TV 2 PLAY'S JOURNEY TO THE CLOUD

Migrating one of Denmark's largest streaming services to the cloud

WHO AM I?

Martin Villumsen

- MSc in Computer Science
- Full stack experience
- Working at TV 2 since August 2017
  - 7 months as DevOps @ central DevOps team
  - Last year as developer / SRE @ TV 2 Play Video Platform team

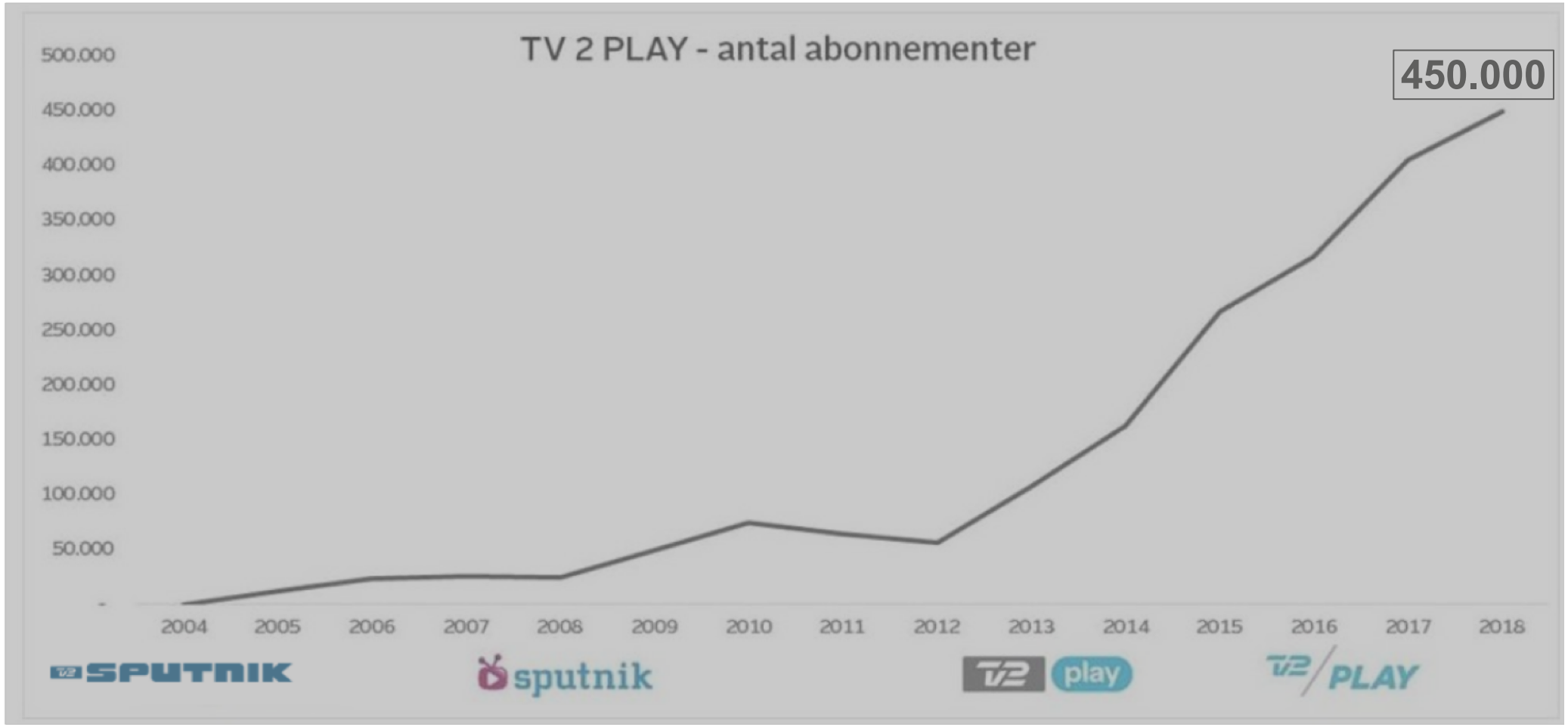
BACKGROUND

TV 2 Play

- One of Denmark's largest streaming services
- Focus on Danish content
- Video on-demand
- TV 2s own live channels
- Targeted ads to HVOD users

BACKGROUND

TV 2 Play



BACKGROUND

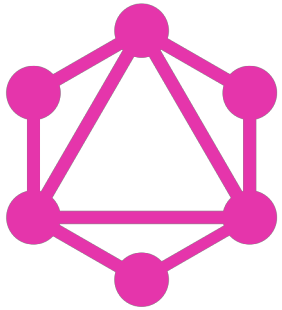
TV 2 Play

- TV 2 Play is developed across five teams
  - Web
  - Mobile Apps
  - Customer and Ad Management
  - **Video Platform & Recommendations**
  - Streaming

BACKGROUND

Video Platform & Recommendation

- 20+ different microservices
  - Java
  - Docker
- GraphQL Gateway serves data to clients



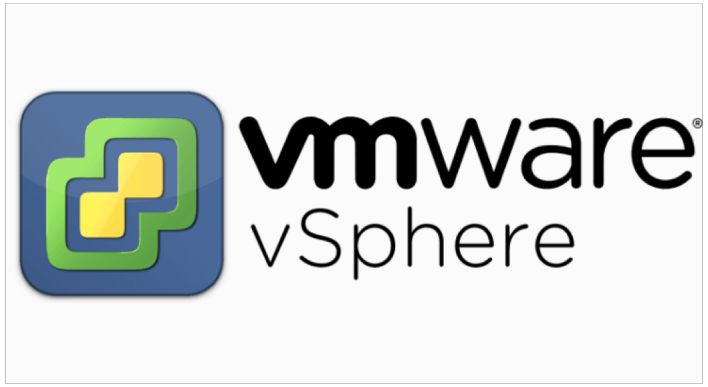
BACKGROUND

Video Platform & Recommendation

- 20+ microservices
  - Java
  - Docker
- GraphQL Gateway serves data to clients
- Handle and delivers data to clients, e.g.:
  - TV Guide
  - DRM
  - Playbacks
  - Navigation and site structure
  - Recommendations

BACKGROUND

TV 2 Play History

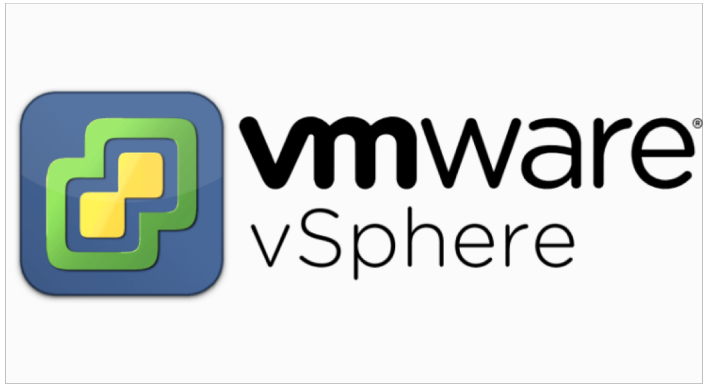


- Started out as a PHP monolith
- Hosted on on-premise datacenter
- Hard to scale

BACKGROUND

TV 2 Play History

- **2015**
  - Microservice architecture
  - Docker
  - Rancher (on-premise)



BACKGROUND

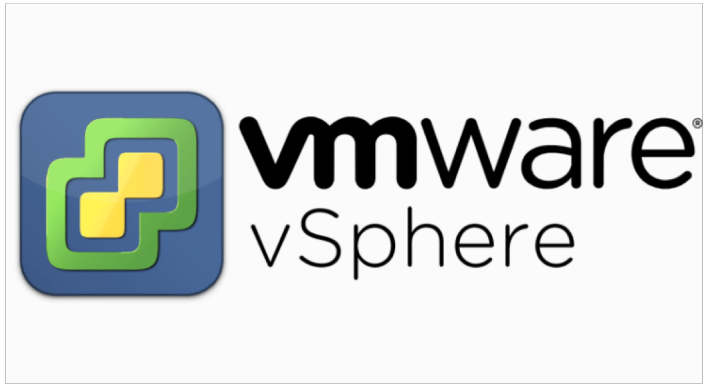
TV 2 Play History

- **2015**
  - Microservice architecture
  - Docker
  - Rancher (on-premise)

- **2017-2018**
  - Growing customer base → capacity issues
    - Christmas Calendar
    - Handball World Cup
    - Badehotellet

BACKGROUND

TV 2 Play History

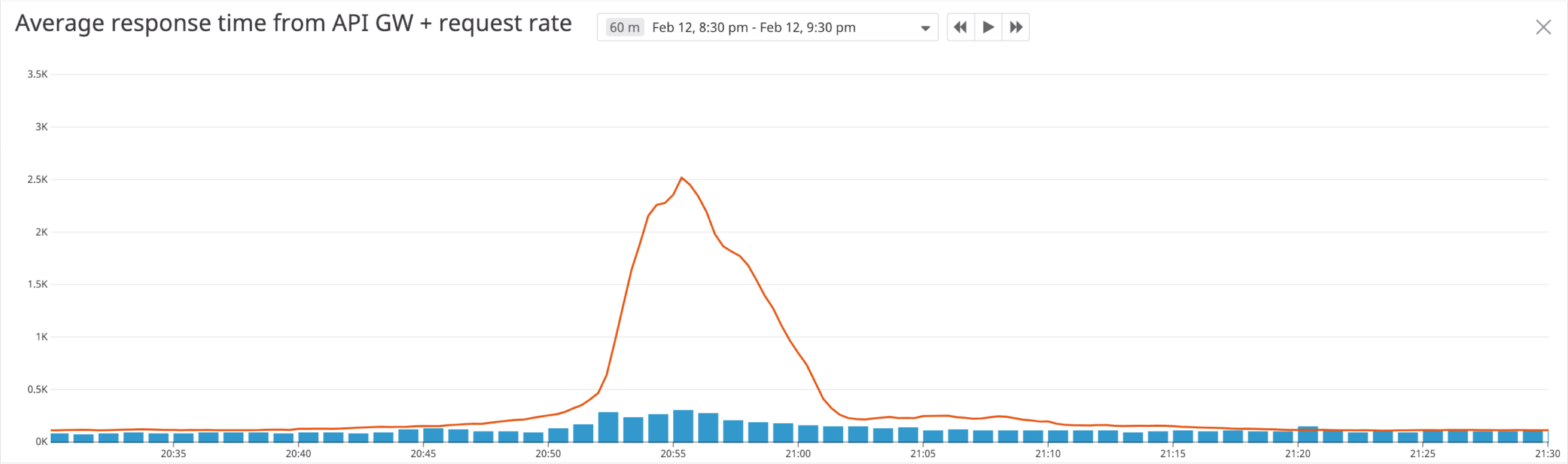


- **2015**
  - Microservice architecture
  - Docker
  - Rancher (on-premise)

- **2017-2018**
  - Growing customer base → capacity issues
    - Christmas Calendar
    - Handball World Cup
    - Badehotellet
  - ”Snigpremiere” → self-denial attack → cascading failures

MOTIVATION

Badehotellet ”snigpræmiere”



MOTIVATION

FIFA World Cup 2018

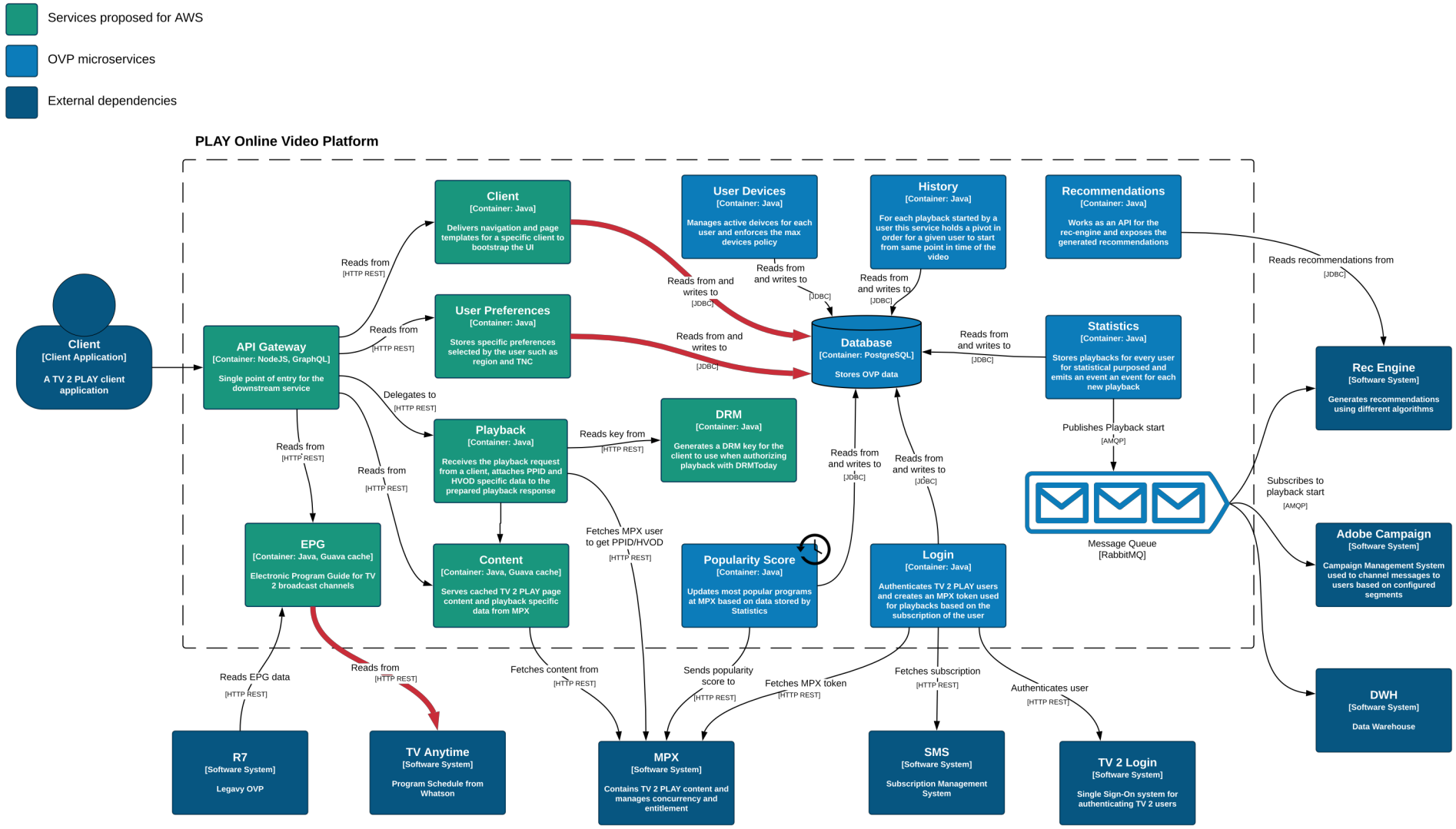
AND SO THE JOURNEY BEGINS…

MIGRATING TO AWS

Planning

- Amazon Web Services
  - Leverage managed services
  - Already hosted another project on AWS
- Tight deadling
  - Planning started in January
  - Started migration in March
- What services to migrate first?

MIGRATING TO AWS



MIGRATING TO AWS

Choosing a container orchestrator

- Previously seen issues with Rancher 1.6 (Cattle)
- Rancher 2.0 was just released
  - Was it good enough for production?

MIGRATING TO AWS

Choosing a container orchestrator

- Previously seen issues with Rancher 1.6 (Cattle)
- Rancher 2.0 was just released
  - Was it good enough for production?

RANCHER®

MIGRATING TO AWS

Choosing a container orchestrator

- Kubernetes
  - Well tested
  - Big community
  - Lots of CNCF project built for/around Kubernetes
  - Learning by doing

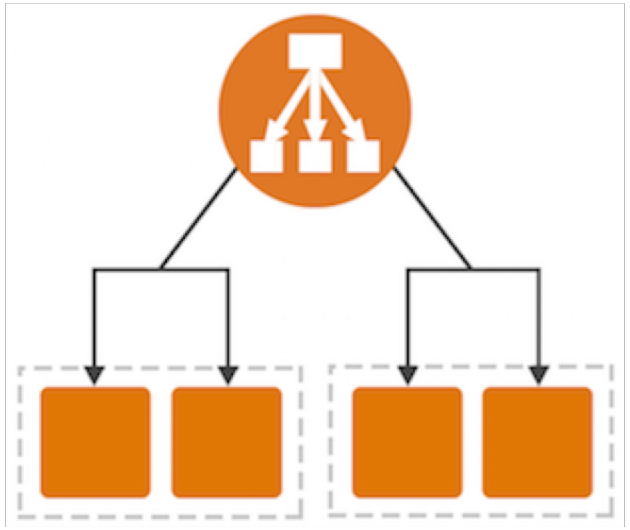
KUBERNETES

Getting started

- Bootstrap sandbox cluster with kops
- Already had docker-compose files for our projects
- Convert with Kompose (Kubernetes + Compose)
  - Great starting point

KUBERNETES

Exposing the services to the world

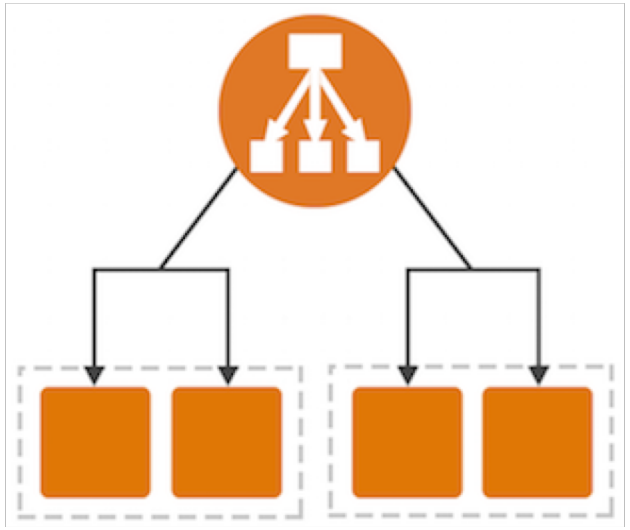


- How can we expose our applications?
- LoadBalancer service not suited for production
- What ingress controller to choose?
- Wanted to leverage AWS ALB
  - → ALB Ingress Controller

KUBERNETES

Exposing the services to the world

- How can we expose our applications?
- LoadBalancer service not suited for production
- What ingress controller to choose?
- Wanted to leverage AWS ALB
  - → ALB Ingress Controller
- Manage DNS with External DNS



KUBERNETES

Handling application configurations

- 3 cluster on separate AWS accounts
  - Different application configurations
- Using .env files in Rancher
- Wanted something similar in Kubernetes
- Helm – Kubernetes Packages Manager
  - Templating
  - Versioning
  - Rollbacks
  - … and more

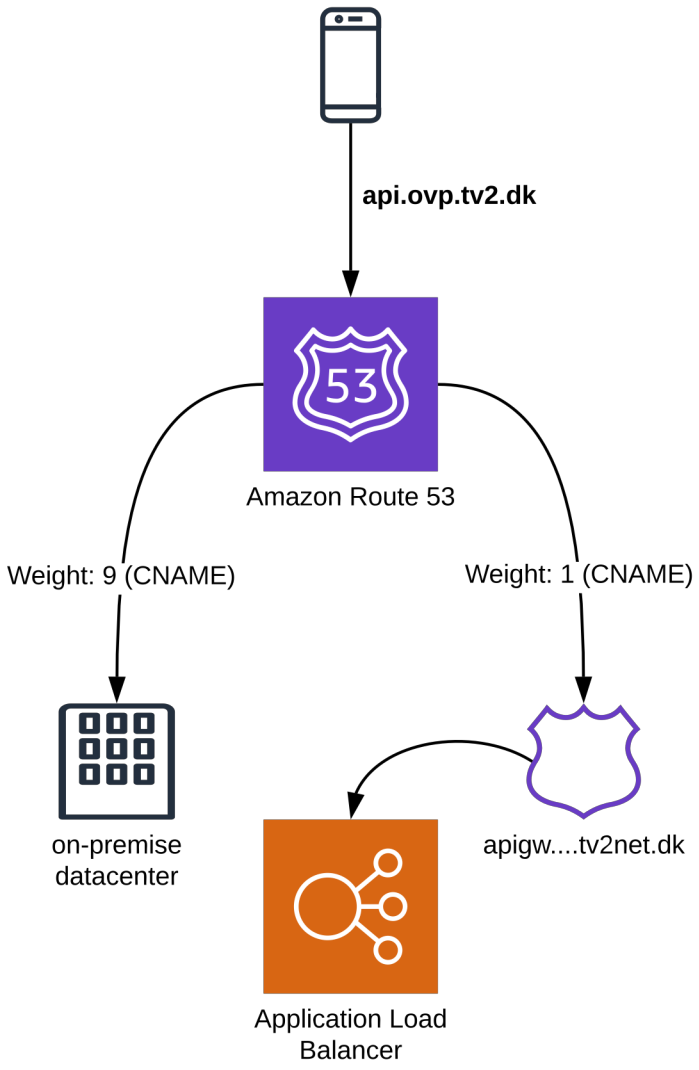
```
.
├── Dockerfile
├── Jenkinsfile
├── README.md
├── build.gradle
├── deployment
│   ├── dev.yaml
│   ├── prod.yaml
│   └── stage.yaml
├── gradle
│   ├── artifactory.gradle
│   ├── dependencies.gradle
│   ├── publishing.gradle
│   ├── repositories.gradle
│   ├── sonarqube.gradle
│   ├── test.gradle
│   ├── version.gradle
│   ├── war.gradle
│   └── wrapper
├── gradle.properties
├── gradlew
├── gradlew.bat
├── settings.gradle
└── src
    ├── integration-test
    ├── main
    └── test
```

KUBERNETES

Going into production

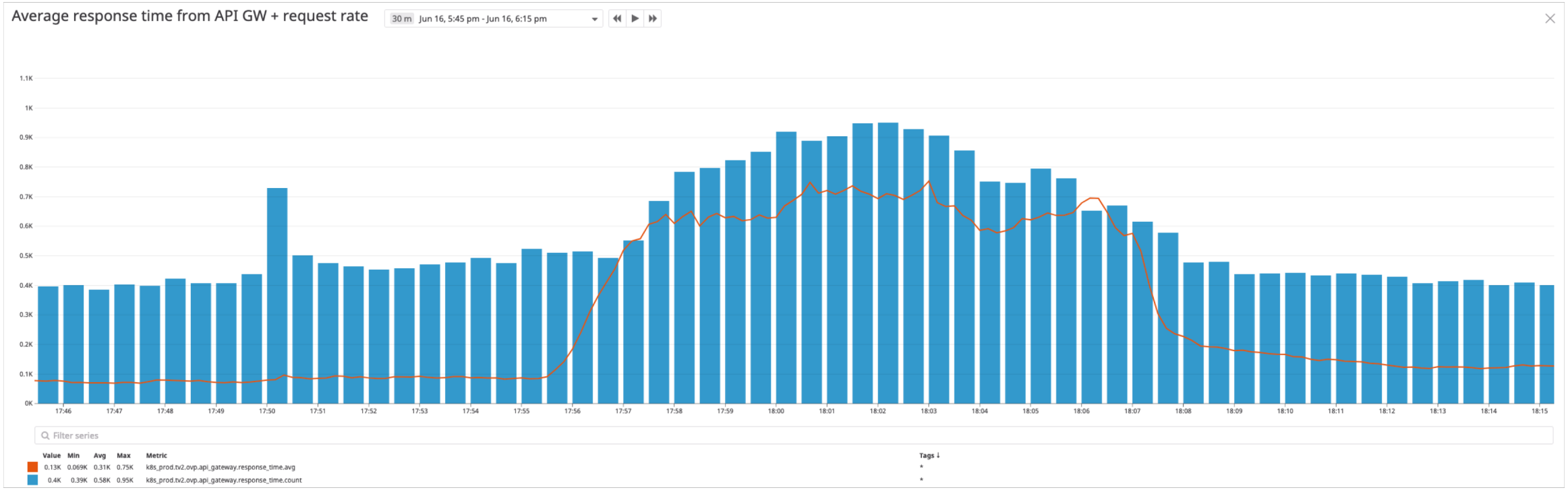
- Services deployed to Kubernetes
- Still missing some features but running out of time

- Shifting traffic to AWS
  - Redelegate DNS to Route53
  - Weighted CNAME records



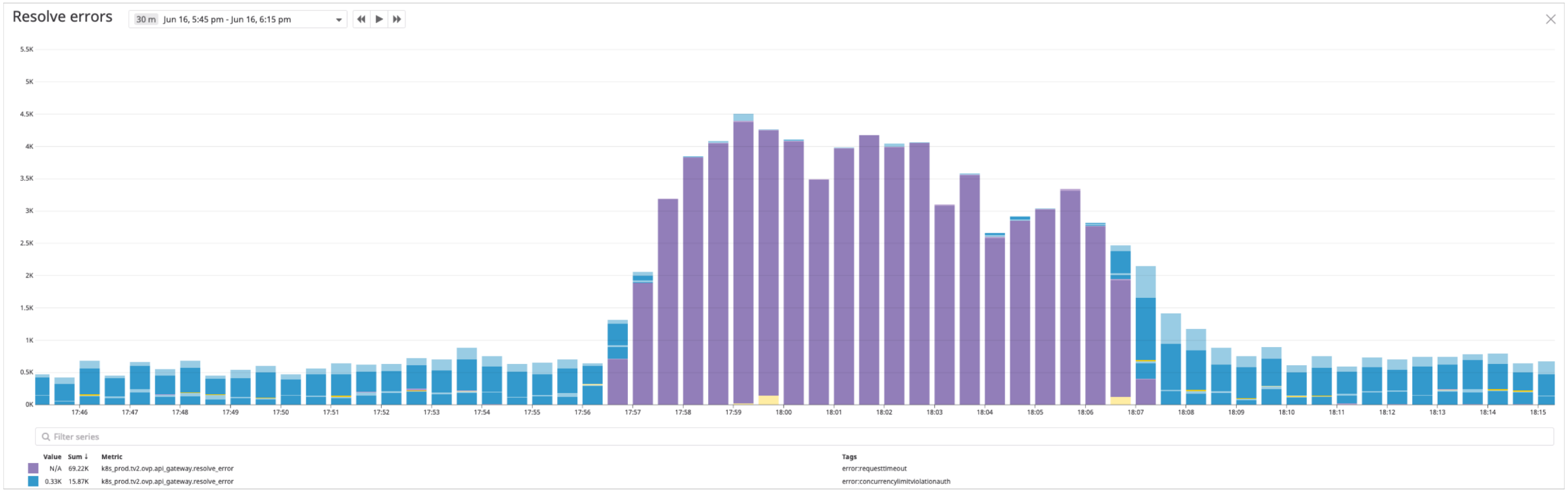
FIRST OUTAGE

FIFA World Cup 2018 Denmark - Peru



FIRST OUTAGE

FIFA World Cup 2018 Denmark - Peru



KUBERNETES

Manual deployment to Kubernetes

- Manual deployment
  - `helm upgrade…`
- Humans make mistakes
  - `helm upgrade release—a tv2/release-b…`
- Need to automate deployment to Kubernetes

KUBERNETES

Using our existing CI/CD setup

- Use existing Jenkinsfiles
- No nice CD solutions for Kubernetes
- Pragmatic approach
  - `helm upgrade…`

KUBERNETES

Using our existing CI/CD setup

```
withCredentials([file(credentialsId: kubeconfig, variable: 'KUBECONFIG')]) {
  def deployResult = sh(returnStdout: true,
                        script: "helm upgrade --install --debug ${releaseName} --values ${deployEnvFile} \
                                 tv2/${releaseName} --set image.registry=${registry},image.tag=${imageTag} \
                                 --namespace ${namespace} --force --wait --kubeconfig ${KUBECONFIG}")
  echo deployResult
}
```

KUBERNETES

Monitoring & log collections

- Been using DataDog since 2016
- Great for metrics and alerting
- Support for a lot of integrations
- DataDog agent deployed as a DaemonSet
- Supports HPA on custom metrics

KUBERNETES

Monitoring & log collections

- On-premise ELK cluster for logs
- Shipping logs from AWS to on-premise
- Filebeat agent deployed as a DaemonSet

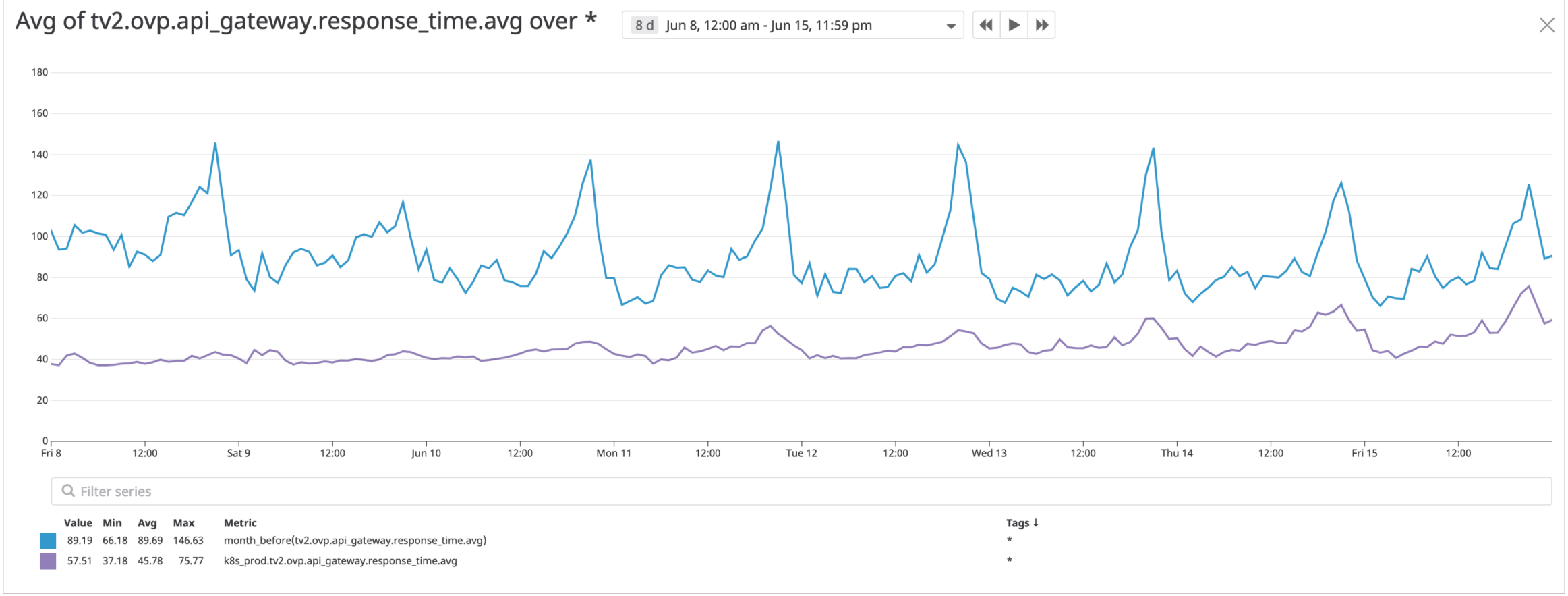
ONE YEAR LATER

What did we achieve

- Survived the FIFA World Cup
- More resilient infrastructure
- Ability to scale
- Deliver faster
- Reduced response times

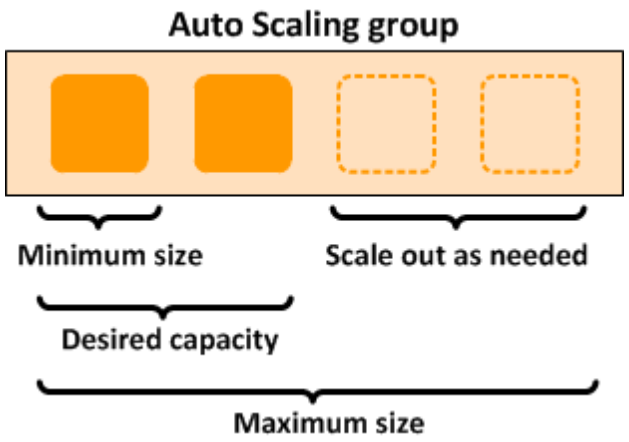
ONE YEAR LATER

What did we achieve



WHAT'S NEXT?

The future of TV 2 Play



- We have come a long way but we still have quite some work to do
- Move remaining services to AWS
- Dynamic autoscaling (pods and cluster)
- Amazon EKS

Amazon EKS

WE'RE HIRING!

THANK YOU!

Martin Villumsen
mvil@tv2.dk
@VillumsenMartin