SBDIO I4.0
Big Data
for Industry

O1.1

Rapporto sulla organizzazione e il deployment dell'architettura distribuita del laboratorio SBDIO I4.0

T3LAB

POR FESR EMILIA-ROMAGNA 2014/2020
UNIONE EUROPEA
Fondo europeo di sviluppo regionale
RegioneEmilia-Romagna

Sommario

1 Obiettivi

Gli obiettivi di questa prima parte del progetto sono i seguenti:

1. definire e implementare l'architettura del primo nodo del cloud federato presso T3LAB;
2. definire le modalità per la federazione del nodo T3LAB con gli altri nodi del futuro cloud federato del progetto.
   La federazione sarà comunque limitata all'autenticazione e non coinvolgerà la gestione delle risorse, che rimarrà locale per ciascun nodo (eventualmente coordinata dallo user agent).
3. installare sul nodo cloud di T3LAB la piattaforma applicativa definita da UniFe, aprendo la strada all'installazione dell'applicazione che sarà sviluppata in accordo con il partner industriale Carpigiani.
   Questa installazione dovrà consentire anche di definire degli strumenti che consentano la sua replicazione sugli altri nodi del cloud federato.
4. indagare le modalità di migrazione di una applicazione da un nodo ad un altro del cloud federato.

Tutto il progetto è basato sull'utilizzo di OpenStack per la realizzazione dei singoli nodi del cloud federato di progetto (in particolare i nodi T3LAB e UniBo).

2 Politica di deployment

OpenStack di per se stesso è orientato alla fornitura di IaaS [1]. Dagli obiettivi indicati in sez. 1 è però evidente che scopo del progetto è anche quello di gestire, in un contesto di cloud federato, applicazioni finali.
Dato poi che ci si aspetta, come è in realtà anche il primo caso di studio considerato, che le applicazioni finali possano essere realizzate tramite l'utilizzo di middleware, la politica di installazione che verrà considerata sarà di tipo layerizzato:

1. su una infrastruttura fisica si installerà un nodo cloud OpenStack;
2. sul nodo OpenStack verrà definito un tenant al quale sarà allocato un pool di risorse virtuali che rappresenteranno la base di definizione della sua infrastruttura (ovviamente il nodo cloud potrà ospitare più tenant);
3. nel contesto del tenant sarà definita una infrastruttura virtuale di computazione compatibile con il pool di risorse allocato al tenant stesso;
4. sull'infrastruttura così definita verrà installata e configurata la piattaforma middleware (emulando una situazione PaaS) sulla quale verrà poi realizzata l'applicazione finale;
5. sulla piattaforma midlleware verrà installata l'applicazione finale.

POR FESR EMILIA-ROMAGNA 2014/2020
UNIONE EUROPEA
Fondo europeo di sviluppo regionale
RegioneEmilia-Romagna

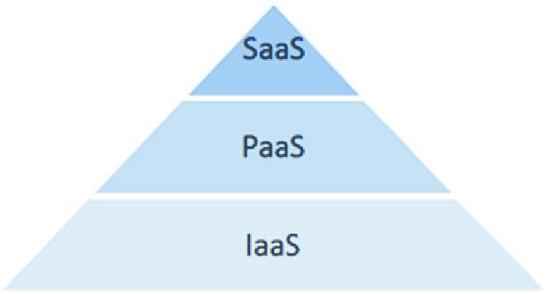

Figura 1 Stack dei possibili servizi cloud

I punti 1 e 2 sono di norma competenza del gestore del cloud, mentre i punti 3, 4 e 5 sono di competenza di un utente del cloud.
In questo stadio del progetto non verranno presi in considerazione pacchetti per l'automazione dell'installazione di OpenStack (e.g. PackStack) e di applicazioni in ambiente OpenStack (e.g. Juju).
Tutti i diversi step di installazione verranno realizzati in forma manuale. Tuttavia, per facilitarne la ripetizione (e.g. su altri nodi del cloud federato), essi verranno per quanto possibile realizzati tramite script.
In ogni caso, ciascuna procedura di installazione, anche quando riconducibile alla semplice attivazione di un script, verrà documentata come tale. Per approfondimenti su OpenStack si consiglia di leggere la documentazione ufficiale [2].

3 Architettura del nodo cloud T3LAB

Il nodo cloud T3LAB si basa su una infrastruttura fisica composta di 4 PC con sistema operativo Linux (Ubuntu 18.04 Server).
OpenStack è strutturato in modo modulare. E' composto cioè da diversi servizi (ogni servizio è contrassegnato da un nome in codice): questi devono essere mappati sui diversi nodi dell'infrastruttura fisica sottostante. I servizi di OpenStack considerati fino ad adesso sono i seguenti:

- OpenStack Compute (nome in codice Nova) [3];
- OpenStack Image Service (nome in codice Glance) [4];
- OpenStack Identity (nome in codice Keystone) [5];
- OpenStack Dashboard (nome in codice Horizon) [6];
- OpenStack Networking (nome in codice Neutron) [7];
- OpenStack Block Storage (nome in codice Cinder) [8].

Figura 2 illustra la struttura fisica (compresa la connettività) del nodo cloud T3LAB e il mappaggio sulle singole macchine dei servizi OpenStack.

POR FESR EMILIA-ROMAGNA 2014/2020
UNIONE EUROPEA
Fondo europeo di sviluppo regionale
RegioneEmilia-Romagna

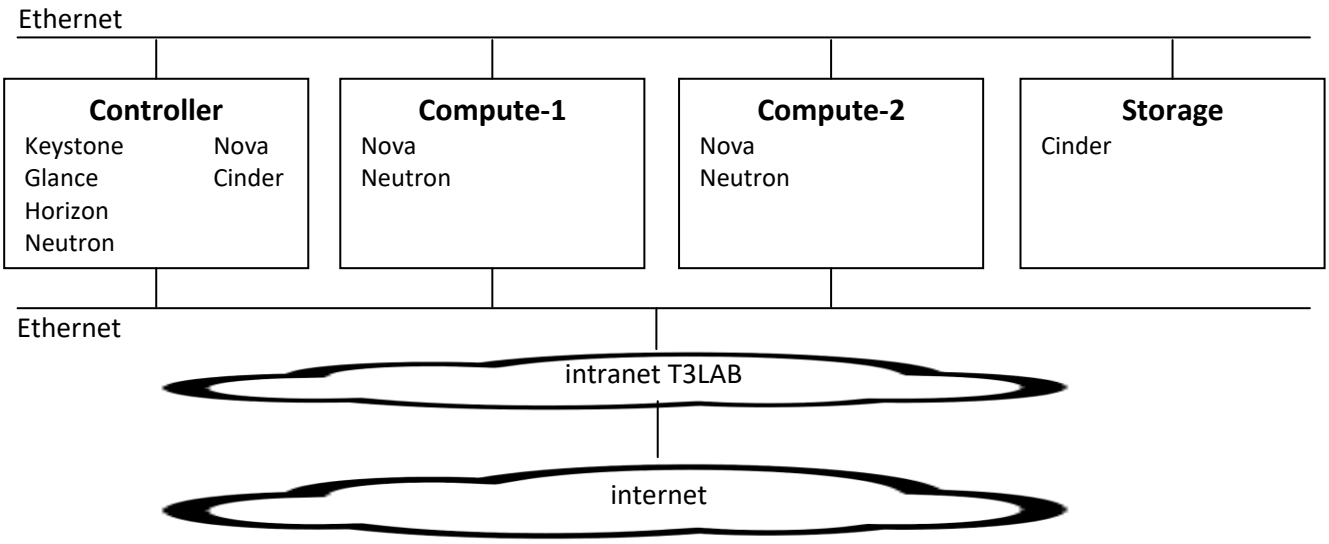

Figura 2 Struttura del nodo Cloud T3LAB

Sull'infrastruttura cloud un cliente può costruirsi una sua infrastruttura virtuale di calcolo composta da macchine virtuali (VM) interconnesse tra loro tramite una rete locale virtuale (in questo momento consideriamo solo infrastrutture virtuali basate su VM e non infrastrutture basate su container). Ogni VM viene dotata di un certo insieme di risorse, processori (core), memoria centrale volatile, memoria di massa permanente a blocchi.
Le VM vengono eseguite sulle macchine fisiche Compute che ospitano il servizio Nova (il servizio Nova sulla macchina fisica Controller è solo uno stub che consente l'orchestrazione dell'esecuzione delle VM).
La memoria di massa di una VM può essere allocata sia localmente alla VM stessa (una piccola parte, dipendente dal sistema operativo, deve comunque essere allocata localmente; questa memoria è strettamente associata alla VM e andrebbe persa con la distruzione della VM) sia sul nodo storage (un volume di memoria allocato sul nodo storage sopravvive, salvo diverse indicazioni date esplicitamente dall'utente, alla distruzione della VM cui e' associato).

4 La piattaforma Elastic Stack e la sua installazione sul nodo cloud T3LAB

La piattaforma Elastic Stack è costituita di 3 componenti principali:

- Elasticsearch: è un server di ricerca, può essere usato per cercare qualsiasi tipo di documento e fornisce un sistema di ricerca scalabile;
- Logstash è un motore per la raccolta, il filtraggio e l'arricchimento di dati che possono poi essere riversati nella base dati di Elasticsearch;
- Kibana è una piattaforma di analisi e visualizzazione dati.

Nel caso specifico, la piattaforma prevede anche un componente aggiuntivo, la sorgente dei dati raw, rappresentato dal database PostgreSQL.
Lo schema generale di funzionamento del sistema è descritto nella figura seguente:

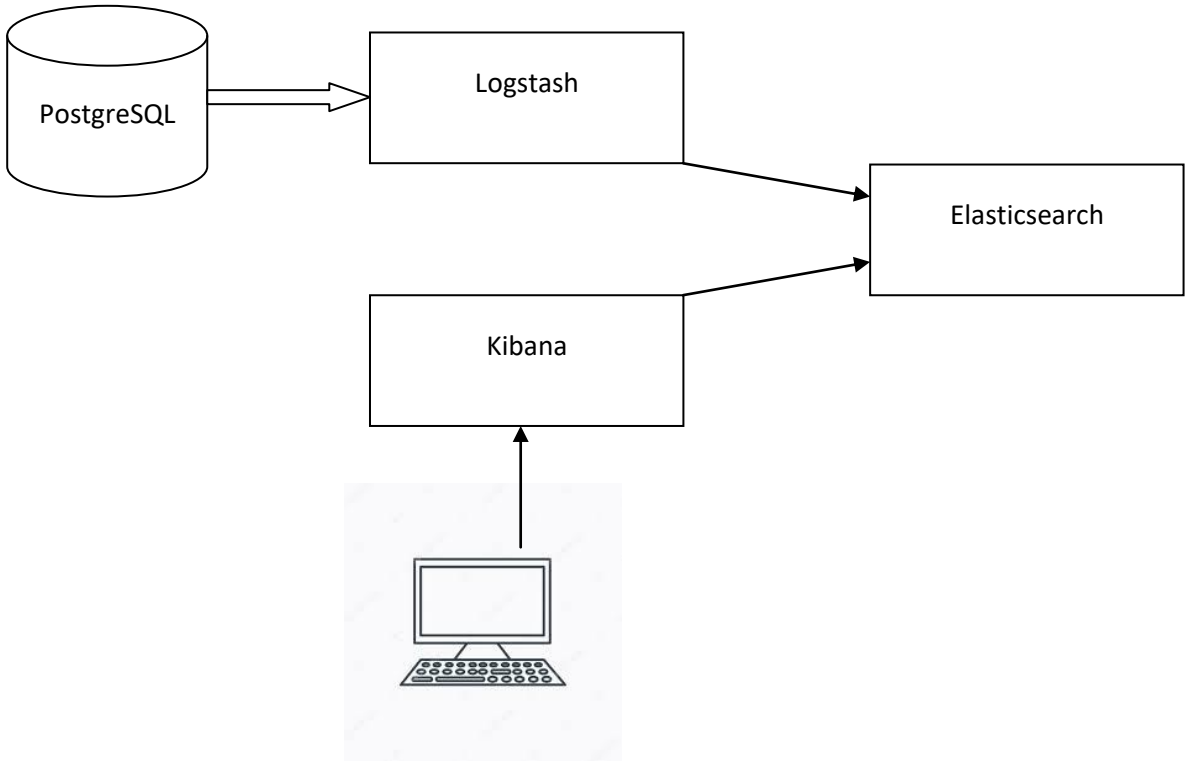

Figura 3 piattaforma Elastic Stack per SBDIOI40

Logstash è configurato per estrarre l'informazione contenuta in PostgreSQL e riversarla in Elasticsearch.
Kibana può quindi acquisire e analizzare i dati da Elasticsearch e presentarli ad un suo client (web, esterno alla piattaforma).
Le modalità in cui i dati sono inseriti in PostgreSQL è in questo momento non rilevante.
L'applicazione finale sviluppabile sulla piattaforma comprenderebbe ad esempio i filtraggi e gli arricchimenti che Logstash potrebbe operare sui dati acquisiti da PostgreSQL prima di riversarli su Elasticsearch. A livello di piattaforma non è previsto nessun filtraggio/arricchimento.
Parte della installazione della piattaforma sono le configurazioni operate su Logstash e Kibana così che questi interagiscano con Elasticsearch.

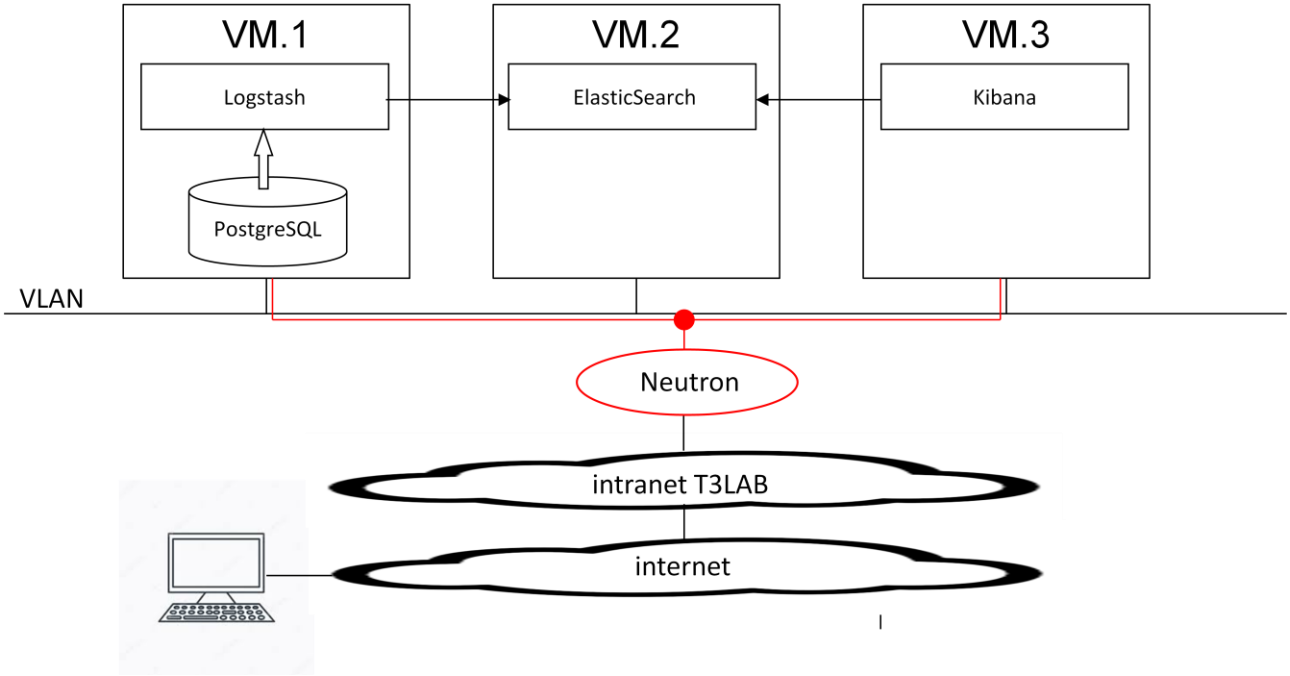

Figura 4 piattaforma Elastic Stack mappata su nodo cloud SBDIOI40 (PaaS)

La messa a disposizione sul nodo cloud T3LAB di una piattaforma integrata (i suoi componenti sono preconfigurati per cooperare tra loro) Elastic Stack è assimilabile ad un approccio PaaS.
L'infrastruttura virtuale per il supporto della piattaforma Elastic Stack sul cloud T3LAB prevede che Logstash (insieme a PostgreSQL), Elasticsearch e Kobana siano ciascuno installato su una diversa VM: l'infrastruttura prevede quindi 3 VM collegate tra loro tramite una VLAN.
L'indirizzo IP delle tre VM su questa VLAN è definito staticamente in modo da facilitare poi la configurazione della piattaforma Elastic Stack.
Di queste 3 VM, quella che ospita Logstash e quella che ospita Kibana devono essere configurate in modo da essere accessibili da internet:

- la VM di Logstash deve essere accessibile per consentire il caricamento dei dati su PostgreSQL;
- la VM di Kibana deve essere accessibile per consentire ad un web-client di accedere all'interfaccia uomo-macchina di Kibana e potere quindi analizzare e visualizzare le informazioni presenti in Elasticsearch.

In realtà sono state previste 2 varianti per l'infrastruttura virtuale su cui si basa la piattaforma:

- secondo la prima variante tutta la memoria di massa di ciascuna VM è allocata all'interno della VM stessa;
- secondo la seconda variante solo il minimo possibile della memoria di massa delle VM è allocato all'interno della VM stessa, mentre la maggior parte viene delegata ad un volume allocato sul nodo di Storage.

POR FESR EMILIA-ROMAGNA 2014/2020
UNIONE EUROPEA
Fondo europeo di sviluppo regionale
RegioneEmilia-Romagna

Le due alternative dovrebbero consentire di indagare il comportamento dinamico del sistema a fronte di carichi di lavoro che, per un bilanciamento ottimale, richiederebbero la migrazione delle diverse VM tra i due nodi di Compute.
In realtà ci sono anche altre tematiche relative all'allocazione delle VM sulle macchine fisiche di Compute che saranno indagate, e.g. la possibilità di specificare che due VM non devono essere allocate sulla stessa macchina fisica: questo sarebbe un aspetto essenziale nel momento in cui un'infrastruttura cloud dovesse essere utilizzata in architetture ridondate.

5 Migrazione di un'applicazione tra nodi del cloud federato

Gli script realizzati per configurare l'infrastruttura virtuale che ospita la piattaforma Elastic Stack e per installare questa piattaforma consentono di replicare facilmente questa installazione sugli altri nodi del cloud federato SBDIOI40.

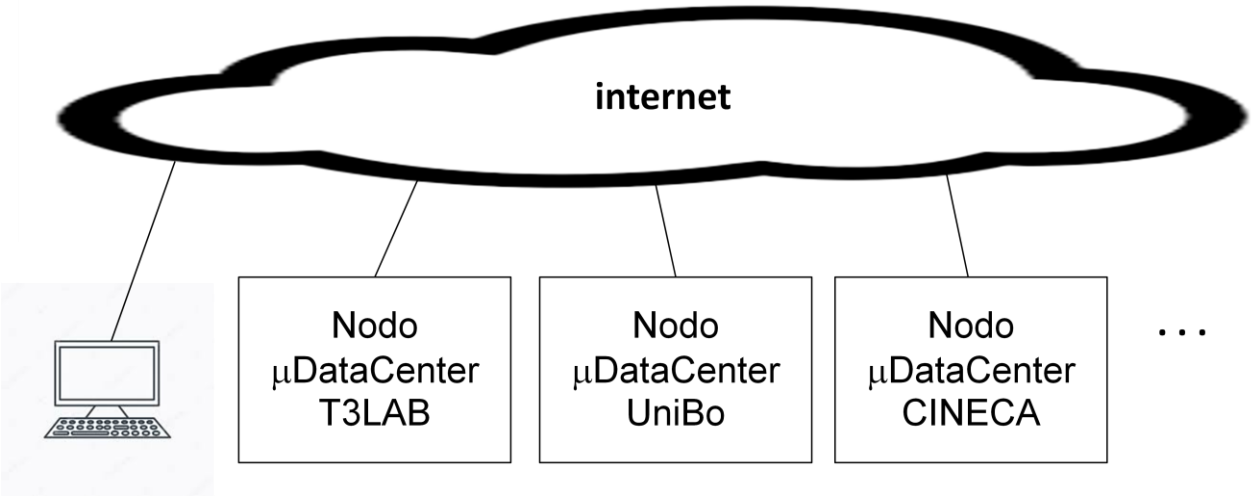

Figura 5 cloud federato SBDIOI40

E' sufficiente che nel nodo destinazione sia già stata creata una infrastruttura virtuale identica a quella del nodo sorgente, infrastruttura che può essa stessa essere create eseguendo lo stesso script che era stato eseguito per la sua creazione sul nodo sorgente.
Uno degli obiettivi del progetto è però quello di rendere possibile anche la migrazione di una applicazione già attiva da un nodo del cloud federato all'altro.
Per ottenere questo risultato ci si è appoggiati ai servizi di Glance, acceduti tramite shell.
Quando si crea una VM è necessario associarle l'immagine del sistema operativo che essa deve eseguire.
In realtà questa immagine può contenere un intero file system.
Una volta che una VM è in esecuzione è possibile richiedere a Glance di effettuare uno snapshot della stessa creando un'immagine del suo file system (per fare ciò è in realtà

POR FESR EMILIA-ROMAGNA 2014/2020
UNIONE EUROPEA
Fondo europeo di sviluppo regionale
RegioneEmilia-Romagna

necessario eseguire una sequenza di operazioni di Glance). Questa immagine potrà quindi essere utilizzata durante la creazione di una VM.
La strategia di migrazione si basa quindi sull'idea di creare (installare) sul nodo cloud target un insieme di VM identico a quello presente sul nodo cloud sorgente, ma dandogli come immagine gli snapshot delle VM che erano in esecuzione sul nodo sorgente: l'operazione di migrazione viene così ricondotta ad una operazione di installazione di una infrastruttura virtuale.

6 Federazione dei nodi cloud del progetto

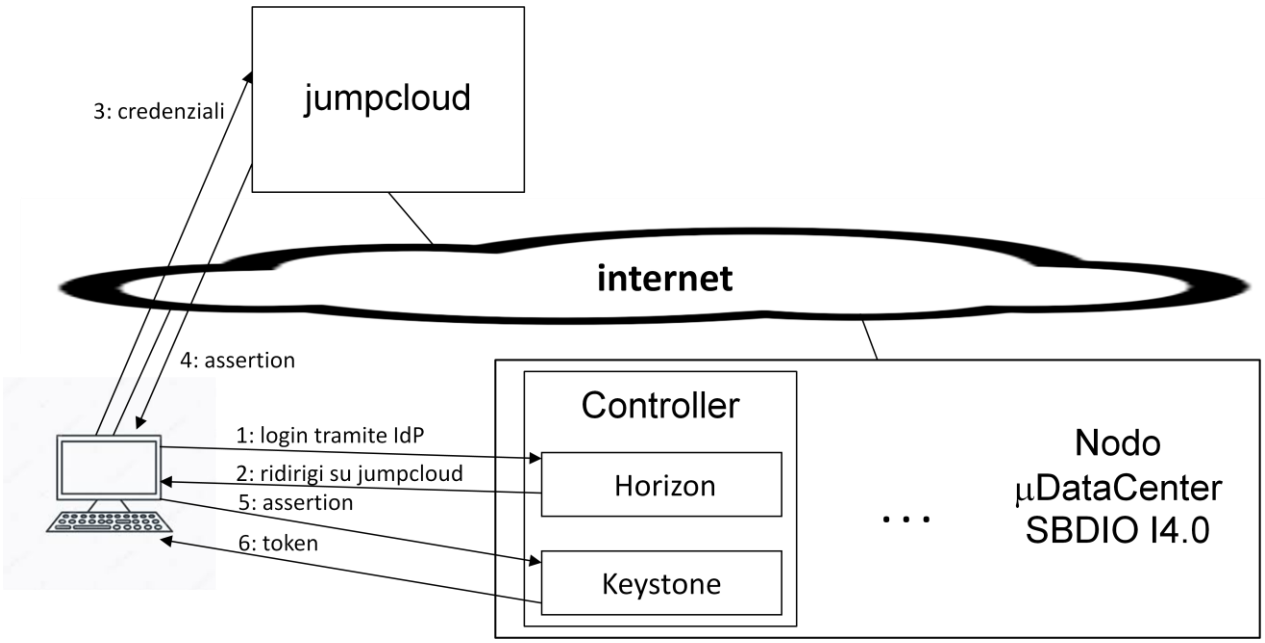

Figura 6 procedura (semplificata) di autenticazione basata su Identity Provider [10] ("An assertion is a formatted statement from the Identity Provider that asserts that a user is authenticated and provides some attributes about him/her")

Il progetto prevede che l'autenticazione degli utenti avvenga in maniera federata. Quando parliamo di autenticazione federata intendiamo che l'autenticazione degli utenti non si appoggi soltanto al servizio di Identity di OpenStack, Keystone, ma che venga delegata ad un Identity Provider esterno (da qui in avanti IdP) [10].
Questa soluzione presenta numerosi vantaggi. In primo luogo i dati sensibili degli utenti del cloud non vengono gestiti direttamente da noi, ma affidati ad un servizio esterno testato, affidabile e sicuro. Inoltre questa tecnica permette agli utenti di accedere a nodi diversi del cloud federato utilizzando le stesse credenziali: basterà che i singoli nodi (i relativi servizi Keystone e Horizon) siano configurati per utilizzare anche un servizio IdP esterno (e quale).

Un'altra possibilità che era stata presa in considerazione in fase di progetto era quella di utilizzare come IdP non un servizio esterno, ma il servizio Keystone di un nodo del nostro cloud federato. Sebbene possa sembrare una soluzione interessante, abbiamo deciso di scartarla per ragioni di sicurezza della gestione delle credenziali e dei dati sensibili.
L'IdP che abbiamo deciso di utilizzare in questa fase di test è jumpcloud.com [9], gratuito fino a 10 utenti (più che sufficienti per le nostre attuali esigenze).
Anche jumpcloud deve essere configurato in modo da accettare le richieste provenienti dal cloud.

7 Script realizzati

Nelle tabelle seguenti sono riportati gli script che T3LAB ha realizzato per il deploy e la gestione del cloud e delle applicazioni che ospita.
Questi script si dividono sostanzialmente in due categorie:

1. script per la gestione dell'infrastruttura del cloud e
2. script per la gestione della singola applicazione.

Il secondo gruppo degli script è strettamente legato all'applicazione con cui devono interagire, per cui quelli sviluppati al momento si occupano di gestire l'applicazione realizzata da UniFe.
Al momento gli script non sono ancora completamente parametrizzati.

7.1 Script per la gestione dell'infrastruttura cloud

Nome script	Descrizione
install_OpenStack.sh	Questo script si occupa di gestire l'installazione di OpenStack su delle macchine fisiche. I servizi che installa sono Keystone, Glance, Placement, Nova, Neutron e Horizon. Si occupa anche di copiare i file di configurazione necessari all'interno delle macchine. **NB:** Questa particolare installazione di OpenStack non supporta la creazione di reti virtuali per le VM, non crea tenant o subnet per le VM.
install_OpenStack_2.sh	Questo script si occupa di gestire l'installazione di OpenStack su delle macchine fisiche. I servizi che installa sono Keystone, Glance, Placement, Nova, Neutron e Horizon. Si occupa anche di copiare i file di configurazione necessari all'interno delle macchine. A differenza dello script precedente, l'installazione che effettua questo script consente di realizzare reti virtuali per le VM. Inoltre l'installazione di Keystone supporta l'autenticazione tramite IdP esterno (JumpCloud). **NB:** Questa particolare installazione di OpenStack non crea tenant o subnet per le VM.
setup_tenant.sh	Script che DEVE essere eseguito dopo lo script **install_OpenStack.sh**. Si occupa di creare alcuni ruoli base per gli utenti, un tenant di default, un semplice flavor, un security_group (le regole che indicano le porte raggiungibili delle VM e i protocolli accettati) di default e una subnet collegata alla rete fisica. Al termine dell'esecuzione lo script lancia una VM di prova sulla rete fisica creata.
setup_tenant_2.sh	Script che DEVE essere eseguito dopo lo script **install_OpenStack_2.sh**. Si occupa di creare alcuni ruoli base per gli utenti, un tenant di default, un

	semplice flavor, un security_group (le regole che indicano le porte raggiungibili delle VM e i protocolli accettati) di default, una subnet collegata alla rete fisica e una subnet virtuale a cui è possibile collegare le VM. Al termine dell'esecuzione lo script lancia due VM di prova: una sulla rete fisica e una sulla rete virtuale.

7.2 Script per la gestione dell'applicazione di UniFe

Nome script	**Descrizione**
build_unife.sh	Questo script ha come scopo quello di creare l'infrastruttura base per UniFe sulla quale poi una persona può operare direttamente per la configurazione dello script Elastic. Istanzia tre macchine virtuali Ubuntu su una subnet apposita usando un o specifico flavor.
delete_unife.sh	Lo script si occupa di eliminare tutte le componenti relative all'applicazione di UniFe presenti sul cloud: elimina il flavor, la rete virtuale, le macchine virtuali e le immagini delle macchine virtuali, riportando il tenant alla situazione antecedente l'esecuzione dello script **setup_unife.sh** (illustrato in seguito).
download_image.sh	Questo script sfrutta gli strumenti CLI di glance e OpenStack per il download in locale delle immagini delle tre macchine virtuali di UniFe, occupandosi anche di crearle a partire dalla tre VM in esecuzione sul cloud. Prerequisito di questo script è, ovviamente, che le tre macchine virtuali esistano sul cloud (create precedentemente a mano o con l'ausilio di altri script).
setup_unife.sh	É lo script che si occupa di creare da zero (una volta che si hanno a disposizione le immagini da caricare nelle VM) l'applicazione di unife. Crea tutti i componenti necessari al funzionamento dell'applicazione.

7.3 Esempio deploy OpenStack

Un esempio di utilizzo degli script precedenti per il deploy di OpenStack:

```
./install_OpenStack_2.sh root123 C85eN6sqdcH9jZnGBOqh 192.168.0.30
192.168.0.31 192.168.0.32 192.168.0.33

./setup_tenant_2.sh root123 192.168.0.30
```

dove i parametri sono, per il primo script rispettivamente

- la password delle macchine fisiche (root123),
- la password dei servizi di OpenStack (C85eN6sqdcH9jZnGBOqh),
- l'indirizzo IP del nodo controller (192.168.0.30),
- l'indirizzo IP del nodo di calcolo 1 (192.168.0.31) e del nodo di calcolo 2 (192.168.0.32) e l'IP del nodo di storage (192.168.0.33)

e per il secondo script

- la password delle macchine fisiche e
- l'indirizzo IP del nodo controller.

POR FESR EMILIA-ROMAGNA 2014/2020
UNIONE EUROPEA
Fondo europeo di sviluppo regionale
RegioneEmilia-Romagna

Al termine dell'esecuzione di questi script avremo OpenStack installato e funzionante sulle nostre macchine.

7.4 Esempio deploy App UniFe

Un esempio di utilizzo degli script per il deploy dell'app di Unife:

```
./setup_unife.sh 192.168.0.30 root123
```

dove i parametri sono

- l'indirizzo IP del nodo controller (192.168.0.30) e
- la password per accedere a ssh sullo stesso (root123).

Questo script prevede che siano presenti sulla macchina sulla quale stiamo lavorando le immagini delle macchine da caricare. Se così non dovesse essere, possiamo recuperarle da altre macchine virtuali in esecuzione (su un altro cloud, per esempio) tramite lo script:

```
./download_image.sh root123 192.168.0.50
```

dove 192.168.0.50 rappresenta l'indirizzo IP di un nodo controller di un'altro ipotetico cloud in cui l'applicazione sta girando, oppure fornire le immagini in altri modi allo script setup_unife.sh (anche manualmente).

8 Bibliografia

[1] «Cloud computing - Wikipedia,» [Online]. Available: https://it.wikipedia.org/wiki/Cloud_computing#Servizi.
[2] «OpenStack Docs: Stein,» [Online]. Available: https://docs.OpenStack.org/stein/.
[3] «OpenStack Docs: OpenStack Compute (Nova),» [Online]. Available: https://docs.OpenStack.org/nova/latest/.
[4] «OpenStack Docs: Welcome to Glance's documentation!,» [Online]. Available: https://docs.OpenStack.org/glance/latest/.
[5] «OpenStack Docs: Keystone, the OpenStack Identity Service,» [Online]. Available: https://docs.OpenStack.org/keystone/latest/.
[6] «OpenStack Docs: Horizon: The OpenStack Dashboard Project,» [Online]. Available: https://docs.OpenStack.org/horizon/latest/.
[7] «OpenStack Docs: Welcome to Neutron's documentation!,» [Online]. Available: https://docs.OpenStack.org/neutron/latest/.
[8] «OpenStack Docs: OpenStack Block Storage (Cinder) documentation,» [Online]. Available: https://docs.OpenStack.org/cinder/latest/.
[9] «Active Directory and LDAP Reimagined - JumpCloud,» [Online]. Available: https://jumpcloud.com.
[10] http://www.gazlene.net/demystifying-keystone-federation.html