

# UNIVERSITÀ DEGLI STUDI DI PERUGIA

## DIPARTIMENTO DI MATEMATICA E INFORMATICA



Sesta esercitazione:

CALCOLO DISTRIBUITO E PARALLELO

---

## CLOUD COMPUTING TRAMITE OPENSTACK

Studente: Manuel Severi



# INDICE

1	Introduzione	1
	1.1 cloud computing	1
	1.2 vantaggi del cloud computing	2
	1.3 tipi di cloud computing	4
	1.4 svantaggi del cloud computing	7
2	Strumenti utilizzati	8
	2.1 OpenStack	8
	2.2 Ubuntu server Edition	11
3	Fase implementativa del progetto	12
4	Fase finale del progetto	24
5	Conclusione	26

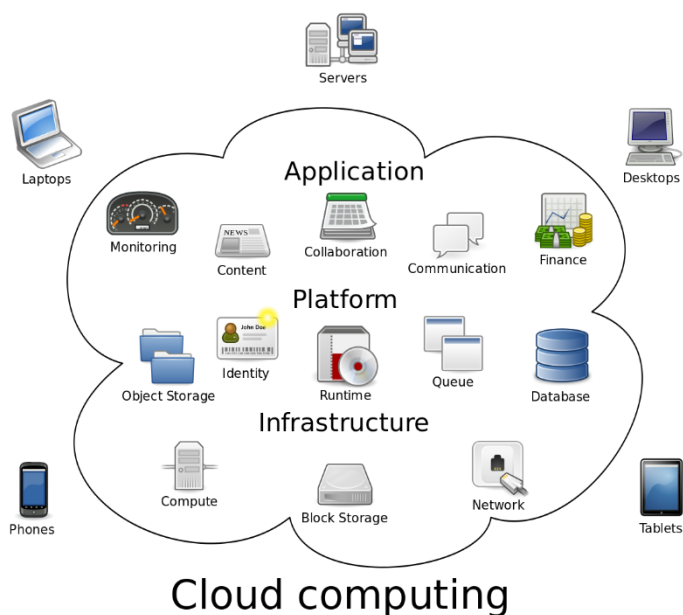
# 1 | INTRODUZIONE

Con la seguente relazione si va ad installare e utilizzare OpenStack, che ci permette di comprendere come funziona il Cloud Computing.

## CLOUD COMPUTING

In informatica con il termine inglese cloud computing (in italiano nuvola informatica) si indica un paradigma di erogazione di servizi offerti on demand da un fornitore ad un cliente finale attraverso la rete Internet (come l'archiviazione, l'elaborazione o la trasmissione dati), a

partire da un insieme di risorse preesistenti, configurabili e disponibili in remoto sotto forma di architettura distribuita. Il cloud computing appunto consiste nella distribuzione on-demand delle risorse IT tramite Internet, con una tariffazione basata sul consumo. Piuttosto che acquistare, possedere e mantenere i data center e i server fisici, è possibile accedere a servizi tecnologici, quali capacità di calcolo, storage e database, sulla base delle proprie necessità affidandosi a un fornitore cloud come ad esempio Amazon Web Services (AWS) oppure Azure di Microsoft. il cloud computing è utilizzato da organizzazioni di ogni tipo, dimensione e settore per una vastissima gamma di casi d'uso,



quali backup dei dati, disaster recovery, e-mail, desktop virtuali, sviluppo e test di software, analisi di Big Data e applicazioni Web destinate ai clienti. Per esempio, le aziende del settore sanitario utilizzano il cloud per sviluppare dei trattamenti più personalizzati per i propri pazienti. Le aziende per i servizi finanziari utilizzano il cloud come base dei propri sistemi di rilevamento e prevenzione di attività fraudolente. I realizzatori di videogame utilizzano il cloud per sviluppare videogiochi online per milioni di giocatori in tutto il mondo.

## VANTAGGI DEL CLOUD COMPUTING

- **AGILITÀ** → Il cloud permette di accedere in modo semplice a diverse tecnologie, così da poter innovare più rapidamente e costruire pressoché qualsiasi cosa tu possa immaginare. A seconda delle tue necessità, potrai aumentare rapidamente risorse quali servizi infrastrutturali, come calcolo, storage e database, fino a capacità per Internet of Things, machine learning, data lake e analisi, e molto altro. Puoi anche distribuire servizi tecnologici in pochi minuti e trasformare le tue idee in implementazioni con una velocità incredibilmente maggiore rispetto al passato. Ciò ti offre la libertà di sperimentare, testare nuove idee per differenziare le esperienze dei clienti e trasformare il tuo business.
- **ELASTICITÀ** → Grazie al cloud computing, non dovrai allocare anticipo una quantità maggiore di risorse di quante siano necessarie così da gestire i picchi nei livelli di attività aziendali in futuro. Piuttosto, potrai allocare la quantità di risorse di cui hai bisogno in realtà. Potrai ridimensionare tali

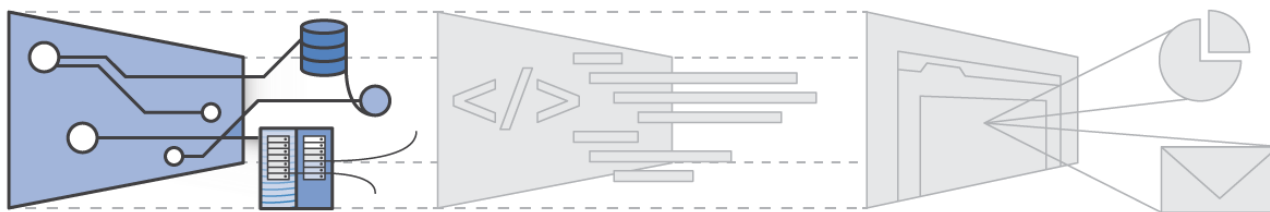
risorse per aumentare o ridurre in modo istantaneo le capacità adattandole alle necessità dell'azienda.

- **RISPARMIO SUI COSTI →** Il cloud permette di evitare spese di capitale (per esempio, per data center e server fisici) in favore di una spesa variabile, pagando solo per le risorse IT realmente consumate. Inoltre, le spese variabili sono nettamente inferiori rispetto a quanto potresti pagare autonomamente, grazie alle maggiori economie di scala.
- **PRESENZA GLOBALE →** Grazie al cloud, puoi espanderti in nuove regioni geografiche e distribuire globalmente in pochi minuti. Fare in modo che le applicazioni siano vicino agli utenti finali riduce la latenza e migliora la loro esperienza.

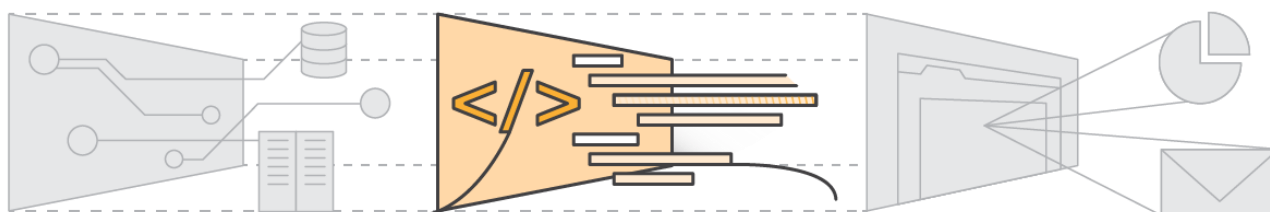
## TIPI DI CLOUD COMPUTING

Il cloud computing offre agli sviluppatori e ai reparti IT la possibilità di concentrarsi sulle attività strategiche, tralasciando attività altrettanto importanti ma onerose a livello di tempo quali, ad esempio, l'approvvigionamento di risorse, la manutenzione e la pianificazione della capacità. Dal momento che il cloud computing sta avendo una sempre maggiore popolarità, sono ora disponibili numerosi modelli e svariate strategie di implementazione in grado di soddisfare le specifiche esigenze di un'ampia gamma di utenti diversi. Ogni tipo di servizio cloud e metodo di implementazione offre diversi livelli di controllo, flessibilità e gestione. Saper distinguere le differenze peculiari tra Infrastructure as a Service, Platform as a Service e Software as a Service, nonché conoscere le varie strategie di implementazione disponibili ti permetterà di scegliere il set di servizi più idoneo alle tue specifiche esigenze di lavoro. Sono disponibili tre modelli di cloud computing principali. Ogni modello rappresenta un aspetto diverso dello stack del cloud computing e questi modelli sono:

- **Infrastructure as a Service (IaaS)**, contiene gli elementi fondamentali di base dell'infrastruttura IT del cloud e in genere consente l'accesso a caratteristiche di rete, computer (virtuali o su hardware dedicato) e spazio di storage dei dati. Infrastructure as a Service offre il massimo livello di flessibilità e controllo di gestione sulle risorse IT e questo servizio può essere paragonato alle risorse IT esistenti attualmente note a reparti IT e sviluppatori.

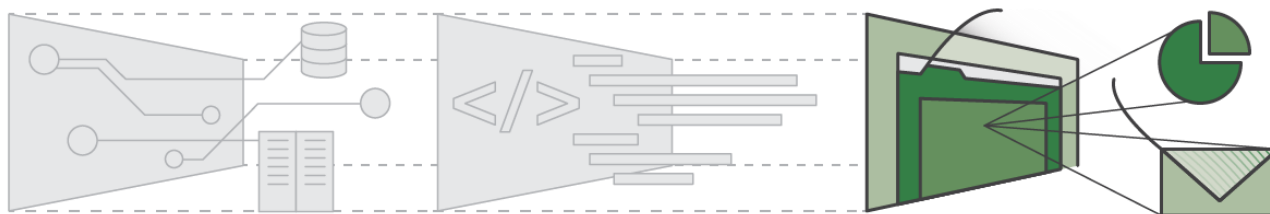


- **Platforms as a Service (Paas)** le organizzazioni non dovranno più gestire l'infrastruttura sottostante (in genere hardware e sistemi operativi). In questo modo sarà possibile concentrarsi sulla distribuzione e sulla gestione delle applicazioni. Questa soluzione è caratterizzata dal massimo livello di efficienza in quanto non è più necessario dedicarsi ad attività quali l'approvvigionamento delle risorse, la pianificazione della capacità, la manutenzione del software, l'applicazione di patch o qualsiasi altro tipo di attività onerosa a livello di tempo e impegno che possa interessare l'esecuzione delle applicazioni.



- Il servizio **Software as a Service (SaaS)** offre un prodotto completo che viene eseguito e gestito dal provider di servizi. Nella maggior parte dei casi, quando si parla di "Software as a Service" si fa riferimento alle applicazioni degli utenti finali. Grazie alla soluzione SaaS non sarà più necessario preoccuparsi delle modalità di gestione del servizio o dell'infrastruttura sottostante. L'utente potrà finalmente concentrarsi esclusivamente sull'uso di un software particolare. Un esempio comune di applicazione SaaS è rappresentato da un'applicazione e-mail basata sul Web che

consente l'invio e la ricezione dei messaggi e-mail senza la necessità di gestire l'aggiunta di caratteristiche al prodotto o di gestire i server e i sistemi operativi in cui il programma viene eseguito.





## SVANTAGGI DEL CLOUD COMPUTING

Diversi sono anche gli svantaggi presenti nel cloud computing, tra i quali si citano, i più importanti:

- **Privacy a rischio:** da più parti si manifesta qualche resistenza per i potenziali rischi che provengono, da una parte dalla completa dipendenza da Internet, dall'altra dalla sicurezza informatica e violazione della privacy a cui sono soggetti i dati memorizzati
- **Dipendenza completa a Internet:** la completa dipendenza dalla connessione con la rete Internet e quindi l'impossibilità della connessione può determinare un blackout di ogni attività;
- **Sicurezza informatica e Violazione Privacy:** premesso che i dati sono memorizzati in server virtuale quindi spesso soggetti ad interventi del Provider, sorge il rischio di manipolazioni diverse per ricerche di mercato, spionaggio industriale o altro. Il rischio sicurezza aumenta con le reti wireless, molto più esposte a casi di pirateria informatica e diventa ancor più complicato, se le Server Farms, dove sono memorizzati i dati personali e sensibili, appartengono ad aziende di residenza diversa da quella dell'utente, per cui in caso di violazione qualsiasi o fallimento, si potranno incontrare serie e fondate difficoltà a procedere legalmente per il recupero dei dati.

## 2 | STRUMENTI UTILIZZATI

### OPENSTACK

OpenStack (a volte indicato come O~S) è un progetto IaaS (Infrastructure-as-a-Service) cloud computing di Rackspace Cloud e NASA. A oggi oltre 500 società si sono unite al progetto tra cui Arista Networks, AT&T, AMD, Avaya, Cisco Systems, Dell, Ericsson, Groupe Bull, IBM, Inktank, Intel, NEC, NetApp, Nexenta, Rackspace Hosting, Red Hat, VMware, Oracle e Yahoo!.



OpenStack è un software libero open source rilasciato sotto licenza Apache che integra il codice dalla piattaforma della NASA Nebula e dalla piattaforma Rackspace. L'architettura di Openstack è suddivisa in moduli, in particolare:

- **OpenStack Compute (Nova)** è un controller per il cloud computing (la parte principale di un sistema IaaS). L'architettura di Compute è progettata per scalare orizzontalmente su hardware standard senza particolari requisiti software o hardware proprietari e di fornire la capacità di integrarsi con i sistemi legacy e tecnologie di terze parti. È stato progettato per gestire e automatizzare il pool di risorse del computer e può funzionare con tecnologie di virtualizzazione ampiamente utilizzate, come pure in configurazioni bare-metal e high-performance computing (HPC).

- **OpenStack Object Storage (Swift)** è un sistema di storage scalabile e ridondante. Gli oggetti e i file sono memorizzati su diversi dischi distribuiti su diversi server nel centro di calcolo. OpenStack si fa carico del compito di assicurare la replica e l'integrità dei dati all'interno del cluster. Il cluster di storage può scalare orizzontalmente semplicemente aggiungendo nuovi server. In caso di guasto ad un server o ad un disco, OpenStack replica i contenuti dai nodi ancora attivi in altri dischi e/o server, al fine di mantenere il livello di affidabilità e ridondanza configurati. Dato che OpenStack effettua le operazioni di replica e distribuzione dei dati sulle differenti unità via software, si può utilizzare hardware commerciale senza dover ricorrere a costose soluzioni proprietarie.
- **OpenStack Block Storage (Cinder)** fornisce uno storage persistente a livello di dispositivi a blocchi per il loro utilizzo da parte delle istanze di OpenStack Compute. Il sistema di storage a blocchi gestisce la creazione, il collegamento e lo scollegamento dei dispositivi a blocchi ai server. I volumi del block storage sono pienamente integrati con OpenStack Compute e con la Dashboard, permettendo agli utenti di gestire i loro dispositivi di storage. Block Storage è particolarmente indicato per ottenere buone performance in casi quali lo storage di database, file system espandibili o per fornire ai server un accesso a storage raw block level. La gestione degli snapshot fornisce funzionalità per il backup dei dati memorizzati su volumi su block storage.
- **OpenStack Networking (Neutron, in precedenza Quantum)** è un sistema per la gestione delle reti e degli indirizzi IP. OpenStack Networking assicura che la rete non sarà il collo

di bottiglia o fattore limitante in un cloud e offre agli utenti una reale gestione self-service anche delle loro configurazioni di rete. OpenStack Networking fornisce differenti modelli di rete per le diverse applicazioni o i gruppi utente. I modelli standard includono flat networks o VLAN per la separazione del traffico. OpenStack Networking gestisce gli indirizzi IP, consentendo l'assegnazione di indirizzi IP statici dedicati oppure tramite DHCP. I floating IP sono indirizzi che consentono di reindirizzare il traffico in modo dinamico a qualsiasi risorsa di calcolo gestita da OpenStack, questa caratteristica può essere utile in caso di manutenzione programmata o in caso di guasto. Gli utenti possono creare le proprie reti, controllare il traffico e collegare server e dispositivi per una o più reti. Gli amministratori possono sfruttare la tecnologia SDN come OpenFlow per consentire elevati livelli di multi-tenancy e massive scale. OpenStack Networking ha un extension framework che consente la messa in campo e la gestione di servizi di rete aggiuntivi, come IDS, bilanciamento del carico, firewall e VPN.

- **OpenStack Dashboard (Horizon)** fornisce un'interfaccia grafica, sia per gli amministratori che per gli utenti, per l'accesso e la gestione delle risorse fornita dal cloud. Horizon può integrarsi con prodotti e servizi di terze parti utili per la gestione. La dashboard si può modificare nell'aspetto, funzionalità particolarmente utile per i fornitori di servizio e operatori commerciali che forniscono istanze di OpenStack. Horizon è solo uno dei vari metodi con cui si può interagire con le risorse fornite da OpenStack.

## Ubuntu Server Edition

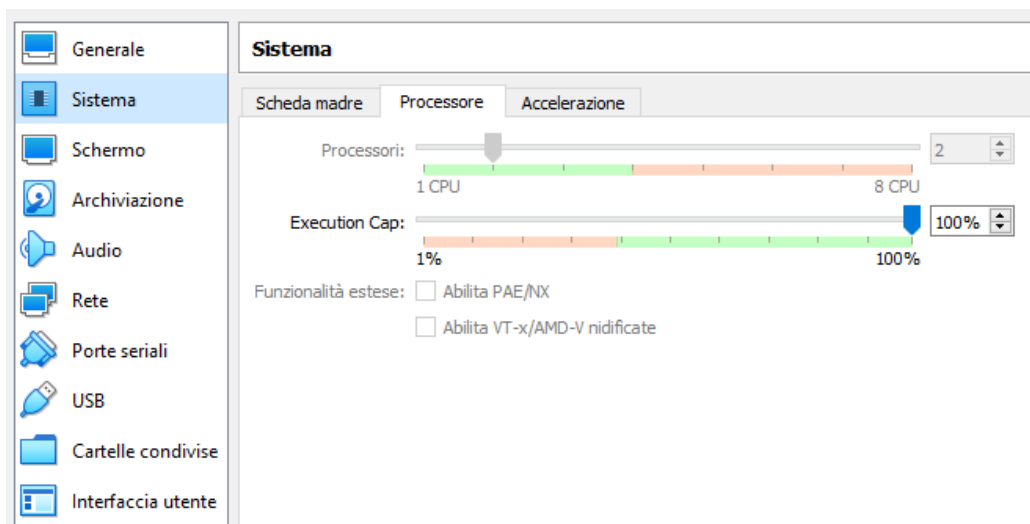
Ubuntu Server Edition è una distribuzione ufficiale di Ubuntu dedicata all'ambito server. Ubuntu Server Edition fornisce un'ottima piattaforma per sviluppare dei Server con numerosissime funzionalità: mail, DNS, web, database o server di file.



Una volta installato, non presenta nessun collegamento verso l'esterno e contiene solo ed esclusivamente il software necessario per rendere il server sicuro. In soli 15 minuti, il tempo richiesto per il completamento dell'installazione della Server Edition, è possibile avere un server LAMP funzionante. Questa caratteristica, disponibile solamente per la Server Edition, è selezionabile in fase di installazione. In questo modo si semplifica il lavoro di installazione e integrazione di ogni singolo componente del server LAMP, un processo che può richiedere ore e molta competenza nell'ambito della configurazione di ogni singola applicazione. I benefici sono una maggiore sicurezza, un tempo di installazione molto ridotto, un ridotto rischio di mal configurare il server (cosa molto diffusa quando vengono usati altri SO per server), semplicità negli aggiornamenti (cerca da solo gli aggiornamenti necessari e chiede all'utente se installarli). Il tutto risulta in un minor costo di gestione. Infine la piattaforma server di Ubuntu ha ricevuto delle certificazioni per IBM DB2 e MySQL.

### 3 | FASE IMPLEMENTATIVA DEL PROGETTO

Per realizzare il seguente progetto ho installato su una macchina la versione di Ubuntu Server. Alla macchina virtuale sono stati assegnati 5GB di RAM, 12 di memoria e 2 processori (come possiamo vedere nella schermata sottostante)



Eseguendo poi l'installazione classica di Ubuntu server per poi proseguire con quella di OpenStack. Per iniziare si esegue l'update della macchina tramite i tre comandi: *sudo apt update*, *sudo apt -t upgrade*, *sudo apt -y dist-upgrade*.

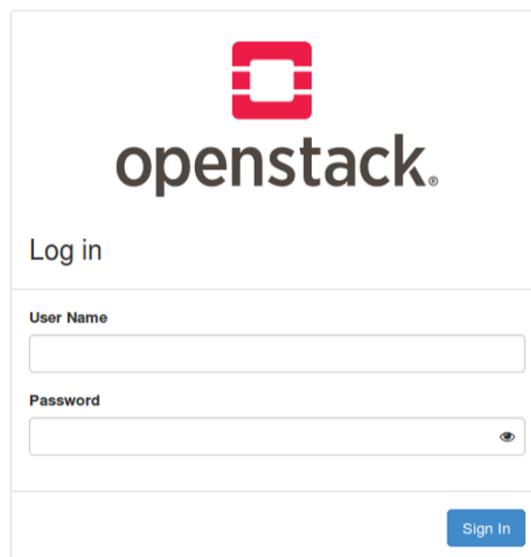
Ora per far funzionare tutti gli aggiornamenti si procede a riavviare la macchina, una volta che il reboot è completato si procede ad aggiungere un utente Stack User tramite il comando *sudo useradd -s /bin/bash -d /opt/stack -m stack* e provvedo ad abilitare tutti i privilegi di questo utente senza password tramite il comando *echo "stack ALL=(ALL) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/stack*. Si switcha sull'utente stack tramite il comando *sudo su -stack*. Una volta posti in essere sull'utente corretto si procede all'installazione vera e propria di open stack andando a clonare tramite il comando git (se non è installato nella

macchina basta usare il comando che verrà consigliato dal terminale) il link ufficiale di OpenStack ovvero: <https://git.openstack.org/openstack-dev/devstack>, si entra nella cartella devstack e si procede a modificare il file local.conf tramite il comando `cp samples/local.conf` con il seguente codice

```
$ HOST_IP="your ip"
$ echo "HOST_IP=${HOST_IP}" >> local.conf
```

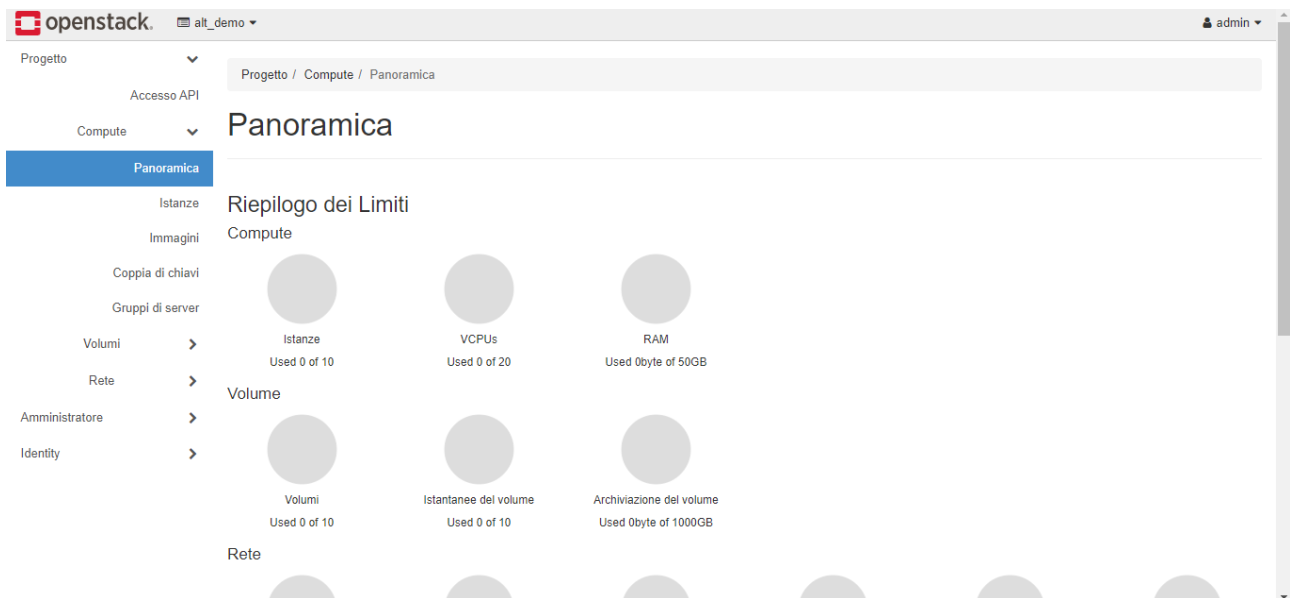
E si procede all'installazione vera e propria una volta scaricato tutto lanciando il file `./stack.sh`.

Se l'installazione non solleva alcun problema, con un browser si procede ad entrare nella pagina di login di OpenStack digitando il seguente indirizzo `localhost/dashboard/auth/login` oppure sostituendo localhost con il proprio ip. Si visualizza la schermata sottostante

The image shows the OpenStack login interface. At the top is the OpenStack logo, which consists of a red square with a white 'O' inside, followed by the word 'openstack' in a dark grey sans-serif font. Below the logo is the text 'Log in'. Underneath this are two input fields: 'User Name' and 'Password'. The 'User Name' field is a simple text box. The 'Password' field is a text box with a small eye icon on the right side to toggle visibility. At the bottom right of the form is a blue button with the text 'Sign In' in white.

Dove come user e password, nel mio caso si usa admin (specificata nel file local.conf precedentemente modificato). Effettuato l'accesso, la dashboard dà un riepilogo del proprio OpenStack attivo. Su una schermata simile possiamo ottenere

informazioni sui vari volumi creati, le network create, le diverse istanze e molto altro ancora



(nella seguente immagine è stato appena installato OpenStack perciò è tutto a zero).

OpenStack dà l'opportunità di virtualizzare diverse macchine virtuali. Nello specifico ho scaricato Ubuntu versione server, e una volta scaricato il sistema operativo, lo vado a caricare dentro OpenStack. Per fare questo, mi reco nella sezione images e creo una nuova immagine.



Image Details

Metadati

Image Details

Specify an image to upload to the Image Service.

Nome dell'immagine

Ubuntu

Image Description

Image Source

File\*

Browse...

ubuntu-18.04.3-live-server-amd64.iso

Formato\*

QCOW2 - QEMU Emulator

Image Requirements

Kernel

Choose an image

Ramdisk

Choose an image

Architettura

Minimum Disk (GB)\*

0

Minimum RAM (MB)\*

0

Image Sharing

Visibilità

Privato

Condiviso

Community

Pubblico

Protetto

Si

No

✕ Annulla

< Indietro

Prossimo >

Create Image

Una volta caricato il sistema operativo bisogna specificare il formato dell'immagine. In questa esercitazione io ho scelto il formato QCOW2-QEMU emulator. Per permettere al tutto di funzionare vado a creare una rete privata che oermetterà all'istanza che sto creando di comunicare con OpenStack. Per fare quello appena descritto basta recarsi nella sezione Network e procedere alla creazione della rete come possiamo vedere.

Creare una Rete

✕

Rete

Sottorete

Dettagli sottorete

Nome di Rete

rete1

Creare una nuova rete. Inoltre, è possibile creare una sottorete associata alla rete nei passaggi successivi di questa procedura guidata.

☒ Enable Admin State ?

☐ Condiviso

☒ Creare una Sottorete

Availability Zone Hints ?

nova

Annulla

<< Indietro

Prossimo >>

pag. 15

## Creare una Rete



Rete

Sottorete

Dettagli sottorete

### Nome sottorete

rete privata

### Origine indirizzo di rete

Alloca indirizzo di rete da un pool

### Pool di indirizzi\*

shared-default-subnetpool-v4 (10.0.0.0/22)

### Maschera di rete

26 (pool predefinito)

### Versione IP

IPv4

### IP gateway

☐ Disabilita gateway

Crea una sottorete associata alla rete. È necessario immettere un "Indirizzo di rete" e un "IP gateway" validi. Se non viene immesso l'"IP gateway", il primo valore di una rete verrà assegnato per impostazione predefinita. Se non si desidera il gateway, selezionare la casella di spunta "Disabilita gateway". La configurazione avanzata è disponibile selezionando la scheda "Dettagli sottorete".


È importante specificare il nome della rete, e diversi parametri della subnetmask, l'indirizzo di pool e la versione ip (abbiamo scelto la ipv4 perché vi ho maggiore praticità). I parametri non specificati sono stati lasciati in maniera predefinita. Ma per permettere di collegarsi online bisogna anche creare un extranet network public come possiamo vedere.

## Crea un router



### Nome router

router1

☒ Enable Admin State 

### Rete Esterna

public

☒ Enable SNAT

### Availability Zone Hints

nova

### Descrizione:

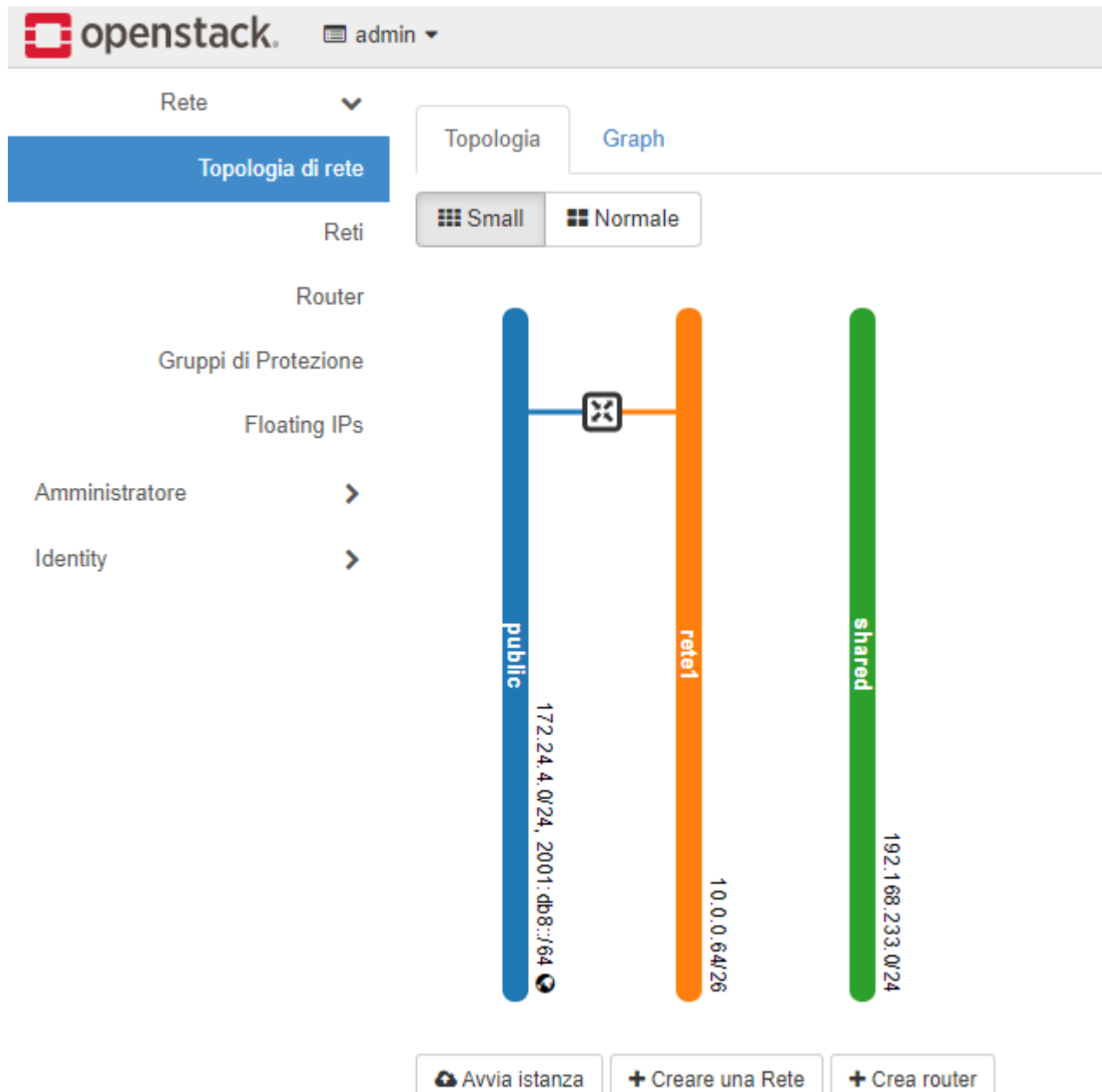
Crea un router con i parametri specificati.

Enable SNAT will only have an effect if an external network is set.

Annulla

Crea router

Una volta finite di configurare le reti possiamo vedere il risultato grazie alle proprietà grafiche di OpenStack che ci mostra una schermata simile.



Ora si procede a creare l'istanza vera e propria nella sezione instances e viene chiesto di settare diversi parametri tra cui:

- Nome dell'istanza (Ubuntu)

## Esegui istanza

Dettagli

Origine \*

Sapore \*

Reti \*

Porte di rete

Gruppi di sicurezza

Coppia di chiavi

Configurazione

Server Groups

Scheduler Hints

Metadati

Fornire il nome host iniziale per l'istanza, la zona di disponibilità in cui sarà distribuita e il numero di istanze. Aumentare il numero per creare più istanze con le stesse impostazioni.

**Instance Name \***

**Descrizione**

**Zona di disponibilità**

**Count \***

Istanze totali (10 Max)

10%

0 Utilizzo attuale

1 Aggiunto

9 Restante

✕ Annulla

< Indietro

Prossimo >

Esegui istanza

- Nella sezione source (in italiano origine) bisogna specificare quale immagine caricare (quella dove è presente Ubuntu Server, precedentemente creata)

Dettagli

**Origine**

Sapore \*

Reti \*

Porte di rete

Gruppi di sicurezza

Coppia di chiavi

Configurazione

Server Groups

Scheduler Hints

Metadati

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.

**Seleziona origine di avvio**

**Volume Size (GB) \***

**Allocato**

Nome	Aggiornato	Dimensione	Tipo	Visibilità
> Ubuntu	12/13/19 5:07 PM	848.00 MB	qcow2	Pubblico

**Disponibile 1**

Click here for filters or full text search.

Nome	Aggiornato	Dimensione	Tipo	Visibilità
> cirros-0.4.0-x86_64-disk	12/13/19 4:45 PM	12.13 MB	qcow2	Pubblico

**Crea nuovo volume**

Si No

**Elimina il volume con l'eliminazione dell'istanza**

Si No

- Ora si procede nella sezione flavor (stranamente tradotta in italiano con sapore) selezionando ds1G dando così alla nuova istanza 1GB di RAM e 10GB di memoria. La sezione attuale è molto importante perché come si può notare dall'immagine, si può gestire la nuova istanza specificando memoria e RAM da assegnarle.

Esegui istanza

Dettagli

Origine

**Sapore**

Reti \*

Porte di rete

Gruppi di sicurezza

Coppia di chiavi

Configurazione

Server Groups

Scheduler Hints

Metadati

I flavor gestiscono la dimensione della capacità di calcolo, memoria e archiviazione dell'istanza.

Allocato

	Nome	VCPU	RAM	Disco totale	Disco root	Disco temporaneo	Pubblico
>	ds1G	1	1 GB	10 GB	10 GB	0 GB	Si

▼ Disponibile 11

Seleziona uno

Q ds1

	Nome	VCPU	RAM	Disco totale	Disco root	Disco temporaneo	Pubblico
--	------	------	-----	--------------	------------	------------------	----------

✕ Annulla

< Indietro

Prossimo >

Esegui istanza

- Nella sezione network (tradotta con reti) procediamo a selezionare la rete su cui far girare l'istanza.

Esegui istanza

Le reti forniscono i canali di comunicazione per le istanze nel cloud.

▼ Allocato 1 Selezionare le reti dall'elenco in basso.

Rete	Sottoreti associate	Condiviso	Stato Admin	Stato
1 > rete1	rete privata	No	Attivo	Attivo

▼ Disponibile 2 Seleziona almeno una rete

Click here for filters or full text search.

Rete	Sottoreti associate	Condiviso	Stato Admin	Stato
> public	ipv6-public-subnet public-subnet	No	Attivo	Attivo
> shared	shared-subnet	Si	Attivo	Attivo

✕ Annulla < Indietro Prossimo > Esegui istanza

- Nella sezione keypair ( in italiano coppia di chiavi) vado ad importare la chiave di sicurezza appena generata sulla macchina virtuale. La chiave è la seguente:

```
+-----[SHA256]-----+
stack@stack:~/mykeys$ ls
cloud.key  cloud.key.pub
stack@stack:~/mykeys$ cat cloud.key.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACmxcA+rENogWVvYA1Ukq2No60H38tQhmISdsZjMF/mg3vvggs1NSDqcEyKxsp5
IIP2ZBYbthjZzR6c6Gxb/jwY0HH9087FDPQ0Gisx24Rq1n912PHSszpMDyEOpfC9IEg4DbPRNsIJQEoms7cUCm92XrVsWaHvxBb4
Fx+zNxbluuCeJyc1jeppBXZ3rEa7sGXemdfLotJ4m3YGkA7v187gEAKLeqX3j6guix8bNLgdhpZaXtk3RdBd0oBSy6asLUKjHLMG
6NLyqh1Aat1sh82yN1QRVNvnuWpLrDAm4Vhc4pLImIPyy3z50PibAI3WQYrJ221R2KGz9L2EU720VSh stack@stack
stack@stack:~/mykeys$
```

Presente come si può evincere dall'immagine sul file cloud.key.pub visto le sue dimensioni e complessità scriverla a mano è praticamente impossibile, copiarla non era possibile quindi è stato utilizzato uno strumento di OCR per permettere di estrapolare il testo dall'immagine. Una volta ottenuto il testo si procede ad inserirlo nella sezione predisposta come si può vedere.

Importa coppia di chiavi

Le coppie di chiavi indicano la modalità con cui si accede all'istanza dopo che questa è stata avviata. Scegliere un nome per la coppia di chiavi che verrà riconosciuto e incollare la chiave pubblica SSH nello spazio fornito.

Key Pair Name \*

Key Type \*

Load Chiave pubblica from a file

Scegli file Nessun file selezionato

Chiave pubblica \*

Content size: 0 bytes of 16.00 KB

Annulla

Importa coppia di chiavi

Una volta creata la chiava (mykey) si procede a selezionarla nell'istanza su cui voglio andar a lavorare.

Dettagli

Origine

Sapore

Reti

Porte di rete

Gruppi di sicurezza

Coppia di chiavi

Configurazione

Server Groups

Scheduler Hints

Metadati

Una coppia di chiavi consente di eseguire SSH nell'istanza appena creata. È possibile selezionare una coppia di chiavi esistente, importare una coppia di chiavi o generare una nuova coppia di chiavi.

+ Crea coppia di chiavi

Importa coppia di chiavi

Allocato

Mostrando 1 oggetti

Nome	Tipo
> mykey	ssh

Mostrando 1 oggetti

Disponibile 1

Seleziona uno

Click here for filters or full text search.

Mostrando 1 oggetti

Nome	Tipo
> key	ssh

Mostrando 1 oggetti

Annulla

< Indietro

Prossimo >

Esegui istanza

- Nella sezione Configuration (configurazione in italiano) vado a scrivere uno script che permette di settare la password del nuovo sistema operativo che si sta installando. Io ho scelto la password root come è possibile notare

Esegui istanza

Attiva/disattiva navigazione

È possibile personalizzare l'istanza dopo l'avvio utilizzando le opzioni disponibili qui. "Script di personalizzazione" è analogo a "Dati utente" in altri sistemi.

Load Customization Script from a file

Scegli file Nessun file selezionato

Customization Script (Modificato) Content size: 71 bytes of 16.00 KB

#cloud-config  
password: root  
chpasswd: {expire: False}  
ssh\_pwauth: True

Partizione disco

Automatico

☐ Configuration Drive

Annulla < Indietro Prossimo > Esegui istanza

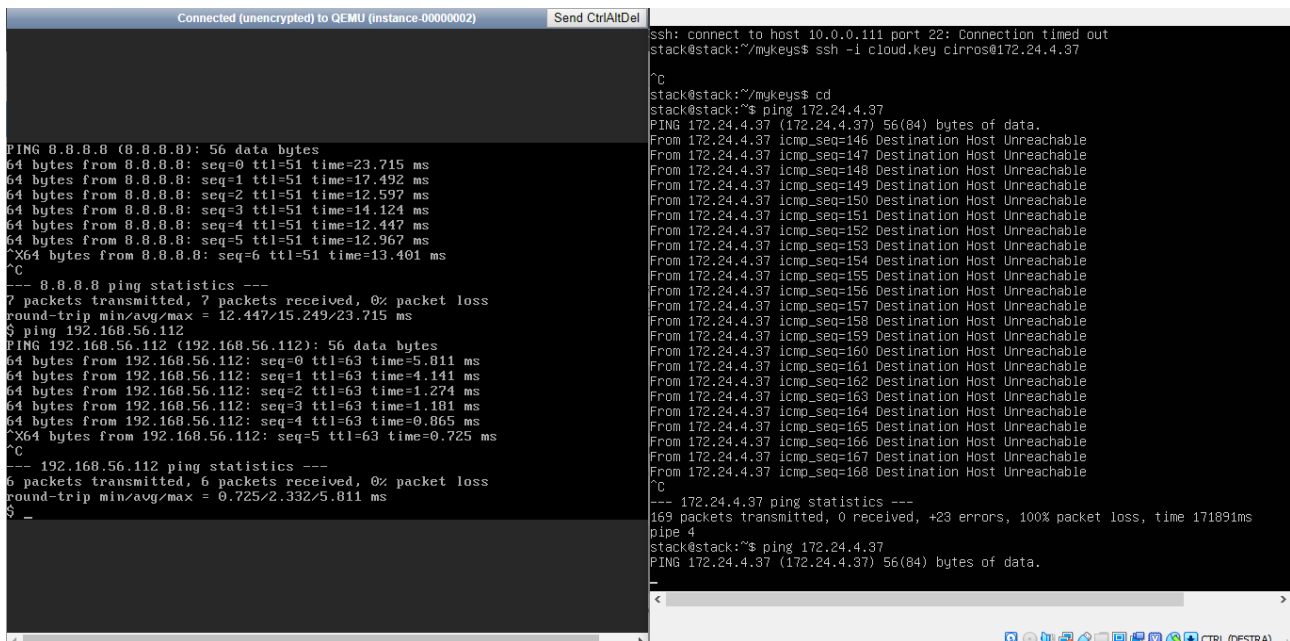
Una volta finito di creare l'istanza si aspetta che questa vada in running per controllare se la configurazione appena spiegata sia corretta. Per controllare se l'immagine si collega ad internet provo a pingare un DNS online nel caso specifico 8.8.8.8 e come si evince dallo screen in basso è andato tutto bene.

```
stack@stack:~/devstack$ ping
Usage: ping [-aAbBdDfhLnOqrRUvV64] [-c count] [-i interval] [-I interface]
        [-m mark] [-M pmtudisc_option] [-l preload] [-p pattern] [-Q tos]
        [-s packetsize] [-S sndbuf] [-t ttl] [-T timestamp_option]
        [-w deadline] [-W timeout] [hop1 ...] destination
Usage: ping -6 [-aAbBdDfhLnOqrRUvV] [-c count] [-i interval] [-I interface]
        [-l preload] [-m mark] [-M pmtudisc_option]
        [-N nodeinfo_option] [-p pattern] [-Q tclass] [-s packetsize]
        [-S sndbuf] [-t ttl] [-T timestamp_option] [-w deadline]
        [-W timeout] destination
stack@stack:~/devstack$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=53 time=20.6 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=53 time=13.0 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=53 time=11.6 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=53 time=12.1 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=53 time=14.4 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=53 time=11.8 ms
```

Una volta che l'istanza creata funziona correttamente, è possibile lavorarci sopra. Per fare ciò basta aprire un terminale sulla macchina virtuale e connettersi all'istanza tramite SSH. Per



effettuare la connessione basta digitare la stringa `ssh -i key.pem debian@ip` dove con `key.pem` si va a specificare il file delle chiavi generato precedentemente, con `ubuntu` si va a specificare il nome dell'istanza creata con Openstack e con `ip` si va a indicare l'indirizzo IP proprio della mia macchina.



```
Connected (unencrypted) to QEMU (instance-00000002) Send CtrlAltDel

PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: seq=0 ttl=51 time=23.715 ms
64 bytes from 8.8.8.8: seq=1 ttl=51 time=17.492 ms
64 bytes from 8.8.8.8: seq=2 ttl=51 time=12.597 ms
64 bytes from 8.8.8.8: seq=3 ttl=51 time=14.124 ms
64 bytes from 8.8.8.8: seq=4 ttl=51 time=12.447 ms
64 bytes from 8.8.8.8: seq=5 ttl=51 time=12.967 ms
^X64 bytes from 8.8.8.8: seq=6 ttl=51 time=13.401 ms
^C
--- 8.8.8.8 ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 12.447/15.249/23.715 ms
$ ping 192.168.56.112
PING 192.168.56.112 (192.168.56.112): 56 data bytes
64 bytes from 192.168.56.112: seq=0 ttl=63 time=5.811 ms
64 bytes from 192.168.56.112: seq=1 ttl=63 time=4.141 ms
64 bytes from 192.168.56.112: seq=2 ttl=63 time=1.274 ms
64 bytes from 192.168.56.112: seq=3 ttl=63 time=1.181 ms
64 bytes from 192.168.56.112: seq=4 ttl=63 time=0.865 ms
^X64 bytes from 192.168.56.112: seq=5 ttl=63 time=0.725 ms
^C
--- 192.168.56.112 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 0.725/2.332/5.811 ms
$ -

ssh: connect to host 10.0.0.111 port 22: Connection timed out
stack@stack:~/mykeys$ ssh -i cloud.key cirros@172.24.4.37
^C
stack@stack:~/mykeys$ cd
stack@stack:~$ ping 172.24.4.37
PING 172.24.4.37 (172.24.4.37) 56(84) bytes of data.
From 172.24.4.37 icmp_seq=146 Destination Host Unreachable
From 172.24.4.37 icmp_seq=147 Destination Host Unreachable
From 172.24.4.37 icmp_seq=148 Destination Host Unreachable
From 172.24.4.37 icmp_seq=149 Destination Host Unreachable
From 172.24.4.37 icmp_seq=150 Destination Host Unreachable
From 172.24.4.37 icmp_seq=151 Destination Host Unreachable
From 172.24.4.37 icmp_seq=152 Destination Host Unreachable
From 172.24.4.37 icmp_seq=153 Destination Host Unreachable
From 172.24.4.37 icmp_seq=154 Destination Host Unreachable
From 172.24.4.37 icmp_seq=155 Destination Host Unreachable
From 172.24.4.37 icmp_seq=156 Destination Host Unreachable
From 172.24.4.37 icmp_seq=157 Destination Host Unreachable
From 172.24.4.37 icmp_seq=158 Destination Host Unreachable
From 172.24.4.37 icmp_seq=159 Destination Host Unreachable
From 172.24.4.37 icmp_seq=160 Destination Host Unreachable
From 172.24.4.37 icmp_seq=161 Destination Host Unreachable
From 172.24.4.37 icmp_seq=162 Destination Host Unreachable
From 172.24.4.37 icmp_seq=163 Destination Host Unreachable
From 172.24.4.37 icmp_seq=164 Destination Host Unreachable
From 172.24.4.37 icmp_seq=165 Destination Host Unreachable
From 172.24.4.37 icmp_seq=166 Destination Host Unreachable
From 172.24.4.37 icmp_seq=167 Destination Host Unreachable
From 172.24.4.37 icmp_seq=168 Destination Host Unreachable
^C
--- 172.24.4.37 ping statistics ---
169 packets transmitted, 0 received, +23 errors, 100% packet loss, time 171891ms
pipe 4
stack@stack:~$ ping 172.24.4.37
PING 172.24.4.37 (172.24.4.37) 56(84) bytes of data.
```

Da qui è possibile fare tutto, come se avessimo un nuovo sistema operativo. Possono essere creati dei file, eseguire dei job ecc. Affinché la sottomissione dei job avvenga correttamente, essendo la versione di ubuntu sprovvista di gcc, è stata installata con il comando classico `apt-get install gcc`.

## 4 | FASE FINALE DEL PROGETTO

Una volta che abbiamo correttamente creato un'istanza di prova, sottomettiamo un generico job. Il job in particolare è l'algoritmo di Mergesort che ordina in maniera crescente un determinato array. Il seguente codice è in c ed è facilmente reperibile online sono solo state apportate leggere modifiche.

```
// ordino due sottoarray di arr[]
#include<stdio .h>
#include<stdlib .h>
#include<time .h>
#include<math.h>
void merge( int arr [] , int l , int m, int r ) {
    int i , j , k;
    int n1 = m - l + 1;
    int n2 = r - m;
    // creo array temporanei
    int L[n1], R[n2]
    for (i=0;i<n1;i++)
        L[ i ] = arr [ l + i ] ;
    for ( j = 0; j < n2 ; j++)
        R[ j ] = arr [m + 1+ j ] ;
    i = 0;
    //indice iniziale del primo sottoarray
    j=0;
    // indice iniziale del secondo sottoarray
    k=1;
    // indice iniziale del sottoarray del merge
    while (1<n1 && j<n2){
        if (L[ i ] <= R[ j ]) {
            arr [k] = L[ i ];
            i++;
        }
        else{
            arr[k] = R[j]
            j++;
        }
        k++;
    }
    while(i<n1){
        arr[k]=L[i];
        i++;
        k++;
    }
}
void mergeSort(int arr[],int l, int r){
    if(1<r){
        int m = 1+(r-1 )/2;
        mergeSort ( arr , l , m);
        mergeSort ( arr , m+1, r );
        merge( arr , l , m, r );
    }
}
```

```

}
int main(){
    clock_t start=clock();
    int arr[1000000];
    for ( int i =0;i <1000000; i++){
        arr [ i ]=rand()%1000000+1;
    }
    int arr size = sizeof(arr)/ sizeof(arr [ 0 ]);
    mergeSort ( arr , 0 , arr size - 1);
    clock_t end=clock();
    printf("tempo %f secondi\n " , (( double )(end-start ))/CLOCKS_PER_SEC);
    return 0;
}

```

Come richiesto, ho provato ad eseguire lo stesso job su più istanze in particolare ne creo una nuova (con lo stesso procedimento descritto) con 2GB di RAM.

Attualmente quindi OpenStack avrà due istanze in stato di running altrimenti se anche una sola non è in stato di running vanno ricontrollate le impostazioni.

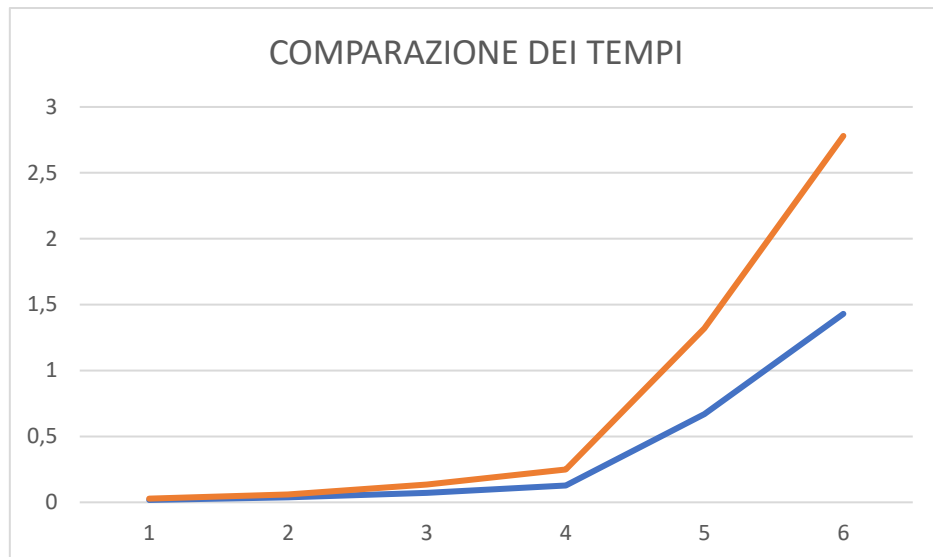
Per concludere vado ad eseguire entrambe le istanze andando a eseguire il programma c con array di diverse dimensioni su entrambe le macchine create.

## 5 | CONCLUSIONE

Per concludere vado a riportare i dati che ho ottenuto dalla seguente relazione. Specificando nella tabella sottostante, i tempi di esecuzione in secondi, la dimensione dell'array, e la RAM concessa.

<b>Dimensione array</b>	<b>T. flavor 1GB</b>	<b>T. flavor 2GB</b>
10000 elementi	0.018	0.01023
20000 elementi	0.037	0.023538
50000 elementi	0.071	0.0643
100000 elementi	0.1274	0.1219
500000 elementi	0.67	0.65
1000000 elementi	1.43	1.35

Come si può evincere dalla tabella, naturalmente più è grande l'array più è il tempo impiegato per effettuare il job in entrambi i casi. Però possiamo notare che assegnando maggiore RAM ad un flavor, più il tempo è basso. È visibile anche dal grafico sottostante dove viene messo a confronto il tempo delle due istanze comparandole



Con la riga di colore arancio vengono rappresentati i tempi della cpu a cui ho dato solo 1GB di RAM mentre con l'altra, quella blu, si vanno a rappresentare i tempi della 2GB.

Considerando comunque che ho utilizzato il MergeSort, un algoritmo che come sappiamo ha costo pari a  $n \log n$ , i tempi sono molto bassi rispetto ad altri algoritmi di ordinamento. È comunque possibile notare come i tempi rimangano più bassi nel secondo caso infatti la linea di colore blu ha un incremento minore rispetto a quella arancio.

In conclusione, maggiore sono i giga che vengono assegnati a una determinata istanza, migliore sarà il tempo di computazione per effettuare qualsiasi lavoro.

Con il presente progetto, per realizzare questa relazione, è stato possibile vedere l'enorme potenziale che ha il Cloud computing. Per capire, quanto sia elevato questo potenziale, attualmente il circuit di Formula 1 utilizza il cloud computing di Amazon (precisamente AWS) per tenere, durante le prove e le gare, traccia dei tempi, posizione e velocità attuale di ogni singola vettura. Questo fa sì che ogni circuito non debba acquistare un enorme centro di calcolo ma grazie alle ottime connessioni che

attualmente sono disponibili le scuderie ed i tifosi hanno a disposizione questi dati in tempo reale.