

goto;

GOTO **AARHUS 2021**

#GOTOaar

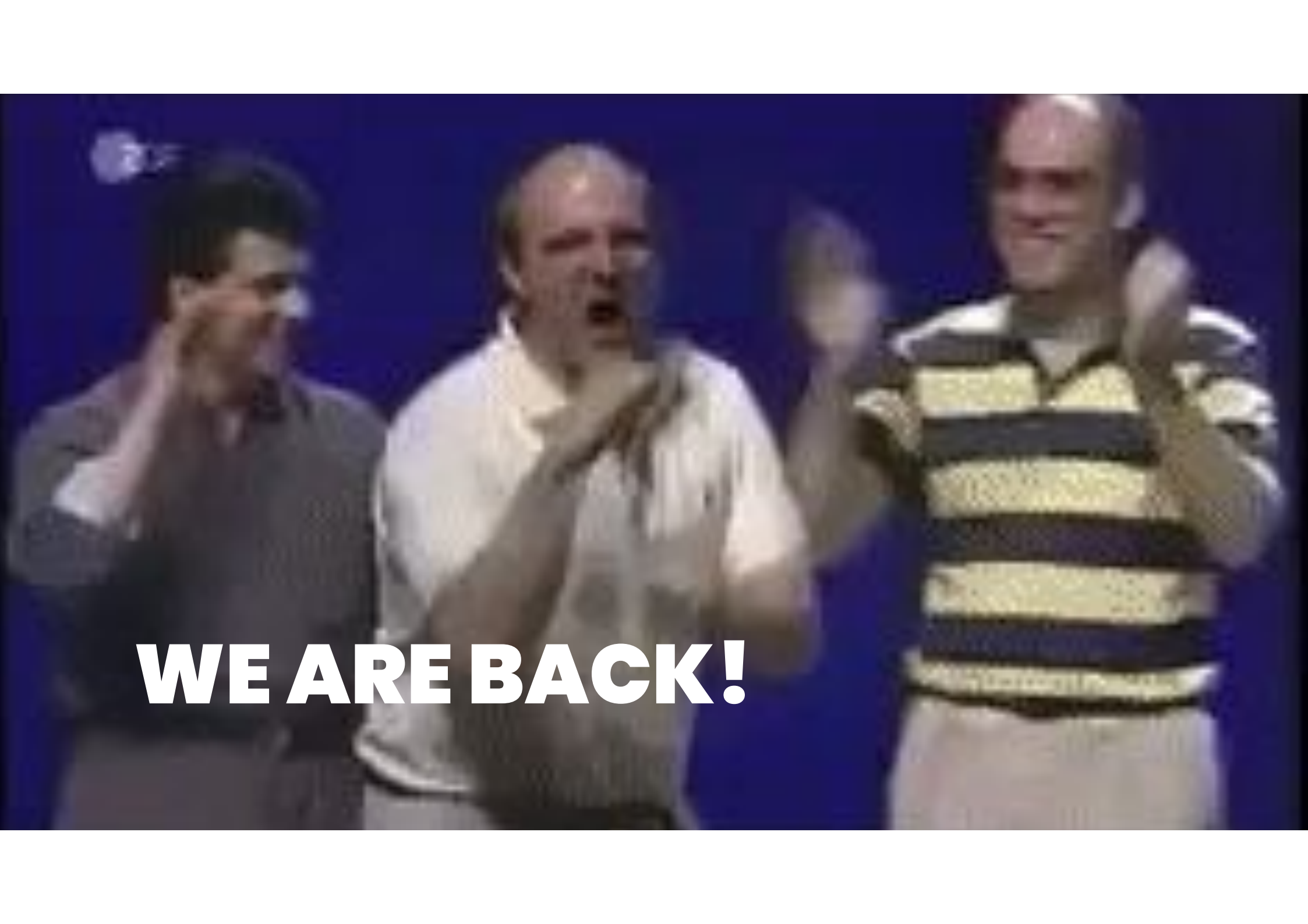
GitOps

What, Why, and Where to go from here?

Kasper Nissen (@phennex)
GOTO Aarhus, June 10-11, 2021



LUNAR



WE ARE BACK!

KASPER NISSEN

LEAD PLATFORM ARCHITECT @lunarbanc

CNCF Ambassador

Community lead at Cloud Native Nordics

Cloud Native Aarhus Organizer

Go Aarhus Organizer

Occasional speaker at Meetups, Conferences

Blog: kubecloud.io

Twitter: @phennex





What to expect from this talk?

After this talk you should have a good understanding of what GitOps is, why you should care, and what the future might hold.

LUNAR[®] at a glance



LUNAR[®]



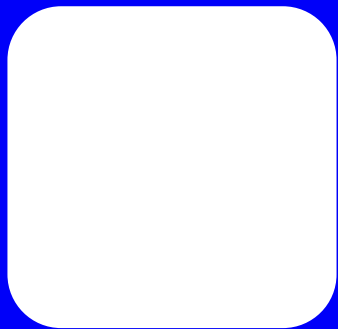
LUNAR[®]

- Founded as Lunar Way in 2015
- Smartphone only challenger bank
- Originally built on top of existing bank
- Received banking license august 2019
- Live with the “real” bank in beginning of 2020
- Best in class UX and support
- Fully cloud based bank
- Present in DK, NO and SE

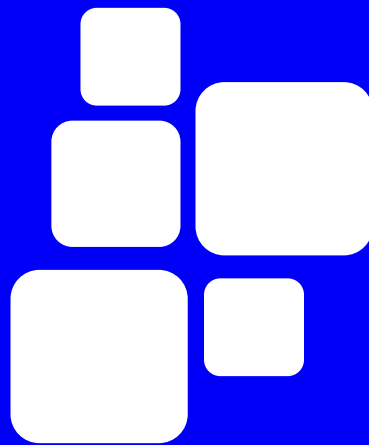
A photograph of three people sitting on wooden benches inside a sauna. On the left is a young, muscular man with dark curly hair and a beard, shirtless and wearing a grey towel. In the middle is a young woman with dark hair tied back, also wearing a grey towel. On the right is an older man with white hair, wearing a dark suit, white shirt, and a striped tie. The man and woman are looking at each other, while the older man looks off to the side with a serious expression. The background consists of vertical wooden planks.

**Banks are boring.
Banks are old.
Banks are mainframes.**

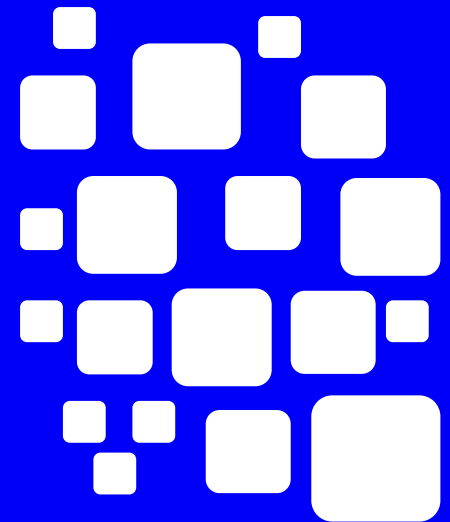
LUNAR[®] TECH JOURNEY



Monolith



Distributed
monolith



Microservices

LUNAR[®] is a proud End User Supporter of



**CLOUD NATIVE
COMPUTING FOUNDATION**

End User Supporter

adidas Adventure Box Technology airbnb amadeus ANOVA Audi AuditBoard

AuthKeys babylon box cookpad cordial cruise CURVE

dailymotion DataGalaxy DB DISCOVER doc.ai DOORDASH Entegral

EQUITYZEN 易酷软件 EVERQUOTE FORM3 GMX GoPro HITRUST

HOBSONS 汇付天下 iHerb King KSAT La Mobilière

LUNAR MAMBU Meltwater MIV monzo myfitnesspal N26

Nasdaq nmatch nielsen onecause payit PDT PARTNERS PostFinance

PriceSpider ProSiebenSat.1 PUSHHER reddit Ricardo RStudio SAP Concur

shopify SIMPLIEXUS snow Sony Entertainment SPRINGER NATURE StateFarm

The New York Times ThermoFisher Scientific THREDUP ticketmaster trivago twitter TWO SIGMA

Ultimate Software Under Armour UNITEDHEALTHGROUP UPSIDER switch verizon media Workspot

workday WP engine zalando zendesk zuora

elephant ostrich giraffe horse lizard jellyfish pig

LUNAR[®]

Let's start with

What?

reconciliation

UK /ˌrek.ənˌsil.iˈeɪ.ʃən/ US /ˌrek.ənˌsil.iˈeɪ.ʃən/

the process of making two people or groups of people friendly again after they have argued seriously or fought and kept apart from each other, or a situation in which this happens

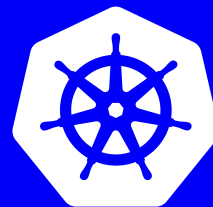
the process of making two opposite beliefs, ideas, or situations agree

Source: <https://dictionary.cambridge.org/dictionary/english/reconciliation>



AN EXAMPLE

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 5
  template:
    spec:
      containers:
        - name: nginx
          image: nginx:1.21.0
          ports:
            - containerPort: 80
```

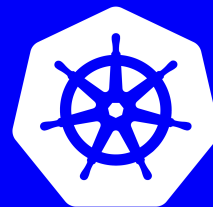


controller-manager

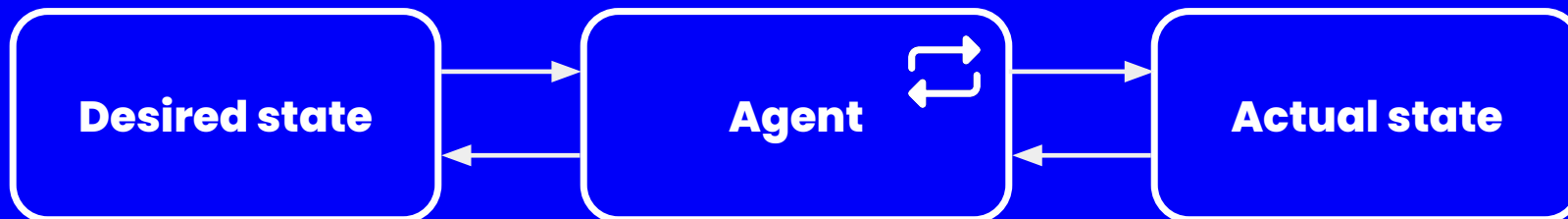
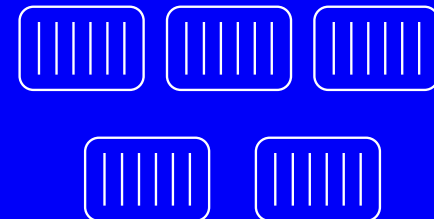


AN EXAMPLE

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 5
  template:
    spec:
      containers:
        - name: nginx
          image: nginx:1.21.0
          ports:
            - containerPort: 80
```



controller-manager



“Compare the **running state of our system with the **desired state** - **continually** - and whenever these get out of sync, force the running state to **converge** to the desired state.”**

- Alexis Richardson, CEO at Weaveworks



The GitOps Working Group is a WG under the CNCF App Delivery SIG.

The focus of the GitOps WG is to clearly define a vendor-neutral, principle-led meaning of GitOps. This will establish a foundation for interoperability between tools, conformance, and certification. Lasting programs, documents, and code are planned to live within the OpenGitOps project.

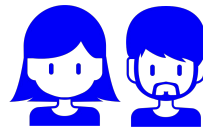
Source: <https://github.com/gitops-working-group/gitops-working-group>

PRINCIPLES

- **Declarative Configuration:** All resources managed through a GitOps process must be completely expressed declaratively.
- **Version controlled, immutable storage:** Declarative descriptions are stored in a repository that supports immutability, versioning and version history. For example, git.
- **Automated delivery:** Delivery of the declarative descriptions, from the repository to runtime environment, is fully automated.
- **Software Agents:** Reconcilers maintain system state and apply the resources described in the declarative configuration.
- **Closed loop:** Actions are performed on divergence between the version controlled declarative configuration and the actual state of the target system.

Source: <https://github.com/gitops-working-group/gitops-working-group#gitops-principles>

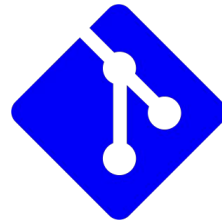
PRINCIPLES



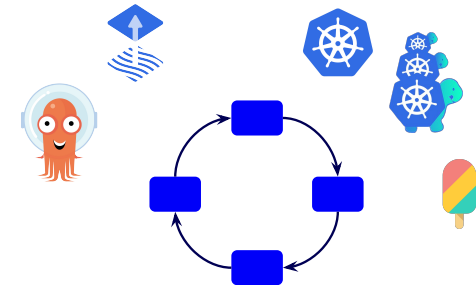
all changes are **audited** and no access to production systems is needed (ideally)



desired state
expressed
declaratively

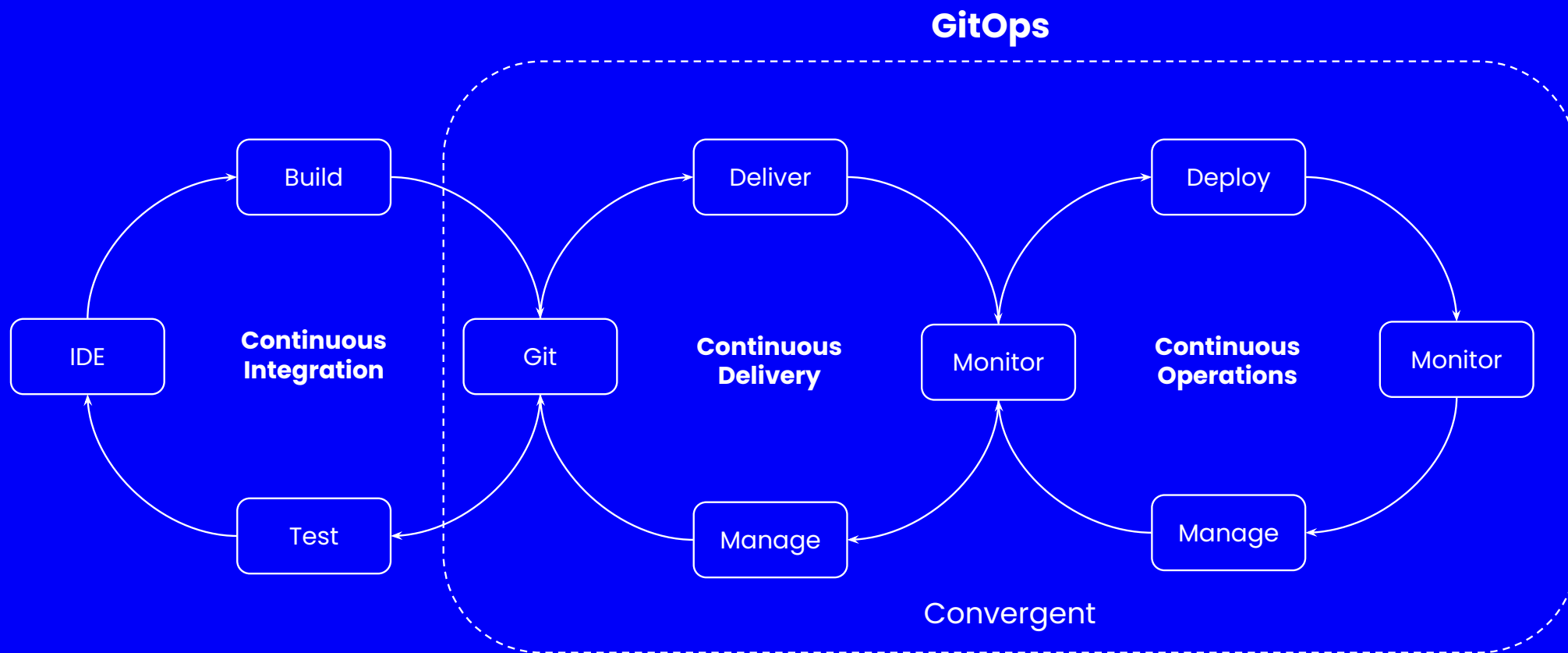


stored in a way that
supports **versioning**,
immutability of versions,
and retains a complete
version history



software agents
continuously, and
automatically, compare
a systems **actual state** to
its **desired state**

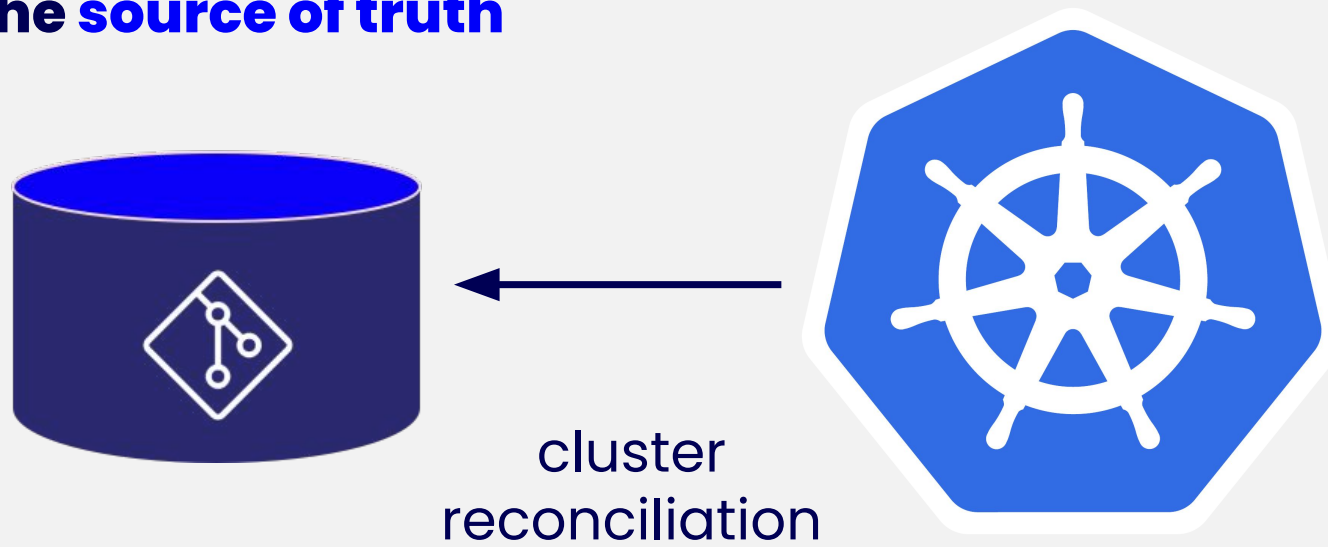
Source: GitOps Working Group Update – Cornelia Davis (KubeCon EU 2021) <https://www.youtube.com/watch?v=eXwrSe2VXHc>



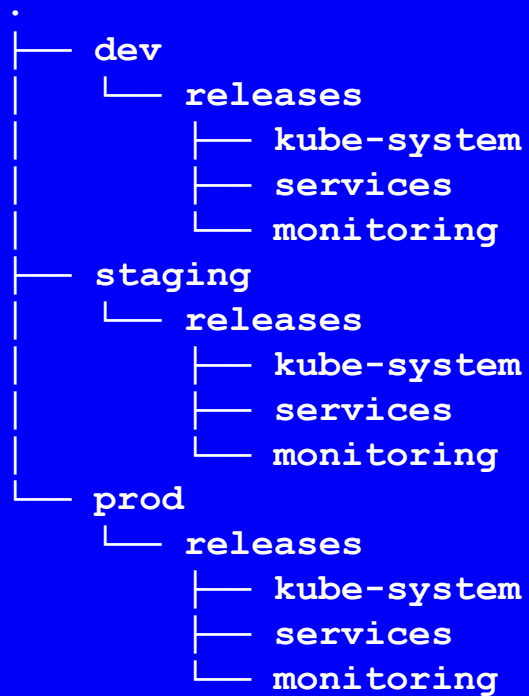
GitOps = CD + CO

Source: GitOps Working Group Update - Cornelia Davis (KubeCon EU 2021) <https://www.youtube.com/watch?v=eXwrSe2VXHc>

The source of truth



CONFIG REPO



Desired state



Software Agents



dev



staging



prod

Actual State

PROJECTS



Argo

<https://argoproj.github.io/argo-cd/>



Flux

<https://github.com/fluxcd/flux2>

Let's continue with

Why?

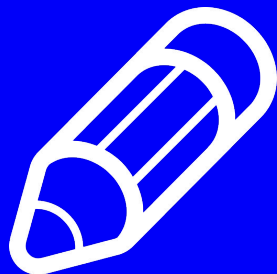
Benefits

- Increased Developer & Operational Productivity
- Enhanced Developer Experience
- Improved Stability
- Higher Reliability
- Consistency and Standardization
- Stronger Security Guarantees



Source: <https://github.com/gitops-working-group/gitops-working-group>

WHY



Audit

An audit log, also called an audit trail, is essentially a record of events and changes.
What happened when?
Who did what? And why?



Least Privileged

Concept in which a user is given the minimum levels of access – or permissions – needed to perform his/her job functions.



Disaster Recovery

Disaster Recovery involves a set of policies, tools and procedures to enable the recovery or continuation of vital technology infrastructure and systems following a natural or human-induced disaster.

AUDIT

Config repository

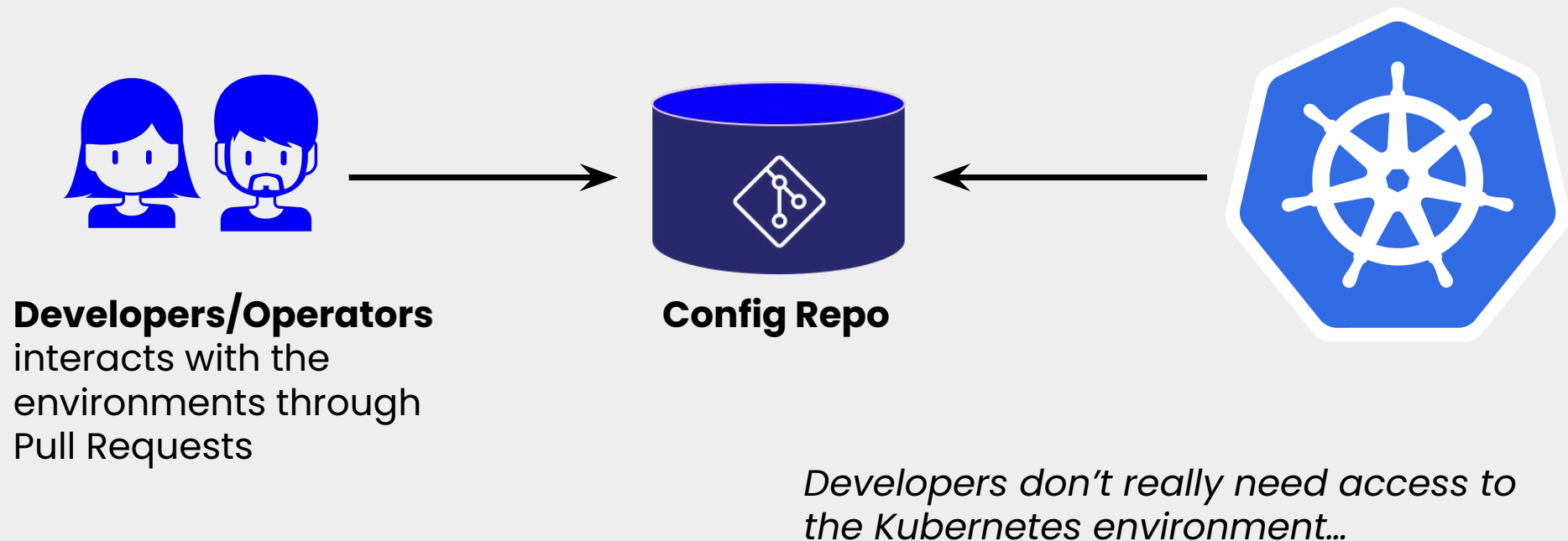
```
$ git log --pretty=format:"%h %ad | %s %d" --date=short
```

```
a20258e573 2021-05-28 | [dev/card-management] release bugfix_app-sync-spam-fade16a9d2-d6669277e3 by ark@lunar.app (HEAD ->
master, origin/master, origin/HEAD)
e201665084 2021-05-28 | [dev/card-payments] release master-1aa5964fc8-d6669277e3 by skw@lunar.app
931c9e7d7b 2021-05-28 | [dev/invest] release feature_onboarding-answers-event-c85113fc14-d6669277e3 by gvy@lunar.app
c3c5155d97 2021-05-28 | [prod/openbanking-connect] auto release master-e1744acbe4-d6669277e3 by bso@lunar.app
ae4891062f 2021-05-28 | [staging/openbanking-connect] auto release master-e1744acbe4-d6669277e3 by bso@lunar.app
a392773670 2021-05-28 | [dev/openbanking-connect] auto release master-e1744acbe4-d6669277e3 by bso@lunar.app
860828e3e8 2021-05-28 | [prod/project-blue] release master-68579c4688-eef57dbcf0 by dtj@lunar.app
fe850ce5ad 2021-05-28 | [dev/movemoney] auto release master-e531fd526a-4b206bf1de by tsj@lunar.app
00167949e7 2021-05-28 | [dev/invest] auto release master-4b2ef32b81-d6669277e3 by gustav-git@pm.me
eccc8a0554 2021-05-28 | [prod/postgresql-controller] auto release master-d4daeac033-6c6ff5b0f3 by jwr@lunar.app
fa0689c5ef 2021-05-28 | [staging/postgresql-controller] auto release master-d4daeac033-6c6ff5b0f3 by jwr@lunar.app
c0e555216f 2021-05-28 | [dev/postgresql-controller] auto release master-d4daeac033-6c6ff5b0f3 by jwr@lunar.app
25df95b06b 2021-05-28 | [prod/houston] release master-25c5735f03-4b206bf1de by sdd@lunar.app
814a5b0e3d 2021-05-28 | [staging/houston] auto release master-25c5735f03-4b206bf1de by sdd@lunar.app
25d5f3f88a 2021-05-28 | [dev/houston] auto release master-25c5735f03-4b206bf1de by sdd@lunar.app
480db23025 2021-05-28 | [platform/flux] auto release master-b3bfd9afb5-6c6ff5b0f3 by kni@lunar.app
7af27ced45 2021-05-28 | [dev/flux] auto release master-b3bfd9afb5-6c6ff5b0f3 by kni@lunar.app
```

[Environment/service] action artifactId by author

Every commit represents the source of truth at a given time

LEAST PRIVILEGE

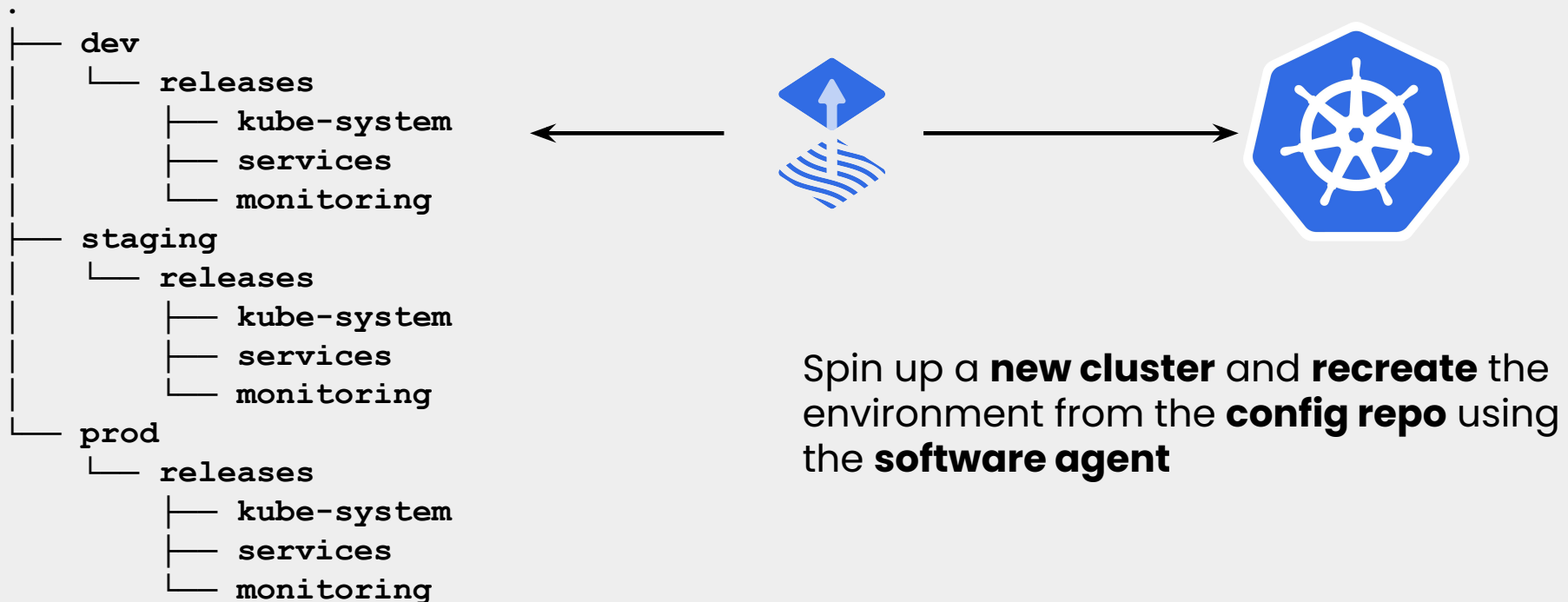


DISASTER RECOVERY

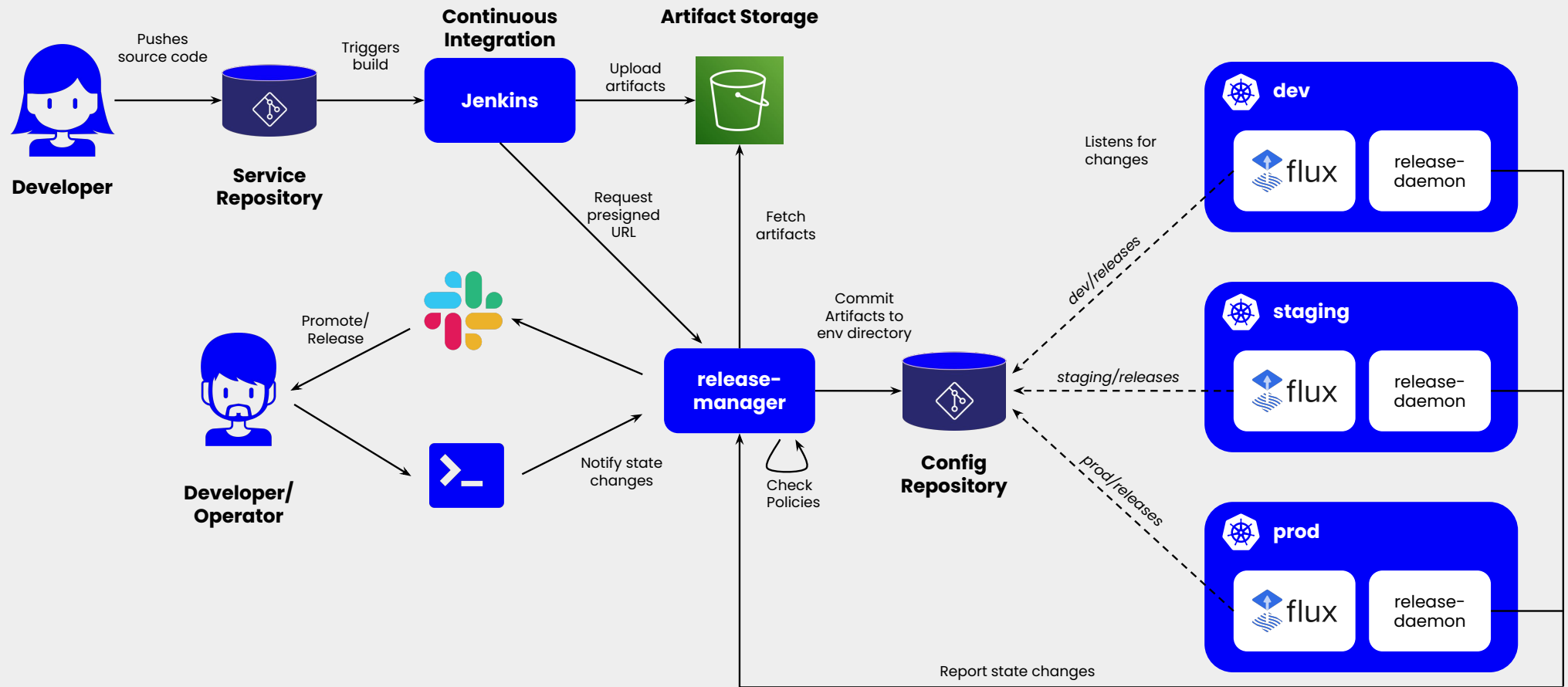


How fast can you
recreate a cluster from
scratch?

DISASTER RECOVERY



LUNAR SETUP



Let's finish with

**Where to go
from here?**

CUSTOM RESOURCES



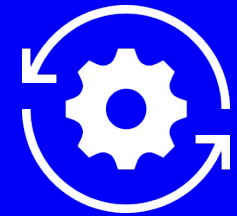
Custom *resources* are extensions of the Kubernetes API

A *resource* is an endpoint in the Kubernetes API that stores a collection of API objects of a certain kind; for example, the built-in pods resource contains a collection of Pod objects.

The **CustomResourceDefinition** API resource allows you to define custom resources.

```
apiVersion: monitoring.coreos.com/v1
kind: Prometheus
metadata:
  labels:
    prometheus: k8s
  name: prometheus
  namespace: monitoring
spec:
  image: quay.io/prometheus/prometheus:v2.26.0
  ...
  retention: 180d
  version: v0.7.0
```

THE OPERATOR PATTERN



Definition

Operators are software **extensions** to Kubernetes that make use of **custom resources** to **manage** applications and their components. Operators follow Kubernetes principles, notably the **control loop**.

Motivation

The Operator pattern aims to capture the key aim of a human operator who is managing a service or set of services. Human operators who look after specific applications and services have deep knowledge of how the system ought to behave, how to deploy it, and how to react if there are problems.

People who run workloads on Kubernetes often like to use **automation** to take care of repeatable tasks. The Operator pattern captures how you can **write code to automate a task beyond what Kubernetes itself provides**.



SOFTWARE AGENTS

All Operators use the controller pattern, but not all controllers are Operators.
















It's only an **Operator** if it's got:

controller pattern + API extension + single-app focus.

Operator is a customized controller implemented with CRD. It follows the same pattern as built-in controllers (i.e. watch, diff, action).

Source: <https://github.com/kubeflow/tf-operator/issues/300>

OPERATOR EXAMPLES

 Portworx Essentials provided by Portworx Free forever cloud native storage solution	 Postgres-Operator provided by Zalando SE Postgres operator creates and manages PostgreSQL clusters running in Kubernetes.	 Postgresql Operator provided by Openlabs Deploys postgresql based applications	 PostgreSQL Operator by Dev4Devs.com provided by Dev4Devs.com Operator in Go developed using the Operator Framework	 Prisma Cloud Compute (Twistlock) Console Operator provided by Palo Alto Networks
 Project Quay provided by Project Quay Project Quay is a private container registry that stores, builds, and deploys containers.	 Project Quay Container Security provided by Project Quay Identify image vulnerabilities in Kubernetes pods	 Prometheus Exporter Operator provided by Red Hat Operator to setup 3rd party prometheus exporters, with	 Prometheus Operator provided by Red Hat Manage the full lifecycle of configuring and managing Prometheus and Alertmanager.	 Pystol provided by Pystol The fault injection platform
 Redis Enterprise provided by Redis Labs, Inc. An operator to run Redis Enterprise Clusters	 Redis Operator provided by Opstree Solutions A Golang based redis operator that will make/oversee Redis standalone/cluster mode setups.	 Ripsaw provided by Red Hat Performance Ripsaw is a benchmark operator to benchmark k8s	 Robin Cloud Native Storage provided by Robin.io Robin Cloud Native Storage operator enables advanced data management capabilities.	 RocketMQ Operator provided by the Apache Software Foundation The RocketMQ Operator manages the Apache

Currently **191 operators**
listed on
[OperatorHub.io](https://operatorhub.io)

Can we also

**manage
software outside
the cluster?**

MANAGE CLOUD RESOURCES



Crossplane.io

Crossplane is an open source Kubernetes add-on that enables platform teams to assemble infrastructure from multiple vendors, and expose higher level self-service APIs for application teams to consume, without having to write any code.



ACK

AWS Controllers for Kubernetes (ACK) lets you define and use AWS service resources directly from Kubernetes.

Similar Cloud Provider solutions exists for Google Cloud and Microsoft Azure

CROSSPLANE

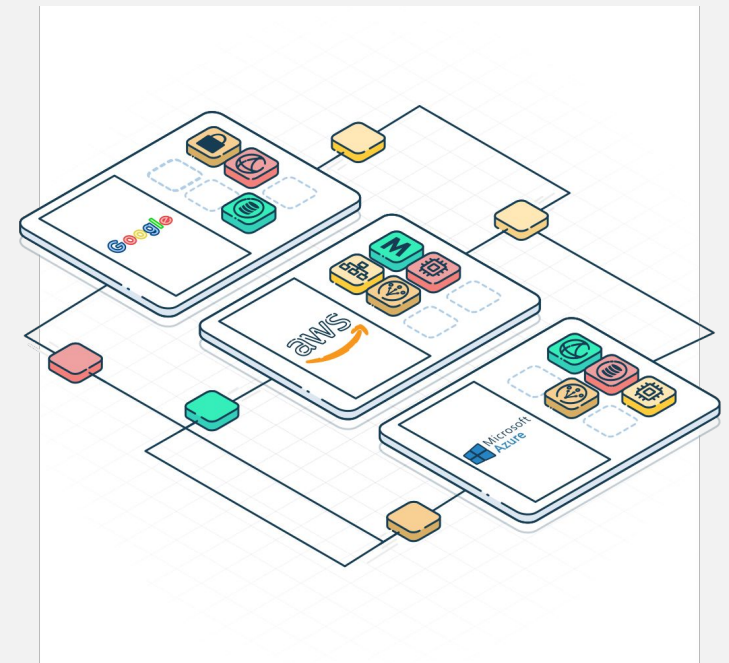


Crossplane is an open source Kubernetes add-on that enables platform teams to assemble infrastructure from multiple vendors, and expose higher level self-service APIs for application teams to consume, without having to write any code.

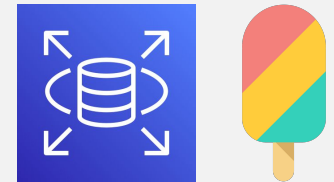
Concepts:

A *Composite Resource* (XR) is a special kind of custom resource that is composed of other resources.

A *CompositeResourceDefinition* (XRD) defines a new kind of composite resource, and optionally the claim it offers.



CROSSPLANE EXAMPLE



As a Software Engineer I need a database for my service.

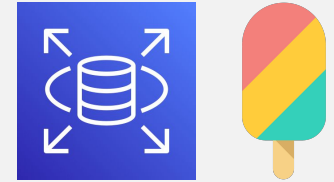


As a Platform Engineer I want certain defaults for how databases are created.

PostgreSQLInstance

```
apiVersion: database.example.org/v1alpha1
kind: PostgreSQLInstance
metadata:
  name: my-db
  namespace: default
spec:
  parameters:
    storageGB: 20
  compositionSelector:
    matchLabels:
      provider: aws
  writeConnectionSecretToRef:
    name: db-conn
```

CROSSPLANE EXAMPLE



CompositeResourceDefinition

```
apiVersion: apiextensions.crossplane.io/v1
kind: CompositeResourceDefinition
metadata:
  name: compositepostgresinstances.org
spec:
  group: database.example.org
  names:
    kind: CompositePostgreSQLInstance
    plural: compositepostgresinstances
  claimNames:
    kind: PostgreSQLInstance
    plural: postgresqlinstances
  connectionSecretKeys:
    - username
    - ...
  versions:
    - name: v1alpha1
      schema:
        openAPIV3Schema:
          type: object
          ...
```

Composition

```
apiVersion: apiextensions.crossplane.io/v1
kind: Composition
metadata:
  name: vpcpostgresinstances...org
  labels:
    provider: aws
spec:
  writeConnectionSecretsToNamespace: crossplane-system
  compositeTypeRef:
    apiVersion: database.example.org/v1alpha1
    kind: CompositePostgreSQLInstance
  resources:
    - name: vpc
      base:
        apiVersion: ec2.aws.crossplane.io/v1beta1
        kind: VPC
        spec:
          forProvider:
            cidrBlock: 192.168.0.0/16
    - name: subnet-a
    - name: dbsubnetgroup
    - name: securitygroup
    - ...
    - name: rdsinstance
```

ACK EXAMPLE



Provisioning a S3 Bucket



```
apiVersion: s3.services.k8s.aws/v1alpha1
kind: Bucket
metadata:
  name: example-service
  namespace: default
spec:
  name: example-service
```

Is that all

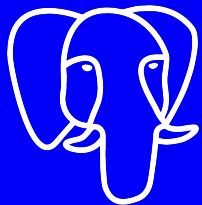
we can manage?

EXAMPLE FROM LUNAR[®]

postgresql-controller

postgresql-controller is a Kubernetes controller for managing users and their access rights to a PostgreSQL database instance. Its purpose is to make a codified description of what users have access to what databases and for what reason along with providing an auditable log of changes.

Available at: <https://github.com/lunarway/postgresql-controller>



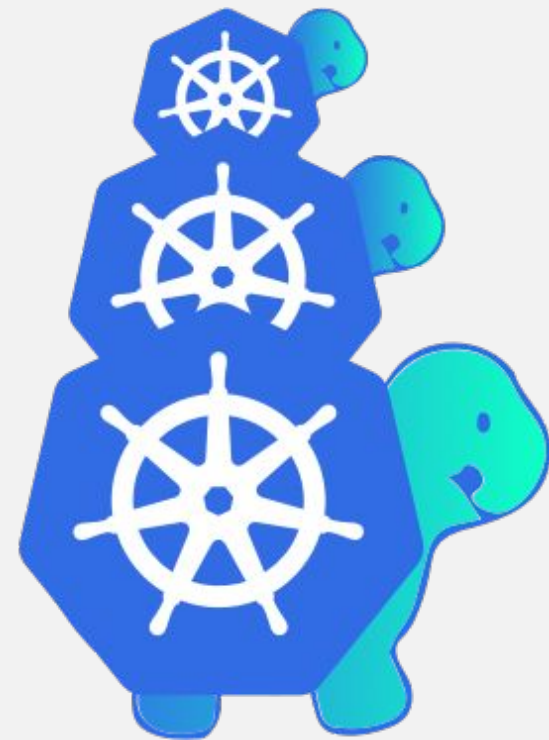
```
apiVersion: postgresql.lunar.tech/v1alpha1
kind: PostgreSQLUser
metadata:
  name: kni
  namespace: dev
spec:
  name: kni
  read:
    - host:
        valueFrom:
          configMapKeyRef:
            name: database
            key: db.host
  allDatabases: true
  reason: I am a developer in Lunar Tech.
```

What about

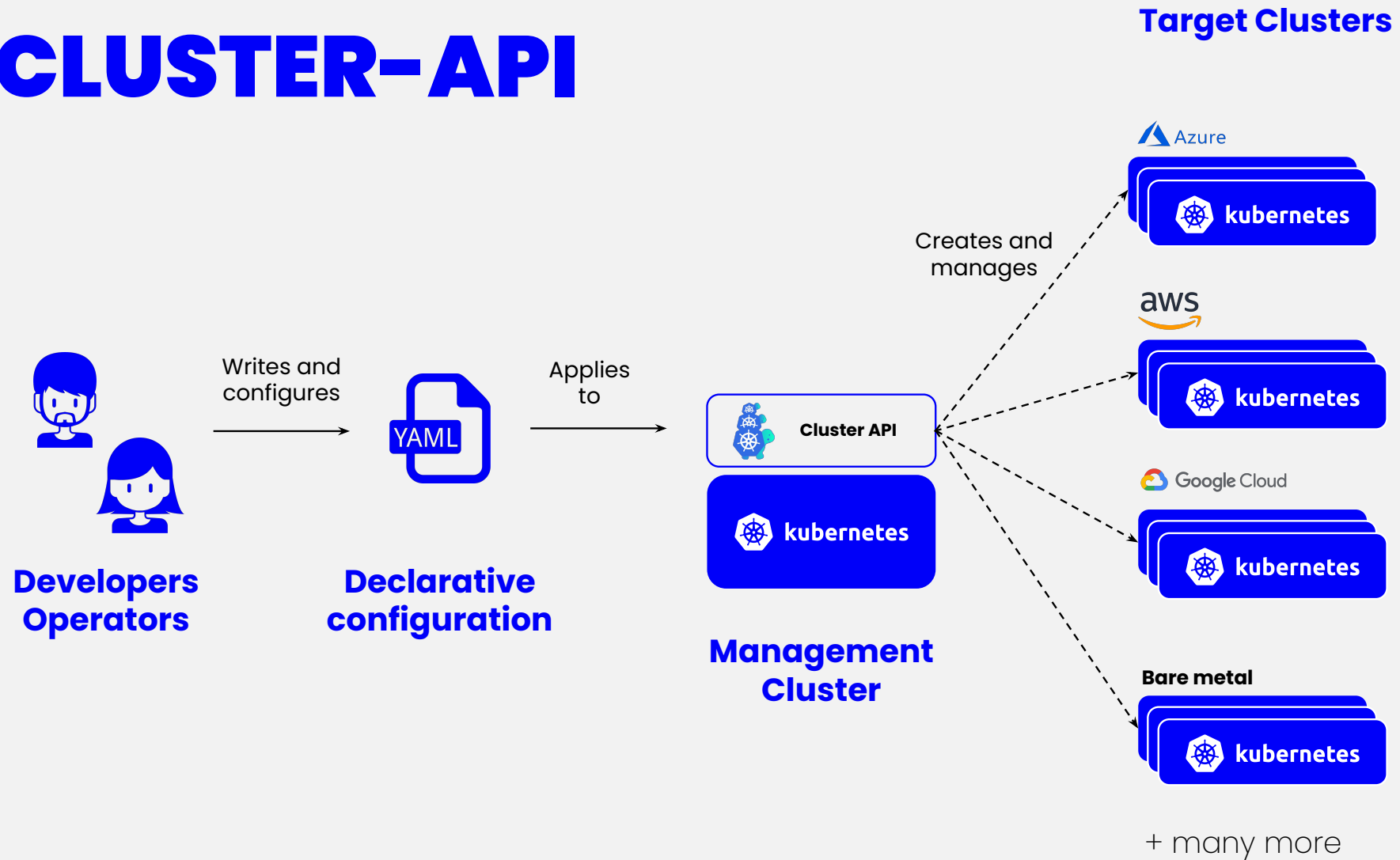
**kubernetes
clusters?**

CLUSTER API

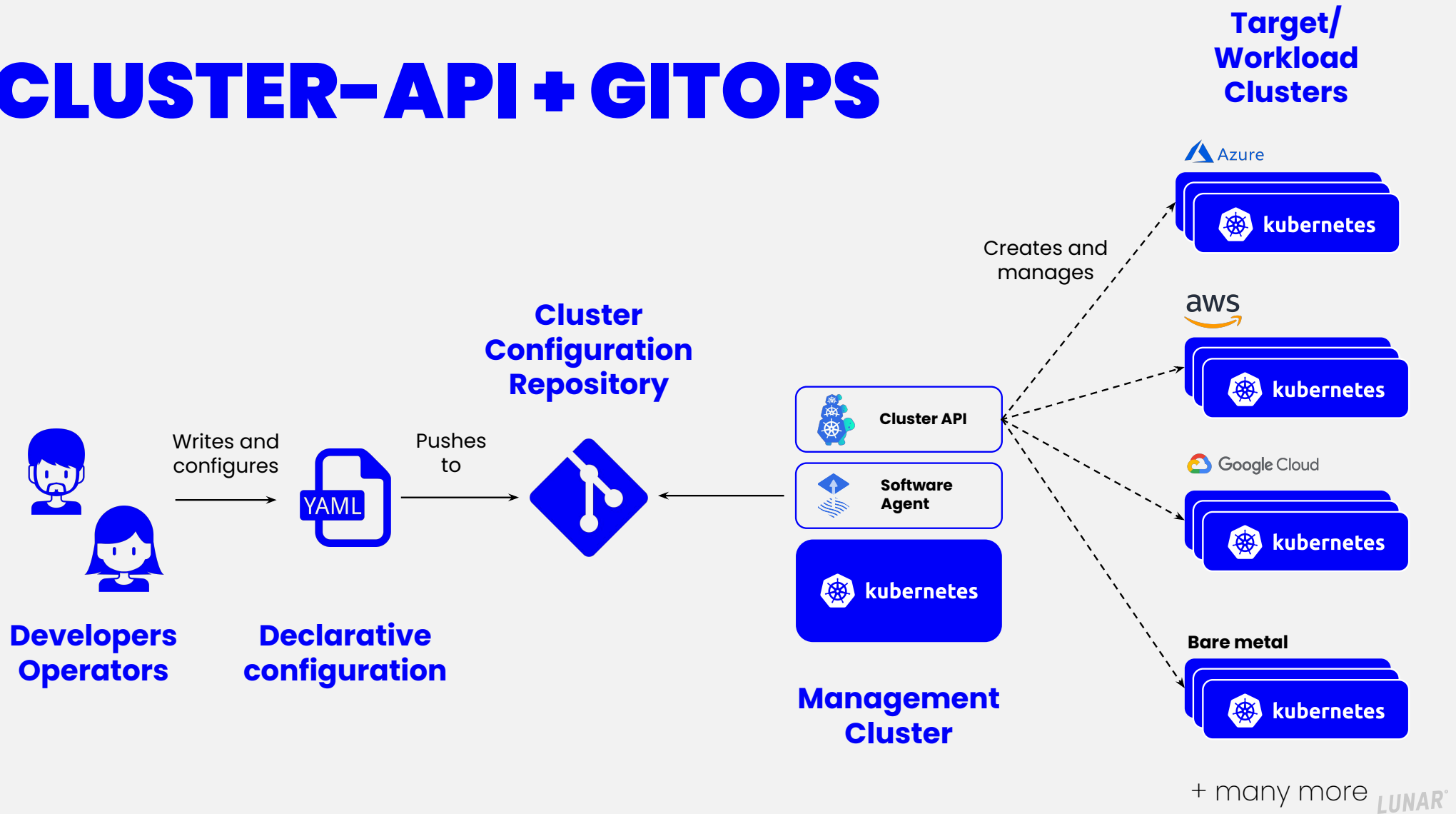
Cluster API is a Kubernetes sub-project focused on providing declarative APIs and tooling to simplify provisioning, upgrading, and operating multiple Kubernetes clusters.



CLUSTER-API



CLUSTER-API + GITOPS



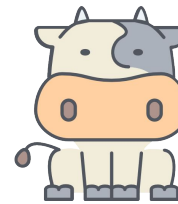
Cattle vs Pets

We stopped doing pet servers...

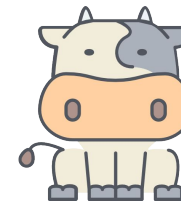
LUCY



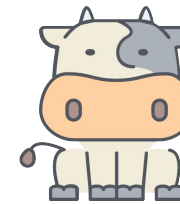
TANGO



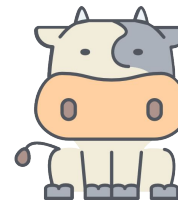
COW-001



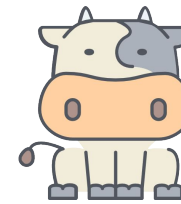
COW-002



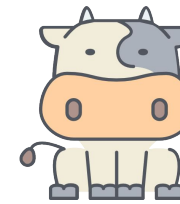
COW-003



COW-004



COW-005



COW-006

Cattle vs Pets

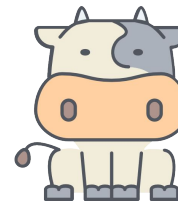
We stopped doing pet servers...
... but created pet clusters.

LUCY

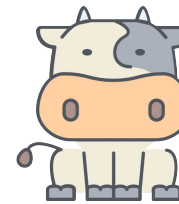


TANGO

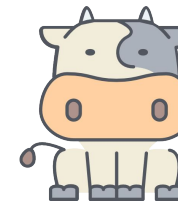
PRODUCTION



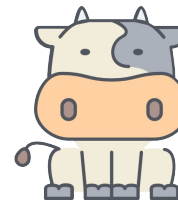
COW-001



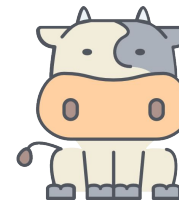
COW-002



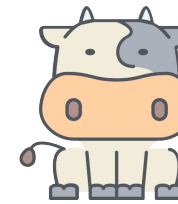
COW-003



COW-004



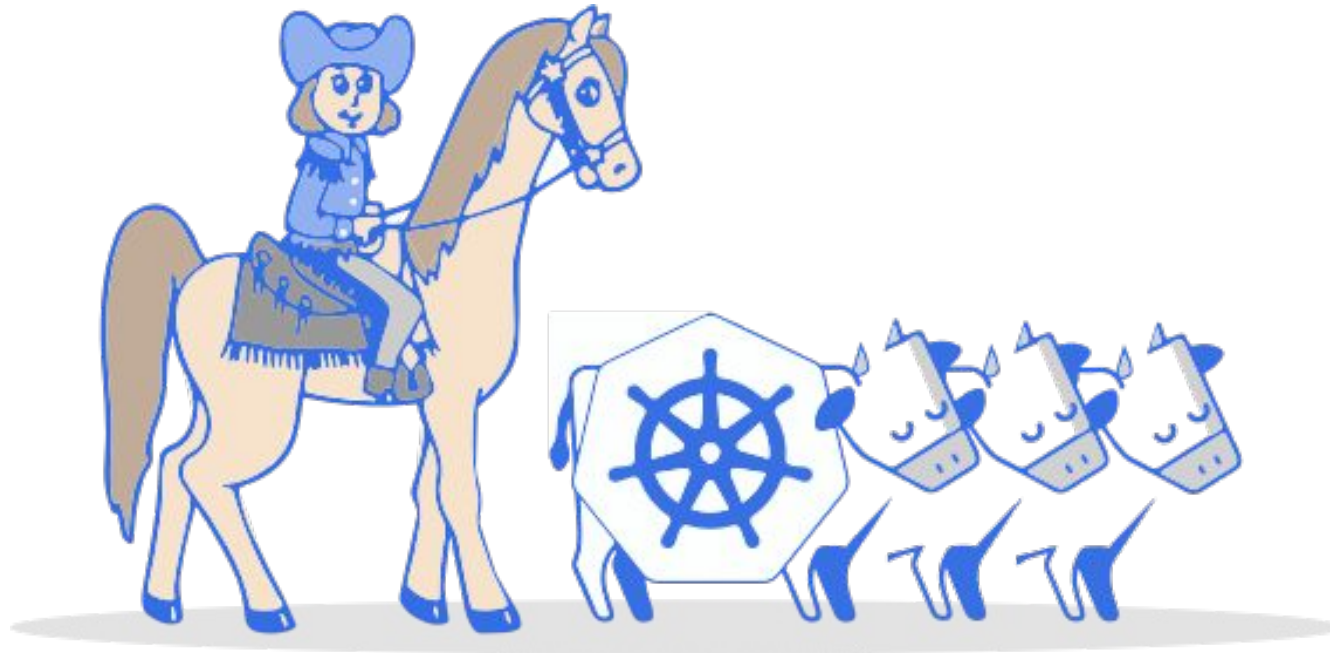
COW-005



COW-006

Cluster API

Clusters as herds... instead of pets.



CONCEPTS

Infrastructure provider

A source of computational resources, such as compute and networking. Cloud Infrastructure Providers include AWS, Azure, and Google, and bare metal Infrastructure Providers include VMware, MAAS, and metal3.io.

Bootstrap provider

The Bootstrap Provider is responsible for:

1. Generating the cluster certificates, if not otherwise specified
2. Initializing the control plane, and gating the creation of other nodes until it is complete
3. Joining control plane and worker nodes to the cluster

Control Plane

The control plane is a set of services that serve the Kubernetes API and continuously reconcile desired state using control loops.

CLUSTER API EXAMPLE (EKS)

Cluster

```
apiVersion: cluster.x-k8s.io/v1alpha3
kind: Cluster
metadata:
  name: eks
  namespace: default
spec:
  clusterNetwork:
    pods:
      cidrBlocks:
        - 192.168.0.0/16
  controlPlaneRef:
    apiVersion: controlplane.cluster.x-k8s.io/v1alpha3
    kind: AWSManagedControlPlane
    name: eks-control-plane
  infrastructureRef:
    apiVersion: controlplane.cluster.x-k8s.io/v1alpha3
    kind: AWSManagedControlPlane
    name: eks-control-plane
```

ControlPlane

```
apiVersion: controlplane.cluster.x-k8s.io/v1alpha3
kind: AWSManagedControlPlane
metadata:
  name: eks-control-plane
  namespace: default
spec:
  region: eu-west-1
  sshKeyName: default
  version: v1.20.4
```

Source: <https://cluster-api.sigs.k8s.io/user/quick-start.html>

CLUSTER API EXAMPLE (EKS)

Machine Deployment

```
apiVersion: cluster.x-k8s.io/v1alpha3
kind: MachineDeployment
metadata:
  name: eks-md-0
  namespace: default
spec:
  clusterName: eks
  replicas: 2
  template:
    spec:
      bootstrap:
        configRef:
          apiVersion: bootstrap.cluster.x-k8s.io/v1alpha3
          kind: EKSConfigTemplate
          name: eks-md-0
      clusterName: eks
      infrastructureRef:
        apiVersion: infrastructure.cluster.x-k8s.io/v1alpha3
        kind: AWSMachineTemplate
        name: eks-md-0
      version: v1.20.4
```

Two arrows originate from the MachineDeployment manifest. One arrow points from the 'kind: EKSConfigTemplate' entry in the 'bootstrap.configRef' field to the EKSConfigTemplate manifest. The other arrow points from the 'kind: AWSMachineTemplate' entry in the 'infrastructureRef' field to the AWSMachineTemplate manifest.

EKSConfigTemplate

```
apiVersion: bootstrap.cluster.x-k8s.io/v1alpha3
kind: EKSConfigTemplate
metadata:
  name: eks-md-0
  namespace: default
spec:
  template: {}
```

AWSMachineTemplate

```
apiVersion: infrastructure.cluster.x-k8s.io/v1alpha3
kind: AWSMachineTemplate
metadata:
  name: eks-md-0
  namespace: default
spec:
  template:
    spec:
      iamInstanceProfile: nodes....sigs.k8s.io
      instanceType: t3.large
      sshKeyName: default
```

Source: <https://cluster-api.sigs.k8s.io/user/quick-start.html>

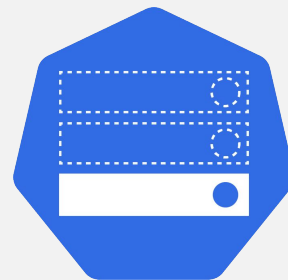
CLUSTER API RESOURCES



Cluster



Machine
Deployment



MachineSet



Machine



Deployment



ReplicaSet

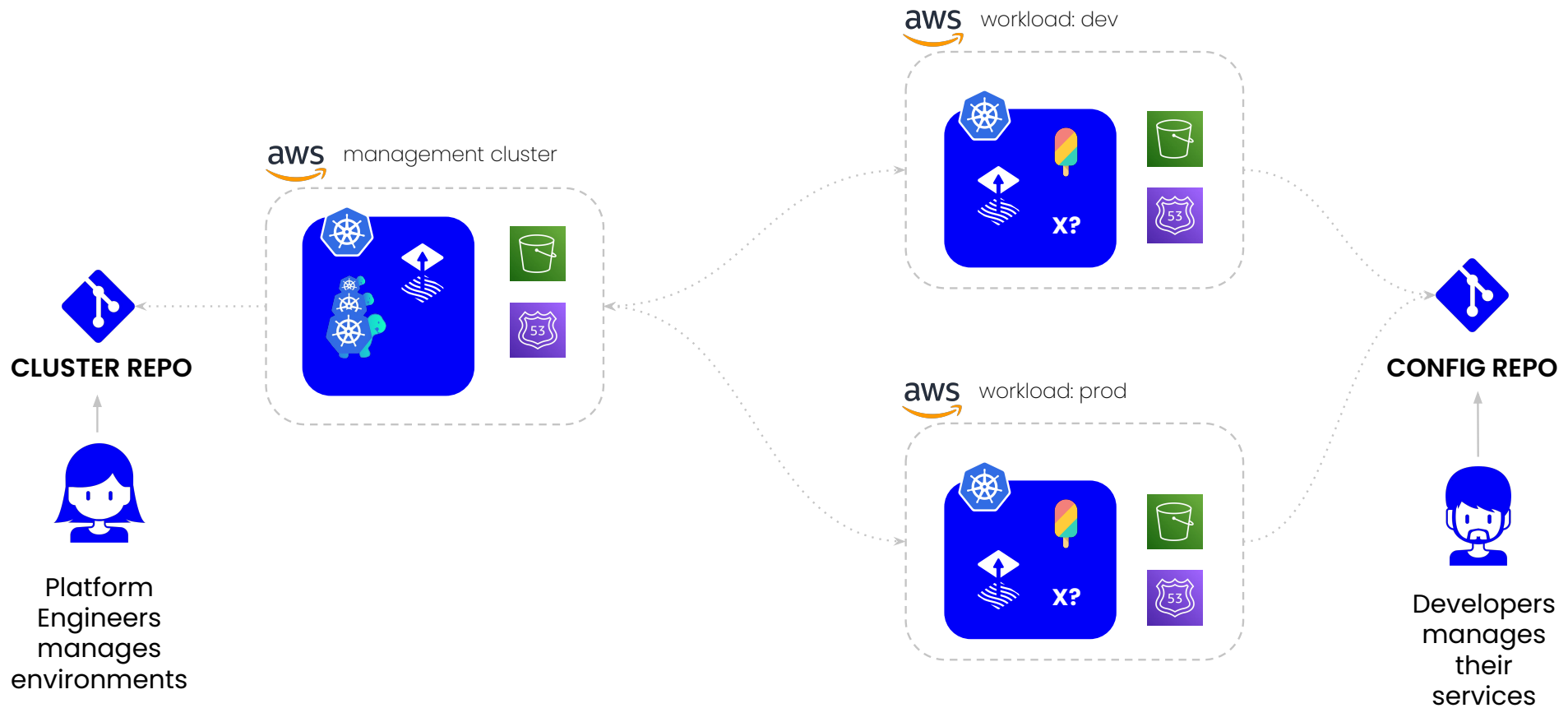


Pod

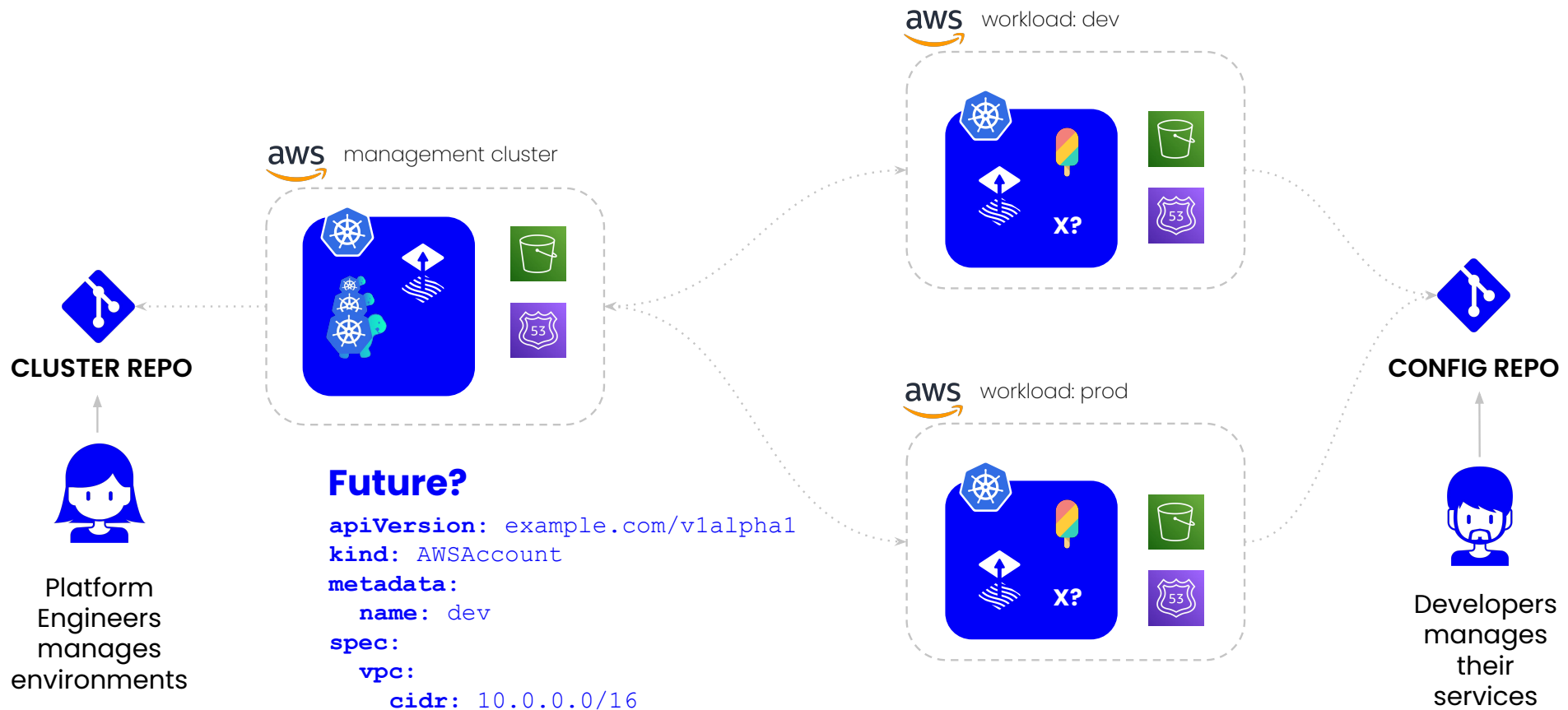
Let's wrap up with

**Putting it all
together**

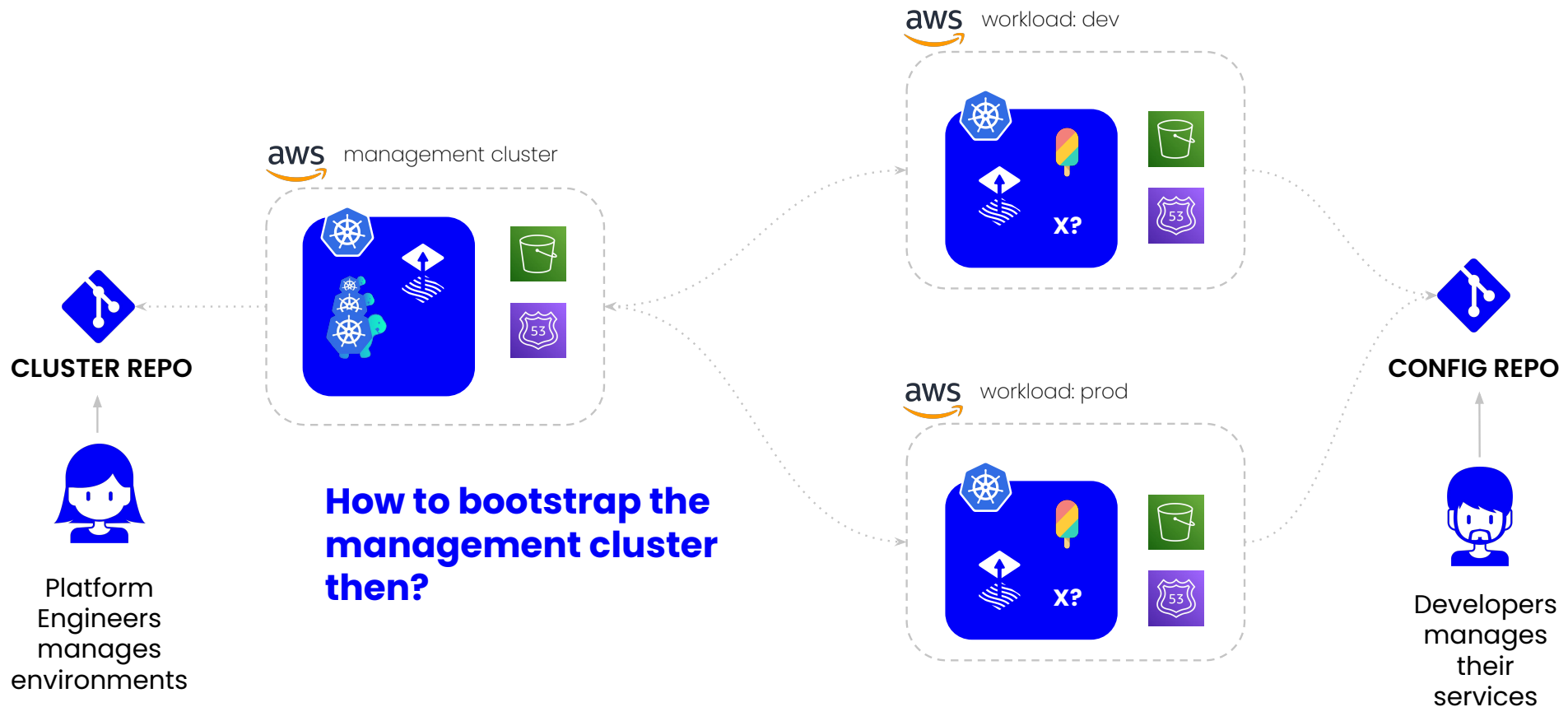
THE GRAND VISION



THE GRAND VISION



THE GRAND VISION



THE GRAND VISION

- Workloads accounts should be fully managed by a set of software agents, operators, and cluster-api through a GitOps Configuration repository.
- Only “real services” should run in the workload environments, plus collectors, and other needed agents.
- All changes to services and infrastructure happens in code, is audited, and can easily be restored in case of disaster.

**Things change,
banks should too...**



Questions?



kni@lunar.app



@phennex

LUNAR[®]

Join us!

jobs.lunar.app



Don't forget to
vote for this session
in the **GOTO Guide app**