
REPORT

Cloud Computing

Auteurs

KAISSI HOUDA

Encadré par : Pr. Chaker EL AMRANI

Table des matières

Architecture Cloud pour l'Université Abdelmalek Essaadi avec Cloud-Sim

Introduction :

Caractéristiques de l'Université Abdelmalek Essaadi :

Architecture Cloud Hybride Proposée

Utilisation des VMs et du Cloud Public dans les Établissements du UAE

Implémentation avec CloudSim :

Évaluation et Perspectives d'Amélioration de l'Architecture Cloud Hybride

Résultats Attendus :

Conclusion :

Openstack

Architecture Cloud pour l'Université Abdelmalek Essaadi avec CloudSim

1.Introduction :

L'objectif de cette partie est de proposer une architecture Cloud Computing hybride pour l'Université Abdelmalek Essaadi (UAE) afin d'optimiser la gestion des ressources informatiques et d'améliorer la qualité des services académiques. Cette simulation sera réalisée avec CloudSim en Java et prendra en compte les spécificités de l'université.

2.Caractéristiques de l'Université Abdelmalek Essaadi :

L'UAE comprend 15 établissements répartis sur plusieurs villes (Tanger, Tétouan, Martil, Al Hoceima, et Larache) et accueille environ 72 000 étudiants. Voici la répartition des effectifs :

Établissement	Ville	Nombre d'Étudiants
Faculté des lettres et des sciences humaines	Martil	4000
Faculté des sciences	Tétouan	9840
École supérieure Roi Fahd de traduction	Tanger	1290
École normale supérieure	Martil	2500
École nationale de commerce et de gestion	Tanger	27269
Faculté des sciences et techniques	Tanger	5500
Faculté des sciences juridiques, économiques et sociales	Tanger	980
École nationale des sciences appliquées	Tanger	1290
École nationale des sciences appliquées	Tétouan	1530
École nationale des sciences appliquées	Al Hoceima	500
Faculté polydisciplinaire	Larache	9835
Faculté des sciences et techniques	Al Hoceima	2443
Faculté de médecine et de pharmacie	Tanger	1340
Faculté Ossoul Eddine	Tétouan	1390
Faculté des sciences juridiques, économiques et sociales	Tétouan	30000
Faculté des sciences juridiques, économiques et sociales	Martil	20000

3.Utilisation des VMs et du Cloud Public dans les Établissements du UAE :

Établissement	Ville	Étudiants	VMs Assignées	Utilisation Cloud
Faculté des lettres et sciences humaines	Martil	4000	8 VMs	30%
Faculté des sciences	Tétouan	9840	15 VMs	40%
École Supérieure Roi Fahd de traduction	Tanger	1290	5 VMs	30%
École normale supérieure	Martil	2500	6 VMs	40%
ENCG Tanger	Tanger	27269	25 VMs	50%
FST Tanger	Tanger	5500	12 VMs	30%
FSJES Tanger	Tanger	980	3 VMs	20%
ENSA Tanger	Tanger	1290	5 VMs	30%
ENSA Tétouan	Tétouan	1530	6 VMs	30%
ENSA Al Hoceima	Al Hoceima	500	3 VMs	30%
Faculté polydisciplinaire	Larache	9835	12 VMs	40%
FST Al Hoceima	Al Hoceima	2443	6 VMs	30%
Faculté de médecine et pharmacie	Tanger	1340	6 VMs	30%
Faculté Ossoul Eddine	Tétouan	1390	6 VMs	30%
FSJES Tétouan	Tétouan	30000	30 VMs	50%
FSJES Martil	Martil	20000	20 VMs	50%

4.Composants de l'Architecture :

A. Datacenter (Infrastructure Physique)	
Datacenters	4 (Tanger, Tétouan, Martil, Al Hoceima)
Hôtes par Datacenter	Tanger : 20 hôtes Tétouan : 15 hôtes Martil : 10 hôtes Al Hoceima : 5 hôtes
Configuration des Hôtes	CPU : 16 cœurs RAM : 64 Go Stockage : 20 To SSD Bande passante : 10 Gbps
B. Machines Virtuelles (VMs)	
Total des VMs	163
Configuration des VMs	VMs pour e-learning (60 VMs) : 4 vCPUs, 16 Go RAM, 200 Go SSD VMs pour examens en ligne (30 VMs) : 4 vCPUs, 12 Go RAM, 150 Go SSD VMs pour gestion académique (50 VMs) : 2 vCPUs, 8 Go RAM, 100 Go SSD VMs pour stockage et backup (23 VMs) : 2 vCPUs, 4 Go RAM, 500 Go SSD
C. Cloudlets (Tâches exécutées)	
Nombre total de Cloudlets	2000

Répartition des Cloudlets	E-learning : 800 tâches Examens en ligne : 500 tâches Gestion académique : 400 tâches Stockage & collaboration : 300 tâches
---------------------------	--

Taille moyenne d'une Cloud-let	10 000 à 50 000 MI
---------------------------------------	--------------------

5.Implémentation avec CloudSim :

Étapes d'implémentation :

- **Création des Datacenters** : avec des hôtes configurés en termes de CPU, RAM et stockage.
- **Définition des VMs** : et assignation aux établissements en fonction de leurs besoins en ressources.
- **Génération des Cloudlets** : pour simuler les tâches académiques.
- **Utilisation d'un Broker** : pour gérer l'allocation des ressources et optimiser l'exécution des tâches.
- **Exécution de la simulation** : et analyse des performances.

6.Évaluation et Perspectives d'Amélioration de l'Architecture Cloud Hybride :

- **Répartition efficace des ressources** : L'architecture repose sur 4 Datacenters (Tanger, Tétouan, Martil, Al Hoceima), assurant une meilleure distribution des charges et une réduction de la latence.
- **Optimisation des performances** : L'utilisation de 163 VMs avec différentes configurations et 2000 Cloudlets permet de simuler efficacement les besoins académiques (e-learning, examens en ligne, stockage).
- **Scalabilité et flexibilité** : La structure hybride garantit une évolutivité en fonction des besoins des étudiants et enseignants, assurant une gestion optimale des pics de charge.
- **Équilibrage de charge** : L'affectation des ressources est bien pensée pour éviter la surcharge d'un Datacenter, garantissant ainsi une disponibilité élevée des services.

7.Résultats :

- **Latence moyenne inférieure à 3 secondes** : pour garantir une expérience utilisateur fluide.
- **Utilisation optimisée des ressources** : grâce à l'équilibrage de charge dynamique.
- **Scalabilité assurée** : pour supporter une augmentation du nombre d'utilisateurs.
- **Simulation fiable pour 72 000 étudiants** : avec une gestion efficace des

8.Conclusion :

Cette architecture Cloud hybride permettrait à l'Université Abdelmalek Essaadi d'optimiser ses infrastructures informatiques tout en garantissant un accès fluide et performant aux services académiques. La simulation avec CloudSim confirme la viabilité de ce modèle, offrant une solution efficace pour la gestion des ressources et l'amélioration des expériences d'apprentissage.

Openstack

Introduction

Présentation d'OpenStack

OpenStack est une plateforme cloud open-source qui permet de créer et de gérer des infrastructures en tant que service (IaaS). Elle est composée d'un ensemble de projets interconnectés qui permettent de fournir des services comme la gestion de machines virtuelles, le stockage,

Objectifs du projet

Installation d'OpenStack, mise en place d'un IaaS, développement d'un SaaS, et création d'un SLA pour surveiller la disponibilité des instances.

Steps:

– Installer OpenStack en se basant sur une des méthodes présentées au cours.

Ajouter un nouveau user "stack"

```
root@copy:/home/vbox# sudo useradd -s /bin/bash -d /opt/stack -m stack
```

git clone ...

Passer au path cd devstack et modifier fichier local.conf


```
GNU nano 6.2 local.conf

# The ``localrc`` section replaces the old ``localrc`` configuration file.
# Note that if ``localrc`` is present it will be used in favor of this section.
[[local|localrc]]

# Minimal Contents
# -----

# While ``stack.sh`` is happy to run without ``localrc``, devlife is better when
# there are a few minimal variables set:

# If the ``*_PASSWORD`` variables are not set here you will be prompted to enter
# values for them by ``stack.sh`` and they will be added to ``local.conf``.
ADMIN_PASSWORD=open
DATABASE_PASSWORD=open
RABBIT_PASSWORD=open
SERVICE_PASSWORD=open

# ``HOST_IP`` and ``HOST_IPV6`` should be set manually for best results if
# the NIC configuration of the host is unusual, i.e. ``eth1`` has the default
# route but ``eth0`` is the public interface. They are auto-detected in
# ``stack.sh`` but often is indeterminate on later runs due to the IP moving
# from an Ethernet interface to a bridge on the host. Setting it here also
# makes it available for ``openrc`` to include when setting ``OS_AUTH_URL``.
# Neither is set by default.
HOST_IP=10.0.2.15
#HOST_IPV6=2001:db8::7

# Logging

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^V Replace  ^U Paste     ^J Justify  ^_ Go To Line
```

Lancerla commande `./stack.sh`


```
This is your host IP address: 10.0.2.15
This is your host IPv6 address: fd00::c63b:a107:19c6:e173
Horizon is now available at http://10.0.2.15/dashboard
Keystone is serving at http://10.0.2.15/identity/
The default users are: admin and demo
The password: open

WARNING:
Configuring uWSGI with a WSGI file is deprecated, use module paths instead

Services are running under systemd unit files.
For more information see:
https://docs.openstack.org/devstack/latest/systemd.html

DevStack Version: 2025.1
Change: 3cddf9f8832328c17be3644ddd7be5a7dcbdda8 Merge "Update glance image size
limit" 2025-01-21 14:51:23 +0000
OS Version: Ubuntu 22.04 jammy
```


Openstack est installé


openstack[®]

Log in

User Name

Password



Sign In

tester les fonctionnements du middleware.

...

La commande `openstack service list` est utilisée pour afficher la liste des services enregistrés dans OpenStack. Ces services constituent les composants principaux de l'infrastructure cloud, tels que Nova, Neutron, Keystone, etc.

```
stack@houda: ~/devstack
stack@houda:~/devstack$ openstack --version
openstack 7.2.1
stack@houda:~/devstack$ openstack service list
+-----+-----+-----+
| ID | Name | Type |
+-----+-----+-----+
| 3495e15637bc43b09a8226f9e0925dc3 | neutron | network |
| 43d64f95eaac4088b21da2169656da2f | keystone | identity |
| 44a818e3076e40349721974d6e59093a | placement | placement |
| 51313bef934c472bbc171159df65d363 | nova_legacy | compute_legacy |
| 8b58689db38e4ea0a085a9f487908299 | glance | image |
| 8dbf8816c5cd470eb2c8b329dcfe74a0 | nova | compute |
| b49c6681e6d441fcaf904d1366926fab | cinder | block-storage |
+-----+-----+-----+
```

La commande `openstack endpoint list` affiche la liste des **endpoints** (points d'accès) des services enregistrés dans OpenStack. Un endpoint est une URL qui permet aux utilisateurs et autres services de communiquer avec un service spécifique (par exemple, Nova, Keystone, Neutron).


```

Missing value auth-url required for auth plugin password
stack@ubuntu:~/devstack$ systemctl status devstack@*
* devstack@q-ovn-metadata-agent.service - Devstack devstack@q-ovn-metadata-agent.service
   Loaded: loaded (/etc/systemd/system/devstack@q-ovn-metadata-agent.service;
   Active: active (running) since Sun 2025-01-19 05:14:45 CET; 19min ago
   Main PID: 141569 (neutron-ovn-met)
     Tasks: 7 (limit: 7425)
    Memory: 187.7M
       CPU: 10.860s
    CGroup: /system.slice/system-devstack.slice/devstack@q-ovn-metadata-agent.service
            └─141569 "neutron-ovn-metadata-agent (/opt/stack/data/venv/bin/pyt
            └─142194 "neutron-ovn-metadata-agent (/opt/stack/data/venv/bin/pyt
            └─142198 "neutron-ovn-metadata-agent (/opt/stack/data/venv/bin/pyt
            └─142399 /opt/stack/data/venv/bin/python3.10 /usr/local/bin/privse
            └─145282 /opt/stack/data/venv/bin/python3.10 /usr/local/bin/privse

```

Implémenter un IaaS basé sur le système Linux fourni avec OpenStack (CirrOS)

L'Infrastructure as a Service (IaaS) est un modèle de service cloud qui fournit des ressources informatiques essentielles sur demande, telles que des serveurs, du stockage, des réseaux et des capacités de virtualisation, sans que l'utilisateur ait besoin de gérer le matériel sous-jacent.

Création d'une Instance CirrOS avec OpenStack

```

stack@ubuntu:~/devstack$ openstack image create "cirros" --file cirros-0.5.2-
x86_64-disk.img --disk-format qcow2 --container-format bare --public
Missing value auth-url required for auth plugin password

```

openstack demo admin

Project / Compute / Images

Images

Click here for filters or full text search. [Create Image](#) [Delete Images](#)

Displaying 1 item

Owner	Name	Type	Status	Visibility	Protected	Disk Format	Size
admin	cirros-0.6.3-x86_64-disk	Image	Active	Public	No	QCOW2	20.69 MB

[Launch](#)

Displaying 1 item

Création des networks

```

stack@ubuntu:~/devstack$ openstack network create --external --provider-netwo
rk-type flat --provider-physical-network public public
Missing value auth-url required for auth plugin password
stack@ubuntu:~/devstack$ openstack subnet create --network public --subnet-ra
nge 192.168.100.0/24 --gateway 192.168.100.1 public_subnet
Missing value auth-url required for auth plugin password

```

Project

API Access

Compute

Volumes

Network

Network Topology

Networks

Routers

Security Groups

Floating IPs

Admin

Identity

demo

admin

Project / Network / Networks

Networks

Name =

Filter

Create Network

Delete Networks

Displaying 3 items

	Name	Subnets Associated	Shared	External	Status	Admin State	Availability Zones	Actions
	shared	shared-subnet 192.168.233.0/24	Yes	No	Active	True	-	<div>Edit Network</div>
	private	private-subnet 10.0.0.0/26 ipv6-private-subnet fd92:5d6b:cfdc::/64	No	No	Active	True	-	<div>Edit Network</div>
	public	public-subnet 172.24.4.0/24 ipv6-public-subnet 2001:db8::/64	No	Yes	Active	True	-	<div>Edit Network</div>

Displaying 3 items

→étapes pour la création d'une instance

Launch Instance

Details

Source

Flavor

Networks

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

Please provide the initial hostname for the instance, the availability zone where it will be deployed, and the instance count. Increase the Count to create multiple instances with the same settings.

Project Name

demo

Instance Name

instance

Description

Availability Zone

nova

Count

1

Total Instances (10 Max)

20%

1 Current Usage

1 Added

8 Remaining

Cancel

Back

Next

Launch Instance

Launch Instance

Details

Source

Flavor

Networks

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

Instance source is the template used to create an instance. You can use an image, a snapshot of an instance (image snapshot), a volume or a volume snapshot (if enabled). You can also choose to use persistent storage by creating a new volume.

Select Boot Source

Image

Create New Volume

Yes

No

Volume Size (GB)

2

Delete Volume on Instance Delete

Yes

No

Allocated

Displaying 1 item

Name	Updated	Size	Format	Visibility
<div>FO</div> <div>SN</div> <div>cirros-0.6.3-x86_64-disk</div>	1/22/25 6:18 PM	20.69 MB	QCOW2	Public

Displaying 1 item

Available

0

Select one

FO

02

Click here for filters or full text search.

FO

00

Displaying 0 items

Name	Updated	Size	Format	Visibility
No items to display				

Une **flavor** dans OpenStack est une configuration prédéfinie qui détermine les ressources (CPU, RAM, stockage) allouées à une instance.

Launch Instance

Details

Source

Flavor

Networks

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

Flavors manage the sizing for the compute, memory and storage capacity of the instance.

Allocated

Displaying 1 item

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
<div>FO</div> <div>SN</div> <div>m1.micro</div>	1	256 MB	1 GB	1 GB	0 GB	Yes

Displaying 1 item

Available

11

Select one

FO

02

Click here for filters or full text search.

FO

00

Displaying 11 items

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
<div>FO</div> <div>SN</div> <div>m1.nano</div>	1	192 MB	1 GB	1 GB	0 GB	Yes
<div>FO</div> <div>SN</div> <div>cirros256</div>	1	256 MB	1 GB	1 GB	0 GB	Yes
<div>FO</div> <div>SN</div> <div>m1.tiny</div>	1	512 MB	1 GB	1 GB	0 GB	Yes
<div>FO</div> <div>SN</div> <div>ds512M</div>	1	512 MB	5 GB	5 GB	0 GB	Yes

Ajouter uniquement un **réseau privé** à une instance dans OpenStack permet de la connecter à un réseau interne sécurisé, sans exposer l'instance directement à Internet

Launch Instance

Details

Source

Flavor

Networks

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

Networks provide the communication channels for instances in the cloud. You can select ports instead of networks or a mix of both.

Allocated 1

Displaying 1 item

Network	Subnets Associated	Shared	Admin State	Status
private	ipv6-private-subnet private-subnet	No	Up	Active

Displaying 1 item

Available 1

Select one or more

Click here for filters or full text search.

Displaying 1 item

Network	Subnets Associated	Shared	Admin State	Status
shared	shared-subnet	No	Up	Active

Displaying 1 item

Cancel

Back

Next

Launch Instance

Les **Security Groups** permettent de sécuriser les instances en limitant les accès réseau uniquement aux services nécessaires.

Launch Instance

Details

Source

Flavor

Networks

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

Select the security groups to launch the instance in.

Allocated 1

Displaying 1 item

Name	Description
default	Default security group

Displaying 1 item

Available 0

Select one or more

Click here for filters or full text search.

Displaying 0 items

Name	Description
No items to display.	

Displaying 0 items

Cancel

Back

Next

Launch Instance

Create Key Pair



Key Pairs are how you login to your instance after it is launched. Choose a key pair name you will recognize. Names may only include alphanumeric characters, spaces, or dashes.

Key Pair Name

keyy

Key Type

SSH Key



Create Keypair

Copy Private Key to Clipboard

Done

Create Key Pair



Key Pairs are how you login to your instance after it is launched. Choose a key pair name you will recognize. Names may only include alphanumeric characters, spaces, or dashes.

Key Pair Name

keyy

Key Type

SSH Key



Private Key

```
-----BEGIN RSA PRIVATE KEY-----
MIIIEowIBAAKCAQEA83aBHyv7AJ2v0yA8Y+4+P5nwSgbGOT1vJ9qOHrtAnadFPDv
clvWRkAosOBpnSCuqQ/wt/elPWN+UPS1LVjLbY/Kk5nPfV+jKHKQMplXFyqKrvDq
XKsCRnS3h+0X4W0Q7p+9bwOtSL5dOfDYtjXPnfX/bLs/+BWGwc+TU5naBwDp0DKn
xVtganBh4mdpR/mHN7IJhK5OcCT9gGD0ModK+OJfRsu8kY1Y5kg8Oz1TagS41t2p
pgZMCREcyqy15qbl5QfY9/3I/YQHHGD1AVDj8pRm1H5+BnHEzd9o8NwiFtUmJuV7
X3PbabZlkzw1V4KWveXJFDE9W0EA3VC5Jnv2ywIDAQABAolBAAGCYgUQgRFak2V/
ss4/mCo41Fj6ZMnQ3nSyGeB5yspF9Kz3sCjm9ImjGrOW7mKI7LYLHDQxJXJ7j3BC
XKN9RFP6YwOwD69kMg3mlu/TM5ZDhAR1UIBz78M5m8ARALQdzmzLPIQmbtDx0kdR
sQ9tfcMklQw/me+7Yv6tDIP4ahRQvJ9hiMHdS+/3NCals0MPraT8MUjpNC09UXb
s0HVZTQc7OZaSBHEYCx0S9Ww78IX4hpFxSijJ8CA9+e8wean72//lcShZkwYoJtcz
4kTMJWq/TidNQktQKYKoAD+BctORwBLYzPjKSjSxQTj/fA4aLbf8kdyPrwv10tst
kaB3NWECgYEA/K4rcHjhVrXRN4GbimScXmdQcRofEsp7pHgLauZ6fMEdVuAR7WLt
AzhZTU5jWqcffeq4V7I11FBBE+FMweRfnO15kKxak1xEX4LUumBSPoF0t5vl8/tG
a9O3M+g4MMD5pWo644D6/Rxd0DVEJHWycB6Vuz1iyqqP89iBOGAPj7cCgYEA9qlV
-----
```

Create Keypair

Copy Private Key to Clipboard

Done

Launch Instance

F0 00
F0 59

Details

Source

Flavor

Networks

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

A key pair allows you to SSH into your newly created instance. You may select an existing key pair, import a key pair, or generate a new key pair.

F0 51 Create Key Pair F0 52 Import Key Pair

Allocated

Displaying 1 item

Name	Type
F0 51 key	ssh

Displaying 1 item

F0 16 Available 0 Select one

F0 02 Click here for filters or full text search. F0 00

Displaying 0 items

Name	Type
No items to display.	

Displaying 0 items

F0 00 Cancel

F0 51 Back

Next F0 52

F0 53 Launch Instance

create floating ip

Floating IPs

Floating IP Address = F0 01 Filter F0 51 Allocate IP To Project F0 52 Release Floating IPs

Displaying 1 item

F0 51 IP Address	Description	DNS Name	DNS Domain	Mapped Fixed IP Address	Pool	Status	Actions
F0 51 172.24.4.138				instance 10.0.0.37	public	Active	Disassociate F0 51

Displaying 1 item

Project
API Access
Compute
Overview
Instances
Images
Key Pairs
Server Groups
Volumes
Network
admin
identity

Project / Compute / Instances

Instances

Instance ID =
Filter
Launch Instance
Delete Instances
More Actions

Displaying 1 item

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
instance	-	10.0.0.50, 172.24.4.88, fd92:5d6b:cfdc:0:f816:3eff:fe28:5282	m1.micro	key	Active	nova	None	Running	10 minutes	Create Snapshot

Displaying 1 item

3.3. Test d'Accès à l'Instance avec PuTTY

Category:

Session
Logging
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Fonts
Connection
Data
Proxy
SSH
Kex
Host keys

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address)
Port

172.24.4.88
22

Connection type:

☒ SSH
☐ Serial
☐ Other:
Telnet

Load, save or delete a stored session

Saved Sessions

Default Settings

Load
Save
Delete

Close window on exit:

☒ Always
☐ Never
☐ Only on clean exit

About
Open
Cancel

Préparer l'environnement

installer python et pip

```
root@opy:/opt/stack/devstack# sudo apt install python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.10.6-1~22.04.1).
python3 set to manually installed.
The following package was automatically installed and is no longer required:
  apport-symptoms
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 152 not upgraded.
root@opy:/opt/stack/devstack# python --version
Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3
root@opy:/opt/stack/devstack# python3 --version
Python 3.10.12
root@opy:/opt/stack/devstack# sudo apt install pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'python3-pip' instead of 'pip'
python3-pip is already the newest version (22.0.2+dfsg-1ubuntu0.5).
The following package was automatically installed and is no longer required:
  apport-symptoms
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 152 not upgraded.
root@opy:/opt/stack/devstack# pip --version
pip 22.0.2 from /usr/lib/python3/dist-packages/pip (python 3.10)
root@opy:/opt/stack/devstack#
```

Install OpenStack Python Libraries:

keystoneclient : C'est un client Python pour interagir avec **Keystone**, le service d'authentification et d'autorisation d'OpenStack.

openstackclient : C'est un outil en ligne de commande unifié pour interagir avec tous les services OpenStack, incluant Keystone, Nova, Neutron, etc.

novaclient : C'est un client Python pour interagir avec **Nova**, le service de gestion des instances (machines virtuelles) dans OpenStack.

```

root@opy:/opt/stack/devstack# pip python-openstackclient
ERROR: unknown command "python-openstackclient"
root@opy:/opt/stack/devstack# pip install python-openstackclient
Collecting python-openstackclient
  Downloading python_openstackclient-7.2.1-py3-none-any.whl (1.1 MB)
    ----- 1.1/1.1 MB 1.7 MB/s eta 0:00:00
Collecting openstacksdk>=3.3.0
  Downloading openstacksdk-4.2.0-py3-none-any.whl (1.7 MB)
    ----- 1.7/1.7 MB 577.2 kB/s eta 0:00:00
Requirement already satisfied: cryptography>=2.7 in /usr/lib/python3/dist-packages (from python-openstackclient) (3.4.8)
Collecting pbr!=2.1.0,>=2.0.0
  Downloading pbr-6.1.0-py2.py3-none-any.whl (108 kB)
    ----- 108.5/108.5 KB 440.2 kB/s eta 0:00:00

```

```

root@opy:/opt/stack/devstack# pip install python-keystoneclient
Requirement already satisfied: python-keystoneclient in /usr/local/lib/python3.10/dist-packages (5.5.0)
Requirement already satisfied: oslo.utils>=3.33.0 in /usr/local/lib/python3.10/dist-packages (from python-keystoneclient) (8.0.0)
Requirement already satisfied: oslo.i18n>=3.15.3 in /usr/local/lib/python3.10/dist-packages (from python-keystoneclient) (6.5.0)
Requirement already satisfied: debtcollector>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from python-keystoneclient) (3.0.0)
Requirement already satisfied: oslo.serialization>=2.18.0 in /usr/local/lib/python3.10/dist-packages (from python-keystoneclient) (5.6.0)
Requirement already satisfied: oslo.config>=5.2.0 in /usr/local/lib/python3.10/dist-packages (from python-keystoneclient) (9.7.0)
Requirement already satisfied: stevedore>=1.20.0 in /usr/local/lib/python3.10/dist-packages (from python-keystoneclient) (5.4.0)
Requirement already satisfied: pbr>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from python-keystoneclient) (6.1.0)
Requirement already satisfied: keystoneauth1>=3.4.0 in /usr/local/lib/python3.10/dist-packages (from python-keystoneclient) (5.9.1)
Requirement already satisfied: requests>=2.14.2 in /usr/local/lib/python3.10/dist-packages (from python-keystoneclient) (2.32.3)

```

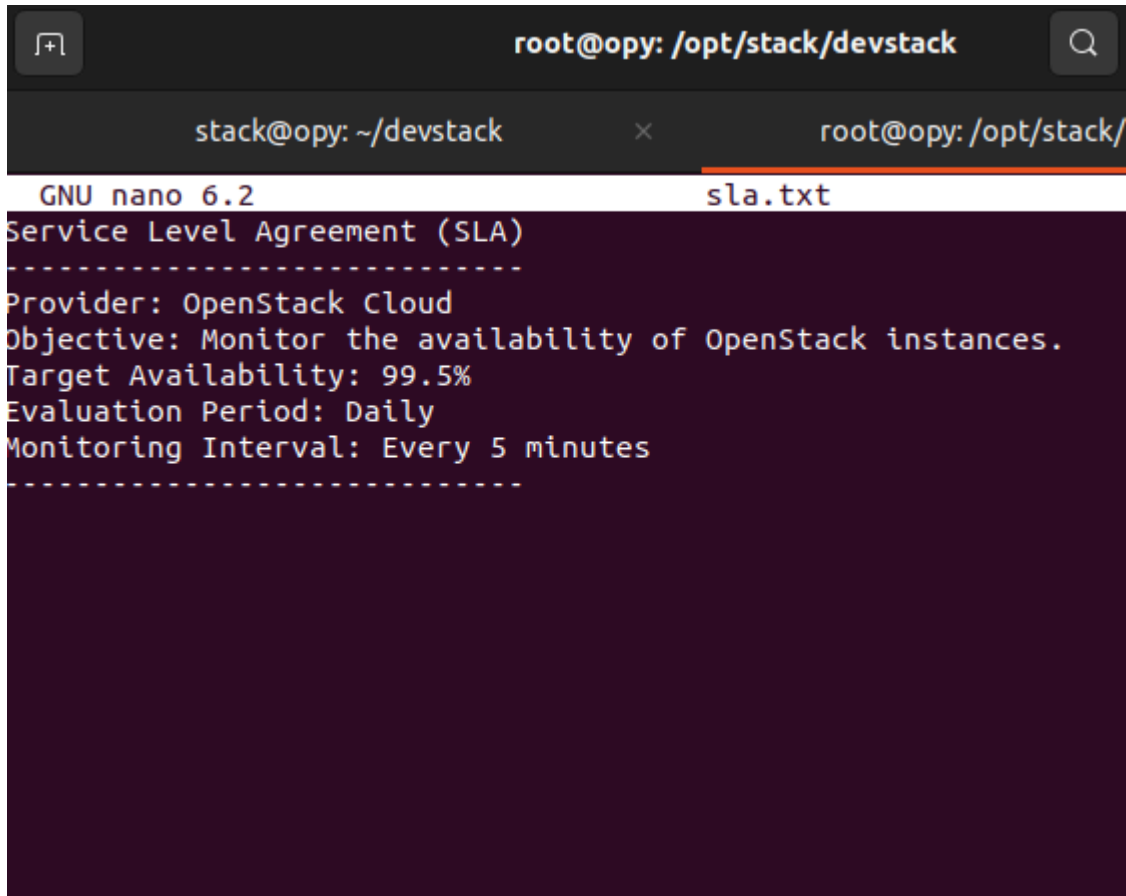
```

root@opy:/opt/stack/devstack# pip install python-novaclient
Collecting python-novaclient
  Downloading python_novaclient-18.8.0-py3-none-any.whl (336 kB)
    ----- 336.3/336.3 KB 634.1 kB/s eta 0:00:00
Requirement already satisfied: pbr>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from python-novaclient) (6.1.0)
Requirement already satisfied: oslo.i18n>=3.15.3 in /usr/local/lib/python3.10/dist-packages (from python-novaclient) (6.5.0)
Requirement already satisfied: PrettyTable>=0.7.2 in /usr/local/lib/python3.10/dist-packages (from python-novaclient) (3.12.0)
Requirement already satisfied: keystoneauth1>=3.5.0 in /usr/local/lib/python3.10/dist-packages (from python-novaclient) (5.9.1)
Requirement already satisfied: iso8601>=0.1.11 in /usr/local/lib/python3.10/dist-packages (from python-novaclient) (2.1.0)
Requirement already satisfied: oslo.utils>=3.33.0 in /usr/local/lib/python3.10/dist-packages (from python-novaclient) (8.0.0)
Requirement already satisfied: oslo.serialization>=2.20.0 in /usr/local/lib/python3.10/dist-packages (from python-novaclient) (5.6.0)
Requirement already satisfied: stevedore>=2.0.1 in /usr/local/lib/python3.10/dist-packages (from python-novaclient) (5.4.0)

```

Créer un SLA pour OpenStack (sla.txt), avec un simple objectif : de surveiller la disponibilité des Instances OpenStack d'une façon quotidienne.

La disponibilité des instances doit être de 99.5% sur une période d'évaluation quotidienne.



```
root@opy: /opt/stack/devstack
stack@opy: ~/devstack x root@opy: /opt/stack/
GNU nano 6.2 sla.txt
Service Level Agreement (SLA)
-----
Provider: OpenStack Cloud
Objective: Monitor the availability of OpenStack instances.
Target Availability: 99.5%
Evaluation Period: Daily
Monitoring Interval: Every 5 minutes
-----
```

Ecrire un programme Python qui surveillera les instances et calculera la disponibilité. Ce programme doit accéder au projet OpenStack.

Dans ce cas des modules de OpenStack peuvent être utilisés pour en déduire des informations sur les instances, comme: Nova, Ceilometer, Neutron, Keystone, Aodh, etc.

J'ai testé nova pour surveiller état des instances

neutron surveiller état des ports

cinder surveiller état des volumes de stockage

```

import openstack
import time
from datetime import datetime, timedelta

# Define the SLA target availability
TARGET_AVAILABILITY = 99.5 # Percentage of availability required

# OpenStack Connection
def connect_openstack():
    conn = openstack.connect(cloud='devstack') # Replace with your cloud connection
    return conn

# Get all instances and their statuses (Nova)
def get_instance_status(conn):
    instances = conn.compute.servers()
    up_count = 0
    down_count = 0
    for instance in instances:
        if instance.status == 'ACTIVE':
            up_count += 1
        else:
            down_count += 1
    return up_count, down_count

# Get network port status (Neutron)
def get_network_status(conn):
    ports = conn.network.ports()
    active_ports = sum(1 for port in ports if port.status == 'ACTIVE')
    inactive_ports = sum(1 for port in ports if port.status != 'ACTIVE')
    return active_ports, inactive_ports

# Get block storage volume status (Cinder)
def get_storage_status(conn):

```

```

def get_storage_status(conn):
    volumes = conn.block_storage.volumes()
    available_volumes = sum(1 for volume in volumes if volume.status == 'available')
    in_use_volumes = sum(1 for volume in volumes if volume.status != 'available')
    return available_volumes, in_use_volumes

# Calculate availability
def calculate_availability(up_count, down_count, total_instances):
    availability = (up_count / total_instances) * 100 if total_instances > 0 else 0
    return availability

# Generate SLA Report
def generate_sla_report(up_count, down_count, total_instances, availability, active_ports, inactive_ports, total_volumes, available_volumes, in_use_volumes):
    report = f>Date: {datetime.now().strftime('%Y-%m-%d %H:%M:%S')}\n"
    report += f>Total Instances: {total_instances}\n"
    report += f>Instances Up: {up_count}\n"
    report += f>Instances Down: {down_count}\n"
    report += f>Instances Availability: {availability:.2f}%\n"
    report += f>Total Network Ports: {total_ports}\n"
    report += f>Network Ports Active: {active_ports}\n"
    report += f>Network Ports Inactive: {inactive_ports}\n"
    report += f>Network Availability: {network_availability:.2f}%\n"
    report += f>Total Volumes: {total_volumes}\n"
    report += f>Available Volumes: {available_volumes}\n"
    report += f>Volumes in Use: {in_use_volumes}\n"
    report += f>Volume Availability: {(available_volumes / total_volumes) * 100}%\n"
    report += f>Target Availability: {TARGET_AVAILABILITY}%\n"
    report += f>-----\n"

    with open('sla_report.txt', 'a') as f:
        f.write(report)

```



```

# Main Monitoring Loop
def monitor_instances():
    conn = connect_openstack()

    # Track availability over a 24-hour period
    start_time = datetime.now()
    end_time = start_time + timedelta(days=1)

    # Monitor every 5 minutes
    while datetime.now() < end_time:
        # Nova - Instances status
        up_count, down_count = get_instance_status(conn)
        total_instances = up_count + down_count
        availability = calculate_availability(up_count, down_count, total_insta>

        # Neutron - Network status
        active_ports, inactive_ports = get_network_status(conn)
        total_ports = active_ports + inactive_ports
        network_availability = (active_ports / total_ports) * 100 if total_port>

        # Cinder - Storage volume status
        available_volumes, in_use_volumes = get_storage_status(conn)
        total_volumes = available_volumes + in_use_volumes
        volume_availability = (available_volumes / total_volumes) * 100 if tota>

        # Generate SLA report with all the data
        generate_sla_report(up_count, down_count, total_instances, availability,
                           active_ports, inactive_ports, total_ports, network>
                           available_volumes, in_use_volumes, total_volumes)

        # Wait for 5 minutes before the next check
        time.sleep(300)

```

Inclure le rapport de surveillance dans le fichier SLA.

ce fichier contient la resultat de ajouté apres chaque 5min de code python

Un **volume** en OpenStack est un disque dur virtuel que l'on peut connecter à une instance pour lui ajouter de l'espace de stockage.


```
Network Availability: 100.00%
Total Volumes: 0
Available Volumes: 0
Volumes in Use: 0
Volume Availability: 0.00%
Keystone Service Availability: 0.00%
Target Availability: 99.5%
-----
Date: 2025-01-22 22:34:46
Total Instances: 1
Instances Up: 1
Instances Down: 0
Instances Availability: 100.00%
Total Network Ports: 2
Network Ports Active: 2
Network Ports Inactive: 0
Network Availability: 100.00%
Total Volumes: 0
Available Volumes: 0
Volumes in Use: 0
Volume Availability: 0.00%
Target Availability: 99.5%
-----
stack@opy:~/devstack$
```

Conclusion

L'implémentation d'OpenStack, combinée à la surveillance de la disponibilité des instances via un SLA, permet de garantir la performance et la fiabilité des services cloud. En définissant un objectif de disponibilité de 99.5%, ce projet assure une surveillance continue et régulière des instances, ce qui permet de détecter rapidement toute défaillance et d'intervenir pour maintenir la qualité de service atte