

Please

**Ask questions
through the app**



Rate Session

Thank you!





Erwin de Gier - Trifork Amsterdam

Optimizing Kubernetes deployments with Helm



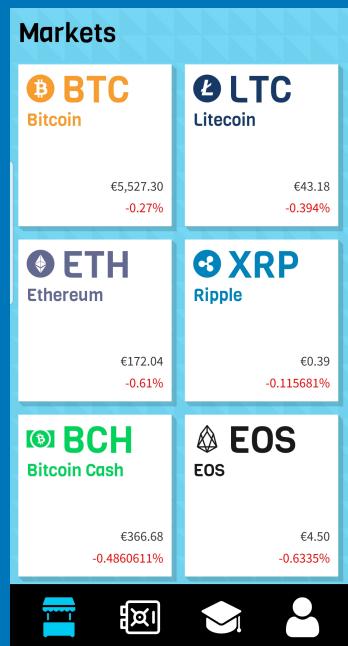
@erwindeg



github.com/erwindeg



edegier.nl



<https://www.weareblox.com>

Project deliverables

EAR

WAR

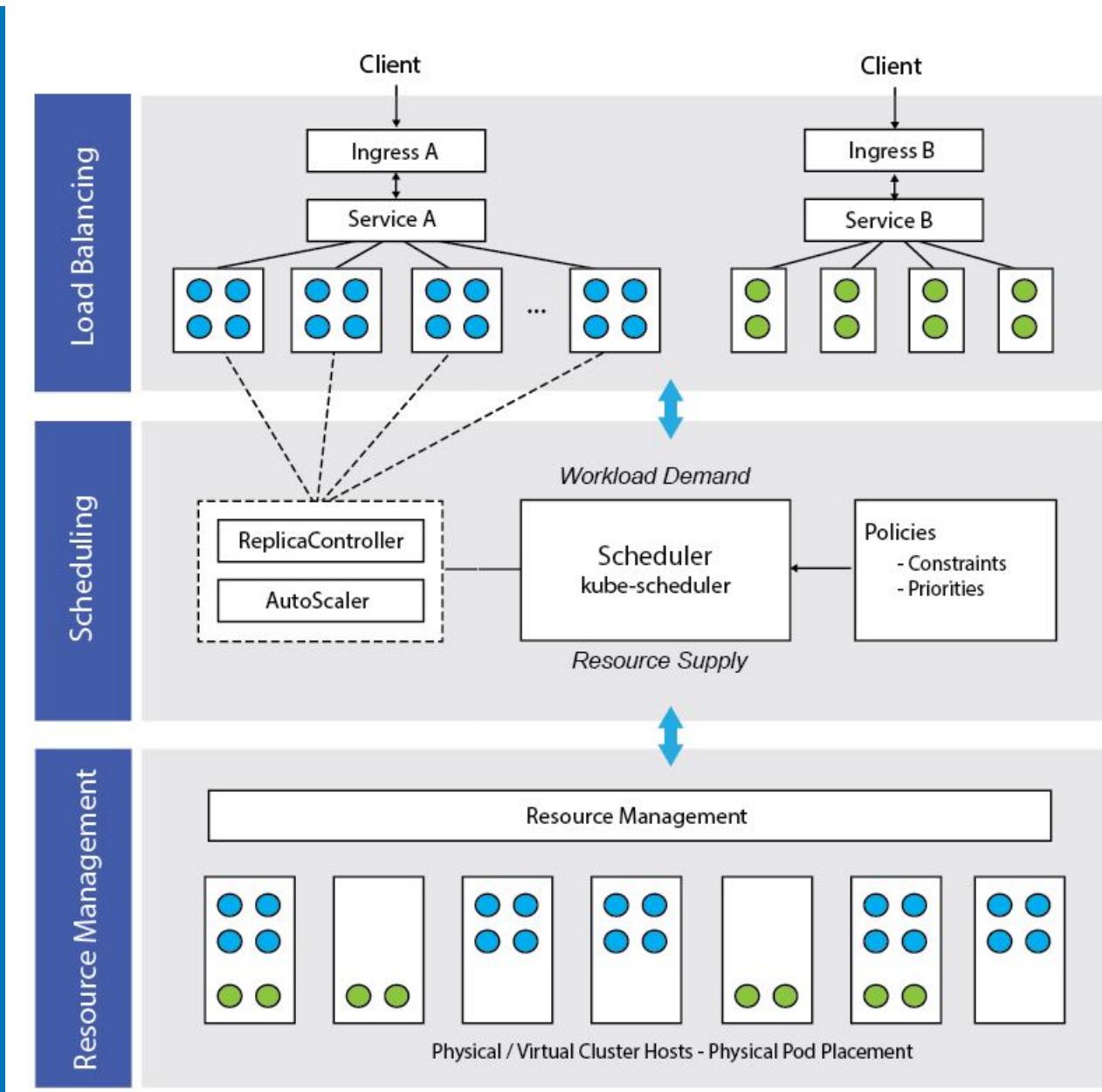
JAR

Docker
Images

YML

Kubernetes

- Container orchestration platform
- Deploy container instances
- Scheduling / load balancing / uptime / etc.



tooling.com/sonar
(ingress)

sonarqube
(Service)

sonarqube-
container
(pod)

postgresql
password
(secret)

sonar.properties
(configmap)

postgresql
(service)

postgresql-
container
(pod)

persistant
volume
claim

```
1 apiVersion: extensions/v1beta1
2 kind: Deployment
3 metadata:
4   annotations:
5     app: sonarqube
6     pod-template-hash: "480031380"
7     release: vested-frog
8     ownerReferences:
9       - apiVersion: extensions/v1beta1
10      blockOwnerDeletion: true
11      controller: true
12      kind: ReplicaSet
13      name: vested-frog-sonarqube-8d44757d4-h4z5j
14      uid: 4dd60448-caf5-11e8-9a20-025000000001
15   labels:
16     app: sonarqube
17   selfLink: /api/v1/namespaces/sonarqube/pods/vested-frog-sonarqube-8d44757d4-h4z5j
18 spec:
19   containers:
20     - command:
21       - sh
22       - -ce
23     - 'mkdir /scripts && cp /tmp-script/startup.sh /scripts/startup.sh && chmod 0755
24     /scripts/startup.sh && /scripts/startup.sh '
25   env:
26     - name: SONARQUBE_JDBC_USERNAME
27       value: sonarUser
28     - name: SONARQUBE_JDBC_PASSWORD
29       valueFrom:
30         secretKeyRef:
31           key: postgres-password
32           name: vested-frog-postgresql
33     - name: SONARQUBE_JDBC_URL
34
35 revisionHistoryLimit: 10
36 selector:
37   matchLabels:
38     app: sonarqube
39     release: vested-frog
40 strategy:
41   rollingUpdate:
42     maxSurge: 1
43     maxUnavailable: 1
44   type: RollingUpdate
45 template:
46   metadata:
47     creationTimestamp: null
48     labels:
49       app: sonarqube
```

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   creationTimestamp: null
5   labels:
6     app: sonarqube
7     chart: sonarqube-0.5.2
8     heritage: Tiller
9     release: vested-frog
10    name: vested-frog-sonarqube
11    selfLink: /api/v1/namespaces/sonarqube/services/vested-frog-sonarqube
12 spec:
13   externalTrafficPolicy: Cluster
14   ports:
15     - name: sonarqube
16       nodePort: 32425
17       port: 9000
18       protocol: TCP
19       targetPort: 9000
20   selector:
21     app: sonarqube
22     release: vested-frog
23   sessionAffinity: None
24   type: LoadBalancer
25   status:
26     loadBalancer: {}
```



DOCS



The package manager for Kubernetes

Helm is the best way to find, share, and use software built for Kubernetes.



source: <http://helm.sh>

SonarQube

localhost:9000/about

Projects Issues Quality Profiles Quality Gates

Search for projects, sub-projects and files... Log in

Continuous Code Quality

[Log in](#)

[Read documentation](#)

0

Projects Analyzed

0 Bugs

0 Vulnerabilities

0 Code Smells

Multi-Language

20+ programming languages are supported by SonarQube thanks to our in-house code analyzers, including:

[Java](#)

[C/C++](#)

[C#](#)

[COBOL](#)

[ABAP](#)

[HTML](#)

[RPG](#)

[JavaScript](#)

[TypeScript](#)

[Objective C](#)

[XML](#)

[VB.NET](#)

[PL/SQL](#)

[T-SQL](#)

[Flex](#)

[Python](#)

[Groovy](#)

[PHP](#)

[Swift](#)

[Visual Basic](#)

[PL/I](#)

Quality Model

Bugs track code that is demonstrably wrong or highly likely to yield unexpected behavior.

Vulnerabilities are raised on code that is potentially vulnerable to exploitation by hackers.

Code Smells will confuse maintainers or give them pause. They are measured primarily in terms of the time they will take to fix.

Write Clean Code

By fixing new issues as they appear in code, you create and maintain a clean code base. Even on legacy projects, focusing on keeping new code clean will eventually yield a code base you can be proud of.

[Read More](#)

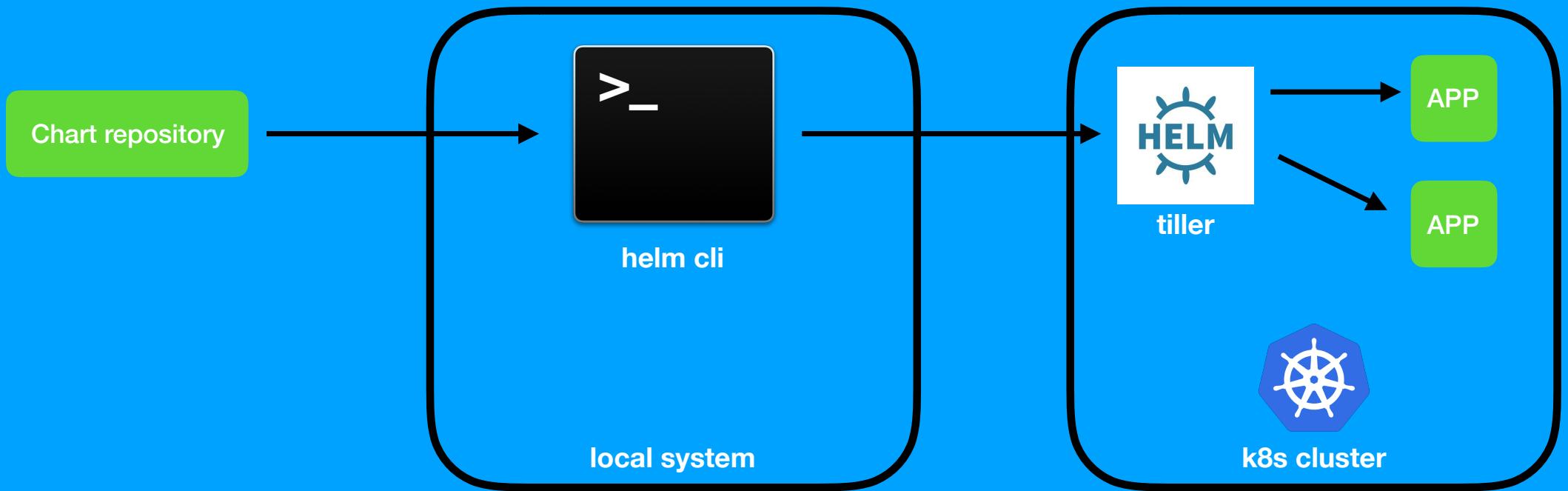
Fix The Leak

The water leak paradigm and the default Quality Gate are based on the leak period - the recent period against which you're tracking issues. For some previous_version makes the most sense, for others the last 30 days is a good option.

[Read More](#)

Prerequisites

- Connection to a Kubernetes Cluster
 - Docker app with Kubernetes
 - Mini-Kube
 - Hosted Kubernetes (GKE, AWS)
- Installation of Helm (homebrew / download)





```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: frontend
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: guestbook
        tier: frontend
    spec:
      containers:
        - name: php-redis
          image: gcr.io/google_samples/gb-frontend:1.0
          ports:
            - containerPort: 80
```

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: frontend
spec:
  replicas: {{ .Values.frontend.replicas }}
  template:
    metadata:
      labels:
        app: guestbook
        tier: frontend
    spec:
      containers:
        - name: php-redis
          image: {{ .Values.frontend.image }}
          resources:
            requests:
              cpu: 100m
              memory: 100Mi
          env:
            - name: GET_HOSTS_FROM
              value: dns
          ports:
            - containerPort: {{ .Values.frontend.containerPort }}
```

Chart package

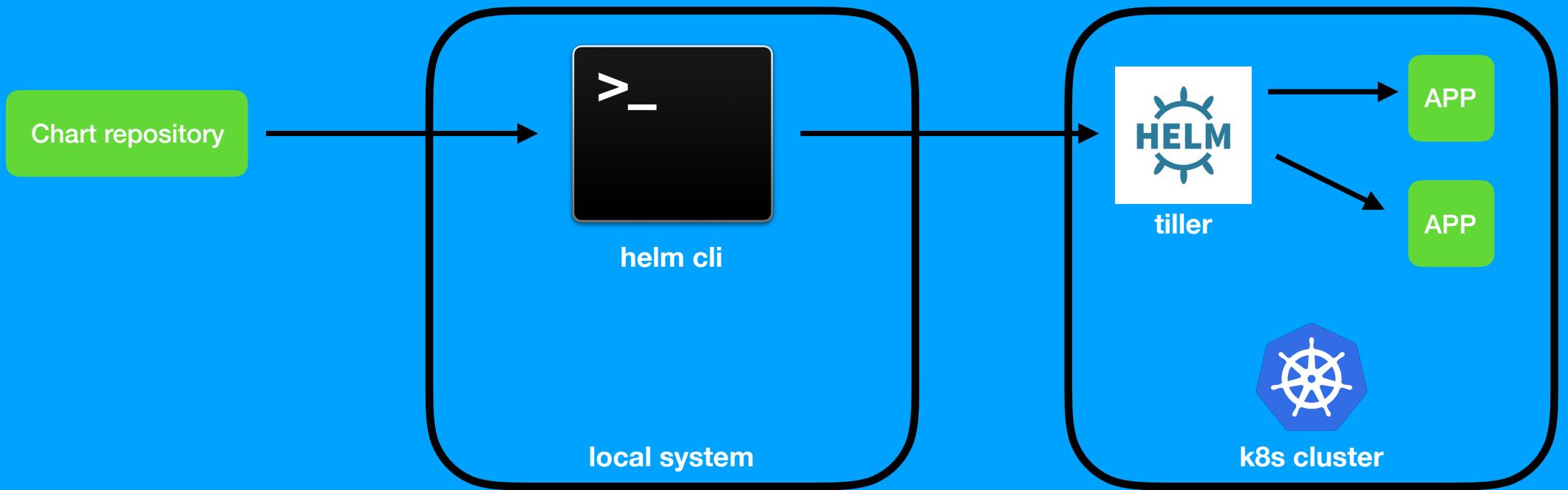
```
templates/guestbook-services.yaml  
templates/guestbook-deployments.yml  
Chart.yml  
values.yml
```

source: https://github.com/erwindeg/k8s-helm/tree/master/helm_chart_guestbook

Guestbook

Messages

Submit



Helm chart repository: <https://github.com/kubernetes/charts>

Public chart repository

- App Store for Kubernetes deployments
- Stable charts for jenkins, sonar, postgres, rabbit mq etc.
- Backed by github.com/helm/charts

Chart config

- Chart values.yml
- Supplying a yml: “-f values.prod.yml”
- Command line variables: “--set frontend.image=”
- Command line file: “—set-file”

```
helm install stable/sonarqube --namespace sonarqube --set  
service.externalPort=9001
```

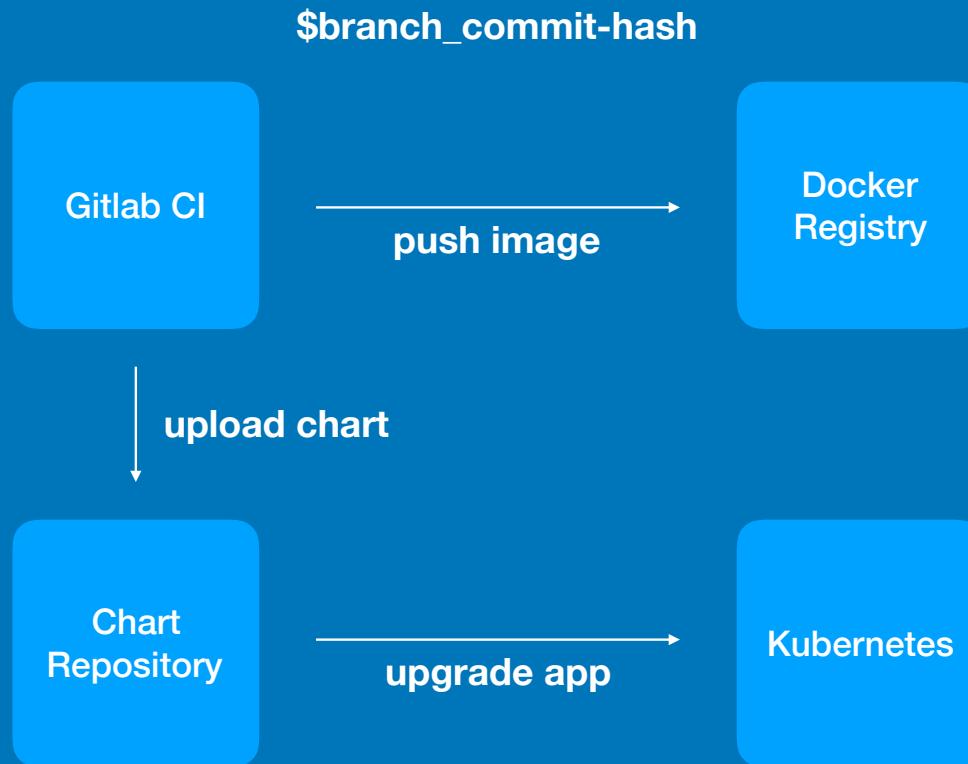
Dependencies

sonarqube/requirements.yaml

```
1 dependencies:  
2   - name: postgresql  
3     version: 0.8.3  
4     repository: https://kubernetes-charts.storage.googleapis.com/  
5     condition: postgresql.enabled  
6   - name: mysql  
7     version: 0.6.0  
8     repository: https://kubernetes-charts.storage.googleapis.com/  
9     condition: mysql.enabled  
10
```

postgresql/Chart.yaml

```
1 name: postgresql  
2 version: 0.8.3  
3 appVersion: 9.6.2  
4
```



```
helm upgrade --install $release-name $chartname --set  
image.tag=$branch_commit-hash -f $appname-values.yml
```

More features

- Dryrun: “helm - -dry-run - - debug”
- Chart quickstart: “helm create - -starter”
- Chart versioning and signing
- Tags & conditions
- Test charts: “helm test <RELEASE NAME>”

Security considerations

- Tiller runs in privileged mode
- RBAC
 - `helm init --service-account <NAME>`
 - restricting namespaces

Alternatives

- Kubectl
- jsonnet templates: Ksonnet, Kapitan
- Declarative: Terraform Kubernetes provider, Kustomize
- Service-mesh approach: Istio
- Build-push-deploy: draft, gitkube, metaparticle, skaffold

Conclusion

- Manage complex applications
- Versioning and updates
- Rollback
- Sharing

Please

Remember to
rate this session

Thank you!

