Cloud Computing

Luigi Santangelo
Department of Computer Engineering
University of Pavia
luigi.santangelo@unipv.it

What we will learn

Part 1:

- What cloud computing is
- Five main cloud computing elements
- Cloud computing from a service model perspective
- Cloud computing from a deployment model perspective
- Advantages and Disadvantages

Part 2:
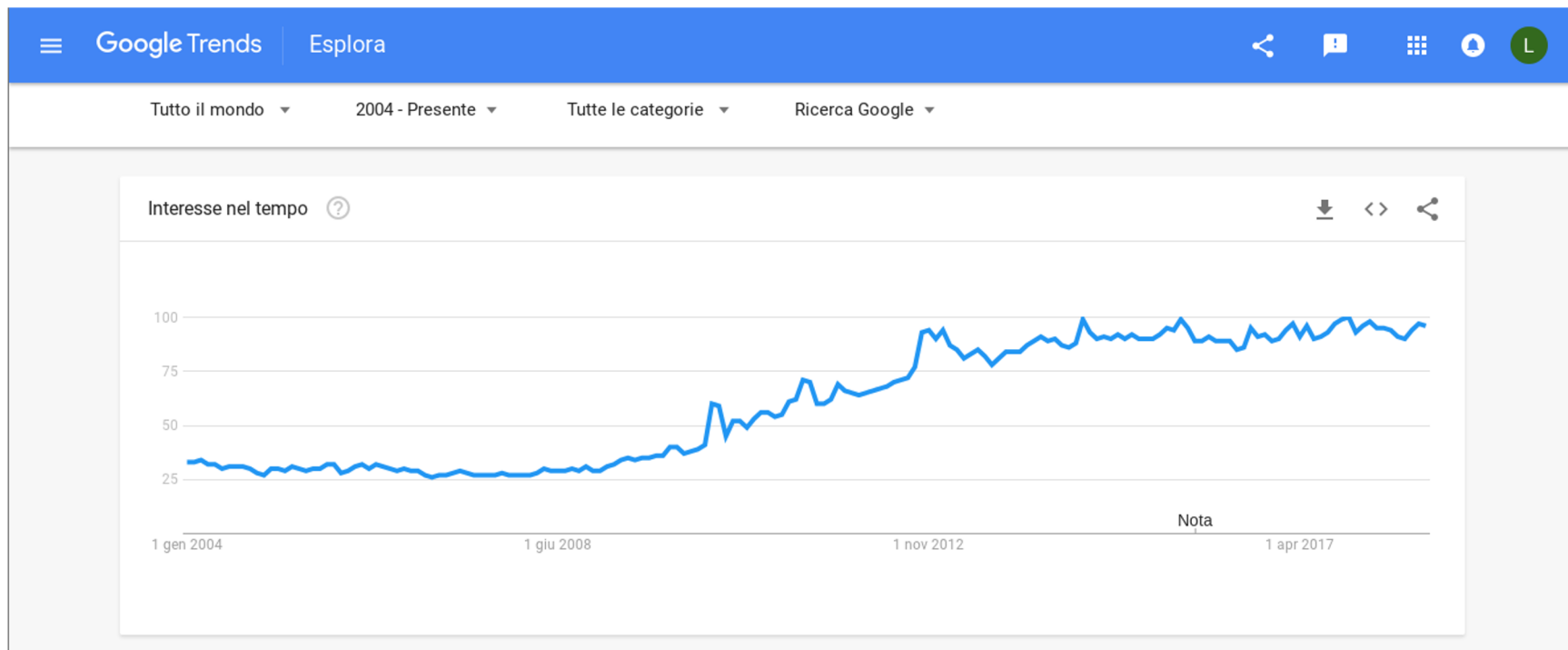
- What Google Cloud Platform is

Part3:

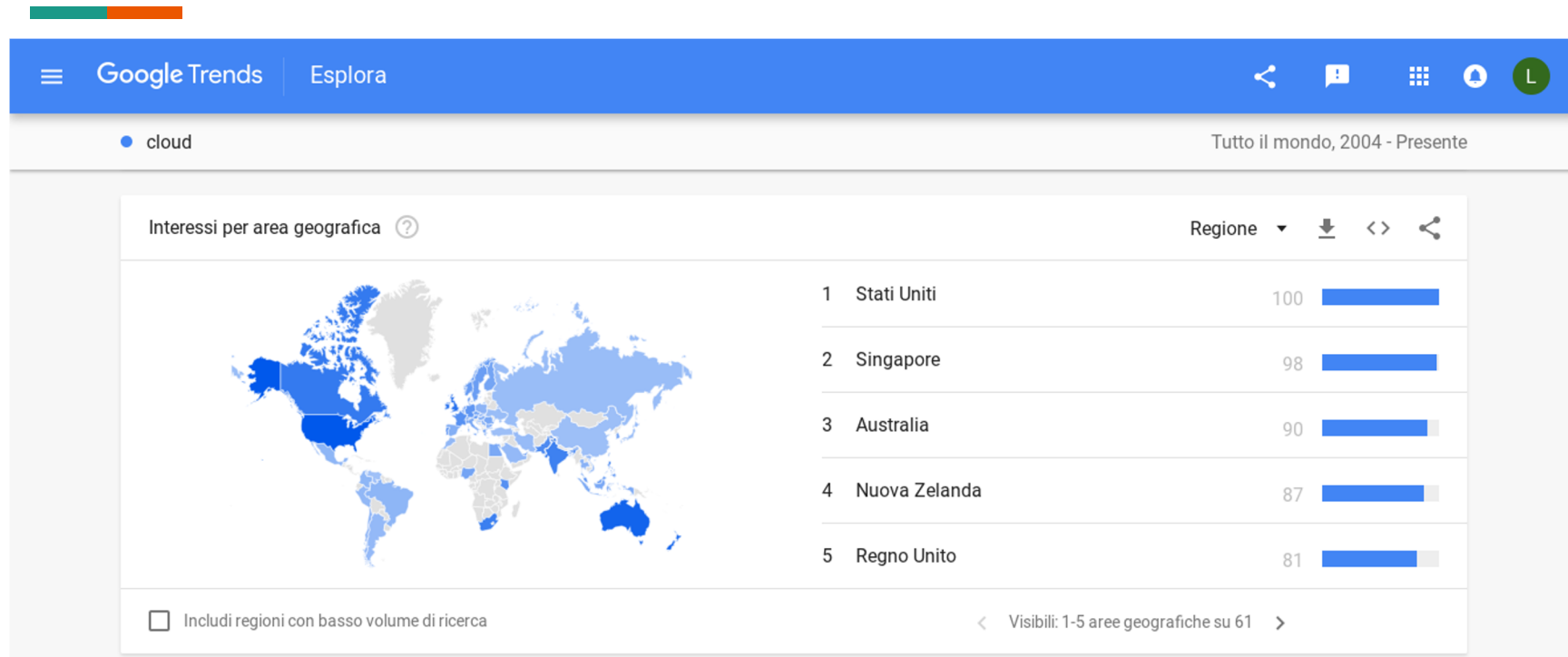- How to create a virtual instance using GCP

Part 1:
Cloud computing

How interest on Cloud is changed

How many times the word "Cloud Computing" has been searched in Google Search Engine by people all around the World from 2004 to 2018

How interest on Cloud is changed

Where the "Cloud Computing" has been searched the most from 2004 to 2018.

Italy is at the 31st place.

When was cloud born?

The term "Cloud" was seen for the first time in 2006, when Amazon Web Services used the word to describe its products.

Since that time, everybody started asking "What is Cloud Computing?"

What is cloud computing

National Institute of Standards and Technology* (2011):

"Cloud Computing is a model for enabling <u>on-demand</u> <u>network access</u> to a <u>shared pool</u> of <u>configurable resources</u> (networks, servers, storage, applications and services) that can be provided with <u>minimal interaction</u> from service provider"

NIST is a U.S. government entity that formally describes standards

Cloud Computing: five elements

1) **On-demand self-service**: a consumer, having a specific need, can get configurable resources (CPU time, network storage, software) in an automatic way, without any human interaction with the provider of these resources.

2) **Network access**: all resources are delivered over the network and can be used by different platform (mobile phones, laptops, PDAs)

Cloud Computing:
five elements

3) Resource pooling: all cloud resources are managed in shared pool and are delivered to the consumers depending on their demand. The physical resources become invisible but in the same time infinite to the consumer, who don't have neither control nor knowledge about where their data are stored or computed. The same physical resource can be shared among different consumers.

Cloud Computing: five elements

4) Measured Service: Cloud computing is based on <u>pay for use</u> model. Consumers pay for just they really use (for example processors by hours or storage by days). Providers make use of an appropriate mechanism to <u>measure the usage</u> of the resources for each individual consumer

Cloud Computing: five elements

5) Scalability and Elasticity: Consumers can also <u>scale-up and down</u> their cloud infrastructure whenever they want. This approach <u>reduces the upfront costs</u>. Indeed, consumers can start with a small resources configuration and scale-up accordingly with the <u>business growth</u>, increasing hardware resources only when there is an increasing in their needs.

Cloud Scalability and Elasticity: Virtualization

To achieve shrinking and growing resource configuration, <u>physical infrastructure has to be virtualized</u>.

Virtualization is not a new topic. Indeed all Operating Systems make use of Virtual Memory. By using <u>Virtual Memory</u>, a developer can run application requiring a quantity of RAM greater than the quantity really available on the system.

Cloud Scalability and Elasticity: Virtualization

There are at least 7 different ways to virtualize the IT infrastructure, depending on the level where the virtualization is applied:

- Full Virtualization
- Para-Virtualization
- Hardware-Level Virtualization
- Operating System-level Virtualization (also known as Container-Based Virtualization)
- Application Virtualization
- Storage Virtualization
- Network Virtualization

Full Virtualization
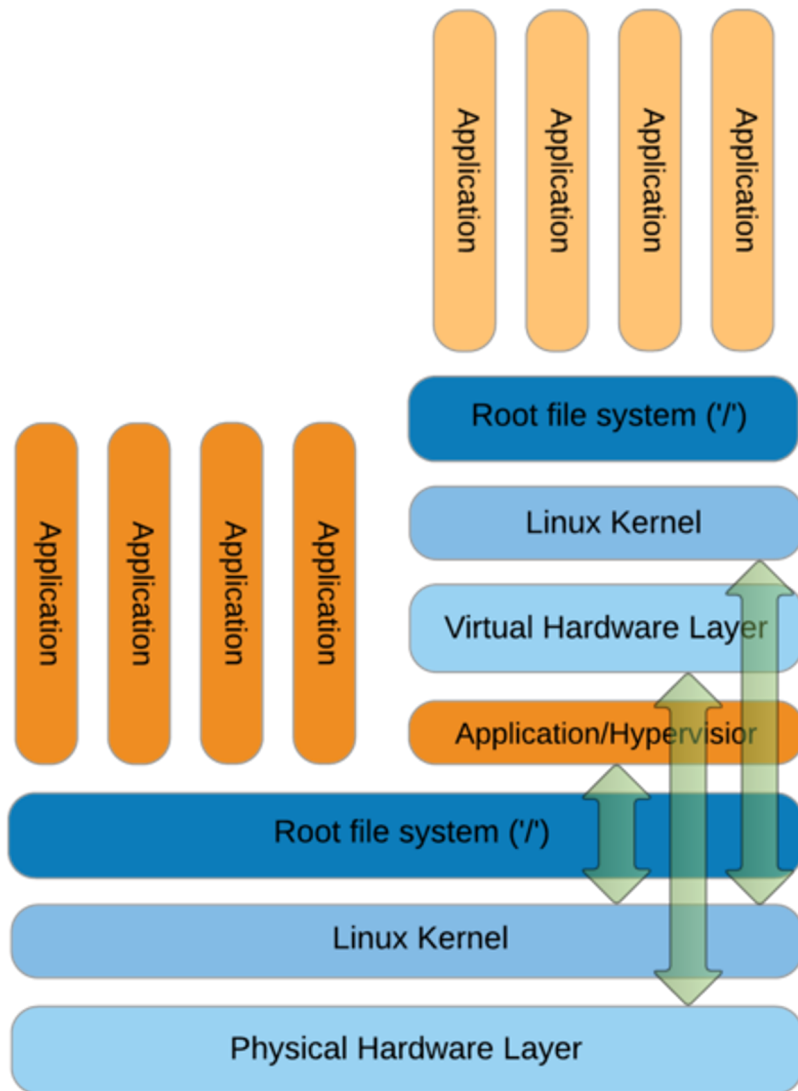
The Hypervisor hides the real OS and creates a virtual hardware layer where the guest OS runs on.

The guest OS is <u>unaware</u> that it is working on a virtual hardware layer.

All interactions among guest OS and hardware are intercepted by the Hypervisor and translated for the physical hardware

Examples: VirtualBox, VMware, Parallels

ParaVirtualization
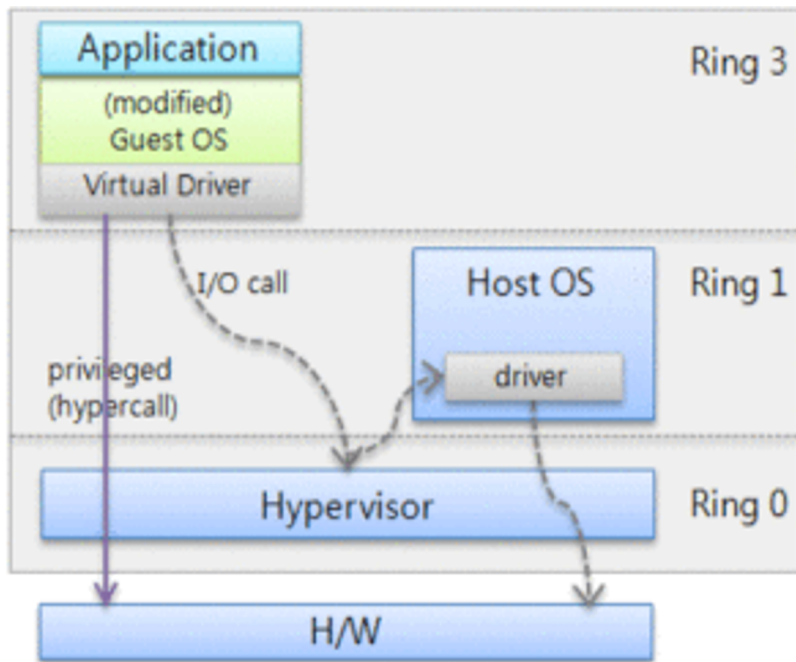
The Guest OS is <u>aware</u> that it is working an a Virtualized Environment, so it can directly talk with the Hypervisor by using hyper call (instead of the system call) and for privileged actions, guest OS can directly access to the hardware layer. In this approach, guest OS need to be modified. Unmodified OS (Win2K and XP) cannot be executed on ParaVirtualization environment.

Examples: VMware

Hardware Level Virtualization
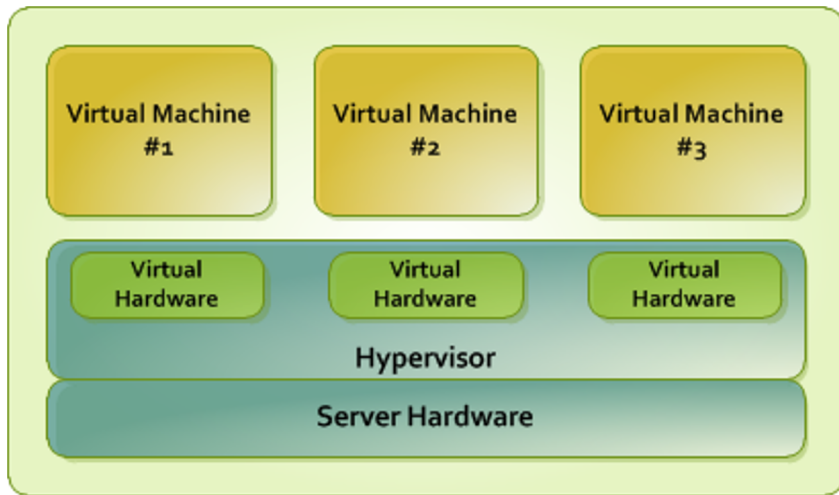
It is quite similar to Full Virtualization but here Hypervisor works directly on the Physical Hardware instead of working on the hosting Operating System

Examples: Xen

Operating System Level Virtualization
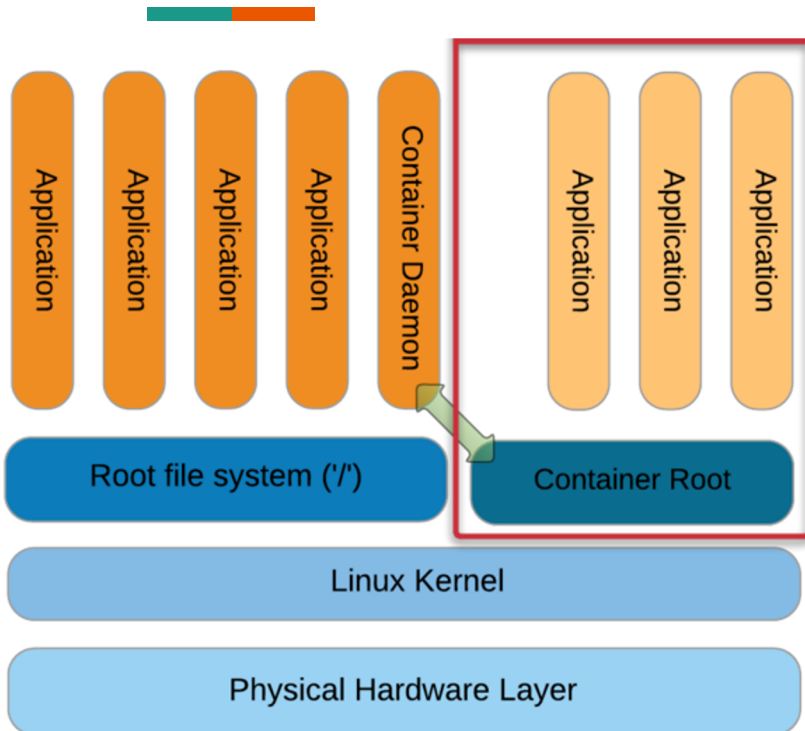
All applications work inside a container which interacts with the Container Daemon running on the hosting OS.

In this approach, the virtualization layer runs on the Operating System

Containers are lighter than Virtual Machine

Examples: Docker, Singularity, Kubernetes

Application Level Virtualization
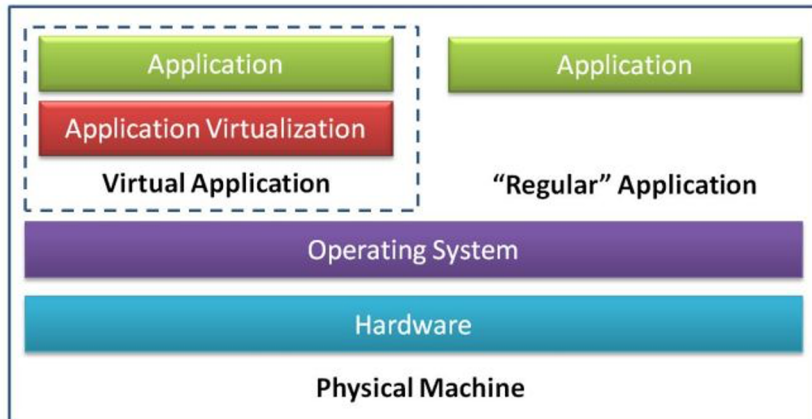
Each application works on its own isolated environment. Different application working on different environment don't interact and conflict. Usually, virtualized application are also accessible from the network.

Example: Citrix

Storage Virtualization

Traditional Disk Mapping — Virtualized Storage Disk Mapping — Virtual Volumes
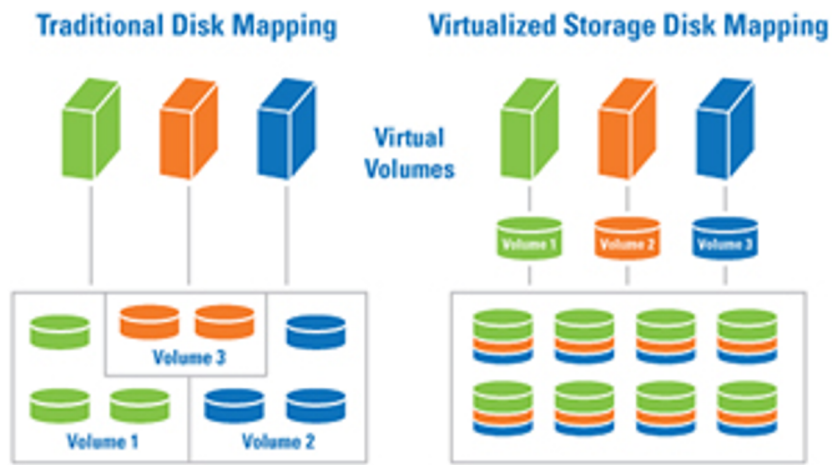
Instead of using different Physical Volumes, each server uses a Virtual Volume. Then, read and write operations are spreaded across all physical volumes. Multiple requests can be executed in parallel, boosting system performance.

Network Virtualization
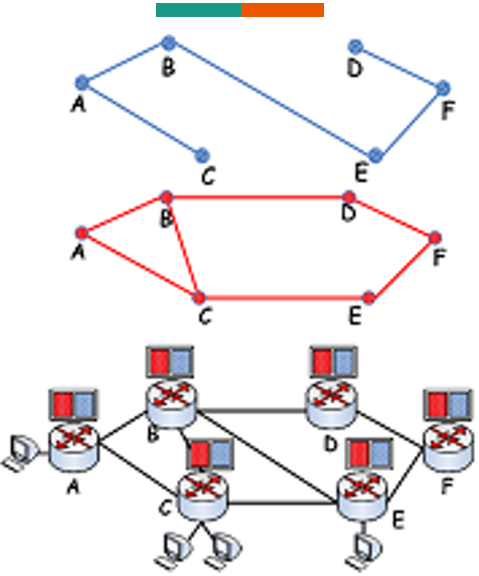
Blue Virtual Network

Red Virtual Network

Physical Network

Multiple subnetworks can be created in the same physical network. This can disable communication among unauthorized nodes belonging to different subnets.

Cloud Computing:
from a service model perspective

From the NIST* original definition of Cloud Computing

- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS)

NIST: National Institute of Standard and Technology

Cloud Computing:
from a service model perspective

As Internet has become more reliable and faster, Cloud Providers started to offer new technology as cloud services. All of this services are identified as Xaas (Anything as a Service)

- Data Storage as a Service (DaaS)
- Blockchain as a Service (BaaS)
- Business Process as a Service (PBaaS)
- DataBase as a Service (DBaaS)
- Function as a Service (FaaS)

Cloud Computing:
from a service model perspective

Software as a Service (SaaS): consumers <u>can just use</u> applications that are deployed on an hosting environment. The application can be <u>accessed through the network</u> using different clients (browser, PDA, etc). Consumers <u>don't have any control</u> on the infrastructure layer, such as network, servers, operating systems, storage. Sometime, just small application configuration settings can be modified.

Examples of SaaS include Google Gmail and <u>Google Docs</u>.

Cloud Computing:
from a service model perspective

Platform as a Service (PaaS): Consumer can use a cloud platform where applications can be <u>developed and deployed</u>. The platform supports the entire <u>software life-cycle</u> and provides programming languages, libraries, services and tools which can be used by developers.

No control on underlying infrastructure, such as network, servers, operating systems or storage, but just on the application deployed by the consumer.

Just a few application-hosting configuration can be set up.

An example of PaaS is <u>Google AppEngine</u>.

Cloud Computing:
from a service model perspective

Infrastructure as a Service (IaaS): Consumers can directly use cloud infrastructure such as processing, storage and network. Consumers can create and destroy all VM they need for deploy and running their own applications.

Consumers cannot manage the underlying cloud infrastructure but can have control over operating system, storage with a limited control over the networking component

An example of IaaS is Amazon EC2 or Google Compute Engine.

Cloud Computing:
from a service model perspective

Data Storage as a Service (DaaS): Consumers store their own data on the cloud. Typical storage interface (as file system) are used. Examples of DaaS include Google Drive.

Cloud Computing:
from a service model perspective

Blockchain as a Service (BaaS): Platforms that include tools for creating blockchain applications

Cloud Computing:
from a service model perspective

Business Process as a Service (BPaaS): Business processes are automated and provided by cloud infrastructure. The company can focus on its business objectives instead of managing technologies and services behind the business services. An example is the Human Resource functions, Procurement or Advertising.

Cloud Computing:
from a service model perspective

DataBase as a Service (DBaaS): Consumers can store and retrieve their data to and from a database, without setting up physical hardware or software. All administrative tasks are managed by the service provider.Examples of DBaaS are Google BigTable, Apache Hbase and Amazon S3.

Cloud Computing:
from a service model perspective

Function as a Service (FaaS): Customers can use a platform where application functionalities can be developed and run on, <u>without building the infrastructure</u> needed to develop and launch an app.

Customers pay depending on the number of times the function has been invoked or the time spent to compute.

This kind of model is usually described as **Serverless Computing**.

Cloud Computing: more about Serverless Computing

How the computer science engineers' work is changed across years.

Let's think about developing an application to validate a credit card:

- Traditional Enterprise Computing: buy a server, pay for the purchase, bring it on site, configure it, set up a network connection, build and deploy the application
- Cloud Computing: buy the virtual server, configure it, build and deploy the application, pay for the server usage
- Serverless Computing: build and deploy the application, pay for the function call

Cloud Computing:
more about Serverless Computing

An analogy to better describe Serverless Computing

- Traditional Enterprise Computing: buy a car to be used whenever
- Cloud Computing: rent a car for the day
- Serverless Computing: take a taxi for the rush

See, for example OpenLambda or IronFunctions

Cloud Computing:
from a deployment model perspective

- Public Cloud
- Private Cloud
- Community Cloud
- Hybrid Cloud

Cloud Computing: four deployment models

Public Cloud: is used by general public cloud consumers. The cloud infrastructure is provided by service providers. Popular public service providers are Amazon, Google, Microsoft

Cloud Computing: four deployment models

Private Cloud: the cloud infrastructure is used and managed by a single organization. This typically happens for different reasons:

- Optimize the utilization of existing in-house resources;
- Reduce the data transfer among local IT infrastructure and public cloud
- Security: data are not accessible by anyone

Example: CINECA (Bologna) infrastructure to deploy Italian Universities information systems for students career, courses and curricula degrees and employees management.

Cloud Computing: four deployment models

Community Cloud: Is the cloud made by different organizations joined together maintaining and sharing the same cloud infrastructure

Cloud Computing: four deployment models

Hybrid Cloud: is a combination of two or more clouds (private, community or public). Amazon has recently developed a new deployment model (Virtual Private Cloud) where the existing IT consumer infrastructure is connected to the Amazon Public Cloud.

Advantages

- Organization can focus on own business instead of spending time building and maintaining IT infrastructure
- No upfront costs
- The infrastructures grows accordingly with the business growth
- No planning for provisioning. The resources are immediately available.
- Reducing costs for purchasing electricity, location, software, hardware
- A private data center may not benefit from the economies of scale that make public clouds financially attractive

Disadvantages

- Downtime
  - outages can make cloud infrastructure inaccessible
  - when internet connection is down, cloud infrastructure is not accessible
- Security and Privacy: data stored in the cloud can be accessed by unauthorized people
- Vulnerability to attack: every component is accessible from Internet. Hackers can attack computing resources and steal data
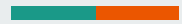
Part 2:
Google Cloud Platform

What Google Cloud Platform is

Google Cloud Platform (GCP) is a suite of cloud computing services offered by Google. The platform provides a series of modular cloud services including:

- **Compute Engine** (IaaS providing virtual machine)
- App Engine (PaaS for application hosting)
- BigQuery (SaaS for large scale database analytics)
- Cloud Datastore (DBaaS providing document-oriented database)
- Storage (IaaS providing online file and object storage)

Google Compute Engine

Google Compute Engine allows users creating Virtual Machine that can be executed on the physical Google infrastructure.

Different configuration of virtual machine can be created depending on the consumer's needs.

Billing is computed depending on the time a virtual machine is left running and the virtual machine configuration.

See https://cloud.google.com/compute/pricing for a detailed cost evaluation

Example of pricing (per hour)

| South Carolina ▾ | | | | Monthly 🔵 Hourly |

Machine type	Virtual CPUs	Memory	Price (USD)	Preemptible price (USD)
n1-standard-1	1	3.75GB	$0.0475	$0.0100
n1-standard-2	2	7.5GB	$0.0950	$0.0200
n1-standard-4	4	15GB	$0.1900	$0.0400
n1-standard-8	8	30GB	$0.3800	$0.0800
n1-standard-16	16	60GB	$0.7600	$0.1600
n1-standard-32	32	120GB	$1.5200	$0.3200
n1-standard-64	64	240GB	$3.0400	$0.6400
n1-standard-96 (Beta) Skylake Platform only	96	360GB	Not available in this region	Not available in this region

GCP Educational Grants

Google provides **free** credits for computer science courses in higher education. University of Pavia has joined with the program, getting (**limited**) credits that can be spent by students on any Google Cloud Platform Product.

For Advanced Computer Architecture, Compute Engine is the single product we will use.

The aim of this course is providing all knowledge about creating and running virtual machine, logging into it and developing, compiling and running an OpenMP application.
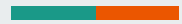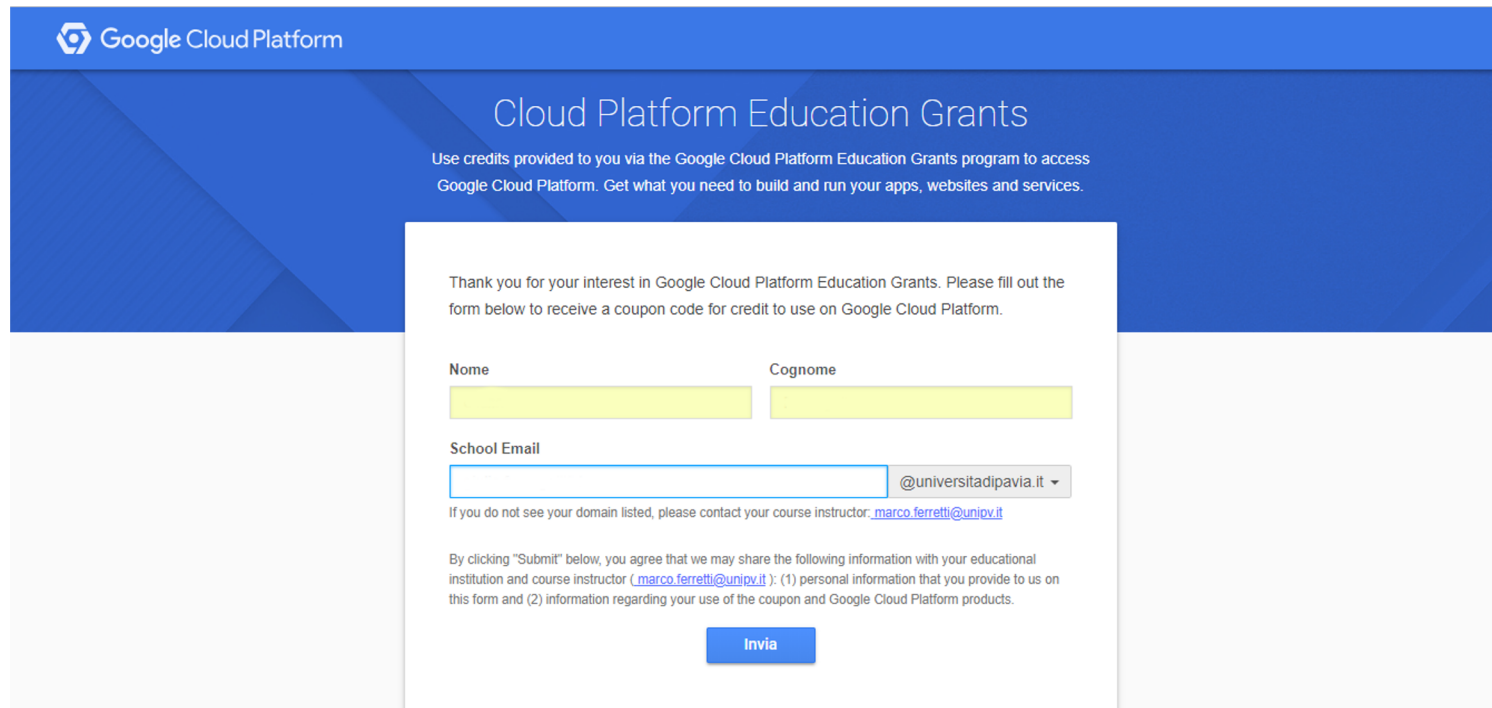
Part 3:
How to create a virtual machine on GCP

Step 1: log into your email account

- all of you have received a mail from the professor to redeem a coupon that offers free access (up to 50$)
- open http://mail.google.com
- log out from all personal accounts
- follow the link in the message and fill in the form
  - firstname.lastnameXX@universitadipavia.it
  - your password (to reset it write to: mailstudenti@unipv.it)
- open the received mail
- follow the link

Step 2: Join the project

- Fill the form with First Name, Last Name and Email address (@universitadipavia.it)
- Press Send button

Step 2: Join the project
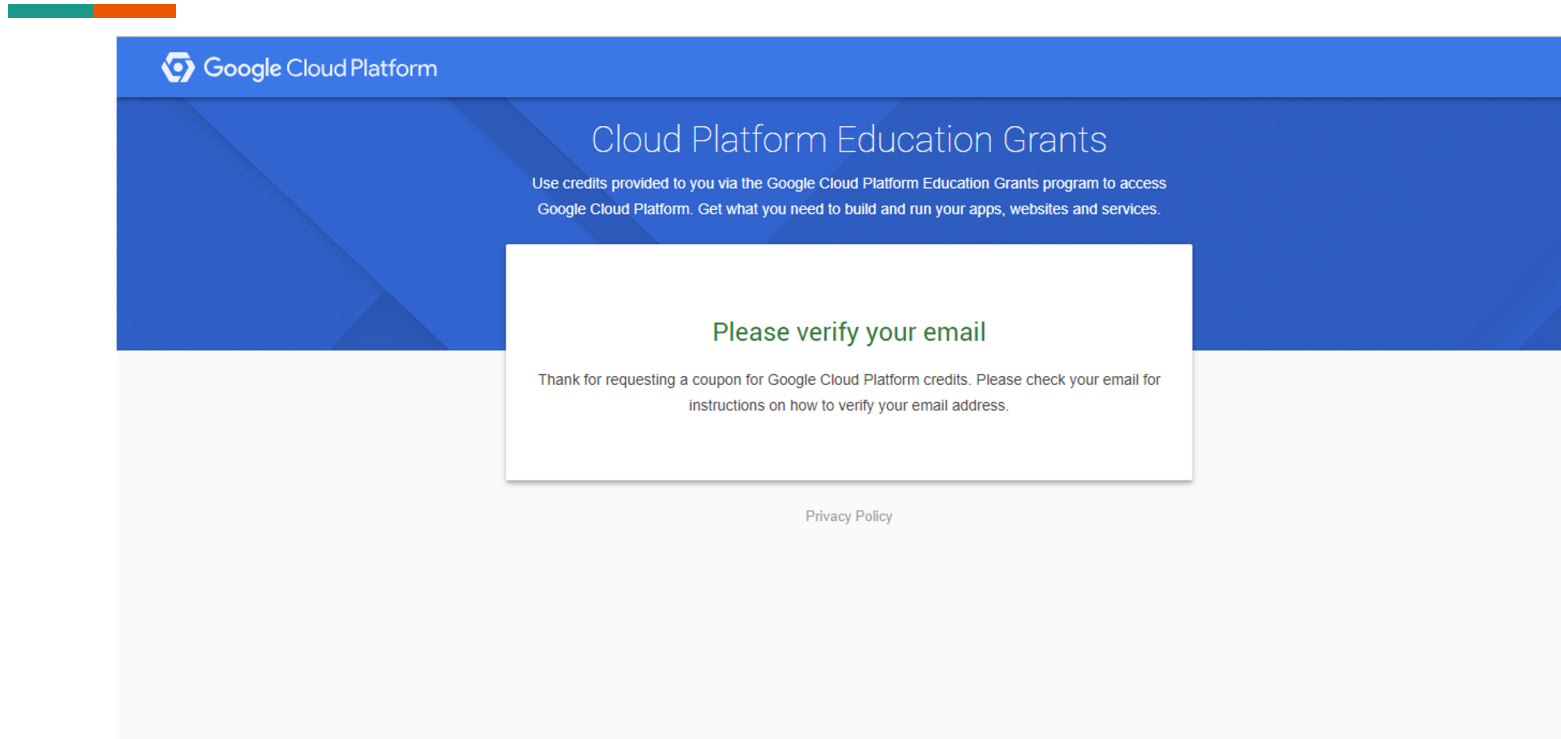
- Check for the incoming mail in your institutional mailbox

Step 2: Join the project

- Follow the link to verify your email address

Step 2: Join the project
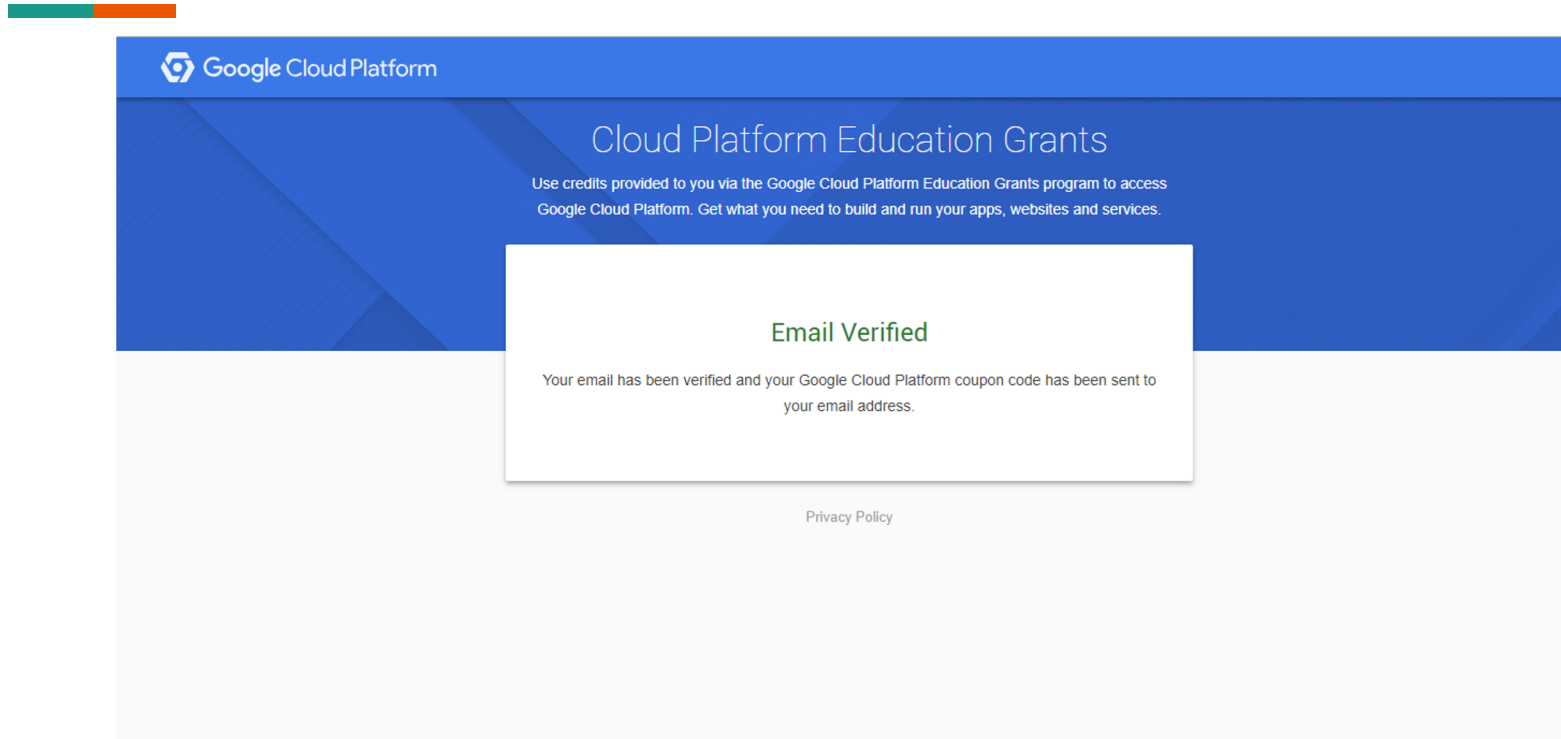
- The email has been verified. A new mail is sent. Check for the incoming mail in your institutional mailbox

Step 2: Join the project

Google Cloud Platform

Dear

Here is your Google Cloud Platform Coupon Code: 00XW-

Click [here] to redeem.

Course/Project Information
Instructor Name: Marco Ferretti
Email Address: marco.ferretti@unipv.it
School: University of Pavia
Course/project: Basi di dati 502479
Activation Date: 08/10/2018
Redeem By: 08/02/2019
Coupon Valid Through: 08/10/2019

If you have any questions, please contact your course instructor as listed above.

Thanks,

- The mail contains the coupon code
- Follow the link to redeem the coupon

Step 2: Join the project

- Check the box for accepting the licence terms
- Confirm

Step 2: Join the project

- You are allowed to use any Google services provided with Google Cloud Console, but we will use just Google Cloud Instances
- You have 50 $ to spend for complete your project.
- There is no possibility to have more credit (don't waste it!!)
- Your credit will be available until the October 2020

Step 3: open google console

- open https://console.cloud.google.com
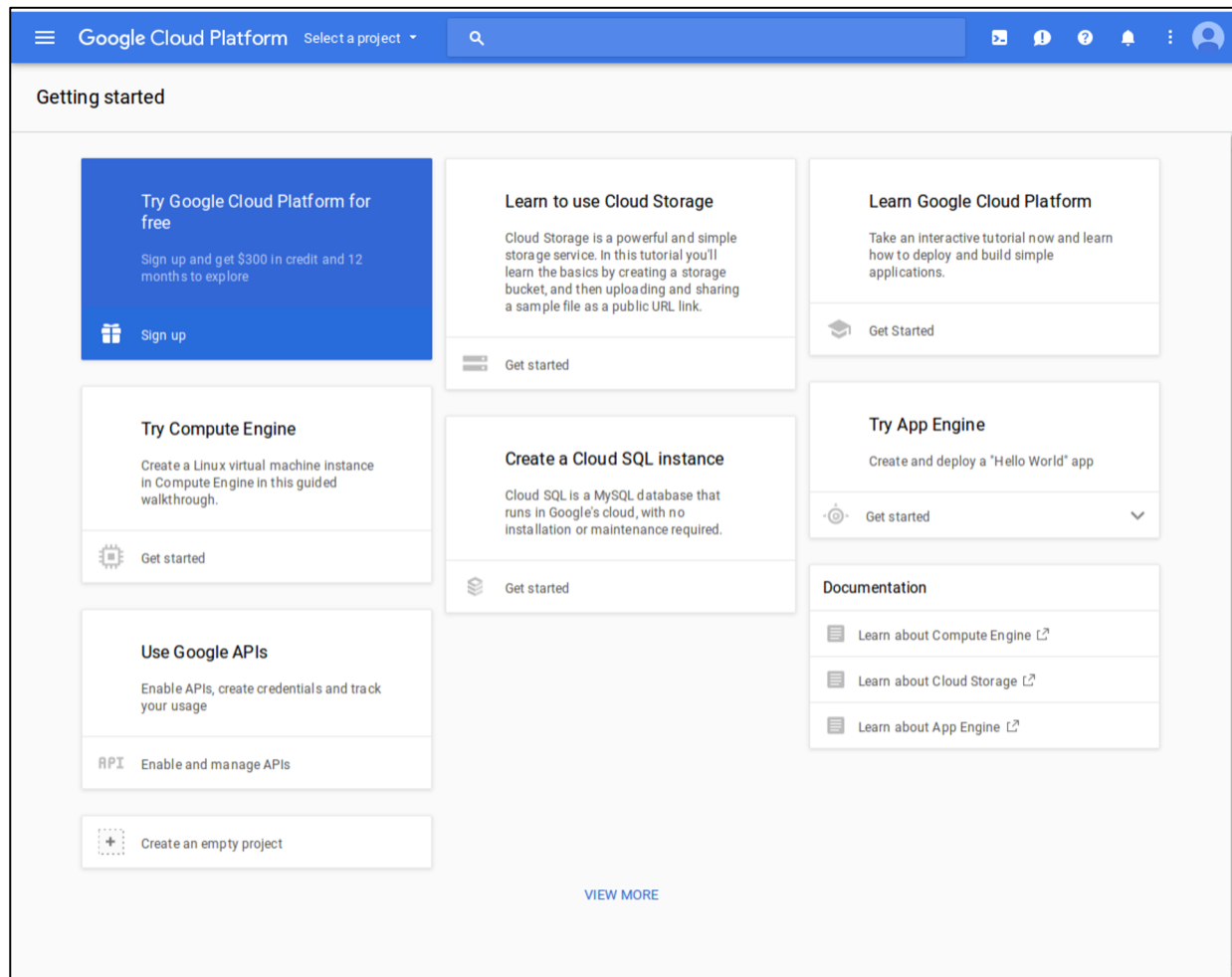
Step 4: getting familiar with the console
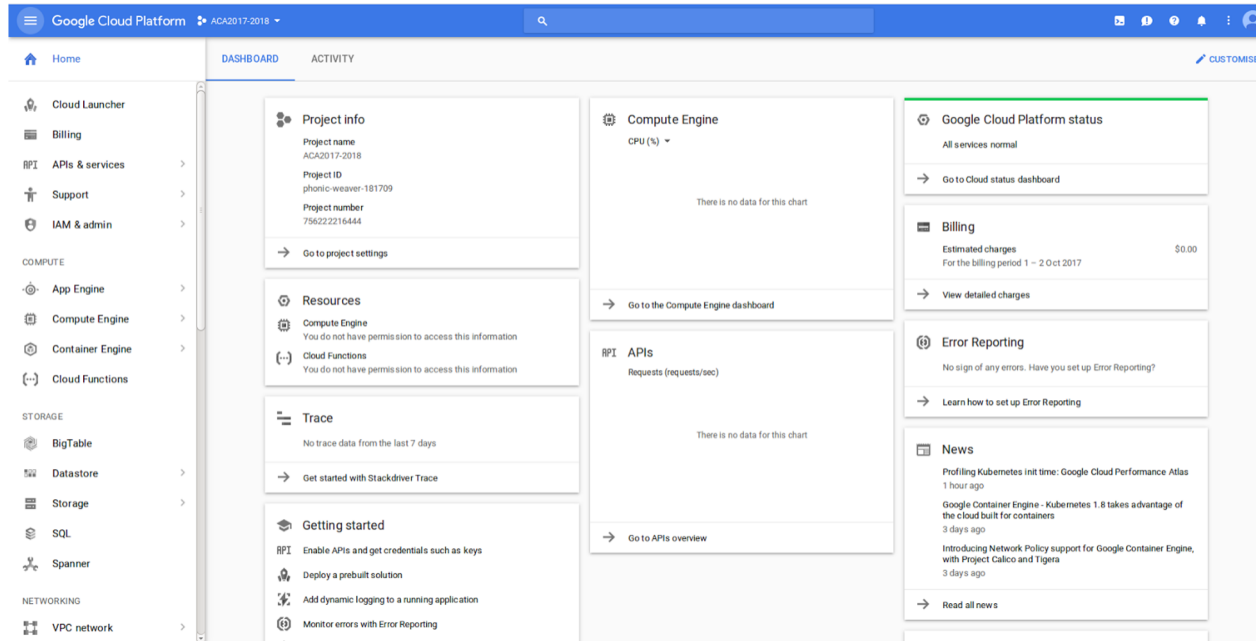
The blu bar shows the *menu bar* icon (leftmost) , the current "project" and your ID (rightmost)

The menu bar shows all products offered by Google.

From the menu bar on the left, select Compute Engine. Compute Engine allows you to use virtual machines that run on Google's infrastructure.

Step 5: creating a VM

VM instances can be:

- created from scratch

- imported from previous created models

- upload from the market

Step 5: creating a VM

Fill the form

Step 6: machine type and zone

The form asks some information about the the instance to be created:

- **name**: the instance ID
- **zone**: The zone determines what computing resources are available and where your data is stored and used. Different zones have different costs.
- **machine type**: specifies the CPU number and the memory size. Depending on the configuration used, the estimated cost per month might change. Start using a small configuration. For example, choosing us-east1-c as zone, and creating a small machine, the cost per month is equals to 14.20$

Step 6: machine type and zone

The form asks some information about the the instance to be created:

- **CPU Architectures**: depending on the selected zone, user can also define the CPU architecture. Example of available architectures are: Broadwell, Skylake, Haswell, Ivy Bridge, Sandy Bridge. The maximum number of cores that a consumer can specify during configuration phase depends on the chosen architecture:
    - Skylake: 96 vCPUs
    - Broadwell, Haswell, Ivy Bridge: 64 vCPUs
    - Sandy Bridge: 16 vCPUs

Step 7: boot disks

The form asks some information about the the instance to be created:

- **Selecting Customize link**: a window allows you to customize the instance configuration. You can define cores number, memory and the CPU platform, and the number and the type of GPU
- **boot disk**: specifies the disk size and the OS distribution. Different boot disk type and size can be used (standard or SSD, from 10GB to 65 TB). Different Operating System can also be used (Debian, Ubuntu, Centos, Windows). We choose any Linux distribution (CentOS 7), standard disk, 10 GB

Step 8: firewall

The form asks some information about the the instance to be created:

- Applications running on the VM use the service account to call Google Cloud APIs. Select the service account that you want to use and the level of API access that you want to allow. Leave this information unmodified.
- **Firewall**: enable both checkboxes if HTTP and HTTPS traffic have to be allowed

Step 9: costs

$14.20 per month estimated
Effective hourly rate $0.019 (730 hours per month)

Item	Estimated costs
1 shared vCPU + 1.7 GB memory	$19.71/month
10 GB standard persistent disk	$0.40/month
Sustained use discount ?	- $5.91/month
Total	$14.20/month

Compute Engine pricing ⬈

⌃ Less

On the left side, an estimation of the cost will be shown

Step 10: configuring network & access

- Select the link Management, disks, networking, SSH keys. a new section will be opened

Step 11: Preemptibility

A preemptible VM is an instance that you can create and run at a much lower price than normal instances but it might be terminated sooner, due to the system demands. The probability that Compute Engine will terminate a preemptible instance for a system event is generally low, but might vary from day to day and from zone to zone. Compute Engine always terminates preemptible instances after they run for 24 hours

Step 12: Network

a. subnetwork: specifies the subnet mask of the local IP (10.142.0.0/20). This will be different depending on the selected region

b. Primary internal IP: the IP address that should be used. Because the IP address is unknown, set the field "Primary internal IP" as Automatic. Warning: this IP cannot be used for public access

c. External IP: represents the public IP address which can be used to log on the instance. Using none, the instance will not be accessible from internet, choose "*Ephemeral*" to use an IP from a shared ephemeral IP address pool (but it changes). Then select Create an IP address to get a *static IP* just putting the name of the instance (during configuration, my instance toke 35.196.98.225)

d. IP forwarding: specifies if the IP forwarding is enabled. Leave it off. it is just used for creating Virtual Private Cloud.

Step 13: SSH key (for Linux users)

a. go to your linux box, open a shell, then write

b. ssh-keygen -t dsa

c. dsa is the encryption algorithm to be used. alternatives are rsa, ecdsa and ed25519.

d. a new pair private/public keys is created. The default location is under the directory .ssh in the home directory. The private key is stored into the file id_dsa, the public key is stored into the file id_dsa.pub

e. do not specify the passphrase with which the private key will be protected

f. open id_dsa.pub, copy the key and past it into ssh keys box. Please note the username displayed on left side. This will have to be used to connect to the instance via ssh

Step 14: Completing

After filling the form, press Create button

Step 15: Starting and stopping VM
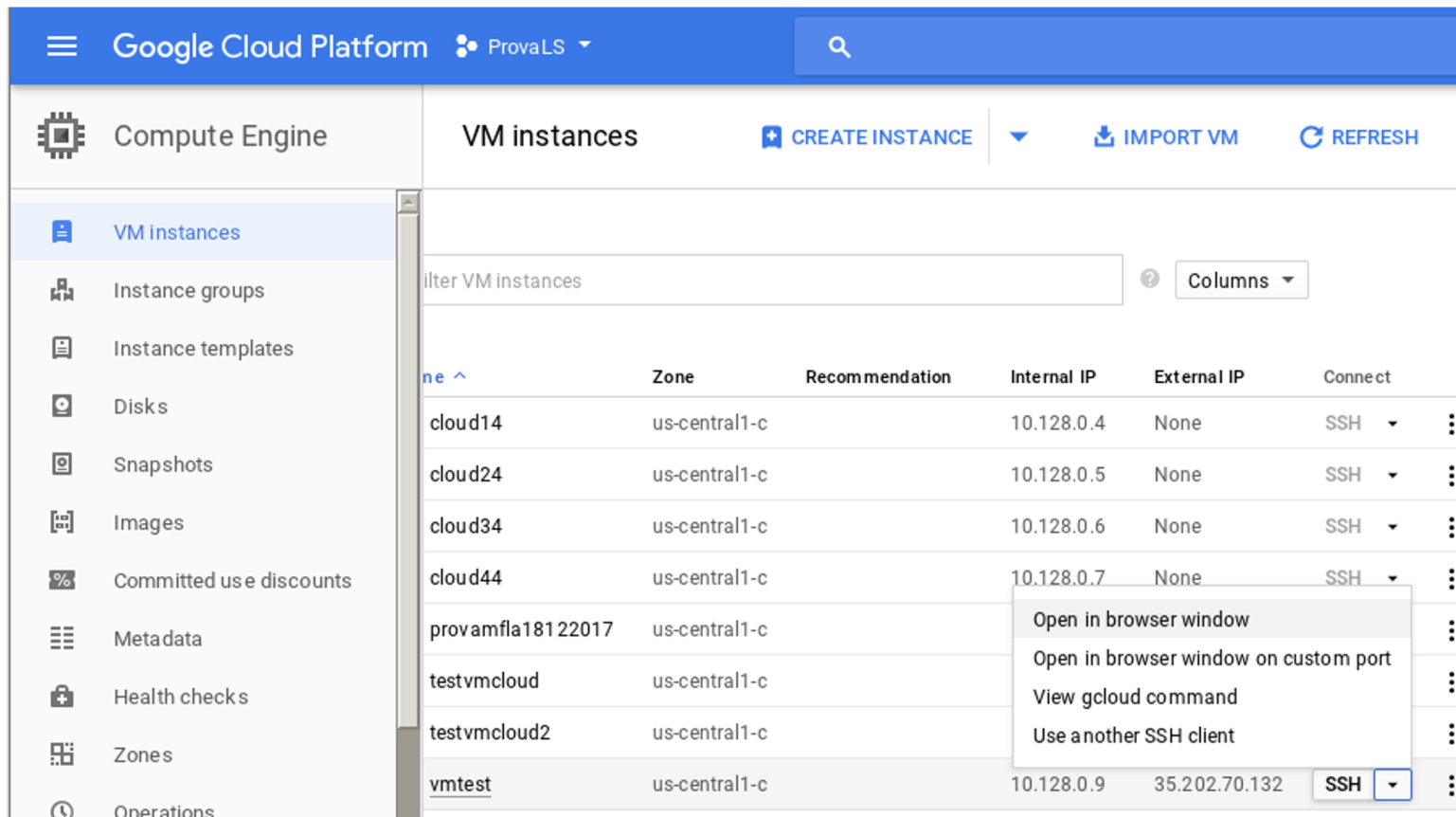
A green icon means the instance is started. You can stop or restart the instance using the three point icone on the right

WARNING: please shut down your VM when you finish working to avoid wasting credits.

WE WILL CONTROL YOU!!!!!

Step 16: Connect from console

A VM can be accessed, directly, from the Console. This allows the user to access without specifying the private key

Step 16: Connect from client (for Linux users) & gcc install

Open a linux terminal and type

- ssh -i /home/luigi/.ssh/id_dsa -l username 35.196.98.225
- uname -a
- cat /proc/cpuinfo
- cat /proc/meminfo
- top
- sudo su
- yum install gcc

Step 17: copy file from local to remote (Linux users)

- scp -i /home/luigi/.ssh/id_dsa -r mylocaldir

  username@35.196.98.225:/absolute/path

Step 13: SSH key (for Windows users)
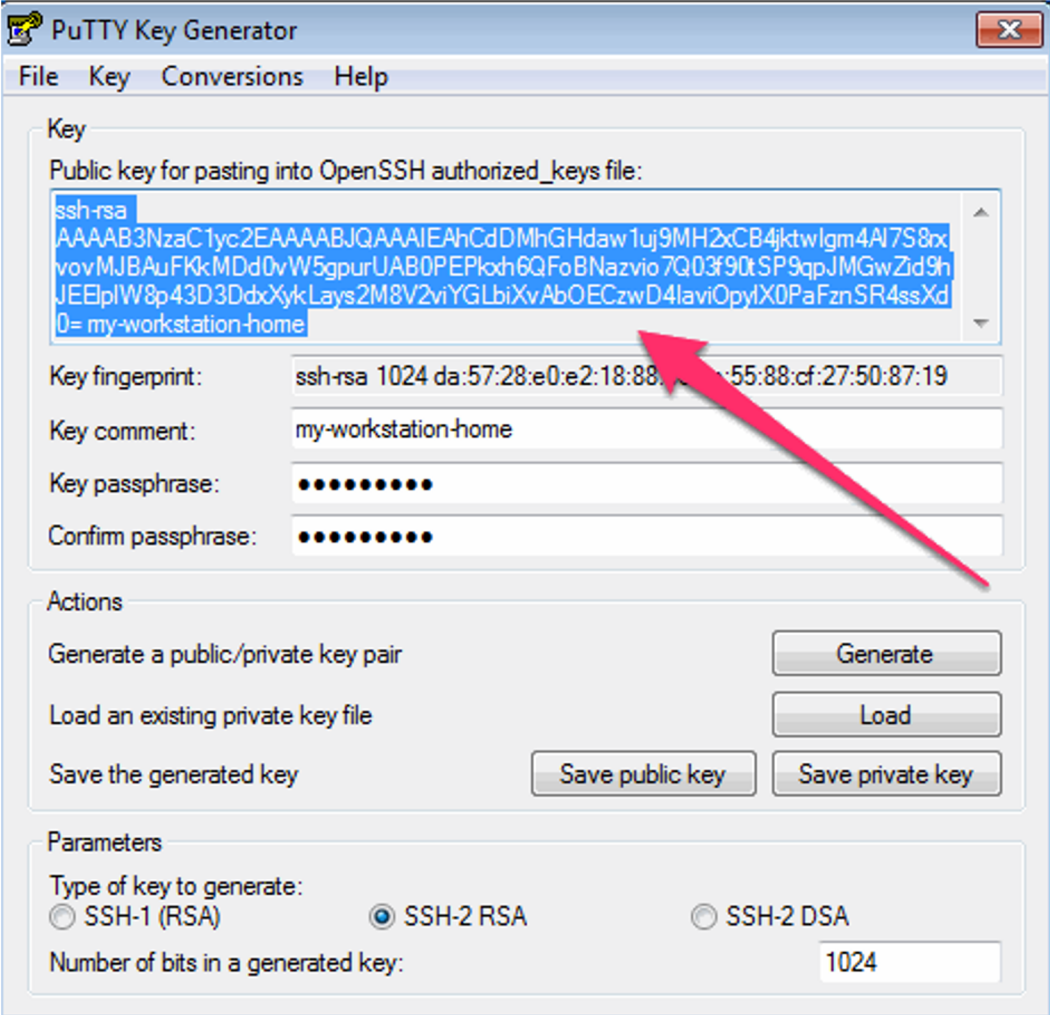
a. open the browser and go to http://www.putty.org

b. download and install the tools puttygen.exe

c. open puttygen.exe

d. select the preferred type of key (for example RSA or DSA) and the number of bits in the generated key (for example 2048)

e. click on Generate Button (to generate the key pair)

f. move randomly your mouse over the blank area (to generate some randomness numbers)

g. Copy the public key shown on top of the window and past it into ssh keys box. Please note the username displayed on left side. This will have to be used to connect to the instance via ssh

h. Select buttons "Save public key" and "Save private key"

i. do not specify the passphrase with which the private key will be protected

Step 13: SSH key (for Windows users)

Step 13: SSH key (for Windows users)

Step 16: Connect from client (for Windows users)

Download from http://www.putty.org the tool named putty.exe

- Specify the destination you want to connect to (the IP address)
- On the left side, select Connection, than SSH, than Auth
- Select the private key to be used
- Click to Open
- uname -a
- cat /proc/cpuinfo
- cat /proc/meminfo
- top
- sudo su
- yum install gcc
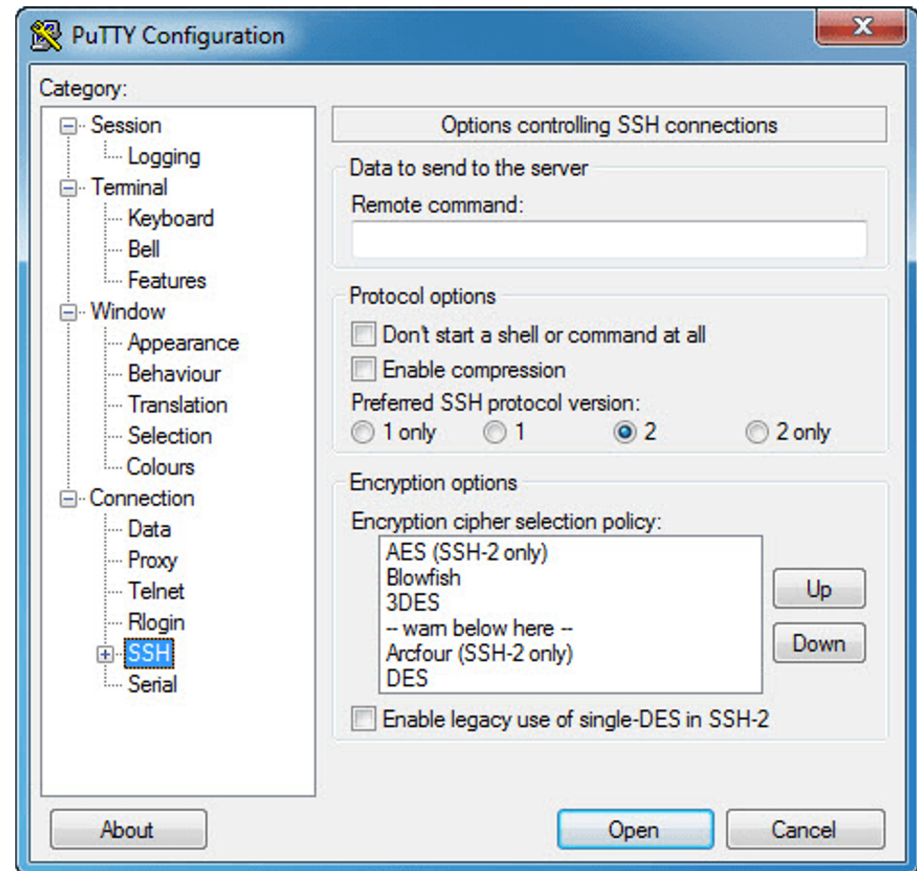
Step 18: copy file from local to remote (Windows users)

Open the link https://winscp.net/eng/download.php

Download the last version of WinSCP

Fill the form with:

- hostname
- port
- username
- private key

Step 18: copy file from local to remote (Windows users)

Then, copy file and directory as needed

Configurations rules for creating new instances

- Don't forget to bring along your laptop
- Start with a small configuration and enlarge it later
  - zone (us-central)
  - operating system (CentOS 7)
  - cores number (1 → 24)
  - memory size (1 GB → …)
  - disk size (10 GB)
- Don't leave your virtual instance running