Servisno orijentisane arhitekture

Predavanje 2: 12 Factor app

Univerzitet u Novom Sadu
Fakultet Tehničkih Nauka

Uvod

- Pravljenje *web-scale aplikacija* nije jednostavan posao
- Rizici su prisutni u svakom apektu razvoja i izvršavanja apliakcije
- Svaku prednost koju možemo iskoristiti da kreiramo bolje *web-sale* aplikacije, ne treba da izbegavamo!
- Za ove stvari obično ne postoji jedno i samo jedno rešenje
- Gledamo kako to rade najbolji, učimo od njih

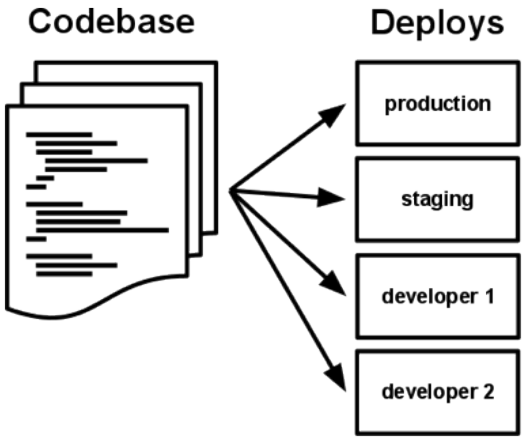
- Srećom po nas, imamo i par prednosti, ako se one tako mogu nazvati
- Problemi koji mogu nastati prilikom razvoja ovakih apliakacija su ugalvnom poznati
- Možemo se osloniti na postojeća rešenja
- Ne izmišljamo toplu vodu, trudimo se da budemo mudri
- Jedno rešenje ili *vodič* je svakako i princip *The 12 Factor App*

Uvod

- *The 12 Factor App* patern je 2011. objavio Adam Viggins
- Aplikacija 12 Faktora je skup principa koji opisuje način pravljenja *web-scale* softvera
- Kada se sledi, ovaj patern omogućava kompanijama nekoliko prednosti:
  - pre svega moguićnost da kreiraju kod koji se može pouzdano isporučiti (release)
  - brzo skalirati
  - održavati na dosledan i predvidljiv način
  - (relativno) lako ispratiti neke greške koje bi inače bilo teško ustanoviti i otkloniti

1. Codebase

- Aplikacija se uvek nalazi, i razvoj se prati nekim od sismtea za kontrolu verzija (SVN, Git, Mercurial, ...)
- Uvek postoji preslikavanje 1-1 *codebase* i aplikacije
- Ako imamo više *codebase-a* onda to nije aplikacija, već distribuiran sistem
- Jedna ista aplikacija može da se pokrene u više različitih okruženja (prod, test, staging, ...)



(https://12factor.net/codebase)

2. Zavisnosti

- U repozitorijumu treba da budu samo kod koji je jedinstven i relevantan za svrhu aplikacije
- Svi ostali ekterni artefakti treba da se referenciraju u manifestu zavisnosti učitanom u memoriju tokom razvoja, testiranja i produkcije
- Ono što želmo da izbegnemo jeste da se artefakti (JAR, DLL, ...) budu prisutni u repozitorojumu sa kodom
- Novi programer može da preuzme kod aplikacije na svoju mašinu i da je pokrene, za šta mu treba samo jezik u kome je aplikacija napisana, i menadžer za upravljanje zavinostima

3. Konfiguracije

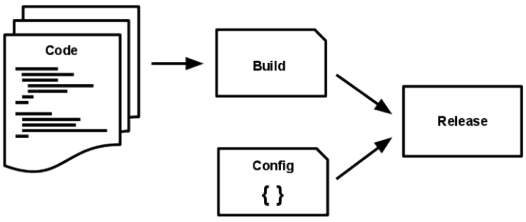
- Konfiguracija aplikacije će verovatno varirati od okruženja do okruženja i primene
- Konfiguracioni elemnte ne treba da budu deo koda u nekakvim promenljivama ili konstantama
- Informacije o konfiguraciji dodaju se ili kao *environtment variable* ili kao posebna konfiguraciona datoteka
- Naravno, možemo u kodu imati podrazumevana podešenja koja će preinačiti neki drugi vid konfiguracije
- Na ovaj načina možemo da centralizujemo i eksternizujemo konfiguracije, nezavisno od okruženja u kome se aplikacija izvršava

4. Povezani servisi

- Tretirati povezane servise kao spoljne resurse
- Povezane servise aplikacija koristi preko mreže kao deo svog normalnog rada
- Korišćenje resursa kao pratećih usluga omogućava fleksibilnost i efikasnost u životnom ciklusu razvoja softvera
- Aplikacija treba da bude u mogućnosti da zameni lokalni servis (npr. loklanu instancu baze podataka), sa nekim sistem kojim upravlja treća strana, bez ikakvih promena u kodu

5. Build, Release, Run

- Kod prolazi kroz tri različite faze: (1) build, (2) release (3) run
- Ove tri faze su eksplicitno rastavljene i nikada se ne spajaju
- Svakai *release* treba da ima svoj jedinstveni ID
- *Build* pokreću programeri aplikacije svaki put kada se primeni novi kod
- *Runtime* sa druge strane se dešva automatizaovano



(https://12factor.net/build-release-run)

6. Procesi

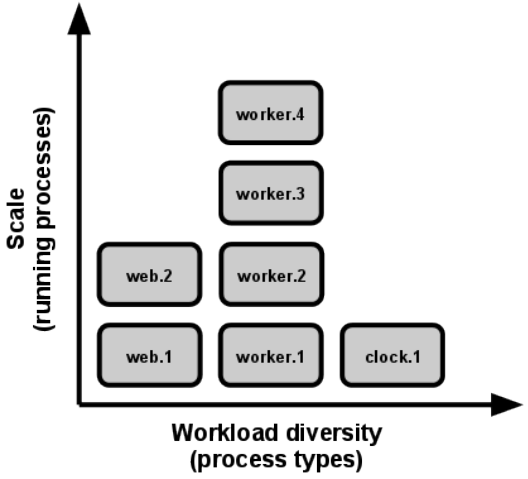
- Aplikacija treba da bude sačinjena od jednog ili više *stateless* procesa
- To znači da nijedan proces ne vodi evidenciju o stanju drugog procesa i da nijedan proces ne vodi evidenciju o informacijama kao što su status sesije ili toka posla
- Skaliranje je dosta jednostavije koristeći ovaj princip
- Ovaj princip omogućava da se instance mogu dodati i ukloniti da bi se odgovorilo opterećenju u datom trenutku

7. Vezivanje portova

- Ovaj princip tvrdi da se servis ili aplikacija može identifikovati na mreži prema broju porta, a ne imenu domena
- Razlog za ovu je vrlo jednostavan, imena domena i IP adrese se mogu dodeliti u hodu ručnom manipulacijom i automatskim mehanizmima
- Osnovna ideja koja stoji iza principa vezivanja porta je uniformna upotreba broja porta najbolji način da se proces otvori ka mreži
- npr. port 80 je port za konvencijlane web aplikacije, 443 je za HTTPS, port 22 je za SSH itd.

8. Konkurentnost

- Organizovanje procesa u skladu sa njihovom svrhom
- Odvajanje procesa tako da se mogu skalirati prema potrebi
- Podrška konkurentnosti znači da se različiti delovi aplikacije mogu skalirati kako bi se zadovoljile potrebe



(https://12factor.net/concurrency)

9. Jednokratna upotreba

- Procesi su jednokratni, što znači da se mogu pokrenuti ili zaustaviti u svakom trenutku
- Maksimizovati robustnost aplikacije sa brzim pokretanjem i koristeći *graceful shutdown* mehanizam
- Niti jedan proces nije specijalan i poseban, sve ih tretiramo isto
- Ophodimo se prema njima kao da u svakom momentu mogu nestati, a ako se to desi vrlo je moguće da ima naredni da prihvati posao

10. Dev/Prod Par

- Omogućiti da su *development*, *staging* i *production* što je moguće identični
- Aplikacija je dizajnirana za kontinuirani *deployment* tako što drži mali jaz na liniji razvoja-produkcija
- Na ovaj način dosta je jednostavnije testiranje i izvrŠvanje i kasniji pronalazak i uklanjanje grešaka

11. Logovi

- Logovi omogućavaju direktna uvid ponašanje aplikacije koja se izvršva u produkciji
- U tradicionalnim web aplikacijama oni se obično zapisuju u datoteku
- Bolji pristup je da logove tretiramo kao *event streams*
- Agregirati sve logove i sortirati prema vremenskoj odrednici
- Logovima se obično bave pridružene aplikacije (npr. sidecar patern)

12. Admin procesi

- Upravljanje ili administrativne operacije treba da se odvijaju u istom kontekstu kao i redovni i tekucći procesi aplikacije
- Administrativni procesi dolaze sa aplikacijom
- Izvršavaju se u istom okružneju kao i aplikacija uz iste konfiguracoine elemente

Dodatni materijali

- 12 factor app
- Graceful shutdown in go
- An illustrated guide to 12 Factor Apps
- What is a 12-factor application and why should you care?

Kraj predavanja

Pitanja? :)