LF NETWORKING

LFN Developer & Testing Forum

LF NETWORKING

LFN Developer & Testing Forum

ONAP: DT and the use of OOM Helm Charts, Argo CD & Istio (Service Mesh)

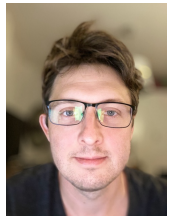
Argofy everything 🦸🦄

@florianbachmann

Recording

- This session will be recorded! Yeah 

The presenter



- Florian Bachmann from Frankfurt
- Technical PO of the "System Team" of TNAP.
- My biggest challenge: "*I try to eat all the cake & chocolate of the world (only the good one)*"
- twitter.com/florianbachmann

- Flori got "Istio / Service Mesh" help from Andreas Geißler
- Architect in TNAP and Master of Helm Charts and Istio
- His biggest challenge: "*To have all Helm Charts Kohn/Istio compliant and somewhere deployed*"
- You can write him an email: andreas-geissler@telekom.de

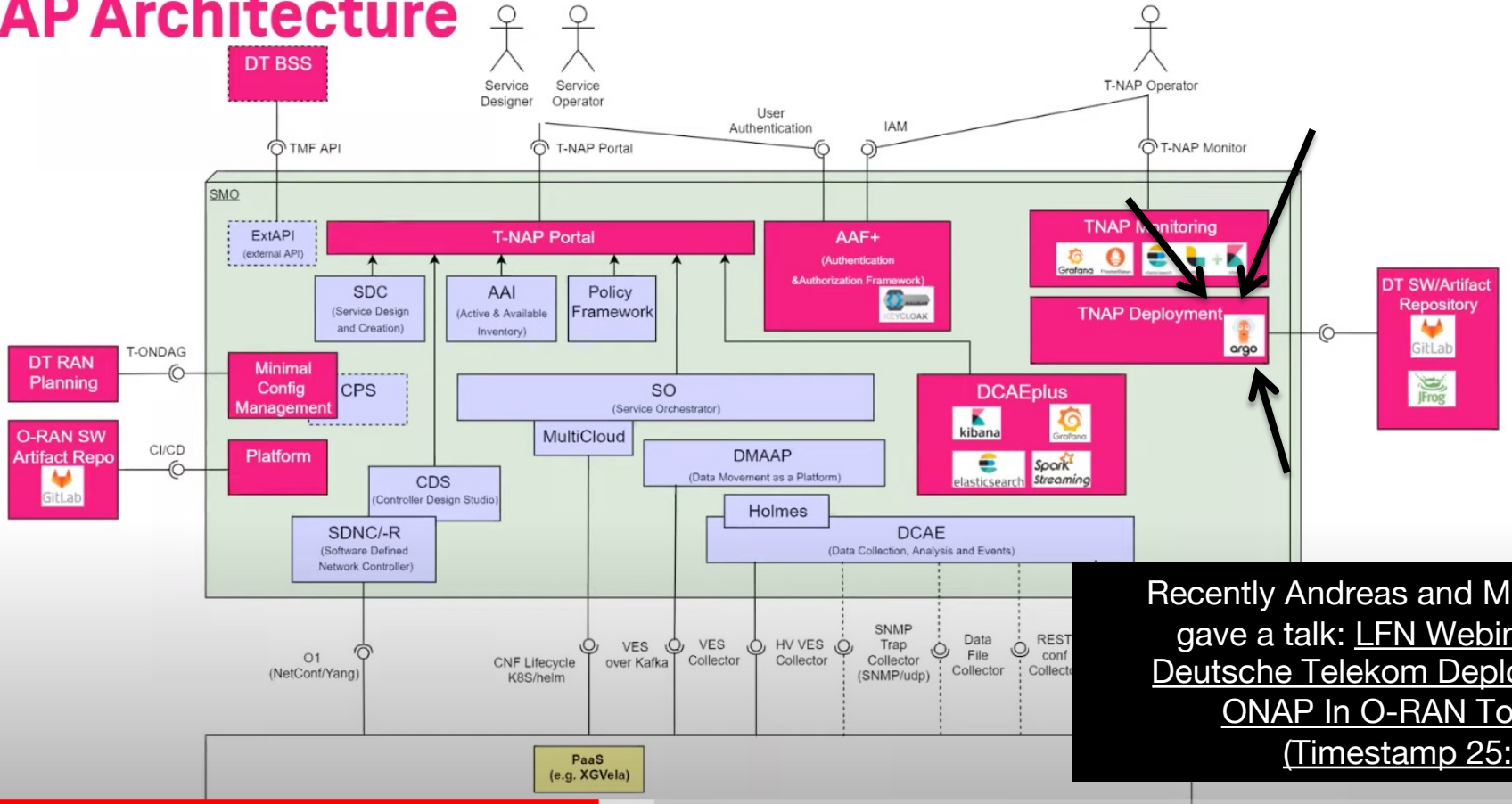
Agenda / Expectation

- What is Argo CD
- Demos Part 1 – How DT deploys ONAP with Argo CD
- Good to know:
  - 12 Factor Apps
  - DevOps
  - GitOps
  - Argo CD
  - Why Argo CD?!
  - Helm Charts & OOM
  - Service Mesh / Istio
  - Current Status: Service Mesh Kohn / TNAP
- Demos Part 2 – Istio
- Outlook: To many cluster, Cluster API for the rescue

ONAP at DT -> called TNAP

LF NETWORKING
LFN Developer & Testing Forum

TNAP Architecture



Recently Andreas and Marc gave a talk: LFN Webinar: Deutsche Telekom Deploys ONAP In O-RAN Town (Timestamp 25:51)

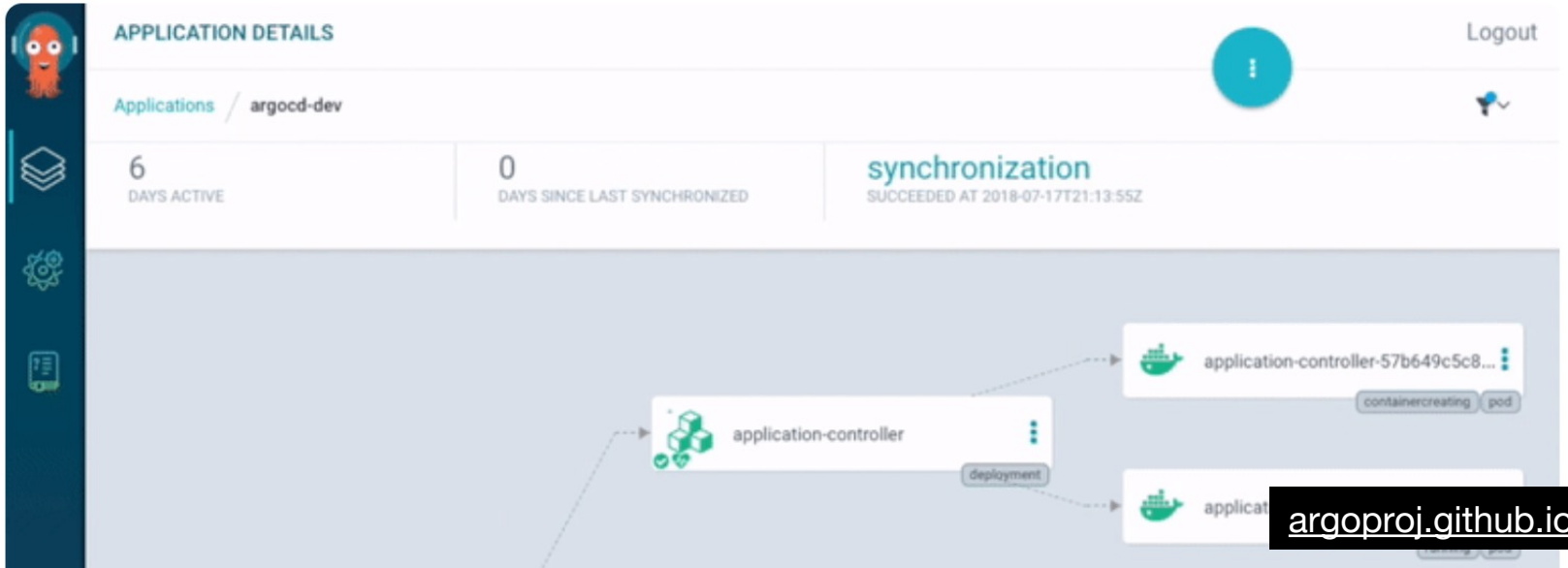
There is no CI/CD
There are CI and CD!

Argofy everything – (fluxyfy is fine as well)

The official Argo CD page

What is Argo CD?

Argo CD is a declarative, GitOps continuous delivery tool for Kubernetes.



argoproj.github.io/cd

Demos Part 1 – How DT deploys ONAP with Argo CD

1. The UI
2. Deploy a Gitlab runner into a k8s cluster
   - Just for learning and getting a feeling
   - start with version 0.33.0 update it later to 0.33.1
   - add it to all `kustomization.yaml`'s
3. Manually trying to change/destroy an existing ONAP component, e.g. `a1policymanagement`

12 Factor Apps (https://12factor.net/)

LFN Developer & Testing Forum

I. Codebase One codebase tracked in revision control, many deploys

II. Dependencies Explicitly declare and isolate dependencies

III. Config Store config in the environment

IV. Backing services Treat backing services as attached resources

V. Build, release, run Strictly separate build and run stages

VI. Processes Execute the app as one or more stateless processes

VII. Port binding Export services via port binding

VIII. Concurrency Scale out via the process model

IX. Disposability Maximize robustness with fast startup and graceful shutdown

X. Dev/prod parity Keep development, staging, and production as similar as possible

XI. Logs Treat logs as event streams

XII. Admin processes Run admin/management tasks as one-off processes

In ONAP/TNAP OOM terms
`localCluster: true`

There is no CI/CD
There are CI and CD!

Let’s define DevOps

DevOps defined

DevOps is a methodology in which teams own the entire process from application development to production operations, hence *DevOps*.

It goes beyond implementing a set of technologies and requires a complete shift in culture and processes.

DevOps calls for groups of engineers that work on small components (versus an entire feature), decreasing handoffs – a common source of errors.

https://glossary.cncf.io/devops/

CLOUD NATIVE GLOSSARY

There is no CI/CD
There are CI and CD!

Let’s define DevOps ✅

Let’s define GitOps

What is GitOps and Why?

- Argo CD helps us, doing GitOps 🦹🚀🦄
  - what is GitOps, and why it is important, was perfectly answered by Cornelia Davis (@cdavisafc) in that talk:
    - KubeCon: GitOps Is Likely More Than You Think It Is - Cornelia Davis, Weaveworks
    - A more recent recorded version of that talk exists as well:
      - GitOps Is Likely More Than You Think It Is (only ~~34~~ 55 views, I was three of them). Both talks are 99% similar, but the first one is more on point.
  - The slides of the two talks can be found here
- No more `helm update`

GitOps Principles

The entire system is described **declaratively**

The canonical desired system state is **versioned** in git

Approved changes can be **automatically applied** to the system

Software agents ensure correctness and perform actions on divergence in a closed loop

GitOps - Cloud Native Agility and Reliability

LFN Developer & Testing Forum

Solution

GitOps is a set of modern best practices for deploying and managing cloud native infrastructure and applications.

Based on our experience operating a full cloud native stack

GitOps manages the whole stack:

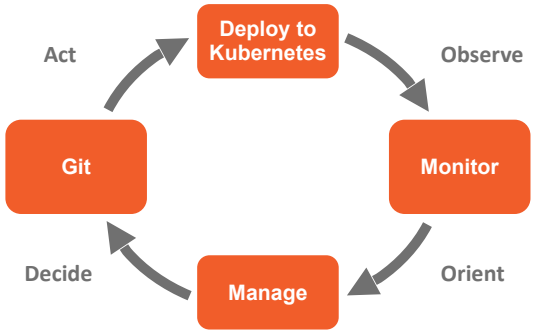
- Cluster and application versioned configuration
- Security and policy enforcement
- Monitoring and observability
- Continuous Deployment of workloads

Benefits

- **Complete platform**: Single platform for infrastructure, core components and applications.
- **Productivity**: Dramatically increase deployments and faster feedback and control loop,
- **Reliability**: Enables cluster and application operator model with standardised tooling.
- **Compliance and Security**: Enforces standard security policy and an audit trail
- **Multi-cloud and on-premise**: Deploy a complete cluster from git with all applications.

Vision

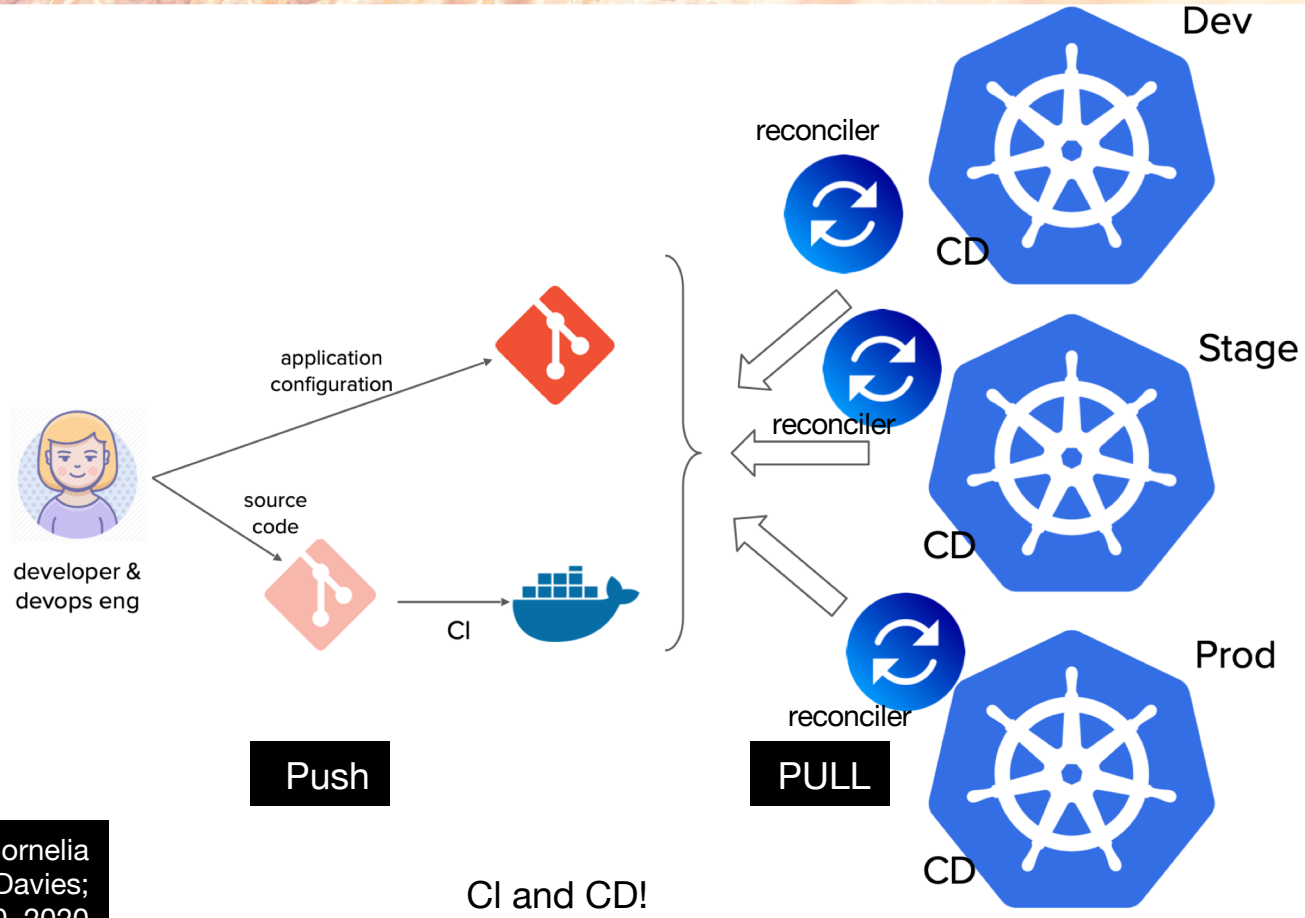
- All application deployments, application operations and cluster management operations under one platform with a common workflow.



Slide by Cornelia Davies; Kubecon 20, 2020

weaveworks

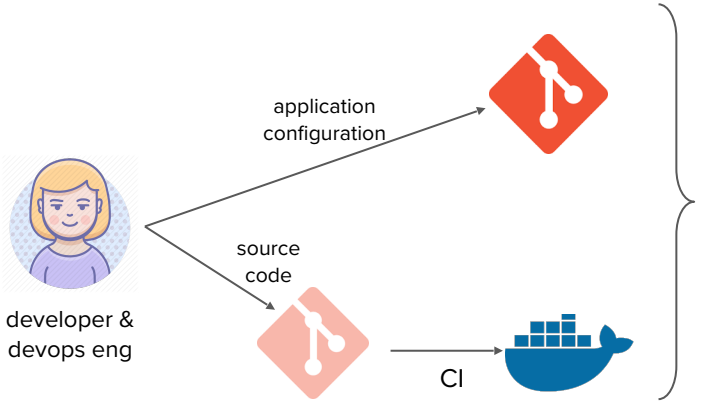
GitOps - Cloud Native Agility and Reliability

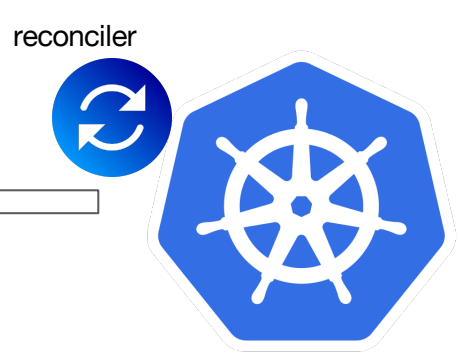


Slide by Cornelia Davies; Kubecon 20, 2020

CI and CD!

Drift detection and remediation





We don't want this ever & no manual `helm update` as well

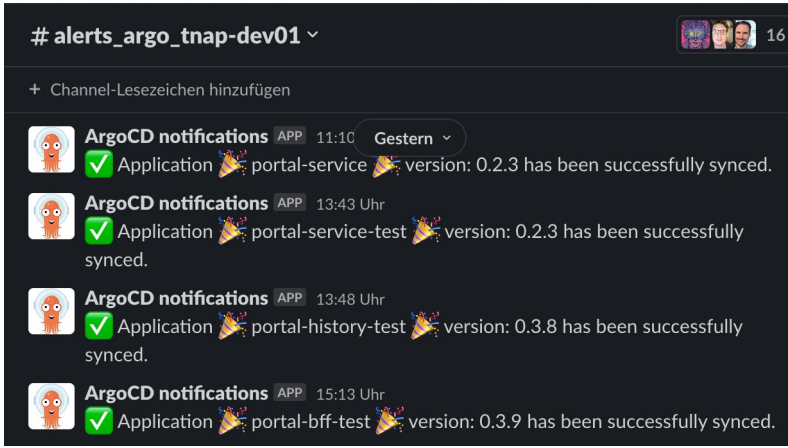
`kubectl apply`

Why Argo?!

- Argo CD - a tool for CD, but what says the CNCF Landscape about it.
  - https://landscape.cncf.io/
- CNCF Landscape - Continuous Integration & Delivery
  - https://landscape.cncf.io/card-mode?category=continuous-integration-delivery&grouping=category
- There are at least 47 tools regarding CI & CD. (checked 11.01.2022)
  - https://landscape.cncf.io/card-mode?category=continuous-integration-delivery&grouping=category
- Why Argo and not Flux
  - Flux CD joins forces with Argo CD project
- And consider: CNCF - The Trail Map
  - https://raw.githubusercontent.com/cncf/trailmap/master/CNCF_TrailMap_latest.png

Outlook: GitOps - were we like to get better

- We got Slack notifications!



- were we like to get better:
  - Pull Request Automatization
    [GitOps Is Likely More Than You Think It Is - Cornelia Davis talk on Youtube] / [Slides] (slide 21 to 24)
    - hint https://github.com/argoproj-labs/argocd-image-updater

OOM from ONAP to TNAP

- OOM from ONAP
  - https://gerrit.onap.org/r/gitweb?p=oom.git;a=tree;h=refs/heads/istanbul;hb=refs/heads/istanbul
- ~~In our case, we split the OOM repo into several repos. One helm chart repo for each ONAP component.~~
- In our case, we have similar to ONAP everything in one git repo, because that is handled better by Argo

A git structure, similar to ONAP

LFN Developer & Testing Forum

onap-oom

Project ID: 71650

Unstar 1 | Fork 0

132 Commits | **14** Branches | **25** Tags | **3.2 MB** Files | **48.8 MB** Storage | **4** Releases

TNAP's oom project. This is the place for all ONAP helm charts needed by TNAP

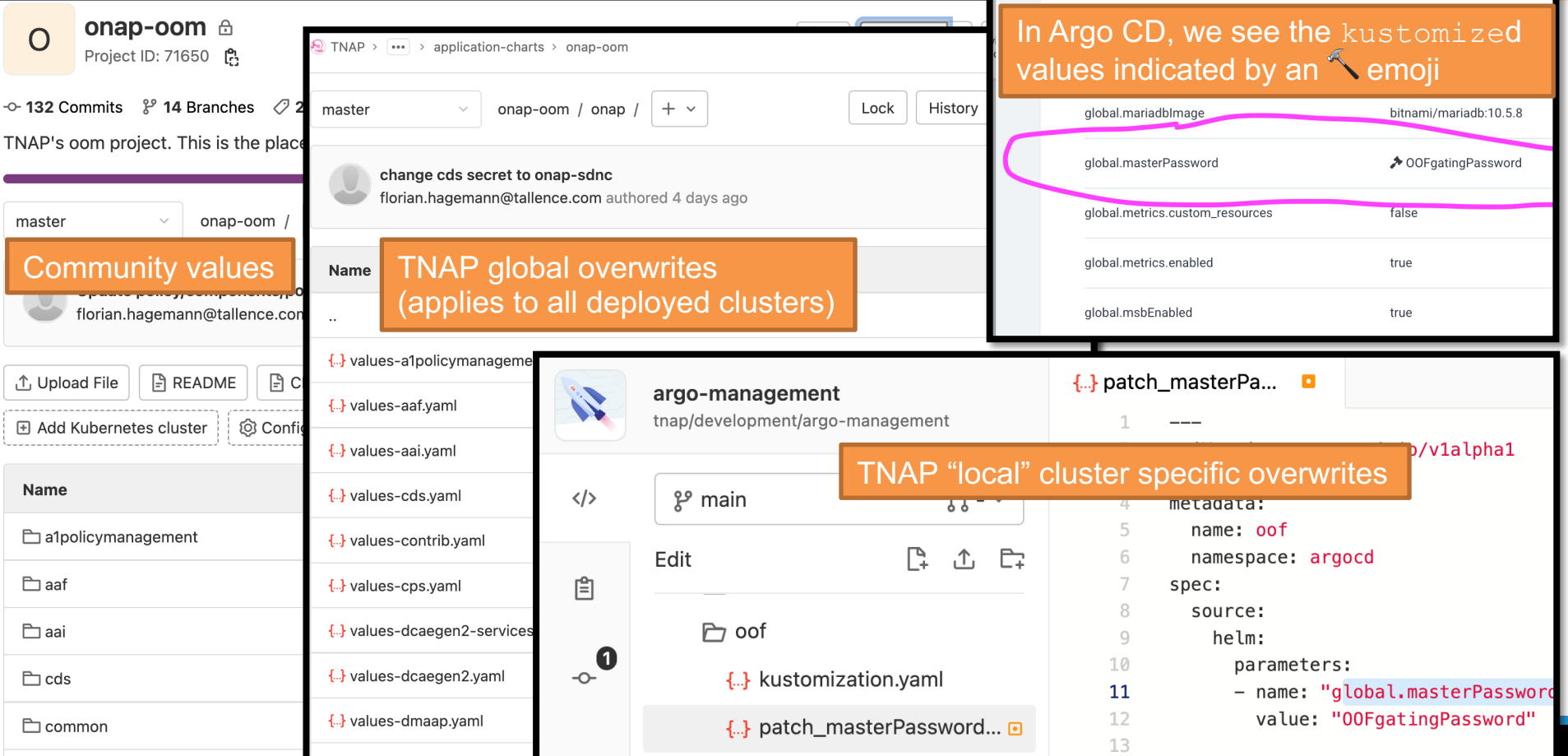
master | onap-oom / | History | Find file | Web IDE | Clone

Update policy/components/policy-api/templates/configmap.yaml,...
florian.hagemann@tallence.com authored 3 days ago — 6cd06fda

Upload File | README | CI/CD configuration | Add LICENSE | Add CHANGELOG | Add CONTRIBUTING | Add Kubernetes cluster | Configure Integrations

Name	Last commit	Last update
a1policymanagement	change kubernetes version to 1.22.5	1 month ago
aaf	change kubernetes version to 1.22.5	1 month ago
aai	Update AAI (https://gerrit.onap.org/r/c/oo...	1 week ago
cds	UPdate CDS charts with patch (https://ger...	3 weeks ago
common	Update according to patch:	1 week ago

Several places to set values (demo



Define service mesh (CNCF definition)

“In a microservices world, apps are broken down into multiple smaller services that communicate over a network.

Just like your wifi network, computer networks are intrinsically unreliable, hackable, and often slow.

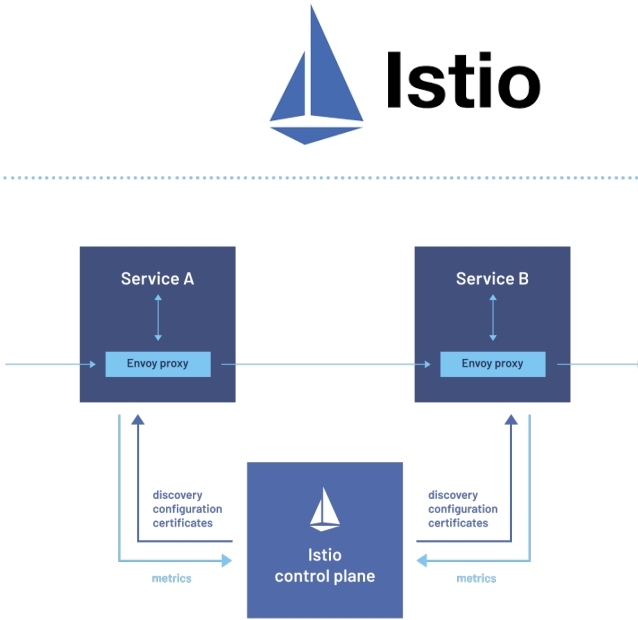
Service meshes address this new set of challenges by managing traffic (i.e., communication) between services and adding reliability, observability, and security features uniformly across all services.”
(hello `localCluster: true`)

https://glossary.cncf.io/devops/

CLOUD NATIVE GLOSSARY

Define Istio service mesh? (advertisement)

- “Simplify observability, traffic management, security, and policy with the leading service mesh.” https://istio.io/
- Istio extends Kubernetes to establish a programmable, application-aware network using the powerful Envoy service proxy.



Define Istio service mesh? (simplified)

- Networking is typically more important than majority of the people think it is
- Service mesh (and actually even Kubernetes itself) is "just" encapsulating complex networking concepts and making it work for specific purpose
- Service Mesh did not invent something that did not exist, but it simplified complex operations and made them more hands-off.
  - From that perspective, it is similar to Docker.
  - Docker did not "invent" containers, but made them simpler to create and manage.
- A service mesh is a way to control the flow of data - it is a way to control how applications interact with each other.
- Service Mesh manages communication between microservices

Why aims TNAP to use Istio/Service Mesh

- Get rid of:
  - Kong & Traefik (used at TNAP for Ingress & RBAC)
  - AAF 🦸🦸🦸🦸
    - 51 config maps with same script `retrieval_check.sh`
  - MSB
  - Message Router
- We want:
  - better RBAC
  - all the other obvious Service Mesh features, like: Observability, Discoverability, Encryption, Traffic MGMT, Canary Releases, A/B/X testing
  - Awesome Service Discovery!

What we did, to enable Istio in TNAP


```
$ kubectl get ns onap --show-labels
NAME      STATUS   AGE     LABELS
default   Active   37m     <none>

$ kubectl label ns onap istio-injection=enabled
```

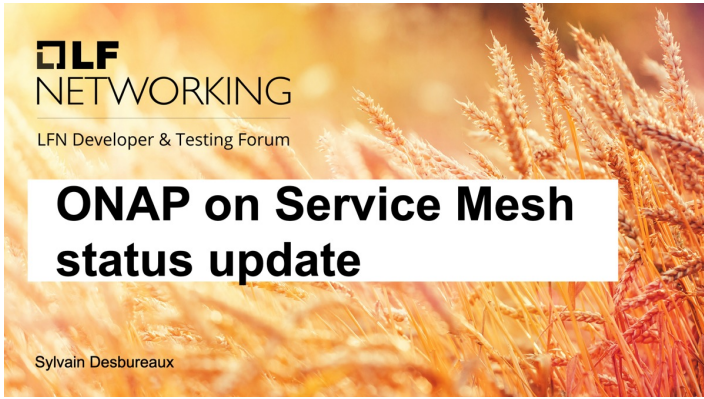
```
# In the OOM YAMLs:
serviceMesh:
  enabled: true
  tls: true
```

Kohn Status of ONAP Service Mesh (1 / 2)

LF NETWORKING
LFN Developer & Testing Forum

Status of SM PoC in Jakarta

- Last status shown in DDF
  2022-01-13 - ONAP: ONAP on Service Mesh status update
- Four steps defined by Sylvain
  - Step 1 - Certificates
  - Step 2 - Authorization
  - Step 3 - simple RBAC
  - Step 4 - full RBAC



ONAP on Service Mesh (Wiki)

Kohn Status of ONAP Service Mesh (2 / 2)

LFN Developer & Testing Forum

All the Service Mesh CRs: https://gerrit.onap.org/r/q/topic:%2522service_mesh%2522

Step 1 - Certificates:

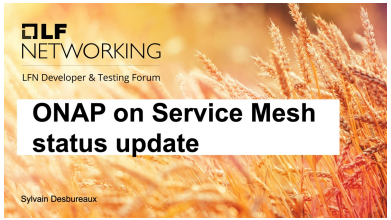
- Deployment of ONAP in “Istio” enabled system
- Requirements:
  a) The component disable AAF integration (if any) (+ disable MSB integration)
  b) The component must listen on HTTP/gRPC (no HTTPs)
  c) The component must talk to other components using HTTP (no HTTPs)
  d) Add Istio Gateway configuration for external component access

Step 2 - Authorization:

- OOM:
  - Create `AuthorizationPolicies` that will authorize some components to talk to others
  - Specific service account per subcomponent must be created (Done)
- Component:
  - Disable MSB integration
  - Internal components disable „basicAuth“

Step 3 - simple RBAC & **Step 4 - full RBAC**:

- `AuthorizationPolicy`



ONAP on Service Mesh (Wiki)

Test environments (in DT Lab)

LFN NETWORKING
LFN Developer & Testing Forum

	ONAP OOM deployment	TNAP Deployment
Installation Method	Helm installation on OOM charts using Integration Chains (chained_ci)	Deployment using kubespray We are using Istio over Helm installation
DB setup	Shared Cassandra, MariaDB-Galera, Postgres	Separate DBs per Component (`localCluster: true`)
Kubespray	2.17	2.18.1
Helm	v3.6.3	v3.7.1
Kubernetes	v1.21.5	v1.22.5
Istio	1.10.2	1.13.1

Step 1: Status of components (1 / 3)

LFN NETWORKING
LFN Developer & Testing Forum

Component	a) remove AAF/MSB	b) HTTP listen	c) HTTP talk	Remarks	CR
Strimzi	✅	✅	✅	Kafka Brokers run with Sidecar	128335 (M)
Cassandra	✅	✅	✅		
MariaDB	✅	✅	✅	Port 4568 Sidecar disabled Port 3306 Peer Authentication disabled	128371 (M) 129175 (M)
Postgres	✅	✅	✅	Patch for ETCD+Postgres	129188 (A)
DMAAP	❌	✅	✅	AAF is still running, 1 open point with DMAAP (the contributor of DMAAP needs to provide the patch)	128715 (A) 129323 (WIP)
AAI	✅	✅	✅		129267 (A)
SDC	✅	✅	✅		122426 (M)

Step 1: Status of components (2 / 3)

LFN NETWORKING
LFN Developer & Testing Forum

Component	a) remove AAF/MSB	b) HTTP listen	c) HTTP talk	Remarks	CR
SO	✅	✅	✅		128994 (A)
MultiCloud	✅	✅	❌	multicloud-k8s/framework-artifactbroker: • SDC Client is not updated (1.3.0 – no http support) • „disable MSB“ has to be checked	129266 (A) MULTICLOUD-1476
CDS	✅	✅	✅		128992 (A)
Policy	❌	❌	❌	• policy-distribution does not handle parameter „isUseHttpsWithSDC“ • policy-clamp-be required „encrypted“ SDC passwd	128543 (WIP) POLICY-4226
CPS	✅	✅	✅		124287 (M)
SDNC	✅	✅	❌	• sdnc-ueb-listener does not handle parameter „isUseHttpsWithSDC“	129471 (A) (Elastic) 129439 (WIP)

Step 1: Status of components (3 / 3)

LFN Developer & Testing Forum

Component	a) remove AAF/MSB	b) HTTP listen	c) HTTP talk	Remarks	CR
DCAEGEN2	🤷	🤷	🤷	- Not started 🤷	🤷
DCAEGEN2-Services	🤷	🤷	🤷	- Not started 🤷	🤷
DCAEMOD	🤷	🤷	🤷	- Not started 🤷	🤷
VFC	🤷	🤷	🤷	- Not started 🤷	🤷
UUI	🤷	🤷	🤷	- Not started 🤷	🤷
NBI	🤷	🤷	🤷	- Not started 🤷	🤷
A1Policy Manager	🤷	🤷	🤷	- Not started 🤷	🤷

TNAP Current State – still at step 1

- Istiofy all tools/ONAP/TNAP components to work with current Istio
- Next TNAP steps:
  1. enable Istio ingress
  2. and have it in parallel with TNAP Traefik & Kong
  3. then migrate all the routes
  4. and then get rid of Traefik & Kong
  5. … see what happens 😀

Next steps after the next steps:
We want to get rid of the envoy proxy & go for eBPF/Cillium (just using plain kernel modules), e.g. https://merbridge.io/

Demos Part 2 – Istio

1. Istio & Kiali

To many cluster

- How to manage & operate(!) all the clusters (and that shows only our LAB environments)
- More cluster = More problems

Project Name	VCPUs	Disk	RAM	VCPU Hours	Disk GB Hours	Memory MB Hours
central-vault	2	40GB	4GB	68,66	1373,17	140613,04
dev01-tesla-test	160	2,3TB	632GB	158,50	2302,70	639757,78
developers	178	4,4TB	388GB	6110,63	154619,41	13639464,68
onap-03-ta5	74	1,4TB	276GB	2540,37	50807,45	9702299,62
onap-daily-istanbul	74	1,4TB	276GB	2540,37	50807,45	9702299,62
onap-daily-jakarta	74	1,4TB	276GB	19,65	392,91	74519,64
onap-daily-master	74	1,4TB	276GB	502,40	10047,92	1918232,78
onap-istanbul	75	1,5TB	278GB	2574,70	51494,03	9772606,14
onap-istanbul-02	74	1,4TB	276GB	2540,37	50807,45	9702299,62
onap-jakarta-01	74	1,4TB	276GB	2540,37	50807,45	9702299,62
onap-test-istio	34	680GB	116GB	29,06	581,17	101336,93
oran-mavenir	309	2,6TB	498GB	10607,77	91659,38	17506323,23
systemteam-dev	94	1,8TB	364GB	3226,96	64539,19	12795786,46
ta5-mgnt	60	1,2TB	232GB	1039,63	20792,51	4116358,62
temp-01-servicemesh	92	1,8TB	360GB	3158,30	63166,01	12655173,42
temp-02-rke-test	52	1TB	168GB	1785,13	35702,53	5905747,60
tnap-cnfs	32	640GB	160GB	1098,54	21970,79	5624521,52
tnap-dev-01	160	2,3TB	632GB	5492,70	79644,10	22216860,00
tnap-dev-02	160	2,3TB	632GB	5492,70	79644,10	22216860,00
tnap-dev-03	92	1,8TB	360GB	3158,30	63166,01	12655173,42
tnap-dev-04	92	1,8TB	360GB	3158,30	63166,01	12655173,42
tnap-dev-monitoring	92	1,8TB	360GB	3158,30	63166,01	12655173,42
tnap-integration	124	1,8TB	488GB	4256,84	63166,01	17154790,63
tnap-monitoring	132	2TB	504GB	4531,47	68658,71	17717242,79
tnap-robot	34	680GB	395GB	1167,20	23343,96	13885537,50
tnap-sys-ks	92	1,8TB	360GB	1520,48	30409,55	6092468,92

Outlook / Next steps

- We try to make ONAP/TNAP deployment independent from the k8s cluster
- ONAP/TNAP should work with and without Istio
- We will fully embrace Cluster API, to have a proper cluster lifecycle management
- The target a two step process (as kind of atomic op):
  - To be able to create k8s with ONAP (but tailored to the ONAP requirements) with ClusterAPI (managed by Argo)
  - To have ONAP managed by Argo deployable to any K8s cluster, that fits the ONAP requirements

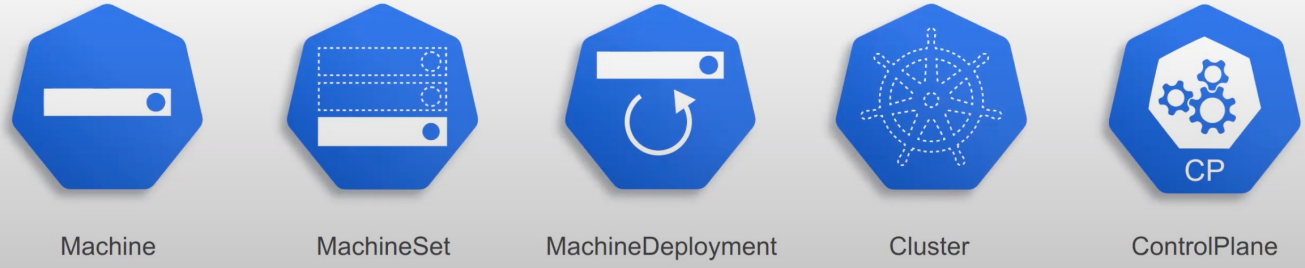
Cluster API teaser

- Provision of declarative APIs for cluster creation, configuration, and management.

Cluster API - Abstractions

Kubernetes

Cluster API

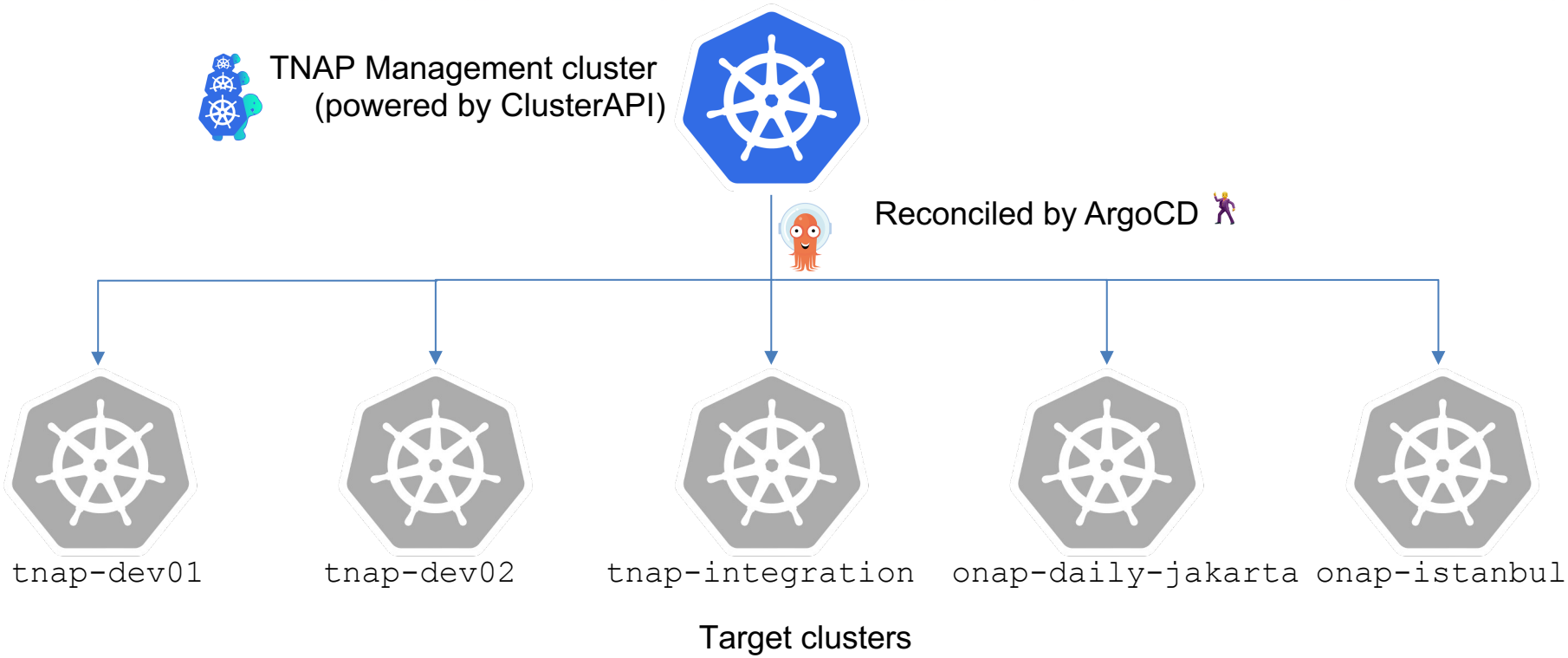


From: Webinar: Cluster API (CAPI) - A Kubernetes subproject to simplify cluster lifecycle management https://youtu.be/A2BBuKx1Yhk?t=1422

If you know `kubectl`, you know `clusterctl`


```
clusterctl init
# Installs the cluster API components in target cluster to
make it into a management cluster
clusterctl upgrade
# Upgrades cluster API and provider components installed in
the management cluster
clusterctl delete
# Deletes provider components from the management cluster
clusterctl move
# Moves Cluster API objects between management clusters
```

Cluster API – what we like to achieve



Thanks!

Questions? 🤔🧐