

FOCUSED CANARY DEPLOYMENT

ELVÉGZETT MUNKA

- Kutatás deployment stratégiák terén
- Kutatás canary szerű deploymentek megvalósítására kubernetesben
- Canary szerű deploymenteket támogató demo elkészítése
 - Saját többkomponensű demo app
 - CI pipeline Docker image-ek legyártására
 - Kubernetes+Istio deployment

DEPLOYMENT STRATEGIES

- Basic: Egyszerre cserélünk le mindent, hátránya hogy még egy alaposan letesztelt rendszer esetén is lehetnek meglepetések, lassú rollback.
- Rolling: Elindít egy példányt az új verzióból, ha jó, akkor leállít egy régit és elindít egy újat. Addig csinálja míg le nem cserélte mindet.
- Canary: Az új verzióra először csak a felhasználók egy kis hányadát irányítjuk, majd fokozatosan növeljük, amíg 100% nem lesz.
 - Lassú, de biztonságos
 - Bonyolult automatizálni

DEPLOYMENT STRATEGIES

- Blue-Green: Két egyforma környezet fut párhuzamosan, és abból egyik a staging, másik a prod. Update alkalmával megcserélik a két környezetet a loadbalancer átállításával.
 - Drága: két prod. környezetet kell fenntartani
 - Triviális és nagyon gyors a rollback
- A/B testing: Ugyan annak a szolgáltatásnak két verziója fut élesben párhuzamosan. Ki lehet vele próbálni funkciókat, lehet kísérletezni.

IMPLEMENTING CANARY IN KUBERNETES

KUBERNETES EMLÉKEZTETŐ

- Elosztott alkalmazásokat lehet vele menedzselni, akár több fizikai, vagy virtuális gépen keresztül.
- Pod \sim konténer (1, vagy néhány konténer összessége, oszthatatlan alapegység)
- Node: Egy virtuális, vagy fizikai gép, amin fut a rendszerünk.
- Service: Egy szolgáltatás absztrakciója, pl.: a frontend, az adatbázis, nem konkrét podokat határoz meg
- Deployement: Ez indítja a podokat, ennél kell beállítani pl.: az image nevét és a volume-okat.

NATIVE KUBERNETES

- Erősen limitált támogatás
- Nem lehet célzottan irányítani a forgalmat
- Csak a podok számának igazításával lehet százalékosan megosztani a forgalmat:
90% stabil – 10% canary -> 9 régi pod, 1 új pod
 - Drága, felesleges podok futhatnak.
 - Nem minden load balancing algoritmusnál fogja a kívánt arányt eredményezni.
 - Blue-Green és A/B így nem valósítható meg.

WEIGHTED LOAD BALANCING

- Súlyozott round-robin terhelés elosztást lehet megvalósítani.
- 90% - 10% arányt el lehet érni 1-1 poddal is.
- Meg lehet valósítani reverse proxyval is.
- Támogatja: NGINX, Traefik, Ambassador, HAProxy, Istio

FOCUSED LOAD BALANCING

ISTIO

- Service mesh
- Load balancing:
 - HTTP headerk alapján
 - Súlyozott round robin
 - Akár IP cím alapján, de nehezebb mint NGINX esetén

NGINX

- Webszerver és reverse proxy
- Load balancing:
 - Akár IP cím alapján
 - Súlyozott round robin
 - HTTP headerk alapján
- Hátránya, hogy plusz egy proxy

IMPLEMENTATION

ISTIO

- A Kubernetes bonyolult, de az Istio még csavar rajta egyet.
- „Beépül a Kubernetesbe és felokosítja.”



Traffic Management

Deploy capabilities like inter-service routing, failure recovery and load balancing.



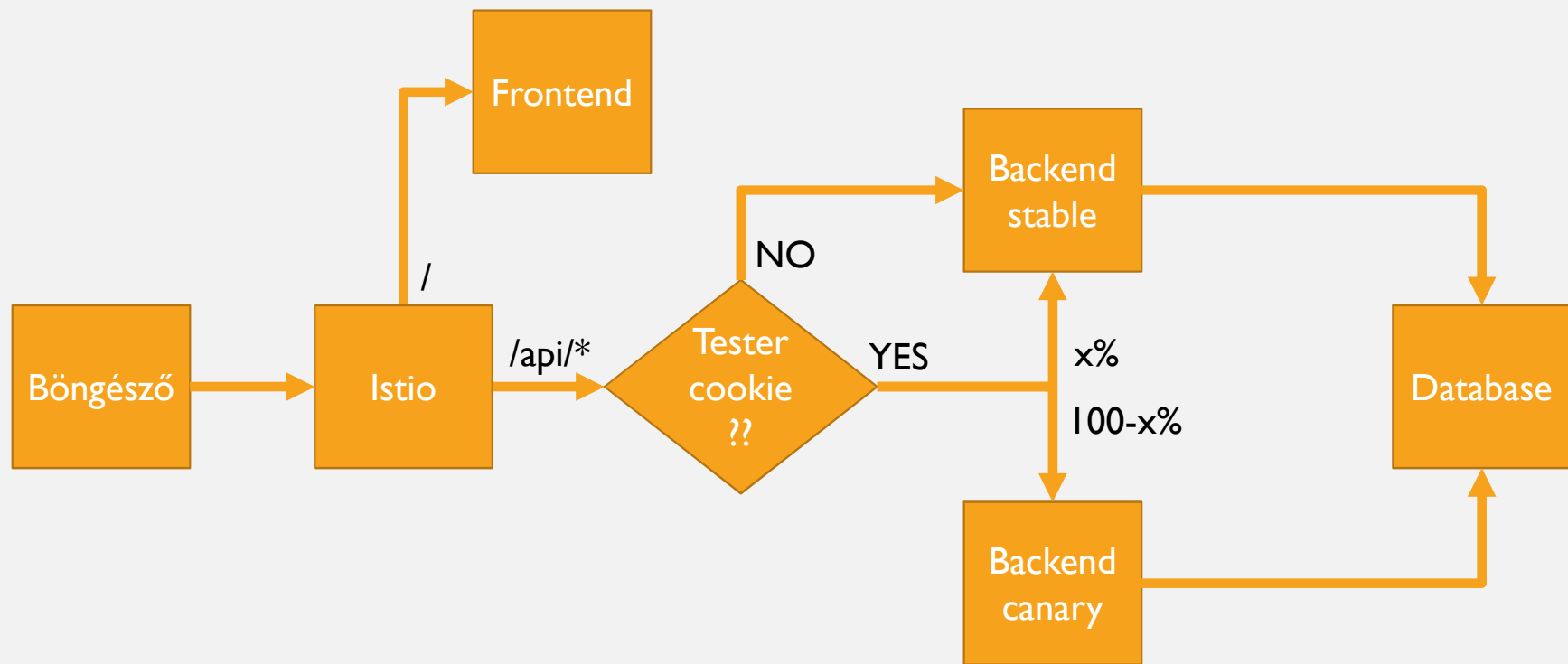
Observability

Provide an end-to-end view of traffic flow and service performance.



Security

Engage encryption, role-based access, and authentication across services.



DEMO

- App ismertetése
 - Front end
 - Kotlin app
 - Multi stage build
 - CI: image build and upload
- Kube/*.yaml fájlok magyarázata
- Kibana
- Helm