

# **AWS at Zonky**

Roman Pichlik

# What is Zonky

- P2P lending platform
- Borrower, Investor
- 1.5billion in loans
- Single ops engineer

# Zonky deployment

- Hybrid
- AWS (4/5 capacity)
- SiteOne (private datacenter, virtualization, networking etc.) (1/5 capacity)
  - DB on baremetal

**Use cases for AWS**

# Zonky homepage

- TV campaign
  - CloudFront, ELB, Beanstalk, RDS
- Issues
  - ELB scalability
- No linux admin
- Quick win

# Server Side Rendering

- Node.js
- Ember.js
- CloudFront, ELB, Beanstalk
- Issues
  - Peaks
  - Pricing hourly
- No linux admin
- Quick win

# Bcrypt offloading

- (String password) -> bcrypt(password)
- Investment peaks
- Duration 60s <
- Issues
  - Peaks
  - Pricing hourly

# Jenkins/Build scale out

- Build, Test and Deploy pipelines
- 0downtime releases
  - Every pull request spin a new instance for tests
- EC2 jenkins plugin
  - Jenkins slaves on top of spot fleet instances



# Developer or test instances

- Zonky in Docker (single instance)
  - On demand by one click
  - backend, frontend branches
  - Heavily used during stabilization
- EC2 spot instances
  - Quick win, but doesn't scale

# Test environments

- Dev, Test, Pre-prod, Stage
  - Same topology as production (HA services)
  - EC2 spot instances
- Issues
  - Expensive
    - Need to implement scale down
  - Fragile
    - EC2 spot instances terminated

# Patterns or use cases for AWS

- Scale up/down use cases
  - Pay for uses time and not reserved time
- Instant environments (Elastic beanstalk) for rapid prototyping

# AWS disadvantages

- Monitoring/BI
  - CloudChecker
  - Lambda for counting running instances
  - EC2 vs EBS price
- Tags limitation
  - run lambda to add tag e.g. on EBS
- Limited automation
  - CloudFormation
  - Ansible, AWS SDK

# Architecture for cloud

- If it's your strategy vertical scalability forget about cloud
- Re-think the lifecycle model
  - Conventional thinking
    - Application layer on top of permanent resources (EC2)
  - Non-conventional thinking
    - Application layer on top of Docker/K8s on top of EC2 spot and reserved instances
- Services can die
- Reactive async rather than pro-active sync
- Producer/Consumer decoupled via a queue

# Open questions for us

- Multi region app deployment vs Single region app deployment
- K8s as a service?
- Vendor lock-in