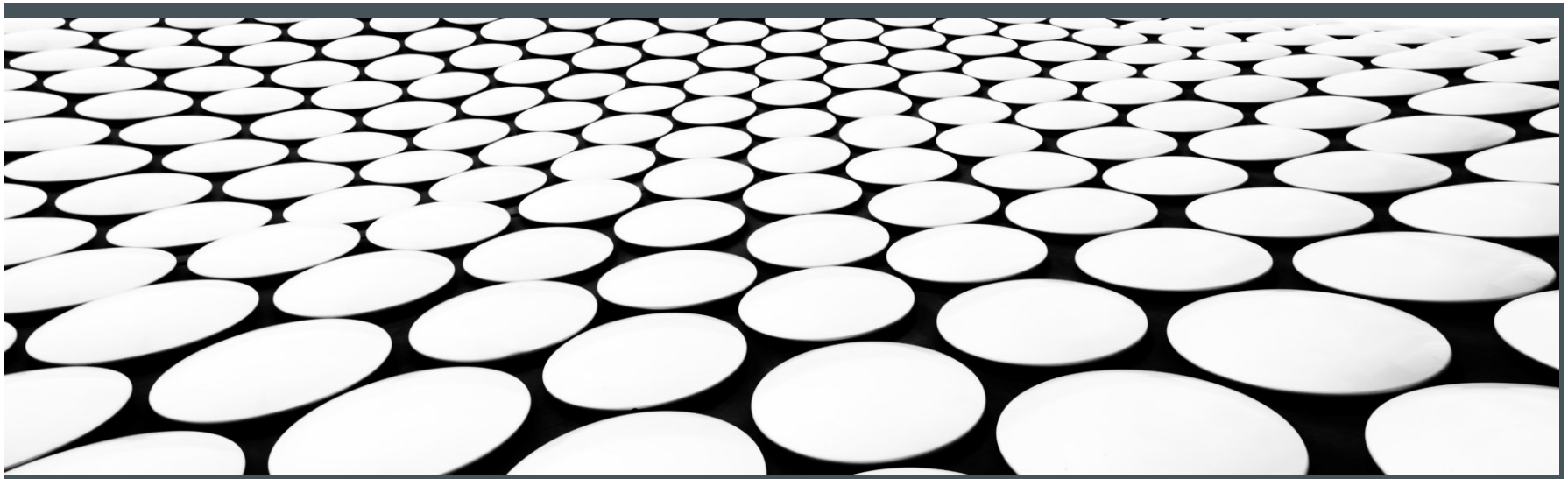
PLATFORM AS A SERVICE (PAAS)

VINCENZO CALABRÒ



Cloud Computing: PaaS

- Utente installa e crea applicazioni sviluppate tramite linguaggi di programmazione, librerie, servizi e tool supportati dal provider
- Utente mantiene il controllo sulla piattaforma installata
  - Ad es. CloudFoundry, Cloudify, WSO2, LAMP (Linux, Apache, MySQL), OwnCloud, piattaforma di test
- Utente non gestisce o controlla l'infrastruttura sottostante che include rete, sistemi operativi, server o storage
  - Non si preoccupa di installare il DB o Apache, o di avere abbastanza memoria
- Utente mantiene il controllo delle applicazioni installate e delle configurazioni dell'ambiente per hosting di applicazioni

Cloud Computing: PaaS

- Livello di astrazione più alto che rende la cloud programmabile

- Piattaforma installata sull'infrastruttura
  - Piattaforma offre un ambiente dove sviluppatori installano e creano applicazioni
  - Nessuna necessità di conoscere l'infrastruttura sottostante
  - Modelli di programmazione multipli e servizi specializzati (ad es., autenticazione, pagamenti) offerti come building block

Amazon ElasticBeanstalk

- ElasticBeanstalk (http://aws.amazon.com/elasticbeanstalk/) comprende una serie di framework forniti da Amazon per il deploy e il management dei servizi all'interno di Amazon Web Services (AWS)
- Fornisce comode interfacce agli sviluppatori dei servizi per il deployment e management dei servizi stessi
- Supporta i più comuni linguaggi di programmazione web
  - Java, C# e .NET, Node.js, PHP, Ruby, Python

Amazon ElasticBeanstalk

- AWS ElasticBeanstalk si basa su
  - **Elastic Compute Cloud (EC2):** permette di partizionare le risorse CPU fornite dal framework tra i servizi esposti
- Permette la configurazione puntuale delle risorse di computazione per dare priorità a specifici servizi o componenti
  - **Simple Storage Service (S3):** fornisce un'interfaccia web service per lo storage e l'accesso ai dati nella cloud
- Amazon garantisce che dati provenienti da una data area geografica (e.g., UE) non siano mai salvati fuori dalla stessa

Amazon ElasticBeanstalk

- Servizi di ridondanza, crittografia e isolamento dei dati
  - **Simple Notification Service (SNS):** fornisce servizi per l'invio e la gestione di messaggi attraverso i più comuni canali (HTTP, email, SMS, …)
  - **Cloud Watch:** fornisce servizi per il monitoring continuo della cloud e dei servizi esposti
  - **Elastic Load Balancing:** fornisce servizi per il load balancing del traffico verso la cloud
- Traffico può essere ridiretto anche verso altre istanze EC2 in rete
  - **Auto Scaling:** coopera con SNS e Cloud Watch permette di definire precise politiche per lo scaling delle risorse
- Permette di affrontare eventuali picchi nelle richieste, minimizzando i costi

Google AppEngine

- AppEngine (https://cloud.google.com/appengine/) permette di fare deploy di servizi direttamente sull'infrastruttura Google
- GoogleAppEngine offre un ambiente per lo sviluppo e hosting di applicazioni web
  - Building block: mail service, instant messaging service (XMPP) e molti altri
  - Supporta servizi scritti in Java, Python e GO (nuovo linguaggio di Google basato su Python)
- Costi di utilizzo basati soltanto sull'utilizzo della rete
  - Nessun costo iniziale, ogni utente ha disposizione 1 GB di storage e 5 milione di page-view al mese
  - Traffico ulteriore viene pagato 0.13€ per GB

Google AppEngine

- Funzionalità principali
  - Supporto alle più comuni tecnologie web
  - PersistentStorage con pieno supporto a query SQL e non-SQL, ordinamenti e transazioni
  - Sistemi automatici di scaling e load balancing
  - API per autenticazione utenti e l'invio di mail attraverso i servizi Google
  - Supporto per scheduled task
- Usati per gestire eventi che si ripropongono a scadenze o intervalli di tempo regolari

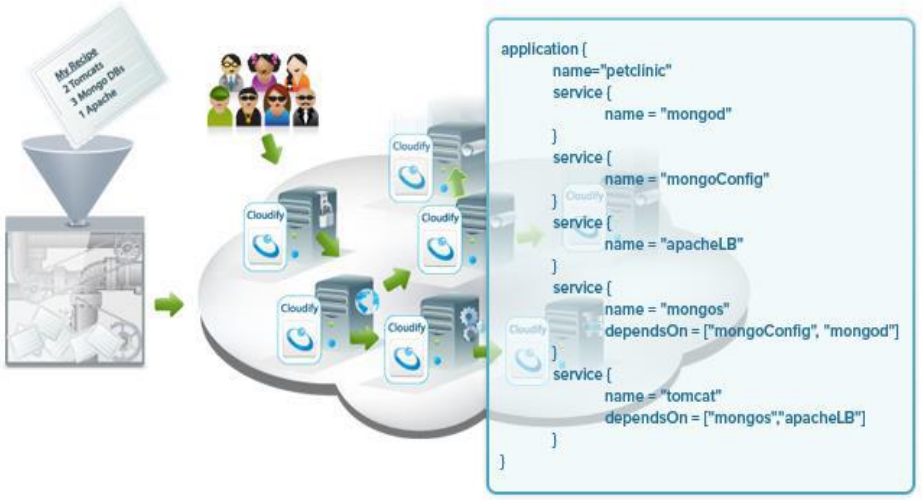
Cloudify

- Cloudify (http://getcloudify.org/) è un framework PaaS open source applicabile a qualsiasi tipo di cloud e stack di applicazioni
- Si basa sul concetto di *recipe*
  - Definiscono i passi per l'installazione, orchestrazione e il monitoraggio dello stack applicativo necessario per il deploy del servizio
  - Ogni ricetta definisce, per ogni applicazione, i servizi di cui necessita e le dipendenze che devono essere soddisfatte
  - Nell'esempio, l'applicazione richiede *mongodb* e *tomcat*
    - È a carico della PaaS attivare i servizi e le VM necessarie

```
application {
  name="petclinic-mongo"

  service {
    name = "mongod"
  }

  service {
    name = "mongoConfig"
  }

  service {
    name = "mongos"
    dependsOn = ["mongoConfig", "mongod"]
  }

  service {
    name = "tomcat"
    dependsOn = ["mongos"]
  }
}
```

Cloudify

- Funzionalità principali:
  - Indipendenza rispetto all'infrastruttura cloud e servizi forniti
  - Funzionalità automatiche per il ripristino e la gestione di nodi compromessi
  - Supporta lo scaling automatico delle risorse in caso di picchi di accessi
  - Gestione automatica del lifecycle dei servizi attraverso le recipe
  - Fornisce servizi di monitoring avanzati per i servizi e l'infrastruttura



Microsoft Azure

- Soluzione PaaS di Microsoft per il deployment di applicazioni, lo sviluppo di nuovi servizi cloud e la vendita di applicazioni (http://azure.microsoft.com/it-it/)

- Supporta la compilazione di applicazioni in qualsiasi linguaggio e sistema operativo

- Supporta ASP.NET, PHP o Node.js ed effettua la distribuzione in pochi istanti tramite FTP, GIT o TFS

Microsoft Azure

- Funzionalità
  - Scalabilità e funzionalità on-demand
  - Servizi mobili
  - Supporto multimediale, dalla codifica e protezione dei contenuti fino allo streaming e supporto dell'analisi
  - Supporta database SQL, archivi di tabelle no SQL, archivi BLOB (Binary Large Object) non strutturati
  - Recovery manager
  - Windows Azure Marketplace
  - Supporto per la connessione dell'infrastruttura locale alla cloud pubblica
  - Supporto per applicazioni che richiedono Big Compute

Cloud Foundry

Cloud Foundry: Un po' di storia

- Piattaforma PaaS *open-source* sviluppata nei linguaggi Ruby e GO
- Inizialmente di VMware, di recente divenuta proprietà della Pivotal Software, una *joint venture* di EMC, VMware e General Electric
- Offerta sia come servizio (tramite il portale *www.pivotal.io*) che come pacchetto software
- Disponibile in una versione totalmente gratuita e *open-source* e in una commerciale

Cloud Foundry: Caratteristiche

- Soluzione *open-PaaS* capace di supportare qualunque linguaggio o *framework*

- Rende possibile per i clienti
  - Il trasferimento delle applicazioni verso altri *cloud provider*
  - Il trasferimento tra *cloud* pubblici e privati senza dover apportare modifiche all'applicazione
  - Differente da PaaS proprietarie tipo Salesforce che usa un linguaggio proprietario per applicazioni (APEX)

Cloud Foundry: Caratteristiche

- Possibilità di ricreare un ambiente omologo ad un *cloud* pubblico in uno privato
- Disponibilità di un insieme esteso di *framework*, linguaggi e tecnologie
- Supporto comunità open source
  - Possibilità di personalizzare ed estendere le funzionalità di base del sistema
- Basato sul concetto di buildpack

Cloud Foundry: Buildpack

- Insieme di *framework*, librerie e software (ad es., interpreti e *web server*) dal quale dipende il funzionamento di una applicazione

- Permettono di estendere Cloud Foundry per il supporto a nuove tecnologie
  - Forniscono flessibilità e capacità di adattare Cloud Foundry alle esigenze e scelte progettuali e di sviluppo del cliente
  - Forniscono supporto runtime per l'applicazione

- Esaminano gli oggetti forniti dall'utente per identificare dipendenze e come configurare l'applicazione per comunicare con i servizi

Cloud Foundry: Buildpack

- Quando una applicazione viene caricata su *Cloud Foundry*, viene indicato il *buildpack* da utilizzare per il corretto funzionamento dell'applicazione
  - Cloud Foundry identifica automaticamente il buildpack da usare e lo installa nel Droplet Execution Agent (DEA) dove viene eseguita l'applicazione
- Il suo contenuto estratto ed installato nell'ambiente predisposto per l'esecuzione dell'applicazione
- Fornito dallo stesso *cloud provider*
  - Fornisce supporto per i linguaggi più comuni
  - Creato e personalizzato dallo stesso sviluppatore che può, così, introdurre un linguaggio non supportato, nuove librerie e interpreti o versioni specifiche degli stessi

Cloud Foundry: Buildpack

Name	Other Supported Languages and Frameworks	GitHub Repo
Java	Grails, Play, Spring, or any other JVM-based language or framework	Java source
Node.js	Node or JavaScript	Node.js source
Ruby	Ruby, Rack, Rails, or Sinatra	Ruby source
Binary	NA	Binary source
Go	NA	Go source
PHP	NA	PHP source
Python	NA	Python source
Staticfile	HTML, CSS, or JavaScript	Staticfile source

Cloud Foundry: Architettura

- Insieme di servizi distribuiti che cooperano per fornire un ambiente PaaS affidabile e scalabile

- Principi
  - Design modulare
  - Separation of concerns
  - Componenti specializzati in una singola funzione o in un insieme correlato di funzioni

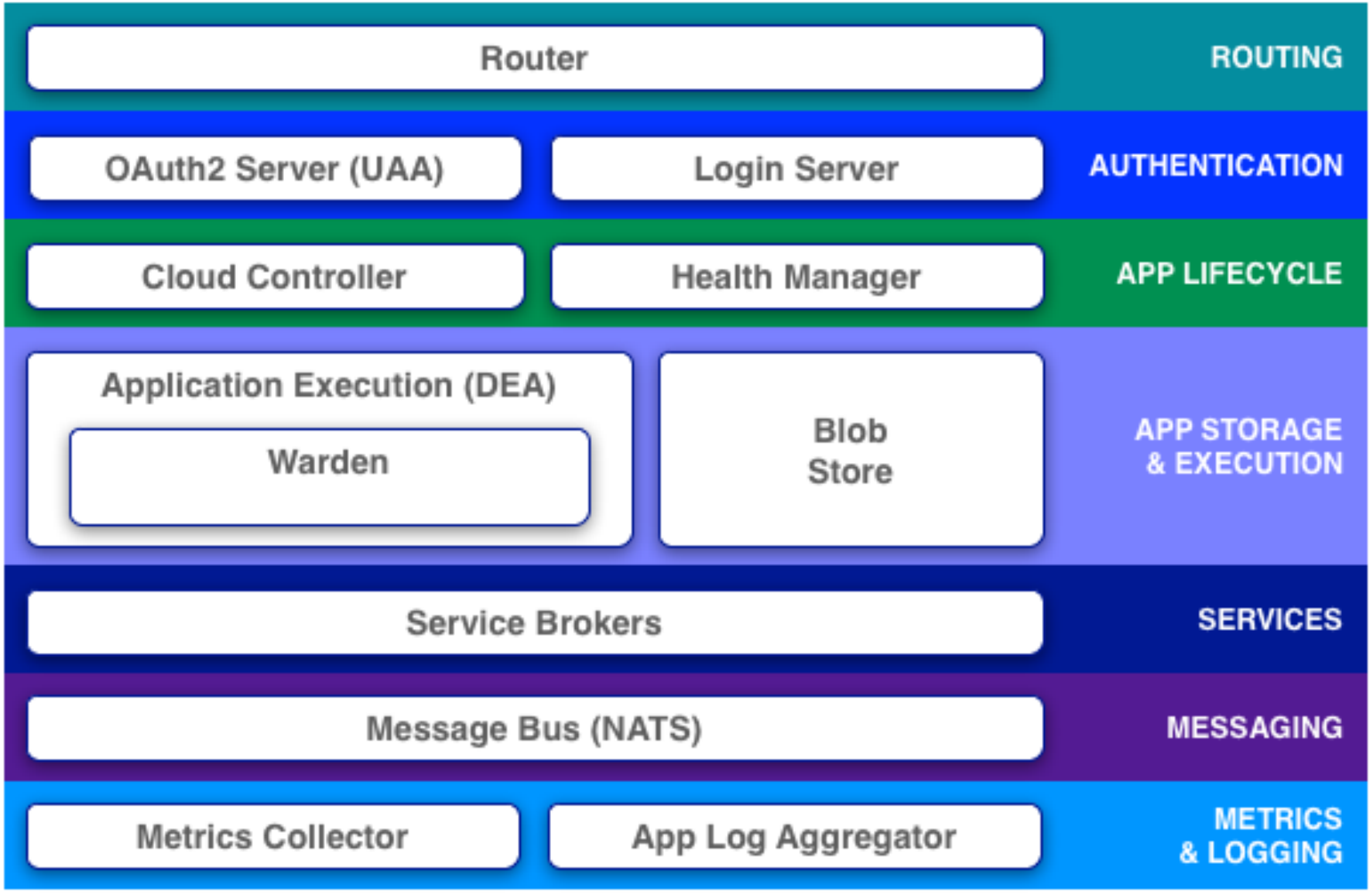
- Fornisce affidabilità, fault tolerance, capacità di determinare in maniera automatica guasti e variazioni di carico, e capacità di intervento automatiche

Cloud Foundry: Architettura

- Architettura monolitica non supporta i requisiti di una cloud

- Architettura modulare
  - Distribuzione dei componenti su più nodi
  - Facilita le operazioni di ripristino dei nodi mal funzionanti
  - Funziona (non ai massimi regimi) anche nel caso di alcuni nodi malfunzionanti
  - Operazioni di scaling mirato
    - Solo i componenti che ne hanno bisogno scalano

Cloud Foundry: Architettura



Cloud Foundry: Componenti Core

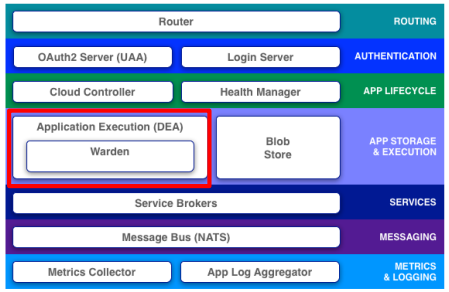
- Cloud Controller
- Droplet Execution Agent (DEA)
- Router
- Health Manager
- Service Brokers
- Message Bus (NATS)

Cloud Foundry: Cloud Controller

- Elemento centrale di Cloud Foundry
- Incaricato di gestire il ciclo di vita delle applicazioni
- Scritto in Ruby e fornisce API REST per l'accesso
- Interazioni con il *Cloud Controller* avvengono tramite *utility* o *plugin* per IDE che implementino le apposite API REST
  - Gestione dei permessi
  - Ciclo di vita delle applicazioni
    - Caricamento di una applicazione, aggiornamento, variazione del numero di istanze e delle risorse allocate, creazione di *orgs, space* e utenze

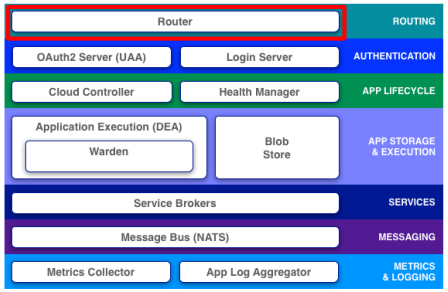
Cloud Foundry: Droplet Execution Agent

- Gestisce istanze di applicazioni, traccia le istanze in esecuzione, fa broadcast dei messaggi di stato
- Istanze di applicazioni vivono all'interno di container Warden
  - Containerization assicura che le applicazioni eseguono in isolamento, ottengono risorse, e sono prodotti da vicini «invadenti»
- Agente attivo sui nodi dedicati all'esecuzione delle applicazioni degli utenti
  - Colleziona informazioni esaminate dal *Cloud Controller* per la selezione del o dei nodi che accoglieranno nuove istanze di un'applicazione



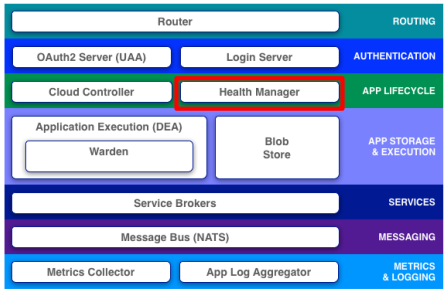
Cloud Foundry: Router

- Demone che ha come compito il monitoraggio delle richieste inoltrate al servizio PaaS e il loro corretto indirizzamento verso i destinatari, solitamente il *Cloud Controller* o un'applicazione in esecuzione su un nodo DEA
- Per individuare il nodo di destinazione, effettua un'interrogazione sulla tabella di *routing*
  - Indice rappresentato dall'*hash* del nome dell'applicazione presente nel campo *HOST* dell'intestazione HTTP
  - Il risultato dell'interrogazione è una lista di nodi capaci di rispondere alla richiesta da cui uno viene selezionato casualmente
- *Router* deve sopportare grandi carichi
  - Creazione di *pool* di nodi *Router* gestito da un *load balancer*
  - Gestione semplice, automatica e online dello *scaling* del *Router*
- Implementato con linguaggio GO (https://golang.org/)



Cloud Foundry: Health Manager (HM9000)

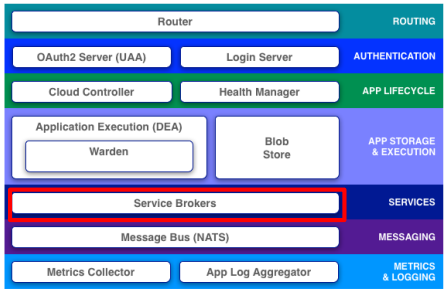
- Monitora applicazioni per determinarne lo stato (ad esempio, running, stopped, crashed), versione e numero di istanze
  - HM9000 aggiorna lo stato attuale di un'applicazione in base ai messaggi heartbeats (presi dal NATS) e droplet.exited rilasciati da DEA che esegue l'applicazione
- Determina lo stato, la versione e il numero di istanze attese dell'applicazione
  - HM9000 ottiene lo stato desiderato da un dump del Cloud Controller database.
- Mappa lo stato attuale con lo stato atteso
  - Ad esempio, se meno delle istanze attese stanno eseguendo, HM9000 richiederà al Cloud Controller di far partire nuove istanze
- Fa eseguire al Cloud Controller azioni per correggere discrepanze nello stato dell'applicazione



Cloud Foundry: Service Brokers

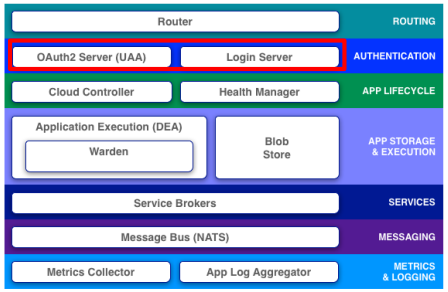
- Applicazioni di solito si basano su servizi esterni (ad esempio, database, third-party SaaS provider, storage, caching)

- *Service broker API* permettono a servizi esterni di essere integrati in *Cloud Foundry* creando un *middleware* chiamato *service broker*
  - Broker fornisce agli sviluppatori un modello di interazione con i servizi semplice, coerente e indipendente dal tipo e dal livello di complessità del servizio
  - Quando uno sviluppatore approvvigiona e associa un servizio a un'applicazione, il service broker del servizio fornisce accesso all'istanza
  - Broker è intermediario tra Cloud Controller e servizio vero e proprio



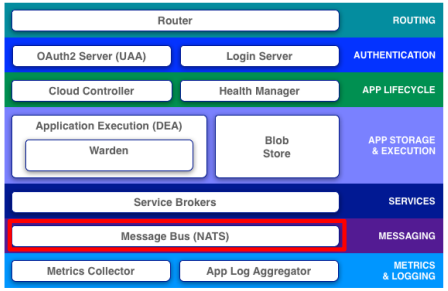
Cloud Foundry: User Account and Authentication (UAA) service

- Concetti
  - OAuth2 (RFC 6749)
    - Framework di autorizzazione che permette alle applicazioni di ottenere un accesso limitato agli account di servizi HTTP (ad es., Facebook, GitHub, DigitalOcean)
    - Delega autenticazione utente al servizio che gestisce l'account utente
    - Autorizza applicazioni di terze parti ad accedere all'account utente
    - Esiste per applicazioni desktop e mobile
  - System for Cross-domain Identity Management (SCIM)
    - Gestione dell'identità utente nella cloud
    - Modelli di autenticazione, autorizzazione privacy
    - Basato su utenti e gruppi



Cloud Foundry: NATS

- Message bus lightweight
- Sistema di messaging distribuito di tipo publish-subscribe
- Usato per comunicazioni interne tra i componenti

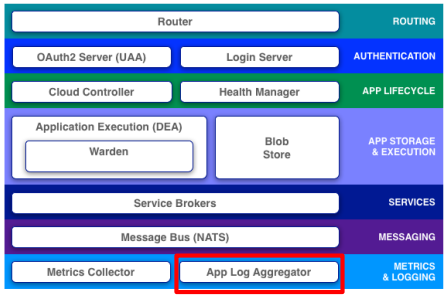


Cloud Foundry: Application Log Aggregator

- Registra in appositi *log* tutte le azioni e gli eventi che interessano le applicazioni o i componenti della piattaforma

- Strumento con memoria limitata

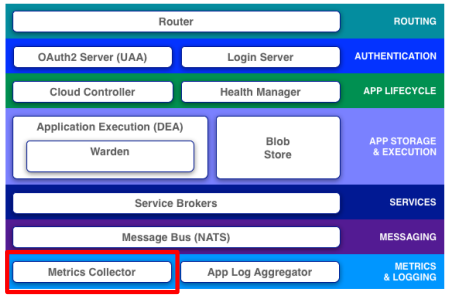
- Per forme di *log* più persistenti è necessario adottare degli strumenti di terze parti verso i quali inviarli secondo il protocollo *Syslog*



Cloud Foundry: Metric Collector

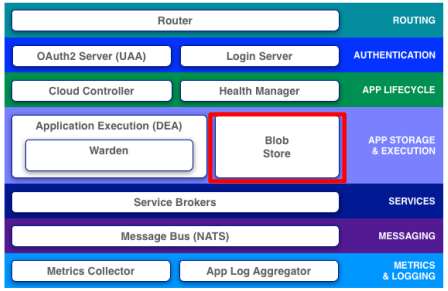
- Scopre automaticamente i componenti dell'infrastruttura monitorando il NATS

- Interroga i componenti sul loro stato attraverso *endpoint* esposti da ogni componente di *Cloud Foundry*
  - */healtz* restituisce un semplice messaggio che comunica il corretto funzionamento del componente
  - */varz* restituisce informazioni più dettagliate sullo stato di esecuzione come, per esempio, la quantità di memoria o di CPU utilizzata



Cloud Foundry: Blob Store

- Contiene
  - Application code
  - Buildpacks
  - Droplets



Conclusioni

- Abbiamo analizzato le caratteristiche di una soluzione PaaS
- Diversi approcci disponibili
- Cloudify fornisce una soluzione OpenSource molto utilizzata

QUESITI?

vincenzocalabro.it