
- ParkMyCloud.
- ! Some of these charge a percentage of your bill

The Quality Management in the Cloud

- Three visions of the quality



Quality of Business (QoBiz)

- A provider-centric quality metric
- It expresses the profit of supplier
  - service price
  - revenue per user
  - revenue per transaction
  - budget
- A good QoBiz expresses the prosperity of the provider
- Decides the pricing policy of the suppliers: rate per hour, rate per month, etc.

Quality of Experience (QoE)

- According to subjective tests
- QoS is not sufficient to ensure user satisfaction
- Quality of experience (QoE) is a subjective metric that expresses quality of service as perceived by the user
- QoE is influenced by several factors:
    - The context of use
    - The personality and preferences of the user
    - The system used to consume the service