Servisno orijentisane arhitekture

Predavanje 11: Mikroservisi i paterni, Obrasci za isporuku (deployment) aplikacija, Cloud computing, Kubernetes

Univerzitet u Novom Sadu
Fakultet Tehničkih Nauka

Obrasci za isporuku (deployment) aplikacija

- Više instanci servisa po jednom hostu
- Jedna instanca servisa po jednom hostu
- Jedna instanca servisa na jednoj VM
- Jedna instanca servisa po kontejneru
- Implementacija bez servera (serverless deployment)
- Platforme za isporuku (deployment) servisa

Platforme za isporuku (deployment) servisa

- Moguće rešenje – Koristiti platforme za instalaciju servisa. One obezbedjuju apstrakciju servisa – dobija se skup imenovanih, visokodostupnih (load balansiranih) instanci servisa
- Primeri:
  - okviri za orkestraciju Docker kontejnera (Docker Swarm, Kubernetes)
  - Serverless platforme – svaki cloud provider nudi neku opciju
  - PaaS cloud varijante raznih provider-a

Uvod

- Cloud computing je model isporuke IT servisa koji omogućava jednostavan, "na zahtev" pristup konfigurabilnim računarskim resursima. Resursima se pristupa putem Interneta.
- Resursi se obezbedjuju u vrlo kratkom vremenu, i čine dostupnim korisniku bez komplikovanog upravljanja
- Obezbedjuje visok nivo apstrakcije procesnih resursa kao i resursa za skladišenje podataka
- Zasniva se na konceptu virtualizacije
- Odlikuju ga odredjeni servisni modeli (modeli usluga), kao i deplyment modeli

Osnovne osobine

- Praktično samosulužni servis “na zahtev”:
  - Korisnik sam može odlučiti koje računarske resurse želi da pribavi, postupak je automatizovan i nije neophodna aktivnost administratora sistema da se novi resurs alocira i dodeli korisniku.
- Heterogeni pristup
  - Pristup resursima je preko mreže (pa i Interneta) i pristupa im se putem standardnih mehanizama, koji obezbedjuju i mogućnost korišćenja iz različitih vrsta klijenata
- Objedinjavanje resursa (resource pooling)
  - Resursi koji se nude se objedinjuju u veće “bazene resursa” koji mogu da opsluže više korisnika (multi-tenancy).

- Različiti fizički i/ili virtuelni resursi se dinamički dodeljuju i oslobadjaju prilagodjavajući se trenutnim potrebama korisnika (on-demand).
- Upotreba servisa se meri i kontroliše
  - Cloud sistemi automatski kontrolišu i optimizuju resurse koristeći odgovarajuće alate za merenje performansi (služe i za obračun naplate usluga)
  - Obezbedjuju predvidljivo pomnašanje računarske platforme usmereno na to da u svakom momentu zadovolji potrebe korisnika

Tipični servisni modeli

- IaaS (Infrastructure as a Service)
- PaaS (Platform as a Service)
- SaaS (Software as a Service)
- CaaS (Container as a Service)
- ...

IaaS

- Obezbedjuje krajnjem korisniku infrastrukturne komponente
  - Procesne module (computing), kapacitete za smeštaj podataka (storage), mrežne komponente i druge osnovne infrastrukturne elemente.
  - Ove resurse korisnik po svom nahodjenju koristi da na njima pokrene softver koji mu je neophodan.
  - Korisnik nema kontrolu niti upravljanje nad hardwareskim resursima ali ima nad OS, alokacijom storage-a, instaliranim aplikacijama i konfiguracijom mrežnih uredjaja (do odredjene granice).
- Primeri: Amazon EC2, GoGrid, iland, Rackspace Cloud Servers, ReliaCloud.

PaaS

- Korisniku se obezbedjuje mogućnost da na infrastrukturi koja je kreirana po korisnikovom zahtevu razvija, instalira i koristi aplikacije koje je kreirao koristeći alate i razvojna okruženja (platformu) koje mu je cloud provider obezbedio.
  - Korisnik ne kontroliše niti upravlja infrastrukturom.
  - Korisnik upravlja svojim aplikacijama i konfiguracijom tih aplikacija (okruženjem, platformom).
- Primeri: Windows Azure, Google App Engine.

SaaS

- Korisniku se obezbedjuje mogućnost da koristi aplikacije koje je ponudjač usluga razvio, na cloud infrastrukturi.
  - Aplikacijama na cloudu moguće je pristupiti preko različitih klijnetskih aplikacija.
  - Korisnik nema pristup niti kontrolu nad infrastrukturnim komponentama, serverima, OS, skladištima podataka, jedino čemu ima pristup je profilno podešavanje aplikacije koju koristi.
- Primeri: Google Apps, Gmail, Flickr, OneDrive ...

CaaS

- Vrlo slično IaaS samo se ne upravlja virtulenim mašinama već kontejnerima
- Postavlja se pitanje dal inam IaaS i dalje treba, kada imamo CaaS
- Primer: Google Cloud, AWS, Dockjer Enterprise, Alibaba, ...

Deployment modeli cloud computinga

- Private cloud
  - Cloud infrastruktura se pokreće i koristi isključivo za potrebe jedne organizacije. Može biti u samoj organizaciji (on-premise), ili u nekom data-centru ponudjača usluge, ali je u potpunosti posvećen jednom korisniku.
- Community cloud
  - Cloud infrastruktura se koristi od strane više organizacija i podržava rad u odredjenim interesnim grupama sa zajedničkim ciljevima.
  - Upravljanje može vršiti neka od organizacija članica ili treća strana

Deployment modeli cloud computinga

- Publiv cloud
  - Cloud infrastruktura se obezbedjuje kao javni servis koji mogu da koriste svi, a upravljanje i kontrolu radi ponudjač usluge.
- Hibridni cloud
  - Cloud infrastruktura predstavlja kombinaciju neke ili svih od prethodnih modela.

Osobine

- Dobre osobine
  - Smanjenje početnih troškova
  - Plaćanje po "pay-as-you-go-modelu"
  - High-end oprema u data centrima
  - Dostupnost resiursa za korišénje
- Loše osobin e
  - Prikriveni troškovi – visoki kasniji troškovi
  - Centralizacija resursa
  - Prebacivanje podataka na jednom mesto gde su resursi – suprotno spram *data locality* principa
  - Da bi u potpunosti iskoristili infrastrukturu apliacije se moraju pisati na specifičan način

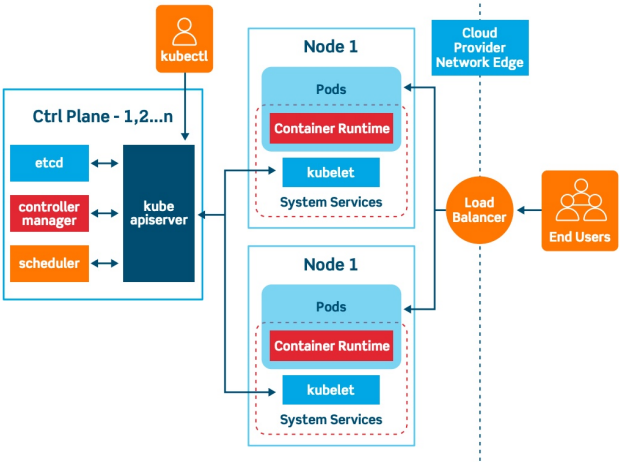
Budućnost

- Cloud na više nivoa
- Obrada podatka lokalno
- Distribuirani cloud – micro cloud
- Dinamičko kreiranje resursa spram potreba stanoništva
- Podela nadležnosti raznih nivao
- Lokalni cloud
- Ad-hoc cloud
- ...

Uvod

- Inicijalnoi razvijen od strane Google-a
- Open source++ implementacija in house orkestratora Borg
- Koristan alat za rad za kontejnerima
- Omogućio upotrebu mikroservisnih aplikacija i automatizaciju posla
- Stanje se opisuje deskritivno koristeći YAML – sitem se snalazi sa ostalim stvarima
- Automatski healtcheck aplikacija, restart i sve ostale stvari
- Vrlo bitan faktor za razvoj cloud native apliakcija!

Arhitektura



(Kubernetes Architecture and Concepts)

- Korisnik specificira sve elemente koristeći YAML
- Sistem se stara o svim ostalim stvarima
- Upravlja kontejnerima – jedinica deployment-a
- Kontejneri koji treba direktno da se vide su deo istog POD-a – app and sidecar
- Vrti se reconcile looop – konstantno uporedjuje trenutno stanje i traženo stanje
- Obavezan alat za cloud native apliakcije
- Lako proširiv
- Postoji ceo ekosistem oko njega

- Aplikcije se pronalaze i vezuju sa servisima koristeći jednosavna sistem labela
- loosely coupled mehanizam
- Servisi u kube terminologij su nesto preko čega se pristuap nekom kontejenru
- Komunikacija nikada ne ide direkt ka kontejneru
- Pruža razne benefiti
- Nije uvek potrebno koristiti kube!
- Nekada je overkill
- Za odredjene primene nikako nije pogodan, ili nije tako jednosavno pogodan

Dodatni materijali

- Building Microservices, Sam Newman
- Microservices • Martin Fowler • GOTO 2014
- What are microservices?
- Microservices patterns

Kraj predavanja

Pitanja? :)