CODATA – RDA Data Schools

Cyber Infrastructures

Cloud Computing

Instructor: Raphael Cobe
@raphaelmcobe / raphaelmcobe@gmail.com

About the Instructor

CODATA – RDA Data Schools

About the Instructor

- Researcher at the Advanced Institute for Artificial Intelligence;
- Associate Researcher at the São Paulo State University - UNESP;
- Member of the SPRACE project (a CMS LHC collaboration);
- Chair of the CODATA-RDA Data School groups since 2018;
- Background on Artificial Intelligence and High Performance Computing;

Materials

Exercises for the Lecture

1. Warmup/Data Exploration using Google Colab;
2. Regression Analysis using Apache Spark on Colab;
3. Setting up a cloud resource using OpenStack;
4. Accessing a Virtual Machine using the SSH protocol;
5. Deploying a Jupyter Notebook Server at an IaaS platform using the conda environment manager;
6. Deploying a Jupyter Notebook Server at an IaaS platform using Docker containers

Acknowledgements 👏

The people that worked hard to get this materials/resources available

1. Rob Quick (Indiana University) for working on the exercises;
2. CODATA-RDA Schools of Research Data Science Co-chairs for reviewing and evaluating the materials;

The setup problem

- Your science computing is too complex!
  - Monte carlo simulation, image analysis, genetic algorithm, ...
- It will take a year (CPU time) to get the results on your laptop, but your **paper is due in a week**.
- What do you do?

CODATA – RDA Data Schools

Option 1: Wait a year

Option 2: Local Clustered Computing

Option 3: Use a High Performance Computing(HPC)

Option 4: Use lots of commodity computers

Option 5: Buy (or Borrow) some computing from a Cloud Provider

These are All Valid Options

- Remember the problem you have one month to publish results for your conference
  - **Option 1**: You will miss your deadline
  - **Option 2**: You might miss your deadline – But if you're lucky you'll make it (or if you know the admin)
  - **Option 3**: If you have parallelized code and can get an allocation you have a good chance
  - **Option 4**: If you can serialize your workflow you have a good chance
  - **Option 5**: You can meet your deadline for a price. Though **academic clouds are becoming more available**.

CODATA – RDA Data Schools

Option 1: Wait a year

Option 2: Local Clustered Computing

Option 3: Use a High Performance Computing(HPC)

Option 4: Use lots of commodity computers

Option 5: Buy (or Borrow) some computing from a Cloud Provider

Computing Infrastructures

- Local Laptop/Desktop – Short jobs with small data
- Local Cluster – Larger jobs and larger data but subject to availability
- HPC – Prime performance with parallelized and optimized code
- HTC – Sustained computing over a long period for serialized
- Cloud – Need deeper permission on an OS and have deeper pockets

Agenda

- Introduction/motivation;
- A few words about virtualization;
- Cloud environments;
  - Public vs Private vs Hybrid
  - Cloud deployment models: SaaS, PaaS, and IaaS;
- Jupyter Notebooks;
- Openstack cloud manager;
- Other virtualization techniques;

Agenda

- Introduction/motivation;
- A few words about virtualization;
- Cloud environments;
  - Public vs Private vs Hybrid
  - Cloud deployment models: SaaS, PaaS, and IaaS;
- Jupyter Notebooks;
- Openstack cloud manager;
- Other virtualization techniques;

Introduction

Cloud Computing

- Complete solution in which all the features of computing are quickly provided to users as the demand increases;
- You can control the features and services offered to ensure high availability, security and quality of service
  - Users get the resources they need, no more, no less.

Introduction

Cloud Computing - Some history

- Utility Computing - John McCarthy
- ARPANET (Advanced Research Projects Agency Network) - 1969
  - “Intergalactic Computer Network” in which everyone on the planet would be interconnected by way of computers;
- Cloud computing concepts: availability and accessibility

Introduction

Introduction

Cloud Computing

- Creates an idea of availability of infinite resources;
- Eliminates the need to acquire and provision resources in advance;
- Offers elasticity, allowing companies to use resources as needed;
- Payment for services is made by the amount of resources used;
- Deeply connected to the Virtualization concept;

Agenda

- Introduction/motivation;
- A few words about virtualization;
- Cloud environments;
  - Public vs Private vs Hybrid
  - Cloud deployment models: SaaS, PaaS, and IaaS;
- Jupyter Notebooks;
- Openstack cloud manager;
- Other virtualization techniques;

Introduction

Virtualization

- Simulated or virtual computing environment instead of a physical environment;
- Include computer-generated versions of hardware, storage devices, and others;
- This allows organizations to partition a single physical computer or server into multiple virtual machines;
  - Interact independently and run different operating systems;
  - Share the resources of a single host computer;

Introduction

Virtual Machines

- It has CPU, memory, disks to store your files and can connect to the Internet if needed;
- Software-defined computers on physical servers, existing only as code;
  - "Virtual" version of a computer, with dedicated amounts of CPU, memory, and storage that are "borrowed" from a physical host computer;
- The virtual machine is partitioned from the rest of the system;

Introduction

Virtual Machines - Motivation

- Create and deploy applications in the cloud;
- Try a new OS (operating system), including beta versions;
- Create a new environment to make running development and test scenarios simpler and faster for developers;
- Back up your existing operating system;

Introduction

Virtual Machines - Benefits

- Remain completely independent of each other and of the physical host computer;
- Hypervisor - allows you to run different operating systems on different virtual machines at the same time;
  - E.g., running a windows virtual machine inside a linux laptop;
- You can move a VM from one hypervisor to another completely different computer almost instantly;

Introduction

Virtual Machines

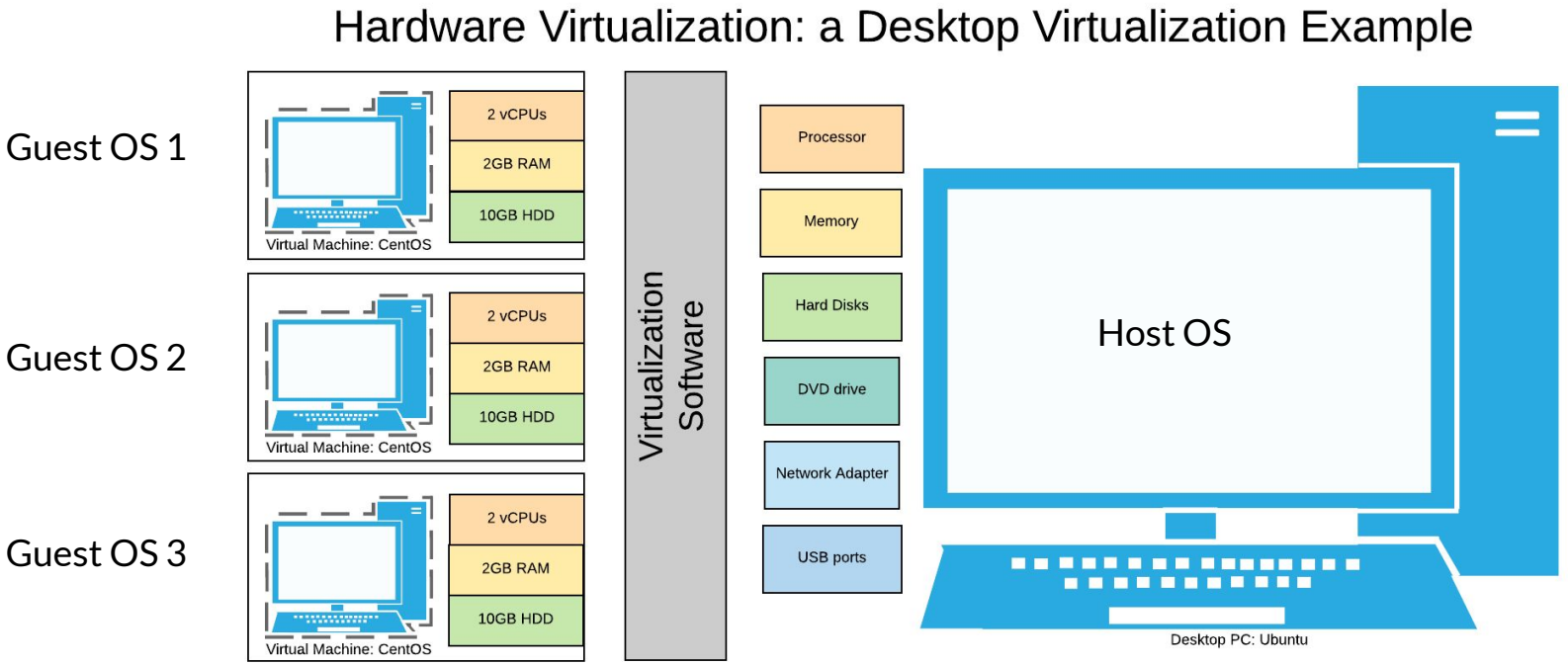


Image from UnixTutorial.org

Introduction

Hypervisors - How does it work?

- Can be run on an operating system (such as a laptop) or installed directly on hardware (such as a server);
- Translates operations on virtual HW into host OS instructions;
  - The host OS then runs the instructions on real HW;
- Can cause performance loss;

Agenda

- Introduction/motivation;
- A few words about virtualization;
- Cloud environments;
  - Public vs Private vs Hybrid
  - Cloud deployment models: SaaS, PaaS, and IaaS;
- Jupyter Notebooks;
- Openstack cloud manager;
- Other virtualization techniques;

Cloud Computing

Cloud Environments

- Abstract, group and share scalable resources across a network;
- Not a technology per se;
- Provide a self-service interface;
- Elasticity and the ability to scale up and down;
- Application programming interfaces (APIs);
- Billing and metering of service usage in a pay-as-you-go model;

Cloud Computing

Cloud Environments - Public, Private and Hybrid

- **Public**: A cloud environment built from resources without an owner, such as an end user, that can be redistributed to other tenants.
- **Private**: Deployed using local resources. It is sometimes preferred for its ability to offer dedicated resources;
- **Hybrid**: Connects web resources and current resources not in the cloud. Allows to extend and grow a company's cloud infrastructure;

Cloud Computing

Cloud Deployment Models - Infrastructure As A Service

- Gives you the highest level of flexibility and management control over your IT resources;
- Users control the applications, data, operating system, and execution environments;
- The IaaS provider provides the virtualization, storage, networking and servers.
- The user does not need to have an on-premise datacenter or worry about maintenance or physical updates;

Cloud Computing

Cloud Deployment Models - Platform As A Service

- Application hardware and software platforms are provided by third parties;
- Mainly aimed at developers and programmers;
  - Allows the user to develop, run and manage applications;
  - A provider hosts the hardware and software components in its infrastructure
- Provides a platform to the user as an integrated solution;

Cloud Computing

Cloud Deployment Models - Software As A Service

- Provides users with a cloud application with the IT infrastructure and platforms underlying it;
- Compatible with software subscription models;
- Service performance is determined by internet connection speed;
- General public oriented:
  - Does not require IT/Software development skills;
- Allows providers to easily deploy new features to customers.

Cloud Computing

Cloud Deployment Models

- Providers:

Agenda

- Introduction/motivation;
- A few words about virtualization;
- Cloud environments;
  - Public vs Private vs Hybrid
  - Cloud deployment models: SaaS, PaaS, and IaaS;
- Jupyter Notebooks;
- Openstack cloud manager;
- Other virtualization techniques;

PaaS Example

Jupyter Notebooks

- Contain both computer code and rich text elements (equations, figures, links, etc.);
- Both human-readable and executable documents;
  - Contain the analysis description and the results (figures, tables, etc..) as well as code that can be run to perform data analysis;
- "*Jupyter*" is an acronym for **Ju**lia, **Py**thon and **R**;
- Replication and reproducibility are two pillars of the scientific method.
  - Jupyter notebooks make it easy to do both;

PaaS Example

Jupyter Notebooks history

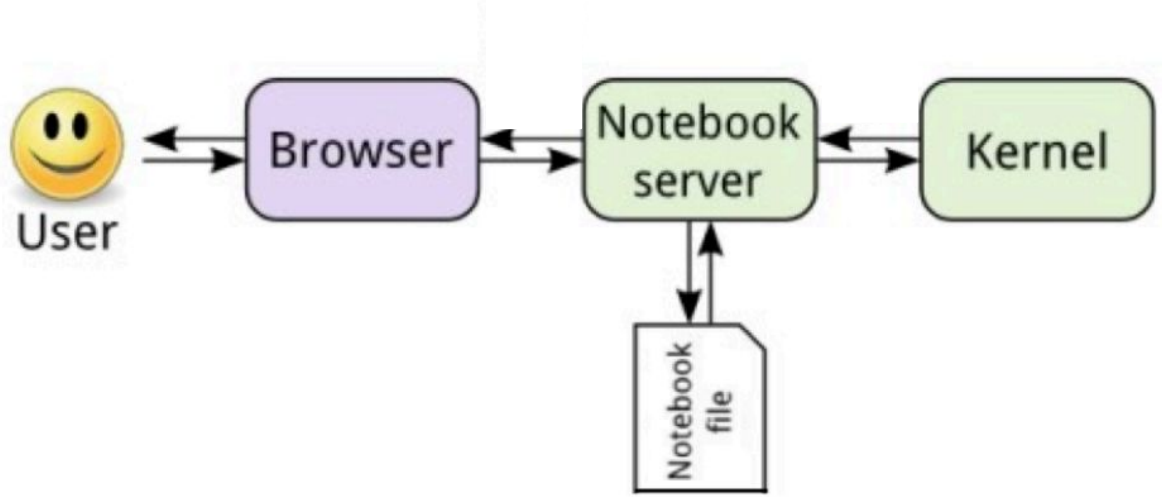
PaaS Example

Jupyter Notebooks - concepts

- Cells:
  - Blocks that can be executed individually and present results;
  - **Code cell**: contains code to be executed in the kernel and displays its output below;
  - **Markdown cell**: contains text formatted using Markdown and displays its output in place
- Kernels:
  - Computational engine that runs the code contained in a notebook.

PaaS Example

Jupyter Notebooks - how does it work?



PaaS Example

Jupyter Notebooks - Google Colaboratory

- Free cloud service hosted by Google
- Encourage research on Machine Learning and Data Science;
- Jupyter notebook environment that requires no setup and runs in the cloud;

PaaS Example

Jupyter Notebooks - Google Colaboratory

- Compatible with codes in Python (2.7 and 3.6) and R 4.1;
- Access to powerful scientific computing resources;
- Free GPU acceleration;
- Pre-installed all major libraries
- Collaboration works like Google Docs;
- Supports bash commands;
- Stored in Google Drive.

PaaS Example

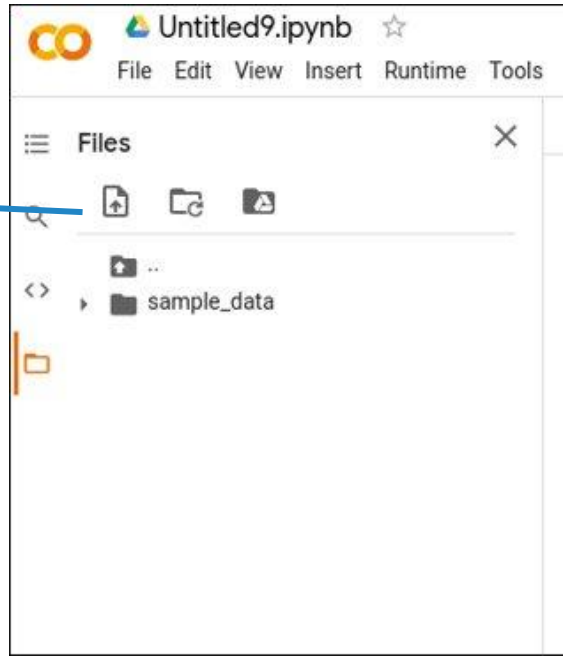
Jupyter Notebooks - Google Colaboratory

- How to access:
  - https://colab.to (Python) or
  - https://colab.to/r (R)
- Bash commands:
  - Prefixing the command with "!" (Python)
  - Use `system("COMMAND", intern=T)` (R)
- Installing new libraries:
  - pip support (!pip install [package name])
  - In R: install.packages('[package name]')

PaaS Example

Jupyter Notebooks - Google Colaboratory

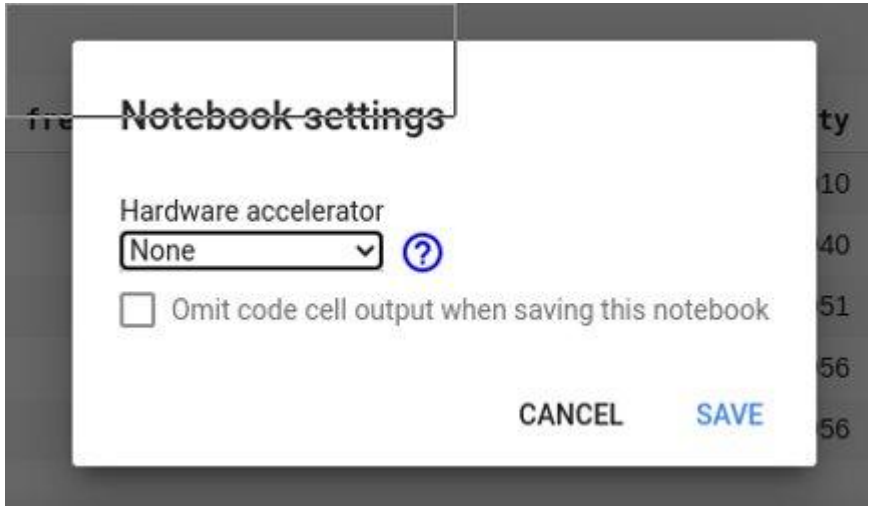
- Uploading data:



PaaS Example

Jupyter Notebooks - Google Colaboratory

- Choosing the Runtime:
  - CPU, GPU ou TPU (limited to 12 hours/day);



Agenda

- Introduction/motivation;
- A few words about virtualization;
- Cloud environments;
  - Public vs Private vs Hybrid
  - Cloud deployment models: SaaS, PaaS, and IaaS;
- Jupyter Notebooks;
- Openstack cloud manager;
- Other virtualization techniques;

Openstack

Some history

2010

NASA + Rackspace develop the basis of OpenStack

2012

OpenStack Foundation established

2014

OpenStack Marketplace opens to showcase maturing ecosystem; “Juno” release seen as enterprise grade

2015

OpenStack Powered interop certification launched

2016 - April

Half the Fortune 100 run OpenStack; Certified OpenStack Administrator program launched

2016

China booms; 86% of telecoms say OpenStack important to their business

2017

OpenStack emerges as one platform for containers, VMs and bare metal

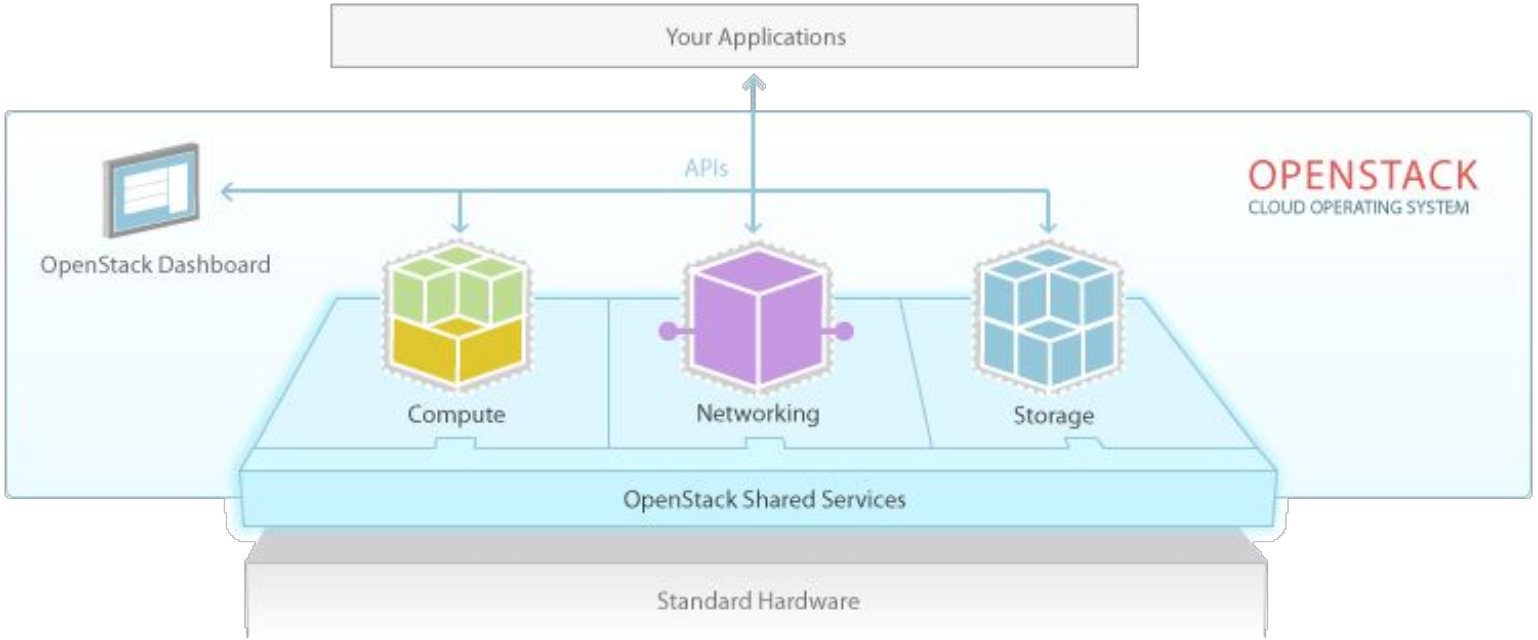
Openstack

Introduction

- Opensource;
- Large project with several components;
  - Virtual Machines (*nova*), Virtual Networks (*neutron*), Cloud Storage (*cinder*, *swift*, *glance*) User self-service interface (*horizon*), User authentication and authorization (*keystone*), etc.;
- OpenStack does not virtualize resources alone, but uses them to create clouds;

Openstack

What is in it?



Openstack

Main concepts

- Project: A container used to group a set of resources such as virtual machines, volumes and images with the same access rights and quota.
- Quota: A per-project limit such as the total number of cores or RAM permitted for a set of virtual machines.
- Flavor: The definition of the size of a virtual machine and its characteristics
  - E.g., 2 core virtual machine with 8 GB of RAM

Openstack

Main concepts

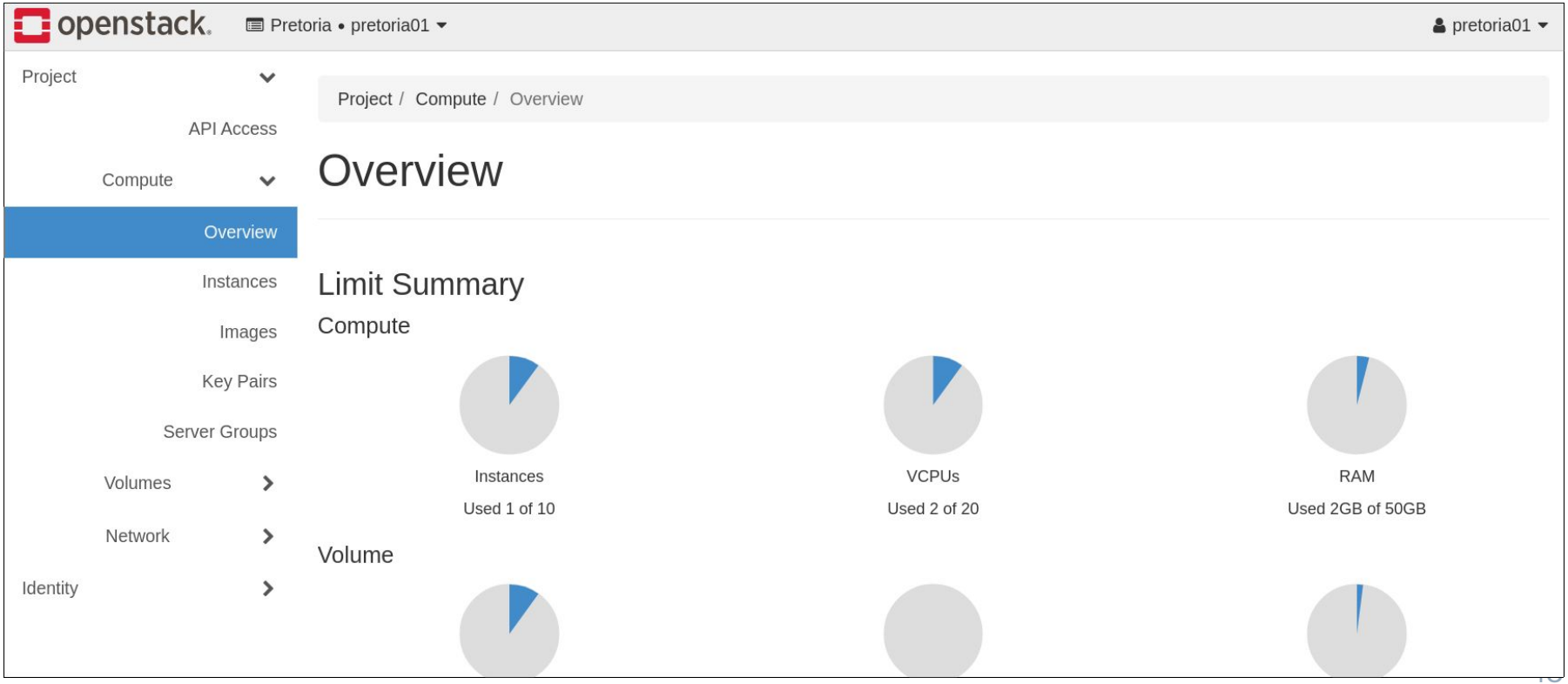
- Image: single file that contains a virtual disk that has a bootable operating system installed on it.
  - Used to create VM instances;
- Volume: block storage devices that you attach to instances to enable persistent storage.
  - You can attach or detach a volume to a running instance. You can also create a snapshot from or delete a volume.

Openstack

Main concepts

- Snapshot: provides a copy of a currently running VM or volume which can be stored into an external service such as “*Glance*”.
  - This can be used to "*pre-configure*" a VM, installing required software beforehand;

Openstack - The horizon interface



Openstack

VM Creation process

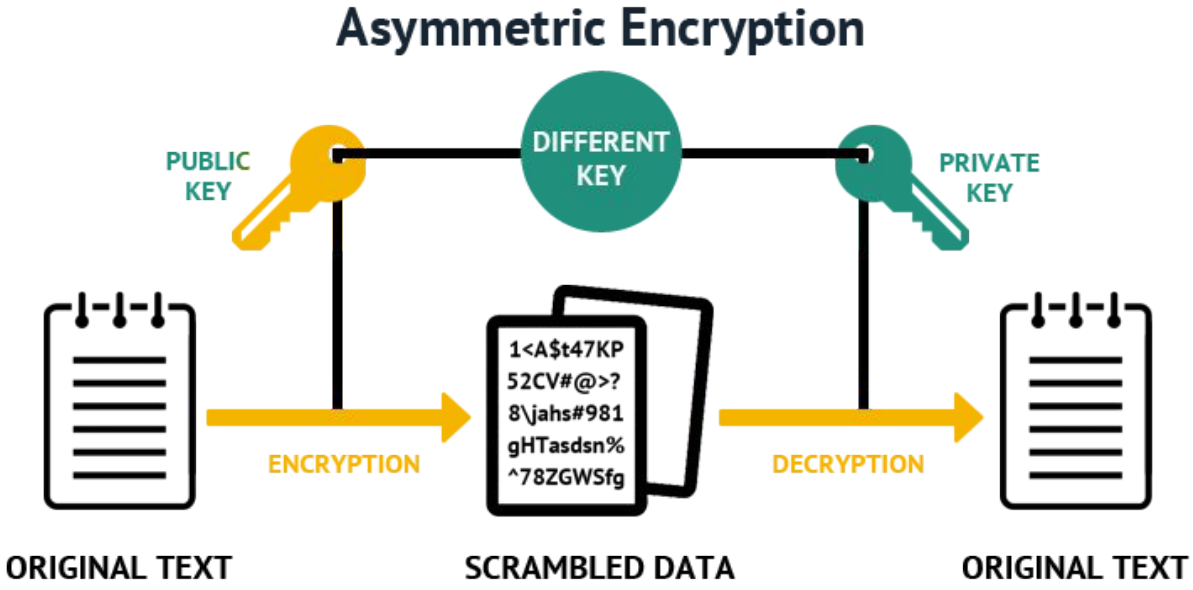
- Creating virtual machines needs a combination of:
  - An image from which the VM should be created;
  - A flavor defining the size of the virtual machine;
  - (optional) A virtual network to connect you VM to;
  - A set of firewall rules (security groups);
  - A keypair to define the authentication for logging in as the VM administrator;

Openstack

CODATA – RDA Data Schools

Accessing remote servers

- Private-Public key pairs;
- Asymmetric Cryptography;



Openstack

Installing your own Openstack

- Packstack - Cloud Proof of Concept
- All-in-one concept cloud;
- Get an Openstack cloud ready in less than 15 minutes;
  - Can run on a notebook (**considering enough resources are available**)

Agenda

- Introduction/motivation;
- A few words about virtualization;
- Cloud environments;
  - Public vs Private vs Hybrid
  - Cloud deployment models: SaaS, PaaS, and IaaS;
- Jupyter Notebooks;
- Openstack cloud manager;
- Other virtualization techniques;

Environment Managers

Conda Environment Manager

- Package and environment management system that runs on multiple operating systems;
- Allows to create different virtual environments
  - Separate software and versions of packages/dependencies
- Benefits:
  - No need for administrator access to the computer to perform library environment modifications

Environment Managers

Conda Environment Manager

- Install from the link;
- Create environments:

```
conda create -n ENVIRONMENT_NAME
```

- (de)Activate environment:

```
conda [de]activate ENVIRONMENT_NAME
```

- Install software to an environment:

```
conda install -n ENVIRONMENT_NAME
SOFTWARE[=VERSION]
```

Environment Managers

Conda Environment Manager

- A lot more to it:
  - Separate channels for distributing softwares;
  - Ability to share your environments;
    - Export/import environments;
- Read the Cheat Sheet

Containers

Docker Containers

- Segregation of processes in the same operating system;
- Maximum isolation possible from the rest of the environment;
- File Systems created from an “image”;
- Makes reproducibility much easier
- Conceptually similar to virtual machines

Containers

Docker Containers - Advantages 😀

- Lightweight because they don't contain an operating system;
  - Have the same performance as code running on the host operating system;
- Up to three times more performance than virtual machines when running on the same hardware.
- Startup time in milliseconds (compared to minutes for a virtual machine);

Containers

Docker Containers - Disadvantages 😔

- Always run on the Linux operating system (containers share the host operating system):
  - Virtual machines can run a different operating system for each virtual machine;
- Containers use process-level isolation
  - potentially less secure than fully isolated virtual machines