Kubernetes Navigation Stories

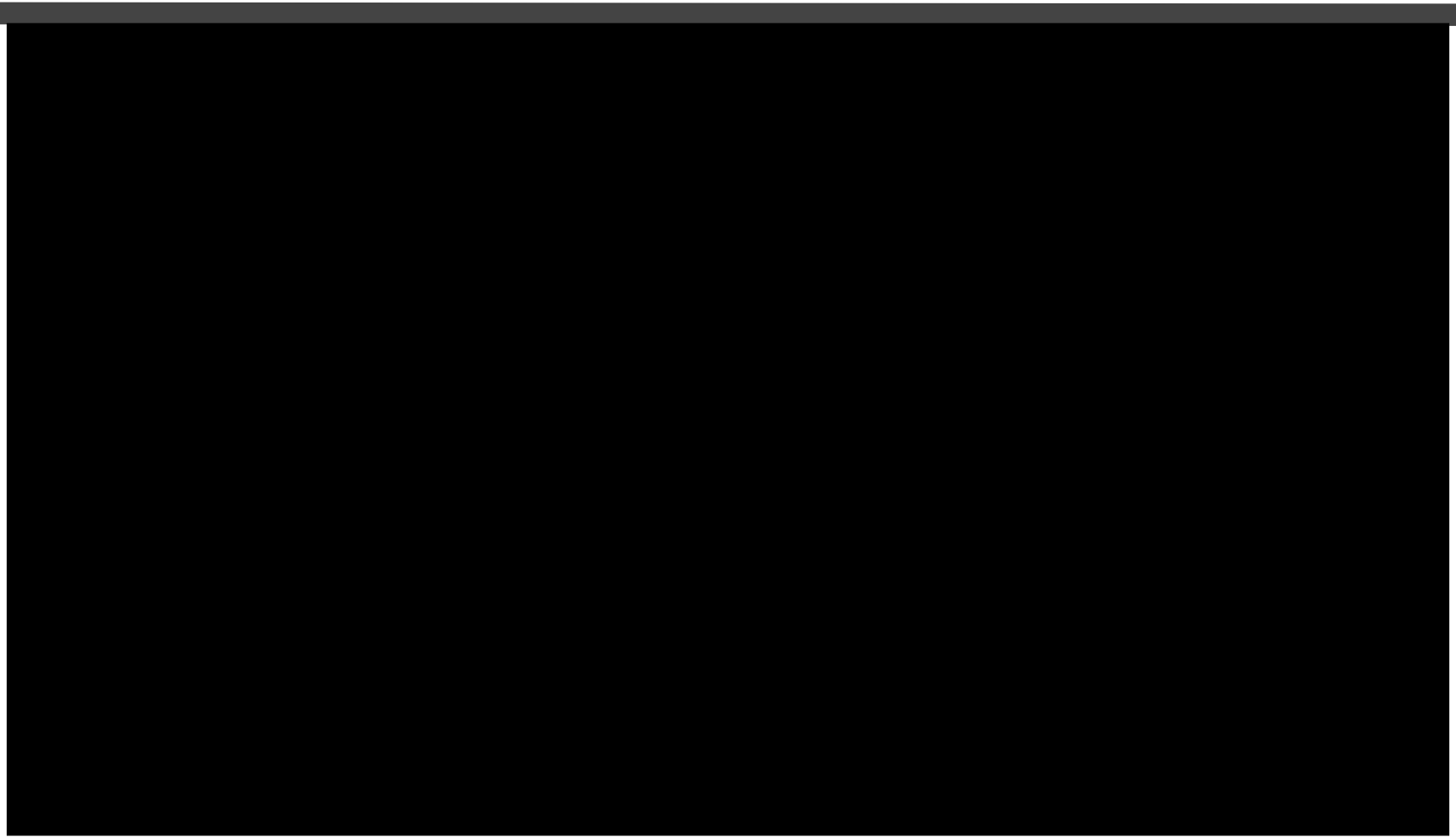
DevOpsStage 2019

Roman Chepurnyi

Director of Infrastructure Engineering at thredUP
Senior Engineering Manager at Hotwire

Oleksii Asiutin

Staff Software Engineer at thredUP
Senior Software Engineer at Toptal



ThredUP Technology

- 70 Software Engineers
- 5 Infrastructure Engineers
- 50 applications
- 100 EC2 nodes

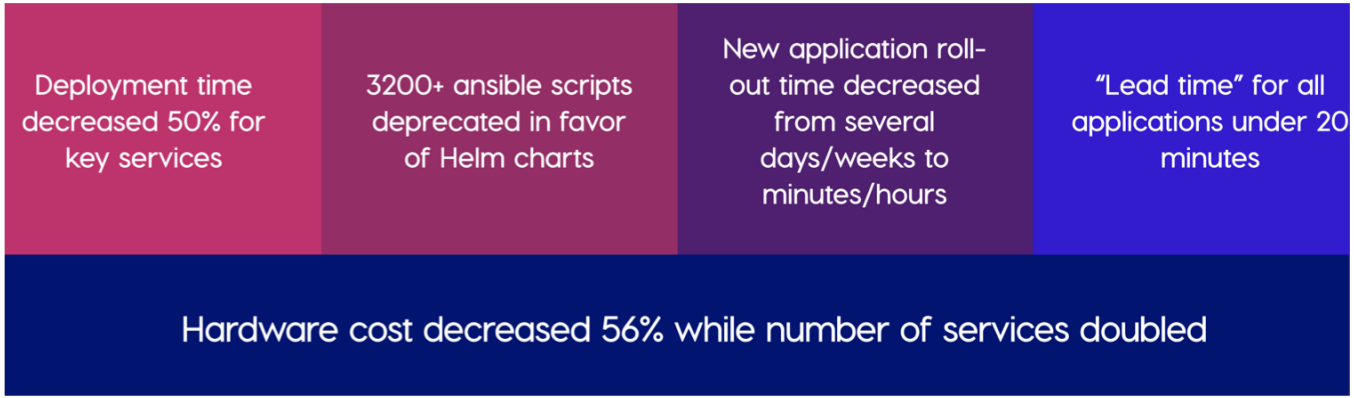
Stack

- NodeJS, react
- Ruby, .NET, Java
- RabbitMQ, SQS
- Redis
- MySQL Aurora

100% in k8s since mid-2018

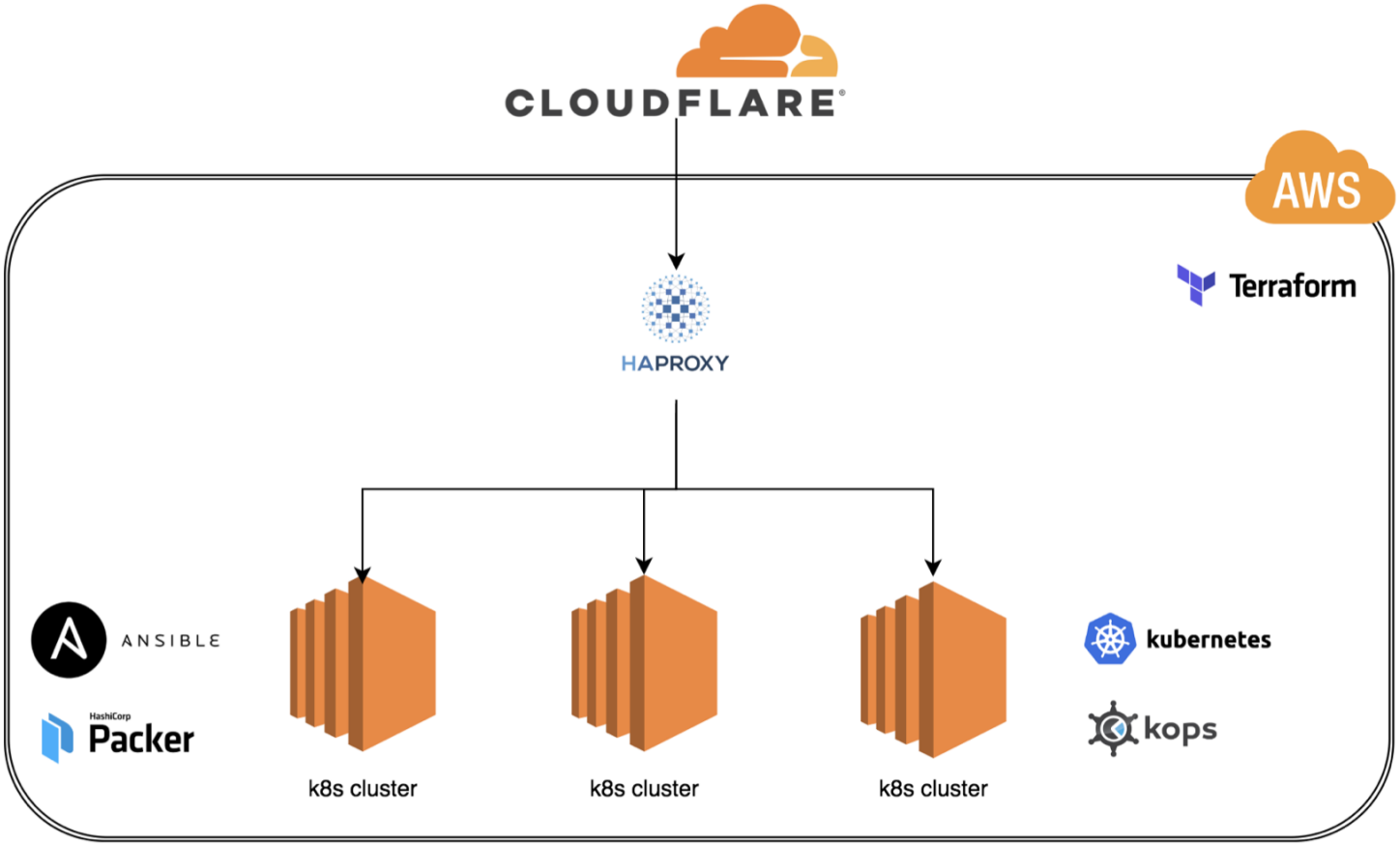
CNCF Case study

https://www.cncf.io/thredup-case-study/



THREDUP

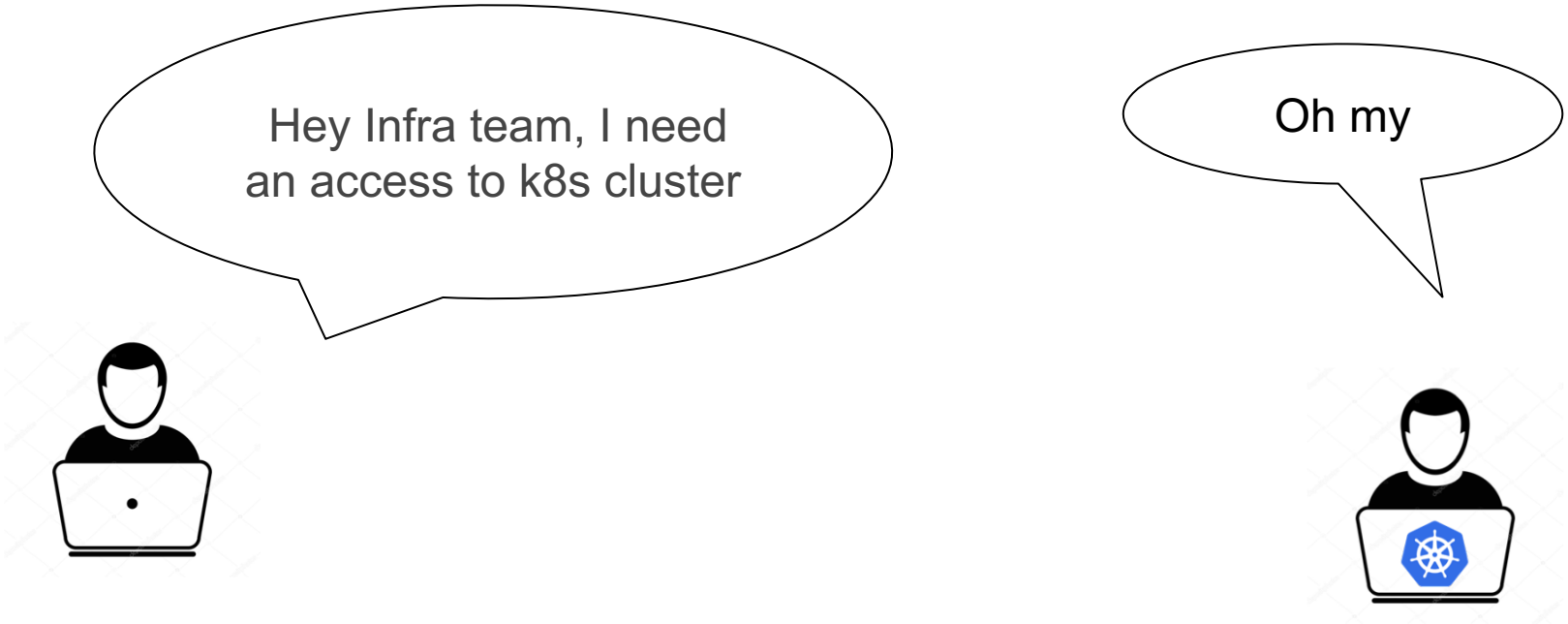
ThredUP Infrastructure



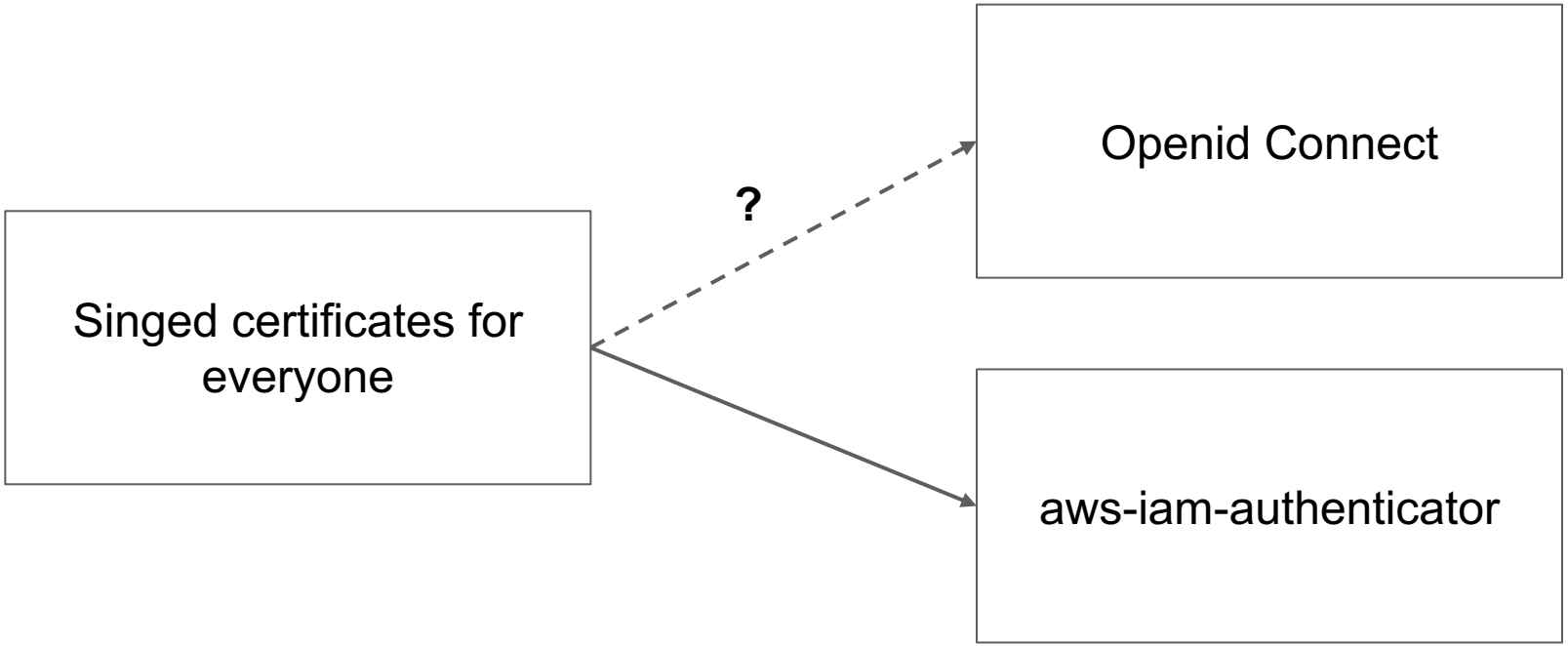
Life after Kubernetes migration

- **Fixing shortcuts and gaps**
  - IAM
  - Secrets management
- **Developers experience**
  - Staging environment
  - Local development
- **Infrastructure optimization**
  - Auto-scaling
  - Spot Instances
  - Security
  - Networking

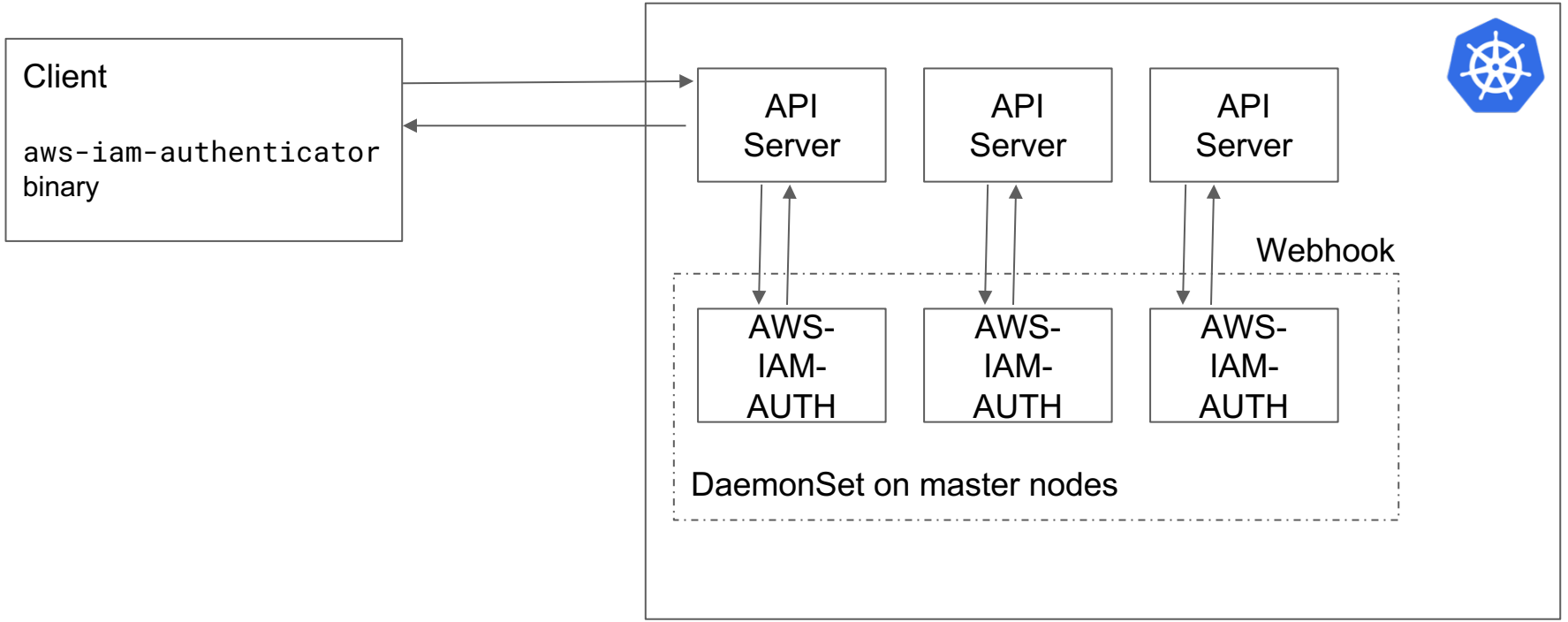
Authentication



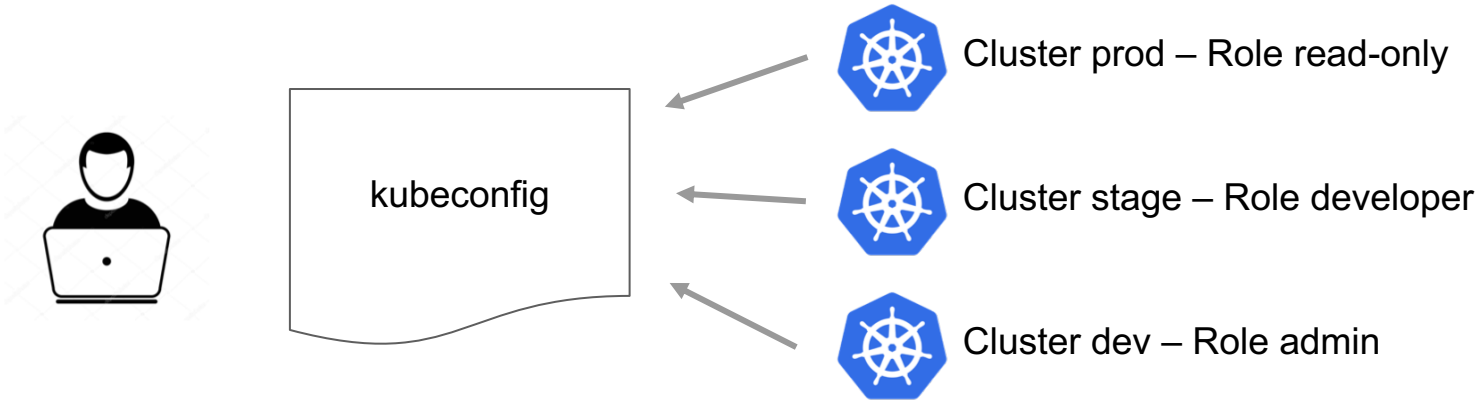
Auth mechanisms



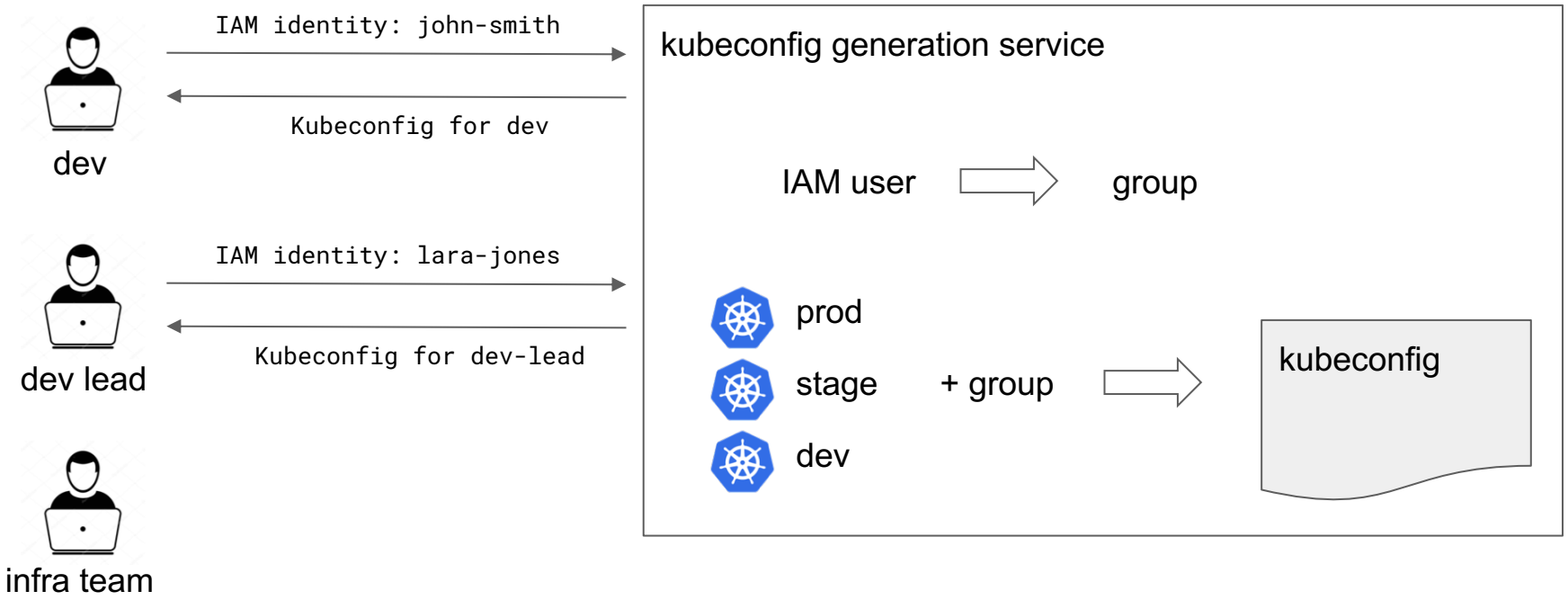
AWS-IAM-Authenticator



AWS-IAM-Authenticator – kubeconfig



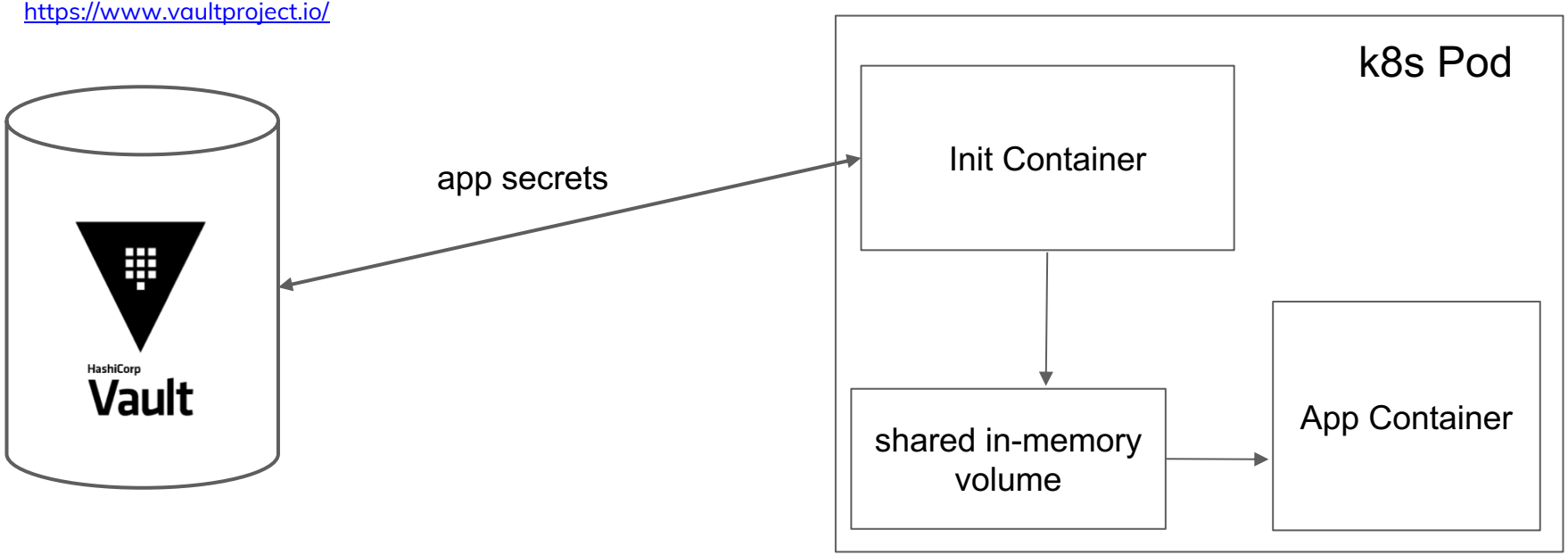
AWS-IAM-Authenticator – kubeconfig generation



Secrets Management

Hashicorp Vault

https://www.vaultproject.io/



https://github.com/cruise-automation/daytona

THREDUP

SOPS – Secrets OPerationS

https://github.com/mozilla/sops

```
# secrets.production.yaml
app_secrets:
 db_username: cart_service
 db_password: supersecret
```

Supported formats:

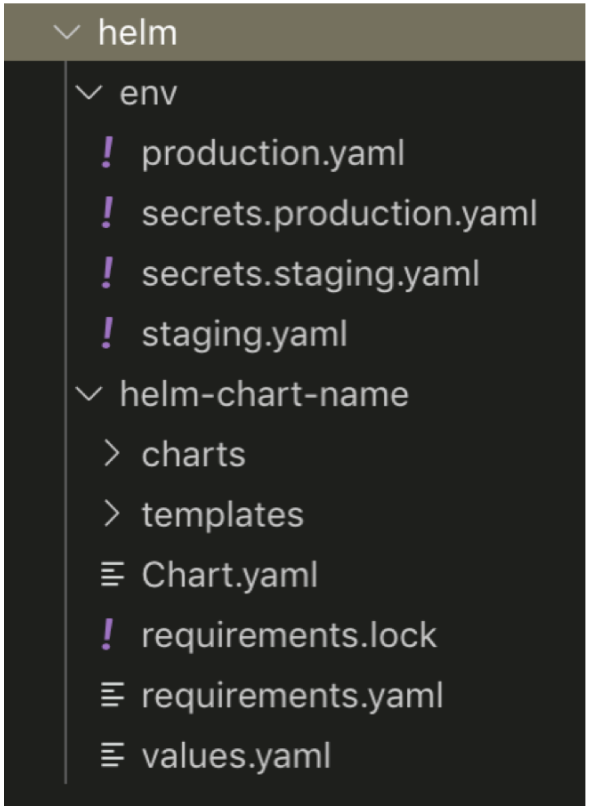
YAML
JSON
.env

SOPS – Encryption

```
$ sops -e --kms <AWS-KMS-ARN> secrets.production.yaml

app_secrets:
    db_username:
ENC[AES256_GCM,data:KuhPWLhijVc/9wa6,iv:V7YS/QglsuYwpmBcTZjOwFz8p10yt+qOcRgg+/OL4Uo=,tag
:jchhWABpUVYK4kpRKlrYPQ==,type:str]
    db_password:
ENC[AES256_GCM,data:TWjWb4up6nx+gSk=,iv:VoI9vnYrIdYxjTmSsqFzbXZ9z8LsZp4ud8LgVocxGAs=,tag
:PVNKEAq3OvWGiUSmM3aHpw==,type:str]
sops:
    kms:
    -   arn: AWS-KMS-ARN
        created_at: '2019-09-26T09:00:30Z'
        enc:
AQICAHhGGWsaRwq5wtMieLutm2hnsC2WqAifhQ6HgfjDUdbvpQE5pwGLIOabNseXxCnNWo0YAAAAfjB8BgkqhkiG
9w0BBwagbzBtAgEAMGgGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQMhPJ/IHKNPgmqzN8vAgEQgDvTzDYH71MH
x5nGWHjzNjpNDjnTw3pgS8IPf26qVhcdrO7Uv1g7yjKsJIVdcD00/hSNCgg6+KgulNgHmw==
    gcp_kms: []
```

SOPS – helm template and secrets storage



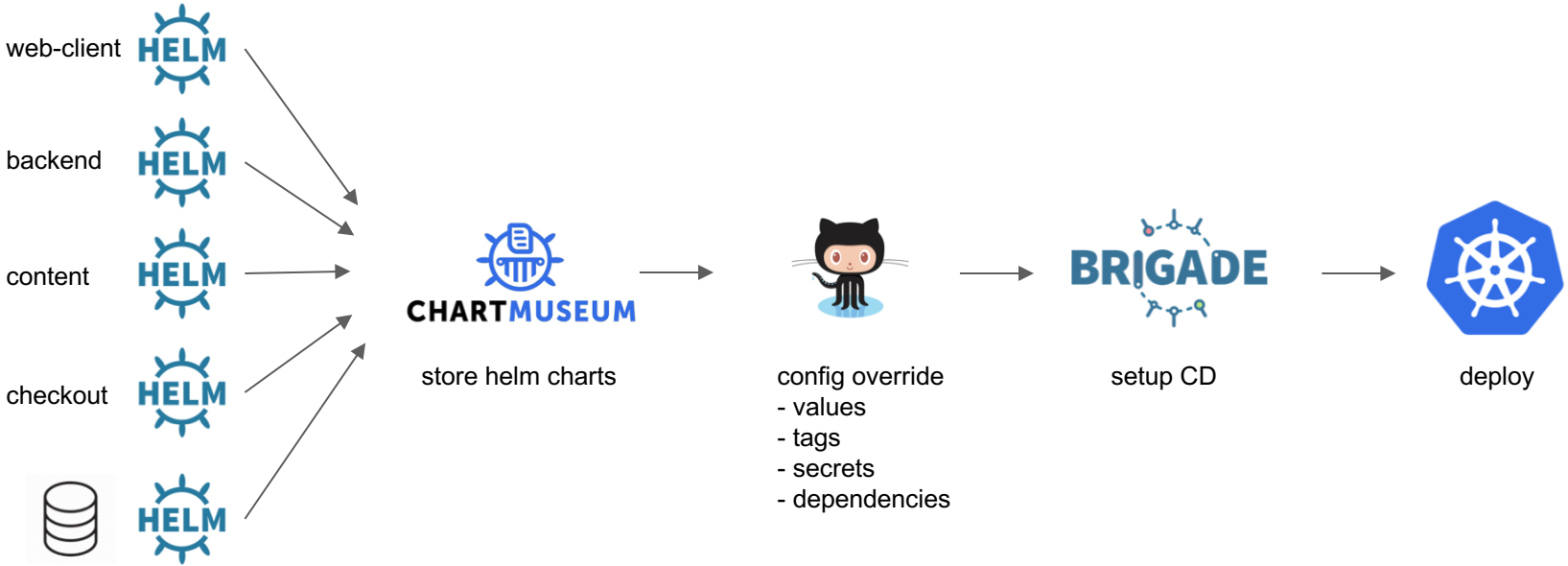

```yaml
apiVersion: v1
{{- with .Values.app_secrets }}
data:
{{ toYaml . | indent 2 }}
{{- end }}
kind: Secret
metadata:
  annotations:
    helm.sh/hook: pre-install,pre-upgrade
    helm.sh/hook-delete-policy: before-hook-creation
  name: {{ template "fullname" . }}
  labels:
    app: {{ template "fullname" . }}
type: Opaque
```

SOPS – Deployment

```
$ sops -d -i ./helm/env/secrets.production.yaml

$ helm upgrade --install --wait --timeout 600 \
               -f ./helm/env/secrets.production.yaml \
               -f ./helm/env/production.yaml \
               app_name ./helm/app_name
```

Staging Environments



THREDUP

Staging Environment

$git checkout -b devopsstage

$git push -u origin devopsstage

wait 4-5 min

use https://devopsstage.threduptest.com/

Local development

When your service has a lot of dependencies (MySQL, Redis, RabbitMQ and 5 other services)

Local Development

```
macbook: Thredup $ git clone git@github.com:thredup/node-proxy.git
Cloning into 'node-proxy'...
...
macbook: Thredup $ cd node-proxy/
macbook: node-proxy (master) $ npm install
added 6 packages from 8 contributors and audited 6 packages in 0.595s
found 0 vulnerabilities
macbook: node-proxy (master) $ npm test
> proxy@1.0.0 test ~/Thredup/node-proxy
...

macbook: node-proxy (master) $ npm start
> proxy@1.0.0 start
> node server.js
```

Local Development with Docker

```
macbook: Thredup $ docker run -it -v ${PWD}:/app -p 3000:3000
node:12-alpine sh
/ $ apk add --no-cache mysql-dev

/ $ npm install
/ $ npm test
/ $ npm start
> proxy@1.0.0 start
> node server.js
```

Local Development with Docker Compose

```
version: "3.7"
services:
  web:
    image: node:12-alpine
    volumes:
      - ./:/app
    ports:
      - "3000"
    environment:
      REDIS_HOST: "127.0.0.1"
  mysql:
    image: ...
    ...
  redis:
    image: ...
```

Local Development with Docker Compose

```
macbook: Thredup $ docker-compose up -d
…
macbook: Thredup $ docker-compose exec web sh

/ $ npm install
/ $ npm test
/ $ npm start
> proxy@1.0.0 start
> node server.js
```

And then you need another service as a dependency ;-)
...and another one
...

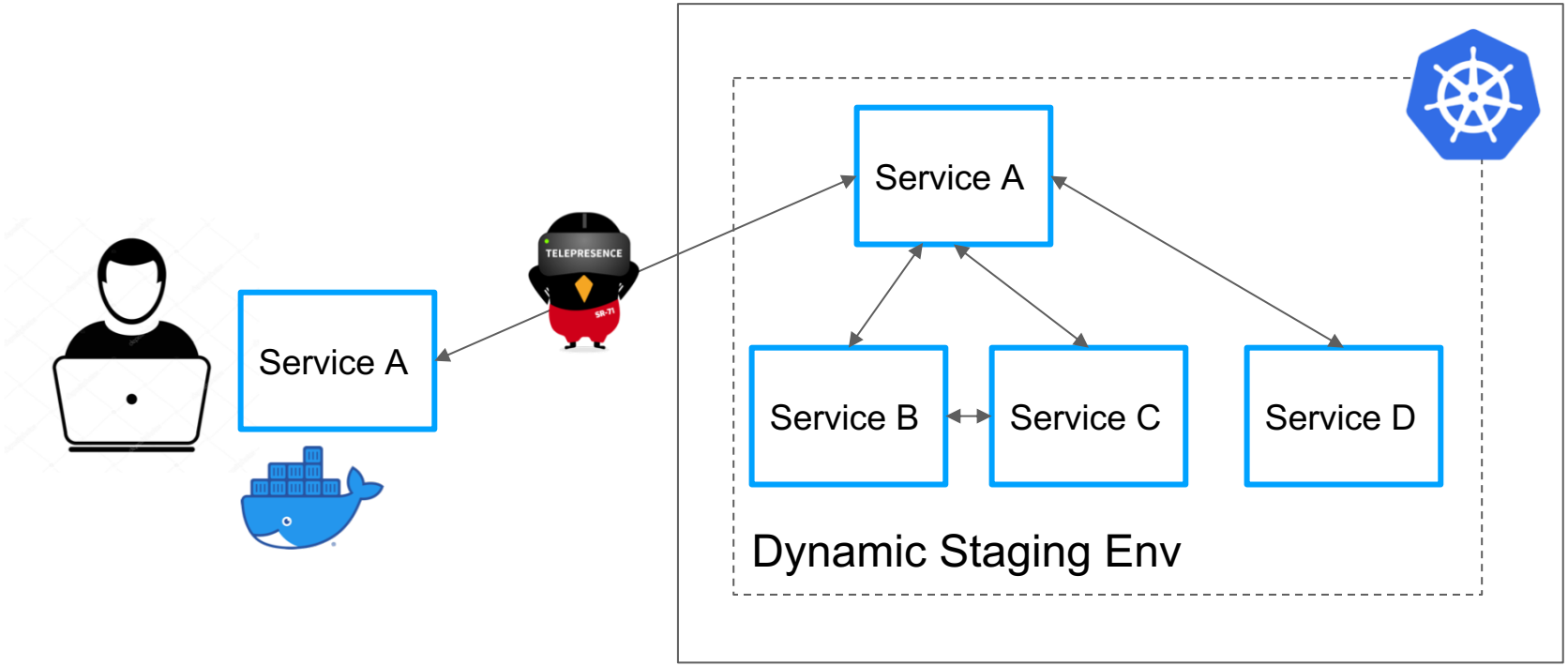
docker-compose.yaml ~ 330 lines

MySQL DB ~25Gb

And you need to keep it
UP TO DATE

Local development - Telepresence

https://www.telepresence.io/



THREDUP

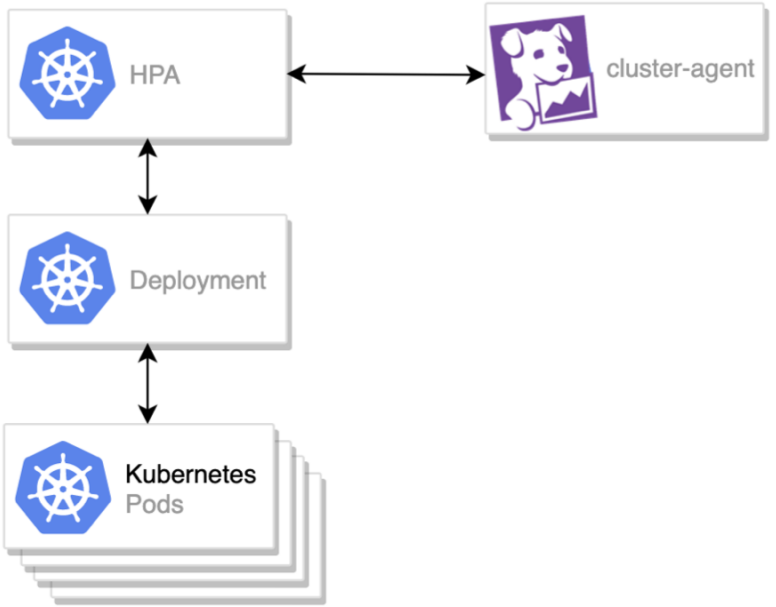
Local development with Telepresence

```
macbook: Thredup $ telepresence --swap-deployment          \
            deployment-name                                \
            --expose 3000                                  \
            --method container                             \
            --docker-run --rm -it -v ${PWD}:/app           \
            000000000001.dkr.ecr.us-east-1.amazonaws.com/cart:latest
...
...
/ $ npm install
/ $ npm test
/ $ npm start
> proxy@1.0.0 start
> node server.js
```

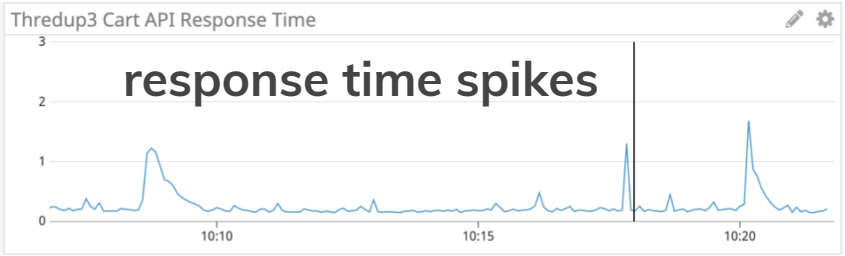
Horizontal Pod Autoscaling (HPA)

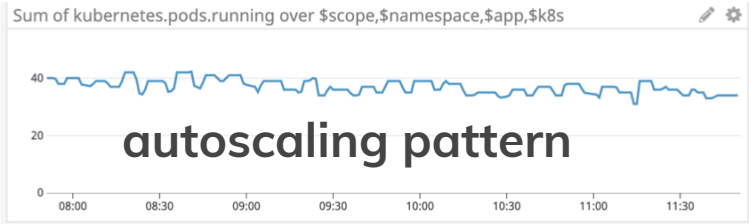
- Do not over-provision
- Be ready for traffic spikes

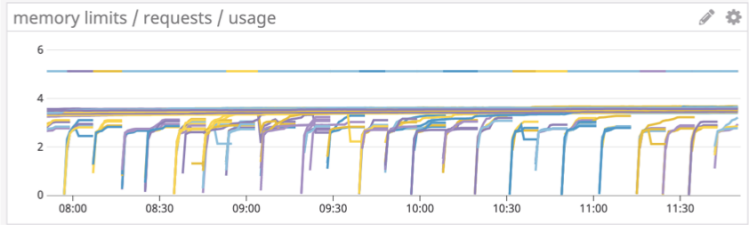
```
metrics:
 - type: External
   external:
     metricName: trace.rack.request.hits
     metricSelector:
       matchLabels:
         env : production
         service : some-service
     targetAverageValue: 10
```

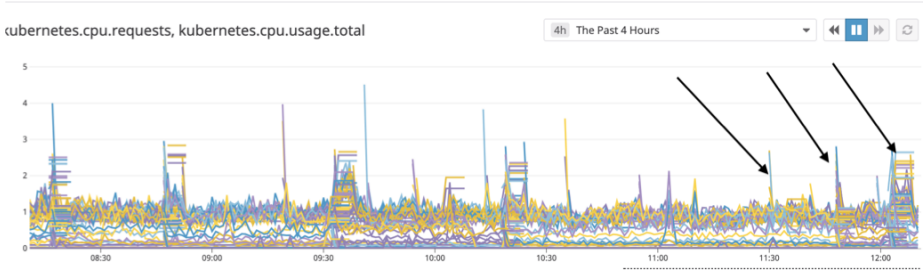


HPA lessons learned









offender pods:
request 1 core
use 3+ cores on start

THREDUP

HPA lessons learned

add warmup script

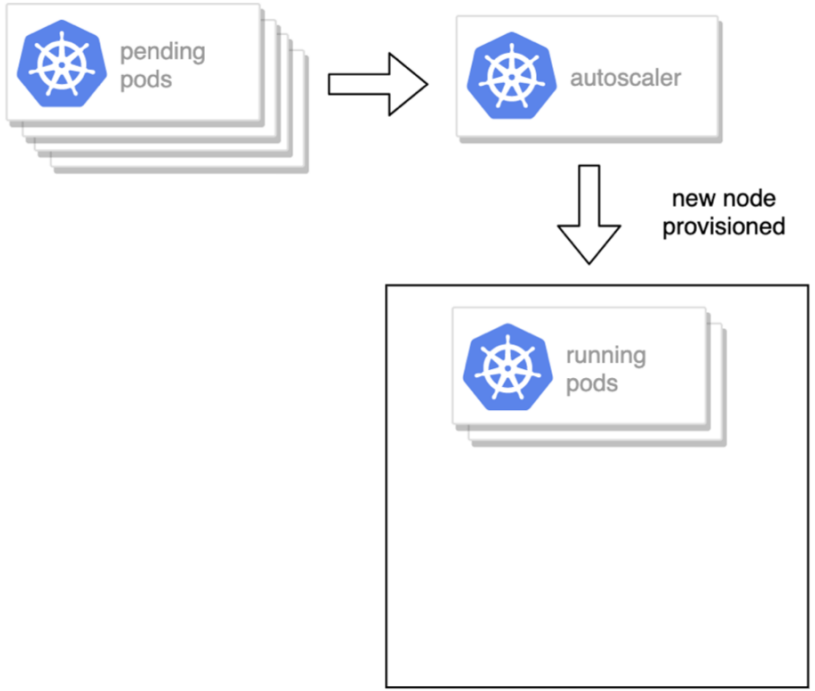
```
lifecycle:
  postStart:
    exec:
      command:
      - /bin/sh
      - -c
      - cd /app/script/warmup && ./warmup.sh
```

update deployment strategy

```
strategy:
  rollingUpdate:
    maxSurge: 25%
    maxUnavailable: 10%
  type: RollingUpdate
```

Cluster autoscaler

- overflow capacity in production
- utilize spot instances



THREDUP

Spot instances and AZRebalance

- spot termination works https://github.com/mumoshu/kube-spot-termination-notice-handler
- except when instance is terminated by Availability Zone

```
Terminating EC2 instance: i-0e685dc2a84b65f63
Cause:CauseAt 2019-07-18T06:09:59Z instances were launched to balance instances in
zones us-east-1a us-east-1e with other zones resulting in more than desired number of
instances in the group. At 2019-07-18T06:11:30Z an instance was taken out of service
in response to a difference between desired and actual capacity, shrinking the
capacity from 4 to 3. At 2019-07-18T06:11:30Z instance i-0e685dc2a84b65f63 was
selected for termination.
```

Spot instances and AZRebalance

```
metadata:
 creationTimestamp: 2017-10-12T16:28:23Z
 generation: 2
 name: m4xlarge
spec:
 image: 405610825889/harden-k8s-x.14-debian-stretch-amd64-hvm-ebs-2019-08-16
 machineType: m4.2xlarge
 maxPrice: "0.20"
 maxSize: 30
 minSize: 5
 role: Node
 rootVolumeSize: 100
 subnets:
 - us-east-1a
 - us-east-1c
 - us-east-1e
 suspendProcesses:
 - AZRebalanceapiVersion: kops/v1alpha2
kind: InstanceGroup
```

Container vulnerability scan

https://github.com/arminc/clair-scanner

jenkins APP 4:22 PM

Vulnerability scan completed:

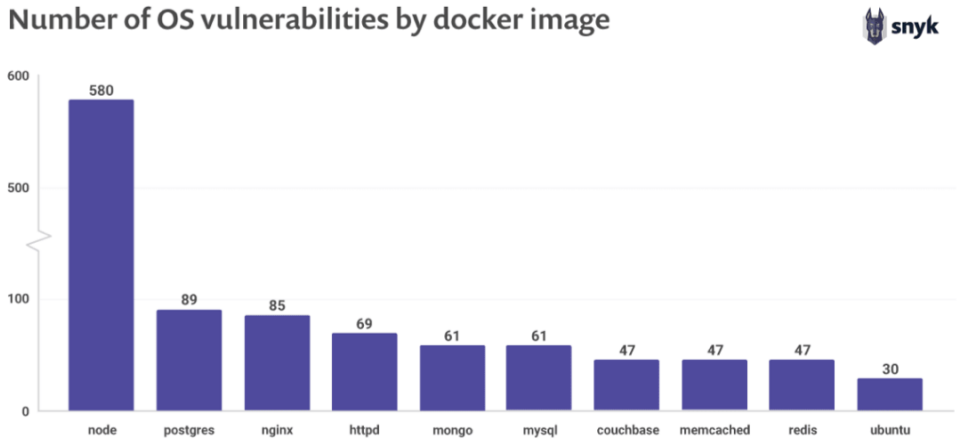
Medium: 7681

High: 2257

Critical:

DefCon1:

```
  16 alpine:v3.3
   9 alpine:v3.4
   4 alpine:v3.7
   2 alpine:v3.8
1015 debian:8
1199 debian:9
   4 ubuntu:14.04
  11 ubuntu:16.04
```



https://snyk.io/blog/top-ten-most-popular-docker-images-each-contain-at-least-30-vulnerabilities/

Container runtime security

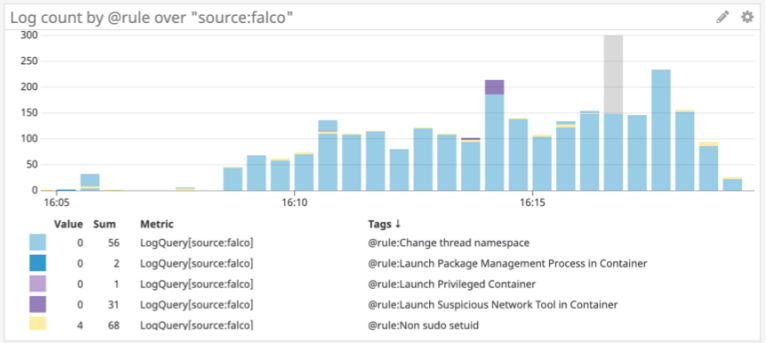
https://falco.org

Sysdig

Falco

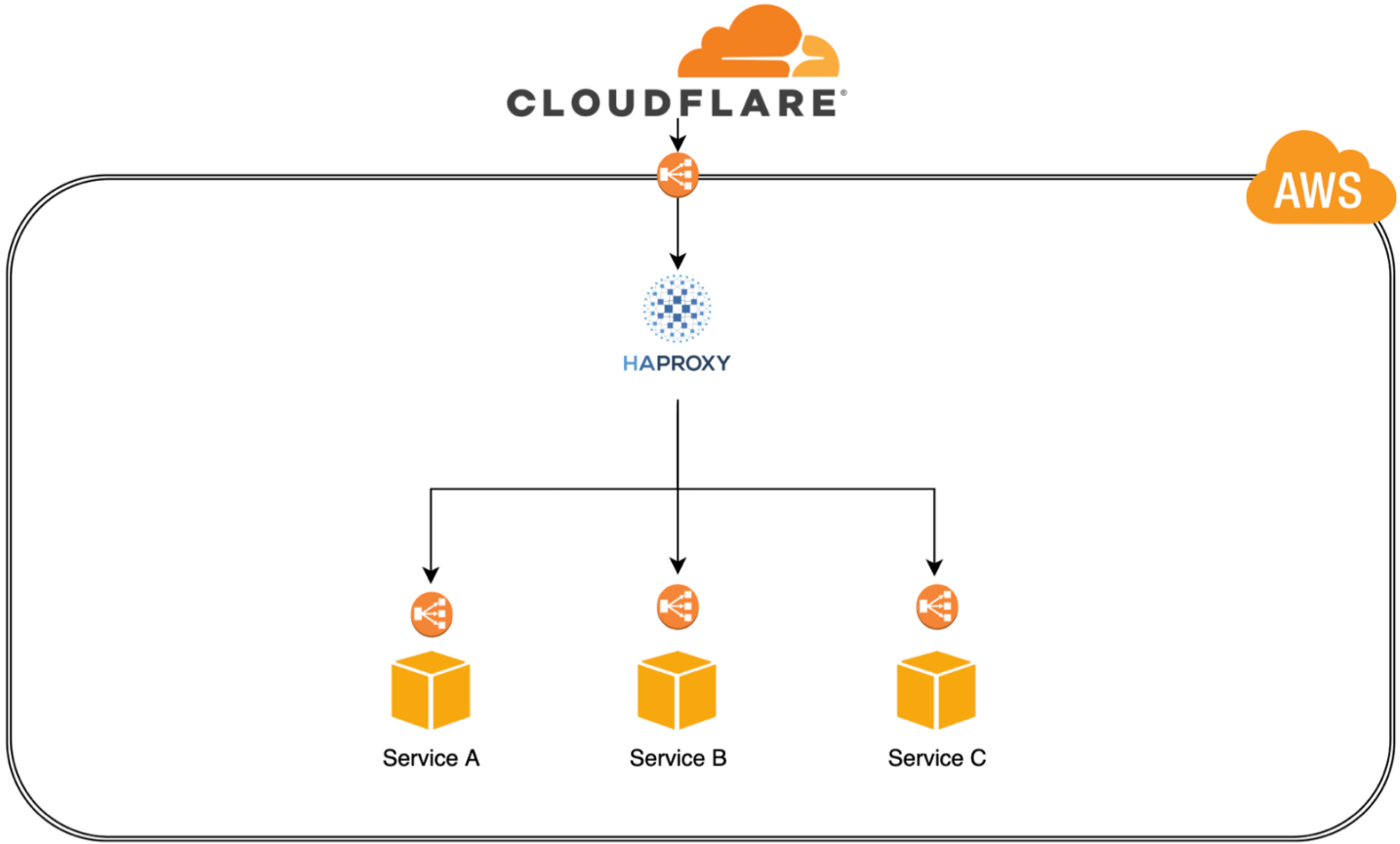
Log count by @rule over "source:falco"

@RULE	COUNT
System procs network activity	2.47K
Non sudo setuid	69
Change thread namespace	56
Launch Suspicious Network Tool in Container	31
Launch Package Management Process in Container	2
Launch Privileged Container	1
Run shell untrusted	1

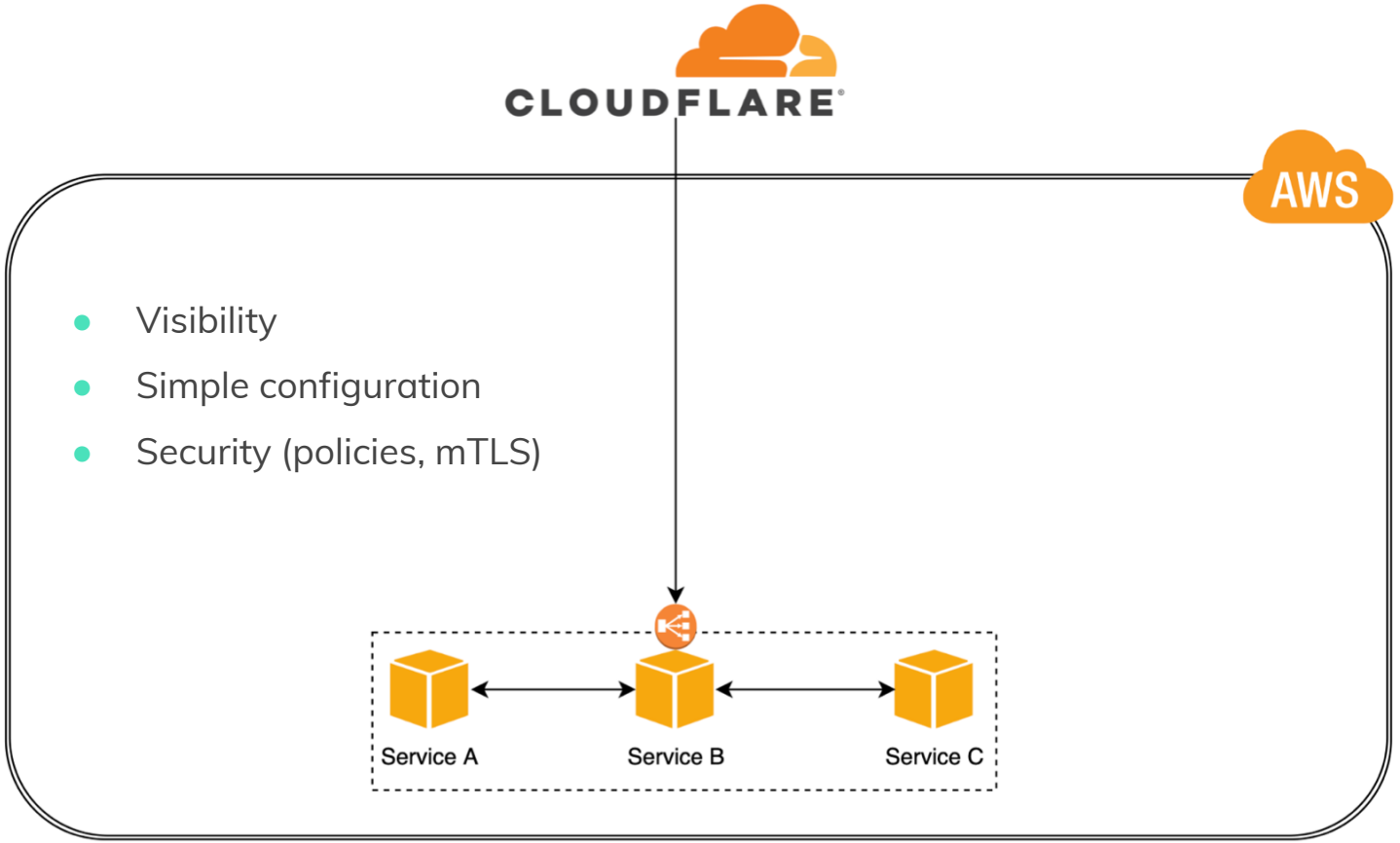


https://snyk.io/blog/top-ten-most-popular-docker-images-each-contain-at-least-30-vulnerabilities/

Service Mesh



Service Mesh



What's next

- Finish Istio rollout
- More security
- Knative builds
- Have fun!

THANK YOU

https://www.thredup.com/devopsstage-2019