Introduction to using cloud and containers for training - OpenStack and Docker oriented view

6.2.2019
Cloud Computing

“Cloud Computing refers to on-demand delivery of computing services – servers, storage, databases, networking, software, analytics and more—over the network.”

“A model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”
Cloud Deployment Models

- **PRIVATE**
  - Provisioned for exclusive use by a single organization
  - Example: organization’s internal cloud system

- **COMMUNITY**
  - Shared by several organizations with similar goals
  - Example: many ELIXIR Node clouds

- **HYBRID**
  - Composition of two or more clouds
  - Example: an academic cloud is extended to public cloud or other academic cloud

- **PUBLIC**
  - Anybody can buy its services
  - Example: Microsoft Azure
IaaS – PaaS – SaaS: the responsibility division

**Infrastructure as a Service (IaaS)**
- CSC’s ePouta/cPouta
- Amazon EC2
- Microsoft Azure

**Platform as a Service (PaaS)**
- CSC’s RAHTI
- CSC’s notebook.csc.fi
- Google AppEngine
- Heroku

**Software as a Service (SaaS)**
- CSC’s Chipster
- Google Web Apps
- Microsoft Web Apps
- DropBox and many other cloud services
Cloud Service Landscape

- This course is strictly about IaaS cloud
- Yesterday’s services in a Virtual Machine are today’s containerized microservices; today’s containerized microservices are tomorrow’s serverless architectures
- IaaS probably has some evolutionary cycles left, too
- Let’s proceed to look at how IaaS is setup!

For example: pouta.csc.fi, Amazon EC2, Microsoft Azure, Google Compute Engine
Typical IaaS Cloud Setup

Frontend Client

- Web UI, CLI, ..

Internet

Cloud Middleware

Compute

Storage
Creating virtual resources in Pouta - User Interfaces

- **Web User Interface** -
  - Suitable for administering individual VMs, keys, images, volumes...
  - The only UI to support Haka federated login

- **CLI tools**
  - Suitable for more elaborate resource provisioning and possibly some lightweight (scripted) software integrations
  - More info at [https://research.csc.f/pouta-install-client](https://research.csc.f/pouta-install-client)

- **Programming APIs**
  - Suitable for building very large systems and stacks
  - Support from individual services (compute, storage) to full-fledged orchestration
  - List of APIs available at [https://pouta.csc.f/dashboard/project/access_and_security](https://pouta.csc.f/dashboard/project/access_and_security)
Workflow For Creating Resources

1. "Give me a virtual machine called VM1 connected to the internal network X."

User

Cloud interface server

Virtualized resources
Workflow For Creating Resources

1. “Give me a virtual machine called VM1 connected to the internal network X.”
2. “OK. It is running.”

Virtualized resources
Workflow For Creating Resources

1. "Reserve public IP address 1.2.3.4 and attach it to VM1."

2. "OK. Done."

Virtualized resources

Network X
Public network Z

1.2.3.4
Workflow For Creating Resources

1. "Connect to 1.2.3.4."

Virtualized resources

User

Cloud interface server

1.2.3.4

VM1

Network X

Public network Z
Things needed to create and access a VM in cPouta

- Access to Pouta Web UI
- One IPv4 address - a public “Floating IP”
- Security Group permitting access from User’s computer
- Ssh key-Based Authentication
  - later the authentication can be changed to password based, but it is not so recommended as password protected key
- SSH client software
- Internet access
Creating a Key pair

Navigate to
Compute>Access and Security>Key Pairs

Click on create Key Pair, name key as lastname_firstname
Storing the key pair

Linux and Mac OS X

- Create .ssh directory in ~ if it's not there already
  
  ```
  mkdir -p .ssh
  chmod 700 .ssh
  ```
- Move key pair to .ssh directory
  
  ```
  cd .ssh
  mv ../Downloads/yourkey.pem .
  ```
- Make key unreadable by other users
  
  ```
  chmod 400 yourkey.pem
  ```
- Protect key with passphrase
  
  ```
  ssh-keygen -p -f yourkey.pem
  ```

Windows

- Download Putty and Puttygen tools if you don't have them

- Load your **private key (yourkey.pem)** into **puttygen** and change it to .ppk format

- Open Putty, load .ppk file under **Connection | SSH | Auth | Private key file for authentication**
  - Provide user name **cloud-user**
  - Provide password which you added to Puttygen (Optional)

---

```plaintext
Storing the key pair

Linux and Mac OS X

- Create .ssh directory in ~ if it's not there already

  ```
  mkdir -p .ssh
  chmod 700 .ssh
  ```

- Move key pair to .ssh directory

  ```
  cd .ssh
  mv ../Downloads/yourkey.pem .
  ```

- Make key unreadable by other users

  ```
  chmod 400 yourkey.pem
  ```

- Protect key with passphrase

  ```
  ssh-keygen -p -f yourkey.pem
  ```

Windows

- Download Putty and Puttygen tools if you don't have them

- Load your **private key (yourkey.pem)** into **puttygen** and change it to .ppk format

- Open Putty, load .ppk file under **Connection | SSH | Auth | Private key file for authentication**
  - Provide user name **cloud-user**
  - Provide password which you added to Puttygen (Optional)
```
Opening VM to internet with security groups

- A Security Group defines a set of cloud level firewall rules for filtering traffic, typically inbound.
- By default Security Groups blocks all incoming connections to your VM. It is good to keep them as closed as possible!
- Security groups define combinations of ports and IP addresses for which the incoming connections are permitted.
- Security groups are created in the web interface and then applied to virtual machines.
- One security group and include several “rules” and be used by several virtual machine.
- One machine can use several security groups and vice versa.
- You can additionally have some VM level firewall rules in conjunction to security group for better security.
- Typical case: allow connections from the IP address of your own computer to port 22 (SSH port).
Security Groups

• Created by navigating to Compute>Access and Security>Create Security Group
  o Several predefined rule sets are available, such as for SSH
  o At bare minimum you need to select the Source IP for the traffic
  o Modify the CIDR field to allow SSH connections only from specific IPs
Creating an Instance

- Navigate to **Compute>Instances** and **Launch Instance**
- Give Instance name as **lastname_firstname_instance**
- Select a Flavor of your choice (**standard.tiny** is a good first choice)
- Select Instance Boot Source as **Boot from image**
- Pick an image - any image
- Navigate to **Access & Security** in same popup. Make sure that the **“SSH - World”** Security Group is selected.
- Populate the Post-Creation script as per
Creating an Instance!

- Add SSH key pair to Web UI
- Create instance as before
- In **Access & Security**, make sure that the SSH key pair is selected
- When connecting to the instance, designate the private key into the session or pre-populate it into an SSH agent prior to making a connection
Exercise Set 1: Creating & Securing Virtual Resources

- **Exercise 1 - Creating a temporary Virtual Machine for testing login**
  - Log in to Cloud Dashboard at [https://pouta.csc.fi/](https://pouta.csc.fi/)
  - Create your own Virtual Machine Instance with disposable password in post creation section
  - Associate Floating IP to Virtual Machine Instance
  - Log in to your VM using SSH or Putty
  - Exit and delete the VM

- **Exercise 2 - Creating an SSH key pair for secure login to an Instance**
  - Create an SSH key pair, storing the private key in a safe place
  - Create new VM Instance using this key pair
  - Associate Floating IP address to VM
  - Log in
Exercise Set 1: Creating & Securing Virtual Resources

• Exercise 3 – Create your own Security Group for securing your virtual resources
  o Create your own Security Group for SSH traffic
  o Start by creating a wrong Security Group rule
  o Attach it to your VM
  o You would be denied access to your VM
  o Modify Security Group again, this time with correct Security Group rule
  o Connect to your VM
Pouta: Hardware Options

- **GPU Flavor***
  - AI, Deep Machine Learning
  - Intensive 3D/Video/Image processing

- **I/O Flavor**
  - Fast local disk needs
  - Hadoop/Spark
  - Clustered Databases

- **HPC Flavor***
  - Scientific Applications
  - Advanced Computing & Software Development

- **Standard Flavor**
  - Normal Computing & Software Development
  - Normal Application/Web & content services hosting

---

*Flavors marked with an asterisk indicate a premium option.*
OpenStack

- CSC’s cPouta/ePouta cloud services are powered by OpenStack.
  - Current OpenStack version used by Pouta services is Newton

- OpenStack is a cloud software that allows end user to create and use their VM instances, networks and storage.

- Fast moving open source project with backing from industrial giants like AT&T, Red Hat, IBM, Intel, HP etc.

- Flexible architecture which may support different types of scalabilities.

- Used by many organizations from research institutes to service/content providers.

- Large customer base augments better availability of expertise, support and chances of continuity.

- Supports Web UI, CLI and REST Interfaces
OpenStack WebUI
### OpenStack CLI

#### Image List

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>4a36f47f-4ffe-4f88-bc0f-1dad6748f4d2</td>
<td>CentOS-6</td>
<td>active</td>
</tr>
<tr>
<td>7add5d46-325a-40d2-8d84-b83c35da383f</td>
<td>CentOS-7</td>
<td>active</td>
</tr>
<tr>
<td>5a9d651b-0be6-44eb-9b6b-9079b91e5af0</td>
<td>Fedora-Atomic-25</td>
<td>active</td>
</tr>
<tr>
<td>c4216e6c-7e3b-45bd-a43b-2873396e0c99</td>
<td>ScientificLinux-6</td>
<td>active</td>
</tr>
<tr>
<td>18eb990c-4e28-7ecb-1904-41e2c8367265</td>
<td>ScientificLinux-7</td>
<td>active</td>
</tr>
<tr>
<td>66f86b65-9f57-4b33-8e3b-9a39d0d8f2ff</td>
<td>Ubuntu-16.04</td>
<td>active</td>
</tr>
<tr>
<td>60c4d78fe-7c8b-4dcb-92f5-faf4e9a74d2</td>
<td>Ubuntu-16.04</td>
<td>active</td>
</tr>
<tr>
<td>b8839b29-41c0-458b-89c1-f86baaaf5c81</td>
<td>demo_snapshot</td>
<td>active</td>
</tr>
</tbody>
</table>

#### Flavor List

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>RAM</th>
<th>Disk</th>
<th>Ephemeral</th>
<th>VCPUs</th>
<th>Is Public</th>
</tr>
</thead>
<tbody>
<tr>
<td>613cbf61-4780-4df1-aa04-4a7384a7c7c6</td>
<td>standard.tiny</td>
<td>1600</td>
<td>80</td>
<td>0</td>
<td>1</td>
<td>True</td>
</tr>
<tr>
<td>03e4b20d-416d-420c-947a-e42f6346f3a8</td>
<td>hpc-gen-24core</td>
<td>240000</td>
<td>80</td>
<td>0</td>
<td>4</td>
<td>True</td>
</tr>
<tr>
<td>73c38f-4780-4df1-aa04-4a7384a7c7c6</td>
<td>hpc-gen-24core</td>
<td>120000</td>
<td>80</td>
<td>0</td>
<td>24</td>
<td>True</td>
</tr>
<tr>
<td>6179e73f-39ab-4ba5-a9ef-9697594f80fd</td>
<td>standard.large</td>
<td>160000</td>
<td>80</td>
<td>0</td>
<td>6</td>
<td>True</td>
</tr>
<tr>
<td>27eb9b2c-426b-4e4f-9ae6-7e170705914b</td>
<td>hpc-gen-16core</td>
<td>460000</td>
<td>80</td>
<td>0</td>
<td>8</td>
<td>True</td>
</tr>
<tr>
<td>8af4d7b3-4780-4df1-aa04-4a7384a7c7c6</td>
<td>hpc-gen-16core</td>
<td>80000</td>
<td>80</td>
<td>0</td>
<td>16</td>
<td>True</td>
</tr>
<tr>
<td>5d4e4eb0-4690-45fe-917f-8f7087241e67</td>
<td>hpc-gen-8core</td>
<td>400000</td>
<td>80</td>
<td>0</td>
<td>8</td>
<td>True</td>
</tr>
<tr>
<td>2d2e5a40-4780-4df1-aa04-4a7384a7c7c6</td>
<td>hpc-gen-8core</td>
<td>80000</td>
<td>80</td>
<td>0</td>
<td>16</td>
<td>True</td>
</tr>
<tr>
<td>a97f76fe-518a-42c4-914b-343e7181d9b9</td>
<td>hpc-gen-8core</td>
<td>200000</td>
<td>80</td>
<td>0</td>
<td>8</td>
<td>True</td>
</tr>
<tr>
<td>ba90a62b-999f-470a-9e62-71e31521999b</td>
<td>hpc-gen-4core</td>
<td>250000</td>
<td>80</td>
<td>0</td>
<td>4</td>
<td>True</td>
</tr>
<tr>
<td>8acbe36b-99d0-468f-82d7-3313ec2a8b2c</td>
<td>standard.large</td>
<td>200000</td>
<td>80</td>
<td>0</td>
<td>4</td>
<td>True</td>
</tr>
<tr>
<td>e7b336e4-178c-463b-8f5a-f242759014d4</td>
<td>standard.medium</td>
<td>50000</td>
<td>80</td>
<td>0</td>
<td>3</td>
<td>True</td>
</tr>
</tbody>
</table>
What is Object Storage

- Object storage is a computer data storage architecture that manages data as objects.
- Each object has three things: Data, Metadata and Globally unique identifier.
- Different from other data storage architectures like File Storage: Data as a file hierarchy and Block Storage: Data as blocks within sectors & tracks.
- Accessed via APIs at application-level, rather than via OS at system level.
- Scalable and Self healing storage.
File Storage vs Object Storage

File Name: CTSCAN_Kapoor
Created by: User1
Created on: 19-09-2017
File Type: DICOM

Object ID: 123456
Patient Name: Shubham
Patient ID: 23242
Physician Name: Dr. John
Prior1: XYZ.DICOM
Self Destruct: 2 Year

File Name: CTSCAN_Kapoor
Created by: User1
Created on: 19-09-2017
File Type: DICOM

Object Storage

Custom Metadata
System Metadata
File Storage
Where Object Storage Fits

On Basis of Data Type

- Storage of Unstructured/ Semi structured Data like Media files, web contents, Backup Archives etc.
- Cold Storage of structured and semi structured data like Databases, Sensor Data, Log files etc.
- Archiving files in place of local tape drives.
- Big Data, large data sets

On Basis of Data Size
Where Object Storage Doesn’t Fit

- Hot Data.
- Relational/OLTP Databases.
- Latency intolerant applications.
- Data with **Strict** consistency requirements.
Object Storage Around us

- Social Media Storage
- Big Data Analytics
- Offsite Backup/Archive
- Static Website Hosting
- Digital Archives
Persistent Data Volumes

- In Pouta, VM:s have only small local (virtual) disk
- Virtual data volumes can be created.
- Volumes can be attached one VM at the time.
- Volume sizes vary between 1-50 TB (or more)
- A project can have several volumes
- Management with web interface or command line client
- In a volume, data is preserved even though the VM is accidentally deleted, or become in accessible.
- Volumes are project specific, not user specific
- No backup!
Pouta : Managing Project

- A Pouta project contains a set of resources: cores, memory, storage, ip-addresses.

- A default project contains:
  - For cPouta: 8 cores, 32 GB memory, 1 TB disk space, 2 floating IP addresses.
  - For ePouta: Negotiated between customer and CSC.

- If needed you can ask for more resources for your project.

- Project members can build one or several VMs and volumes based on the granted resources.

- When VMs and Volumes are active they are consuming billing units (even if no one is using them).

- Project members can manage other members’ machines and volumes too.

- Your CSC account can be a member in many cPouta projects.
Pouta : VM States

- **Active** – Consumes billing units regardless of the real usage.
- **Shut off** – Not active, but still reserves the resources. Consumes still billing units.
- **Suspended** – Temporarily suspended. Current state saved. Can be revoked. Consumes billing units
- **Shelved** – VM is shut off, resources are freed & state is saved. Can be later on revoked if resources are free (un-shelved). Does not use billing units.
- **Terminate** – Removes the Virtual Machine.
cPouta in action

S3 protocol

WWW interface
https://pouta.csc.fi

S3 client
Virtual machine
WWW server

Pouta Object Storage
1 TB

Data volume

Mounted as a local disk

cPouta project member

SSH terminal

External user

https:
Building Application Stack on Pouta VM
Installing software to your VM

- The VM images include just the basic linux tools.
- You can/must add the tools you need with using tools like
  - **System level repository installation:**
    - Centos and Scientific linux: `sudo yum`
    - Ubuntu: `sudo apt`
    - add missing libraries and linux commands and many applications too
  - **Compile codes or download pre-compiled binaries.**
  - Install Docker and use Docker images.
  - Use Conda!
sudo command for system administration

The default user, cloud-user, does not have superuser rights, but can do admin operations with **sudo**.

**sudo linux-command-to-execute**

- Repository installations
- System libraries and directories
- User accounts

* e.g.
  
  ```
  sudo reboot
  sudo yum install nano
  sudo nano /etc/yum.conf
  sudo useradd teppo
  ```
Repository installation in CentOS and RedHat with yum

System wide installation of libraries and tools
Many application software are also available this way

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>yum help</td>
<td>List subcommands and options</td>
</tr>
<tr>
<td>sudo yum install package</td>
<td>Install a package from repository</td>
</tr>
<tr>
<td>sudo yum update</td>
<td>Update one or all packages in the system</td>
</tr>
<tr>
<td>yum provides filename</td>
<td>Check what packages include the defined file</td>
</tr>
<tr>
<td>yum search term</td>
<td>Search package names and descriptions</td>
</tr>
<tr>
<td>sudo yum localinstall package</td>
<td>Install locally available rpm file</td>
</tr>
<tr>
<td>sudo yum remove package</td>
<td>Remove a package</td>
</tr>
</tbody>
</table>
Repository installation in Ubuntu with apt-get

System wide installation of libraries and tools
Many application software are also available this way

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>apt-get --help</code></td>
<td>List commands and options</td>
</tr>
<tr>
<td><code>sudo apt-get install package</code></td>
<td>Install a package from repository</td>
</tr>
<tr>
<td><code>sudo apt-get update</code></td>
<td>Update one or all packages in the system</td>
</tr>
<tr>
<td><code>apt-file filename</code></td>
<td>Check what packages include the defined file</td>
</tr>
<tr>
<td><code>apt-cache search term</code></td>
<td>Search package names and descriptions</td>
</tr>
<tr>
<td><code>sudo yum localinstall package</code></td>
<td>Install locally available rpm file</td>
</tr>
<tr>
<td><code>sudo apt-get remove package</code></td>
<td>Remove a package</td>
</tr>
</tbody>
</table>
Conda /Bioconda

- Easy way to install software tools together with their dependencies
- Bioconda repository contains over 700 bioscience tools
- Does not need superuser privileges
- For installing conda and browsing bioconda packages, check bioconda home page:

  https://bioconda.github.io/

- Once you have conda installed, you can install application software with commands like:

  conda create -n aligners bwa bowtie hisat star
  source activate aligners
  bwa