## Metode de upgrade in Kubernetes

Student:

Eusebiu Rizescu

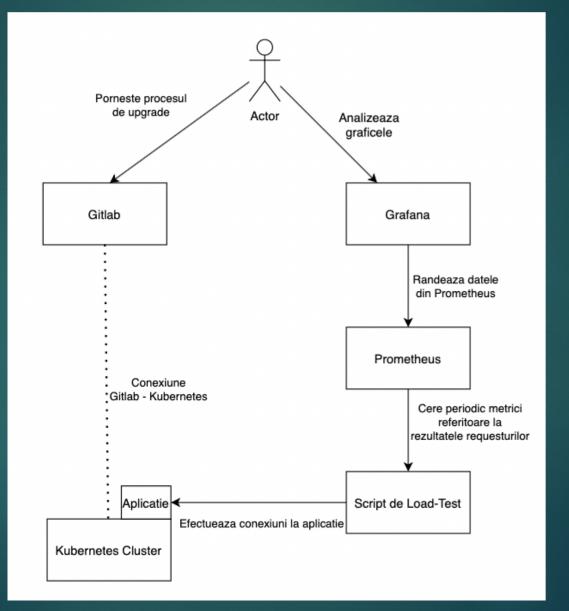
Coordonator științific:

Conf.dr.ing. Radu-Ioan Ciobanu

#### Metode de upgrade analizate

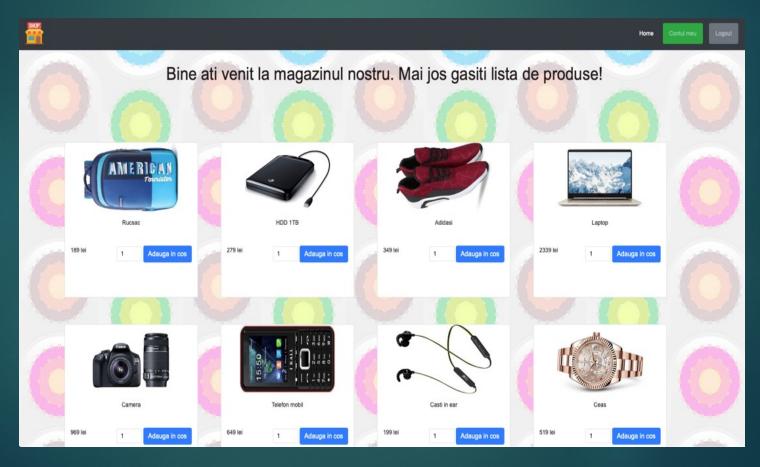
- Recreate
  - Sters Versiunea1 si creat Versiunea2
- Rolling Update
  - Updatat la Vers2 pe rand, cate o instanta a aplicatiei
- ▶ Blue/Green
  - ▶ Pornit complet Vers2 si apoi mutat traficul de pe V1 pe V2
- Canary
  - X% din traffic pe Versiunea2
- A/B Testing
  - Un subset de useri pe Versiunea2

### Arhitectura testare



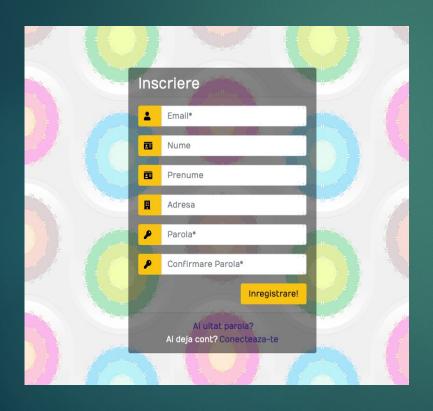
# Aplicatia

- Aplicatie web, scrisa in Flask
- Aplicatia consta intr-un magazin online



## Inregistrare si Autentificare

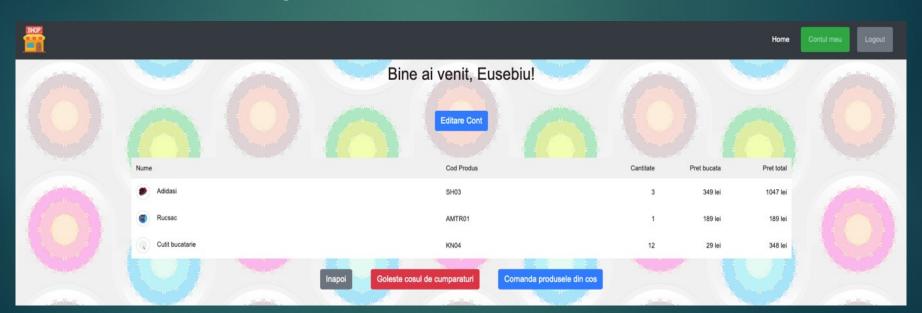
Pentru a lansa o comanda, utilizatorul trebuie sa aiba cont si sa fie autentificat





## Pagina utilizatorului

- De aici utilizatorul poate:
  - comanda produsele din cos
  - goli cosul de cumparaturi
  - ▶ Edita / sterge contul



#### Clusterul de Kubernetes

- Creat cu ajutorul Minikube
- Ruleaza pe local
- Gitlab.com este configurat sa interactioneze cu acest cluster

# Scriptul de load-test

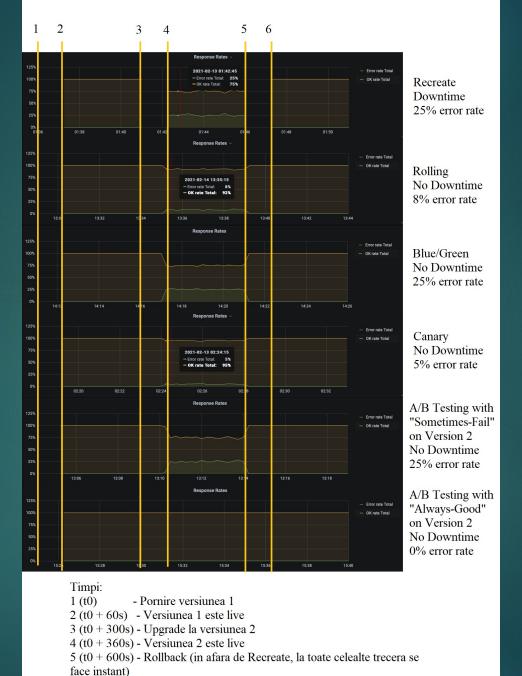
- Simuleaza trafic real
- Lanseaza permanent requesturi catre aplicatie
- Expune metrici cu rezultatele requesturilor, care vor fi citite de Prometheus
- Nota: Acest script este configurat astfel incat 25% din requesturile care ajung in Versiunea 2 sa fie eroare

### Prometheus si Grafana



# Masuratori

Metoda/Parametru	Downtime la upgrade	Rata de eroare dupa upgrade	Downtine la downgrade	Resurse in plus necesare
Recreate	Da	25%	Da	Nu
Rolling update	Nu	8%	Nu	+33%, podul cu versiunea noua
Blue/Green	Nu	25%	Nu	+100%
Canary (20%)	Nu	5%	Nu	+33%, podul pentru canary
A/B testing (Always- Good pe Versiunea 2)	Nu	25%	Nu	+100%
A/B testing (Sometimes-Fail pe Versiunea 2)	Nu	0%	Nu	+100%



6 (t0 + 660s) - Versiunea 1 este live (la Recreate)

#### Demo

- Demo pentru metoda Canary
- https://vimeo.com/manage/videos/569074606