

GITOPS, IAC & PULUMI

-

(CLOUD) INFRASTRUCTURE AS CODE DONE RIGHT (?)
... A JOURNEY FROM KUBECTL APPLY TO GIT PUSH

ANDREAS TELL

CADEC 2023.01.19 & 2023.01.25 | [CALLISTAENTERPRISE.SE](https://callistaenterprise.se)

CALLISTA

AGENDA

- GitOps
 - Definition
 - Implementation
- Infrastructure as Code (IaC)
 - Pulumi
- Demo
- Wrap Up

WHAT'S GITOPS?

A set of principles for operating and managing software systems with
Git, CI/CD (Automation) and IaC (Infrastructure as Code)

Infrastructure as Code (IaC)

CADEC 2021

is the managing and provisioning of infrastructure through machine-readable definition files* instead of through manual processes.

2020 : GitOps

2021 (Oct)

* Proprietary syntaxes, JSON, YAML
or
General Purpose Programming
Languages

CLOUD NATIVE
COMPUTING FOUNDATION

1.0 published

WHAT'S GITOPS?

A set of principles for operating and managing software systems with
Git, CI/CD (Automation) and IaC (Infrastructure as Code)
... leverages existing and widely adopted best practices
... with a strong “Kubernetes-affinity”

2017 : First coined by CTO of Weaveworks Inc.

2020 : GitOps working group founded 

2021 (October) : GitOps principles v1.0.0 published

GitOps Principles

v1.0.0

2021-10-08
[HTTPS://OPENGITOPS.DEV/
BLOG/1.0-ANNOUNCEMENT/](https://opengitops.dev/blog/1.0-announcement/)

IaC ←

1 Declarative

DESIRED STATE
The aggregate of all configuration data that is sufficient to recreate the system so that instances of the system are behaviourally indistinguishable...

A **system** managed by GitOps must have its desired state expressed **declaratively**.

Git ←

2 Versioned and Immutable

Desired state is **stored** in a way that enforces immutability, versioning and retains a complete version history.



“Embedded CD process”
a.k.a. “Reconciler”

3 Pulled Automatically

Software agents automatically pull the desired state declarations from the source.



4 Continuously Reconciled

Software agents **continuously** observe actual system state and **attempt to apply** the desired state.

IMPLEMENTING GITOPS - PULL & PUSH PIPELINES



GitOps - "Push Pipeline"

"Core Infra"
Networking
IAM
DB
Kubernetes Cluster

GitOps - "Pull Pipeline"

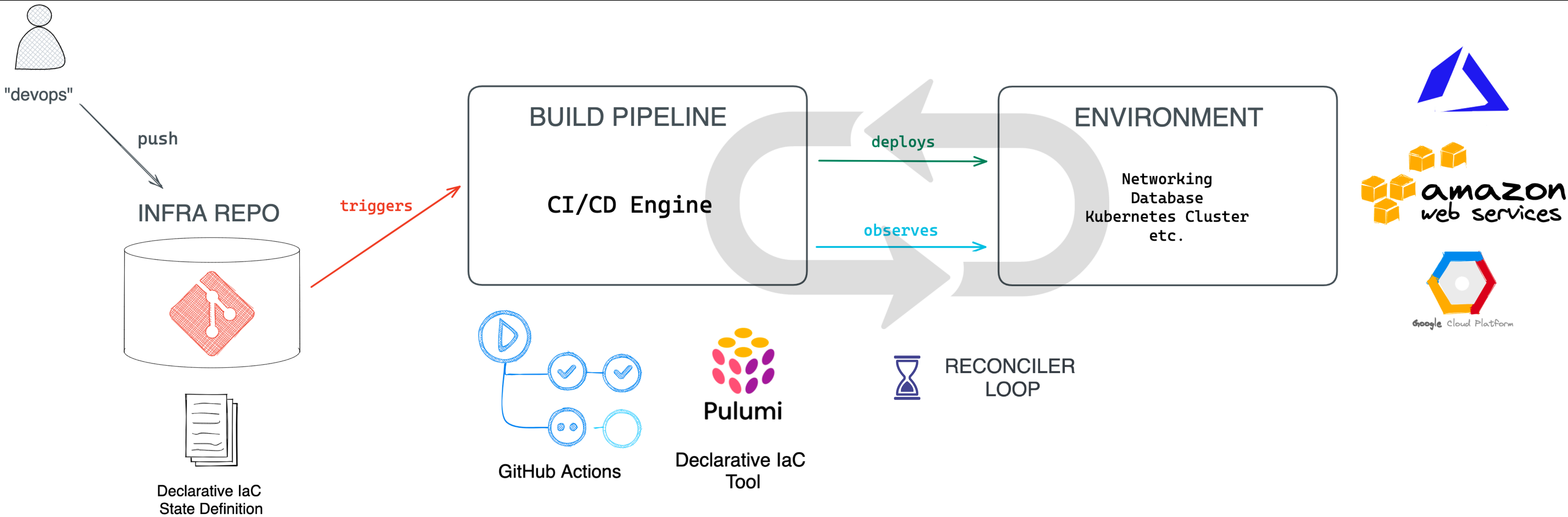


GitOps Principles
v1.0.0

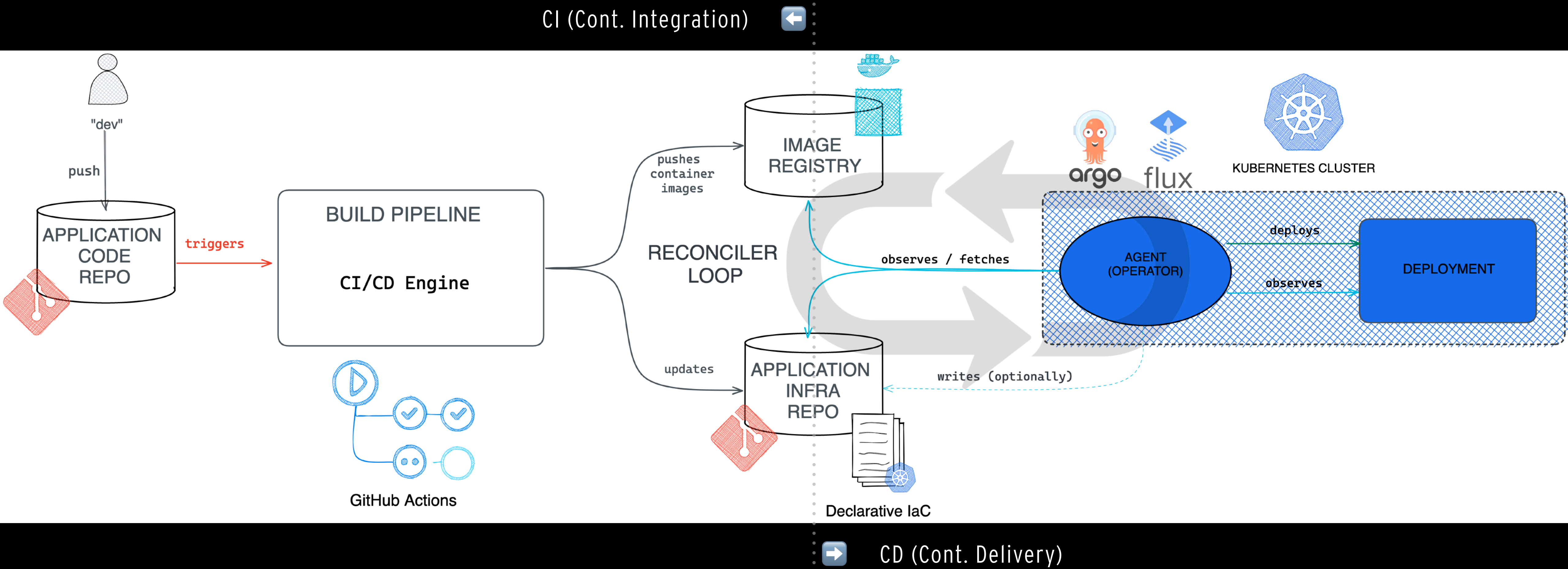
- 1 Declarative**
A system managed by GitOps must have its desired state expressed *declaratively*.
- 2 Versioned and Immutable**
Desired state is *stored* in a way that enforces immutability, versioning and retains a complete version history.
- 3 Pulled Automatically**
Software agents automatically pull the desired state declarations from the source.
- 4 Continuously Reconciled**
Software agents *continuously* observe actual system state and *attempt* to apply the desired state.

PNG

IMPLEMENTATION : GITOPS STYLE - "PUSH PIPELINE" - SETUP CORE INFRA



IMPLEMENTATION : TRUE GITOPS - "PULL PIPELINE"

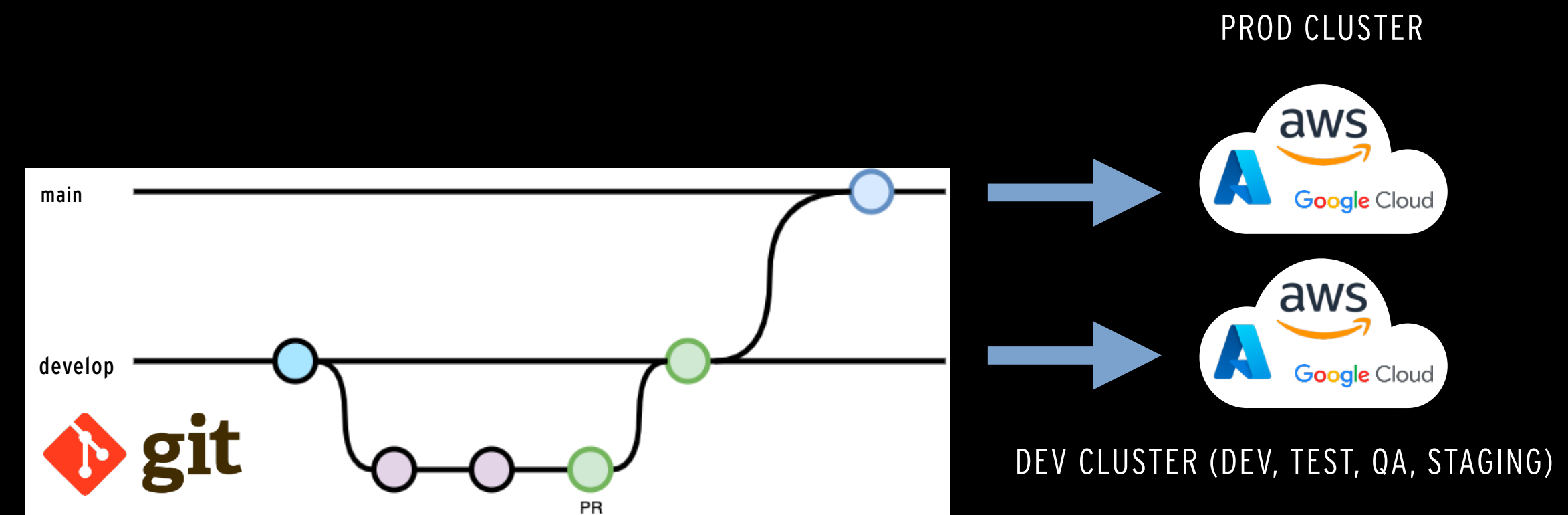


Purpose: Application Cont. Deployment



IMPLEMENTING GITOPS - "OPS BY PULL REQUEST"

- Operations become “a function of Git interaction” (always)
 - Bonus: no need for RW access to infrastructure (Cloud, Kubernetes etc.)
- Branches protected by Pull/Merge requests (e.g. “main” -> “Production”)
 - control gate / quality assurance
- Dedicated config repos
 - Segregation of configuration based on intended usage, access accordingly



IMPLEMENTING GITOPS - "OPS BY PULL REQUEST"

- Operations become “a function of Git interaction” (always)
 - Bonus: no need for RW access to infrastructure (Cloud, Kubernetes etc.)
- Branches protected by Pull/Merge requests (e.g. “main” -> “Production”)
 - control gate / quality assurance
- Dedicated config repos
 - Segregation of configuration based on intended usage, access accordingly



CLAIMED BENEFITS / VALUE PROPOSITION

GitOps =>

Increased Developer & Operational Productivity

Shorter MTTR

Improved DX

Improved collaboration

Improved consistency, predictability and repeatability

Higher Reliability

Improved Security

Reduce Cost

Increase Speed

Minimize Risk

| AGENDA

- GitOps
 - Definition
 - Implementation
- **Infrastructure as Code (IaC) - Pulumi**
- Demo
- Wrap Up

IAC TOOLS FOR CLOUD

Single cloud / Vendor Proprietary

AWS CloudFormation

AWS CDK



Azure Resource Manager (ARM)

Azure Bicep



Google Cloud Deployment Manager



...

Multi cloud / Multi purpose

Pulumi



Terraform



Crossplane



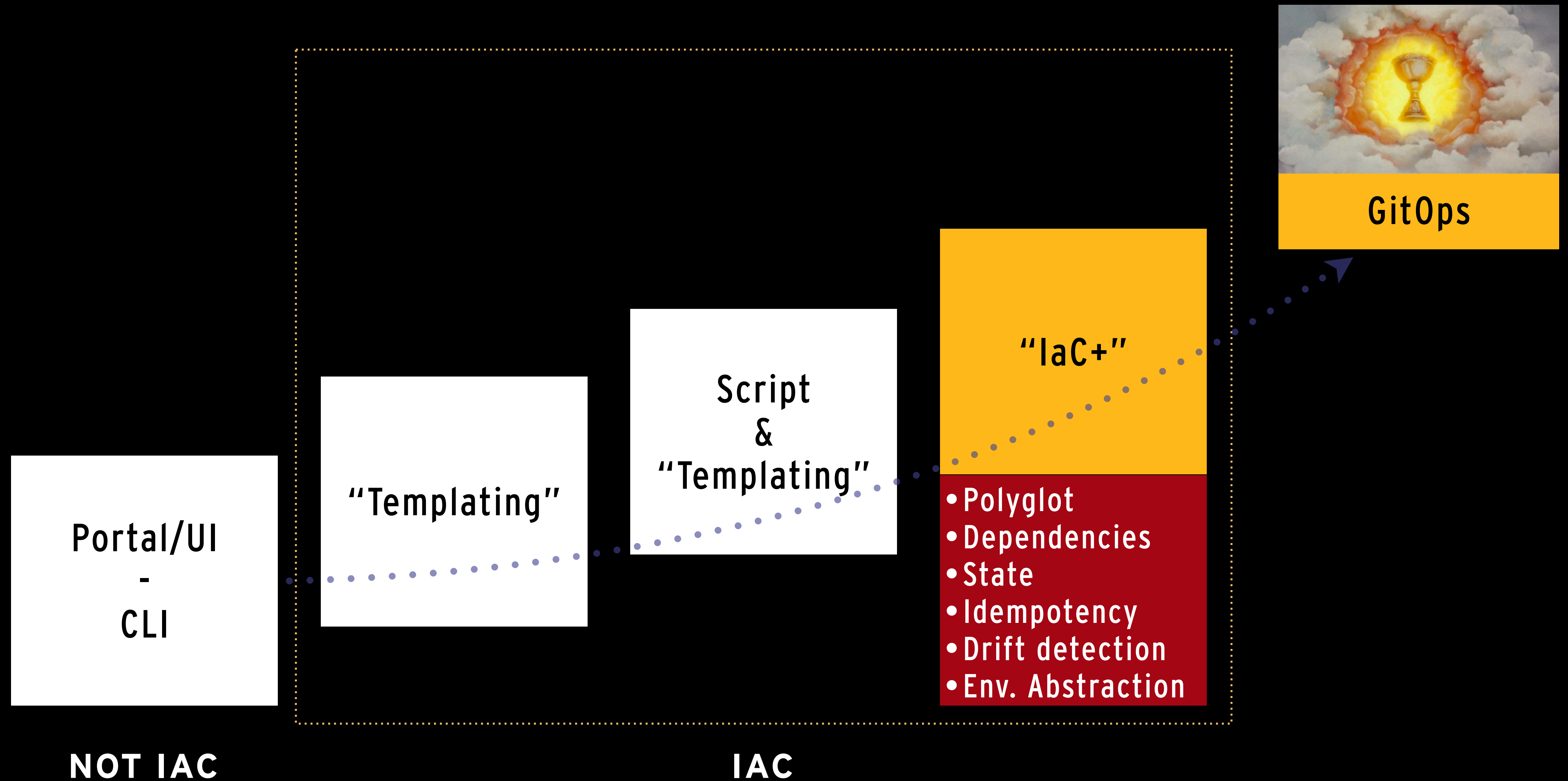
Ansible



Red Hat

...

PROVISIONING CLOUD INFRA - A STEPWISE PROGRESSION



PULUMI - INTRO

- Infrastructure as Code tool for creating, deploying, and managing infrastructure

- “modern” & “traditional”

- Supports 70+ providers

- Open source

- Used by

- Atlassian

- Meta

- Mercedes-Benz

- ...

v1.0 : 2019-09

Your Cloud, Your Language, Your Way

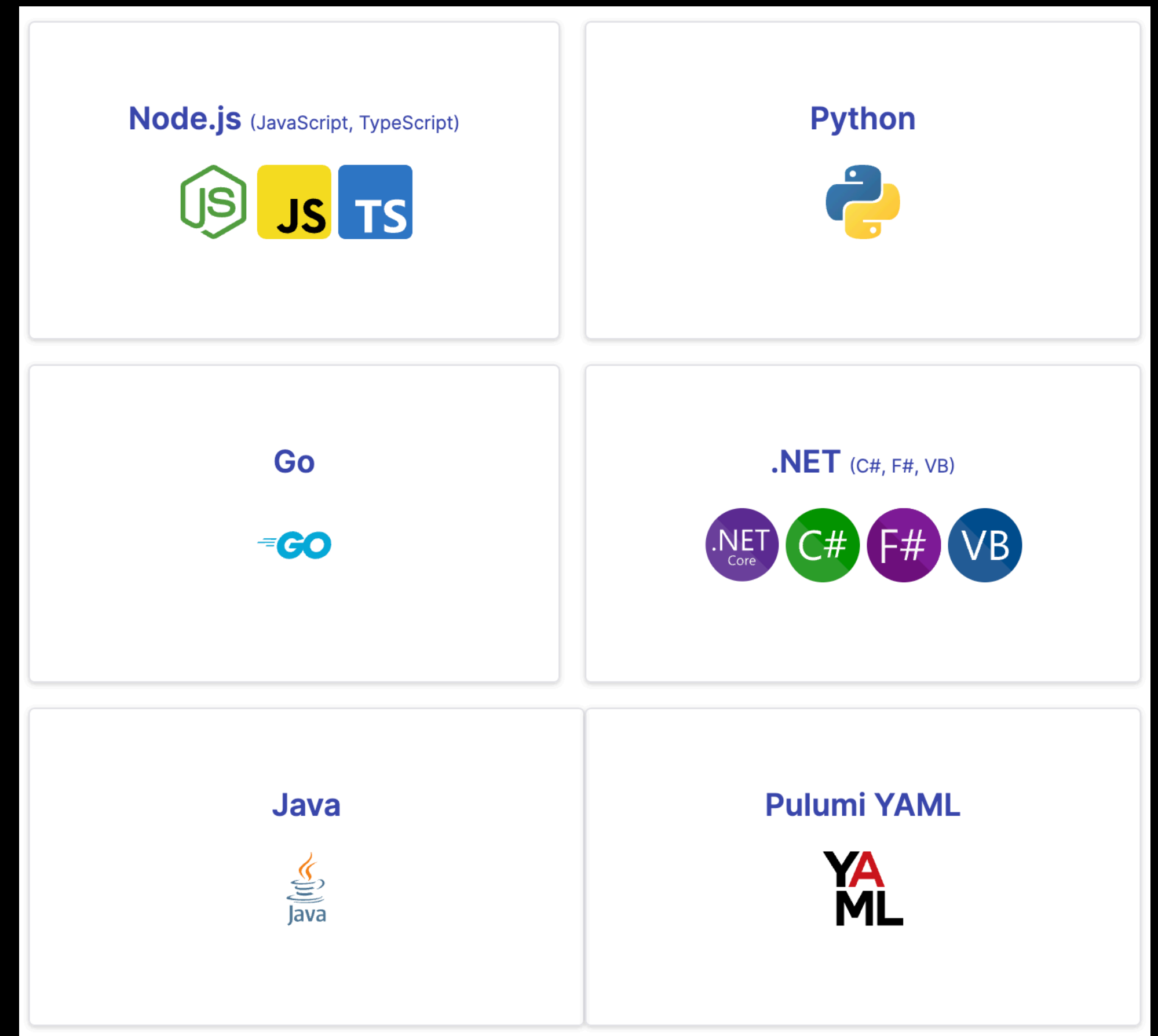
The diagram illustrates the Pulumi ecosystem, organized into several categories:

- Providers:** A large central box lists various cloud providers including AWS, Microsoft Azure, Google Cloud, Kubernetes, HELM, ORACLE, Alibaba Cloud, DigitalOcean, VMware vSphere, linode, openstack, HETZNER, EQUINIX, METAL, and fastly.
- Languages:** A box on the left lists programming languages and frameworks such as node (JS, TS), python, .NET Core, C#, VB, F#, Java, Scala, Kotlin, Clojure, GO, and YAML.
- CI/CD & DevOps:** A box on the right lists tools like Travis CI, CircleCI, codefresh, Azure DevOps, Bitbucket, Azure Pipelines, Octopus Deploy, and Jenkins.
- Build:** A category containing IDEs (VS Code, IntelliJ), Tools (Python, TS, List, Docker, Maven, Bowler), and Packages (python, npm, docker, JFrog).
- Verify:** A category containing packages like python, npm, docker, and JFrog.
- Deploy:** A category containing Foundation (aiven, docker, @mailgun, okta, RANCHER, spotinst), Data (CloudAMQP, MINIO, mongoDB, MySQL, PostgreSQL, elastic, kafka, RabbitMQ, databricks), Network (CLOUDFLARE, NS1, dnssimple, Kong), and Others (SignalFx, splunk, DADOOG, New Relic, GitHub, GitLab, Auth0, PagerDuty, slack).
- Scale:** A category containing Architecture (docker, Kubernetes, AWS Lambda, Amazon S3) and SCM (git, GitLab, GitHub, ATLISSIAN).
- Manage:** A category containing Team (Azure Active Directory, G Suite, okta, SAML) and other management tools.

At the bottom, there is a Twitter handle @PulumiCorp, a website URL <https://pulumi.com>, and the Pulumi logo.

| PULUMI GOODNESS

- Multi-cloud
- “Infrastructure as Software”
- “Native providers”
 - Same day access to new features
 - » <https://www.pulumi.com/blog/pulumiup-native-providers/>
- Has properties to support “a GitOps style workflow”:
 - Can observe infrastructure & detect drift
 - Idempotent
- Environment abstraction
- Built in secrets management



Why use a programming language to describe infrastructure?

- Familiarity (for Devs)
 - Syntax
 - Tooling - IDE
- Auto complete (IDE)
- Type safety
- Modularity
 - Reuse
- Logical constructs
- Testability

Cadec 2021 : AWS Cloud Development Kit

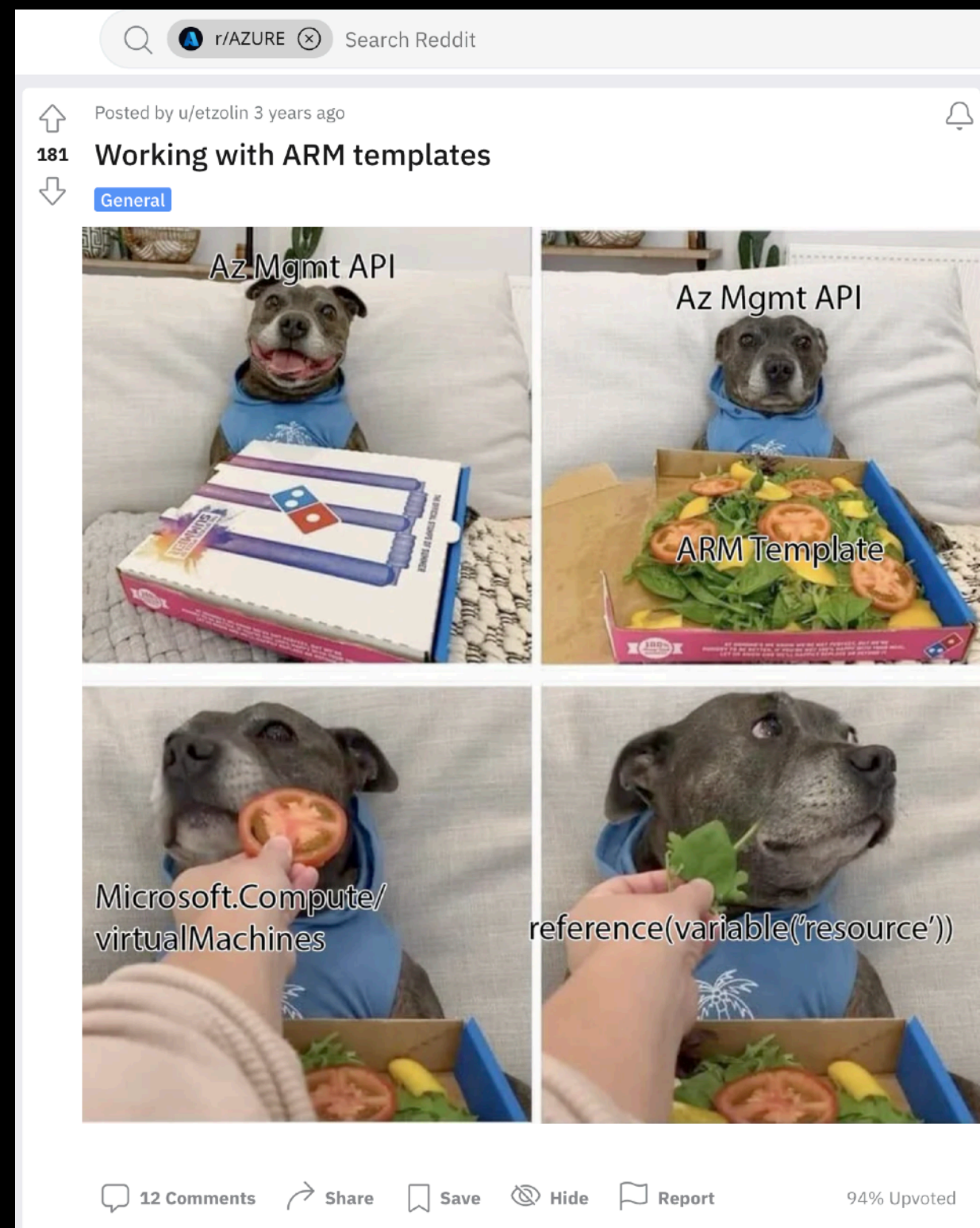
PROS & CONS	
CLOUDFORMATION	CDK
<p>+</p> <p>General perks of Infrastructure-as-Code</p> <p>-</p> <p>Syntax of YAML and JSON</p> <p>No type safety</p> <p>Demands good knowledge of underlying services</p>	<p>+</p> <p>Existing IDE</p> <p>Type safety</p> <p>Abstraction</p> <p>Export self-made constructs</p> <p>Improved readability</p> <p>Logical constructs</p> <p>-</p> <p>Risk of erroneous configuration (abstraction)</p> <p>Some constructs require work-arounds</p>

CALLISTA

IAC EXAMPLE - AZURE STORAGE ACCOUNT

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "location": {
      "type": "string",
      "defaultValue": "[resourceGroup().location]"
    },
    "storageAccountName": {
      "type": "string",
      "defaultValue": "[concat('storage', uniqueString(resourceGroup().id))]"
    }
  },
  "resources": [
    {
      "type": "Microsoft.Storage/storageAccounts",
      "apiVersion": "2021-06-01",
      "name": "[parameters('storageAccountName')]",
      "location": "[parameters('location')]",
      "sku": {
        "name": "Standard_LRS"
      },
      "kind": "StorageV2",
      "properties": {
        "accessTier": "Hot"
      }
    }
  ]
}
```

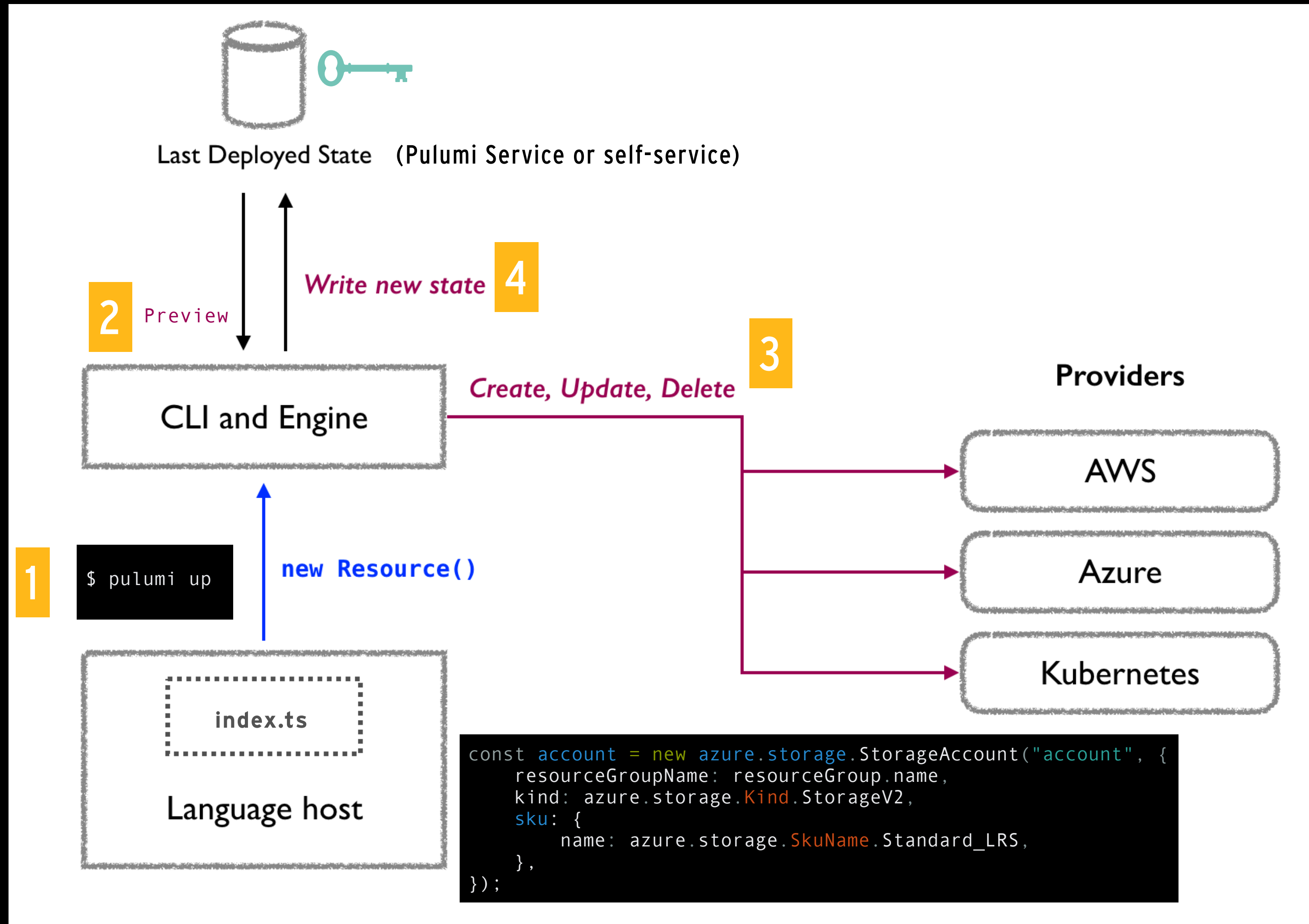
ARM Template



```
const account = new azure.storage.StorageAccount("account", {
  resourceGroupName: resourceGroup.name,
  kind: azure.storage.Kind.StorageV2,
  sku: {
    name: azure.storage.SkuName.Standard_LRS,
  },
});
```

Pulumi Typescript

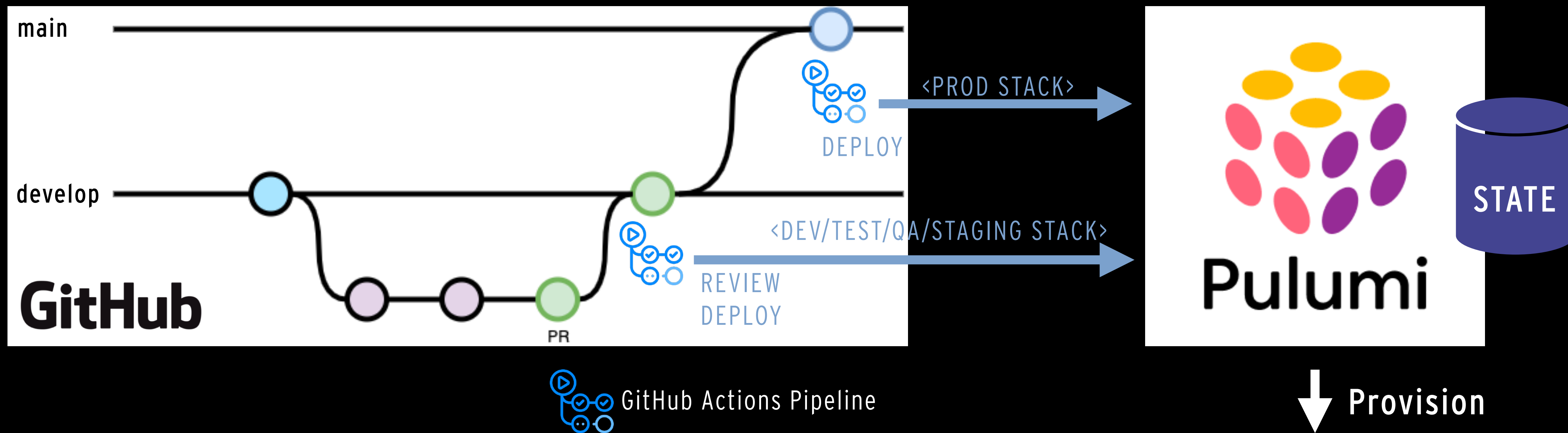
PULUMI ARCHITECTURE



AGENDA

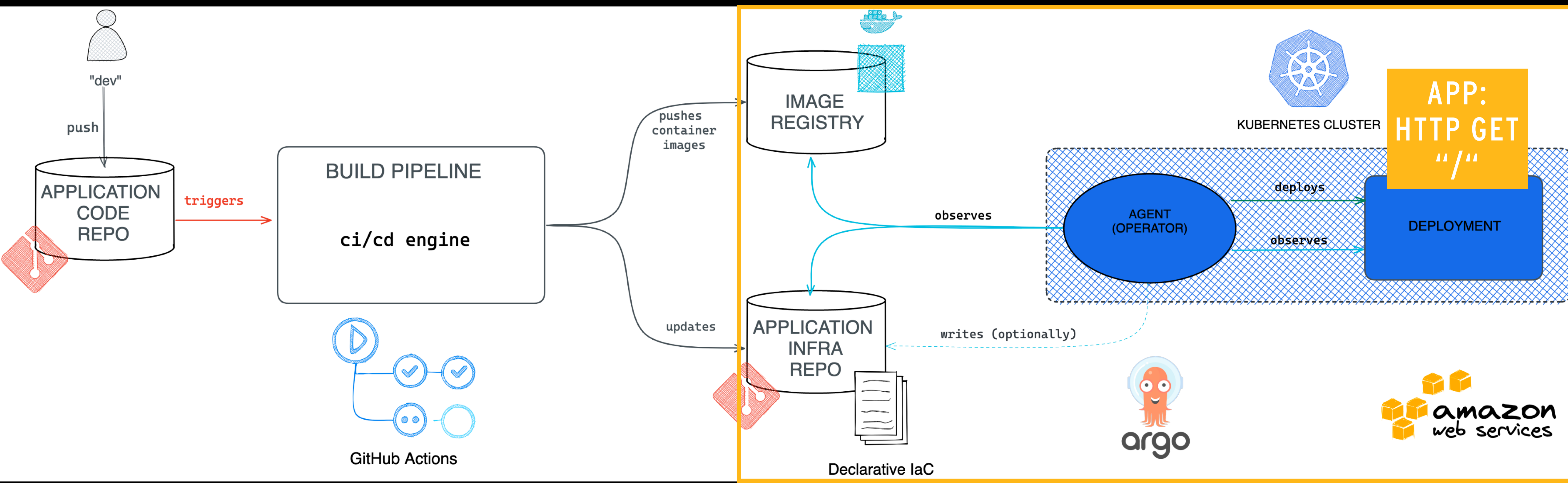
- GitOps
 - Definition
 - Implementation
- Infrastructure as Code (IaC) - Pulumi
- **Demo**
- Wrap Up

DEMO - STEP 1 - INFRA SETUP



1. Execute "Push Pipeline" to setup core infra (K8S Cluster etc.)
2. Install Argo CD for GitOps "Pull Pipeline" for a demo app

DEMO: OUTCOME FROM STEP 1 - ARGO CD DEPLOYS AND TRACKS DEMO APPLICATION



PULUMI PROGRAM

```
1 config:
2   aws:region: eu-north-1
3   aws:defaultTags:
4     tags:
5       project: "cadec23"
6       env: "staging"
7   eks-gitops:desiredClusterSize: "2"
8   eks-gitops:eksNodeInstanceType: t3.small
9   eks-gitops:maxClusterSize: "6"
10  eks-gitops:minClusterSize: "2"
11  eks-gitops:vpcNetworkCidr: 10.0.0.0/16
12  eks-gitops:isMinikube: "false"
```

./Pulumi.staging.yaml

```
8 // Grab some values from the Pulumi configuration (or use default values)
9 const config = new pulumi.Config();
10 const minClusterSize = config.getNumber("minClusterSize") || 2;
11 const maxClusterSize = config.getNumber("maxClusterSize") || 6;
12 const desiredClusterSize = config.getNumber("desiredClusterSize") || 2;
13 const eksNodeInstanceType = config.get("eksNodeInstanceType") || "t3.small";
14 // Problem : no available/free pods if choosing to too small EC2 instance,
15 // see: https://github.com/aws-labs/amazon-eks-ami/blob/master/files/eni-max-pods.txt
16 const vpcNetworkCidr = config.get("vpcNetworkCidr") || "10.0.0.0/16";
17 const isMinikube = config.requireBoolean("isMinikube");
18
19 // Create a new VPC
20 const eksVpc = new awsx.ec2.Vpc("eks-vpc", {
21   enableDnsHostnames: true,
22   cidrBlock: vpcNetworkCidr,
23 });
24
25 // Create the EKS cluster
26 const eksCluster = new eks.Cluster(`eks-cluster-${pulumi.getStack()}`, {
27   // Put the cluster in the new VPC created earlier
28   vpcId: eksVpc.vpcId,
29   // Public subnets will be used for load balancers
30   publicSubnetIds: eksVpc.publicSubnetIds,
31   // Private subnets will be used for cluster nodes
32   privateSubnetIds: eksVpc.privateSubnetIds,
33   // Change configuration values to change any of the following settings
34   instanceType: eksNodeInstanceType,
35   desiredCapacity: desiredClusterSize,
36   minSize: minClusterSize,
37   maxSize: maxClusterSize,
38   // Do not give the worker nodes public IP addresses
39   nodeAssociatePublicIpAddress: false,
40   version: "1.24",
41 });
```

./Index.ts

```
68 // Export some values for use elsewhere
69 export const kubeconfig = pulumi.secret(eksCluster.kubeconfig); // K8S credentials
70 export const vpcId = eksVpc.vpcId;
71 export const argoCDUrl = setupArgo();
```



EKS - Provision #34

Cancel workflow

Summary

Jobs

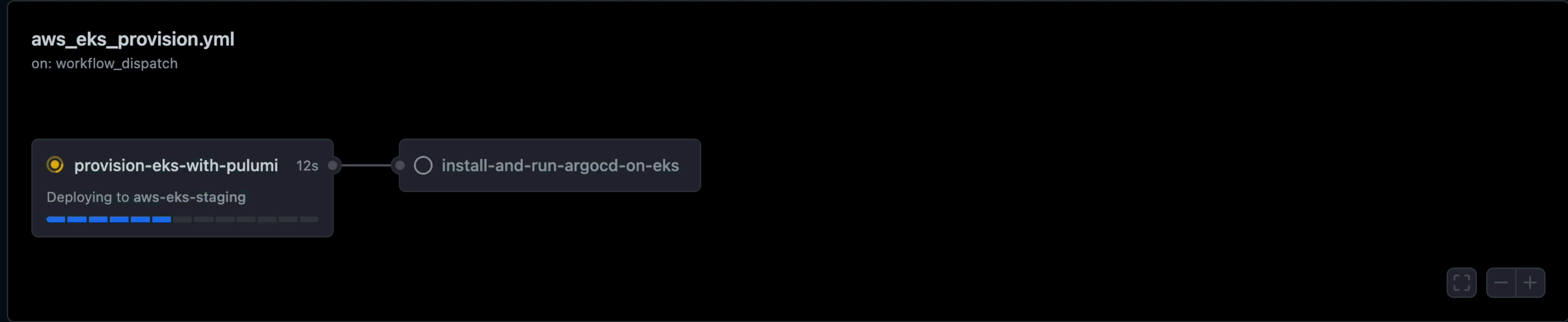
provision-eks-with-pulumi

Run details

Usage

Workflow file

Manually triggered now	Status	Total duration	Artifacts
andtell 736e043	In progress	-	-





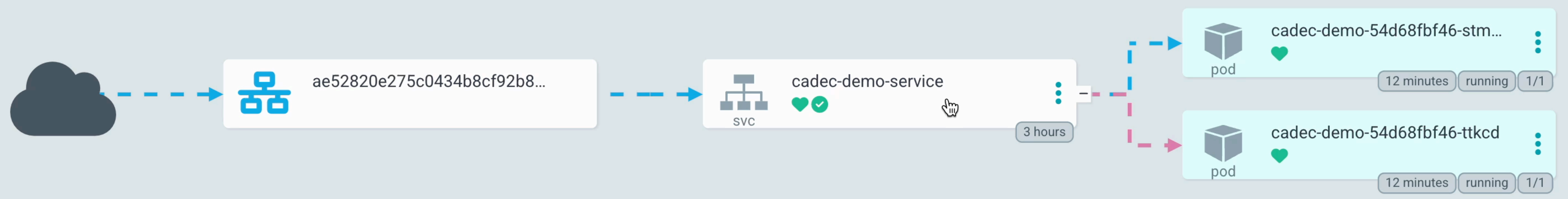
- APP DETAILS
- APP DIFF
- SYNC
- SYNC STATUS
- HISTORY AND ROLLBACK
- DELETE
- REFRESH

APP HEALTH **Healthy**

CURRENT SYNC STATUS **Synced** [MORE](#)
To **develop (6288ed8)**
Author: andtell <andtell@users.noreply.github.com> -
Comment: Bumping image version to 2d044074

LAST SYNC RESULT **Sync OK** [MORE](#)
Succeeded 12 minutes ago (Sun Jan 22 2023 19:39:03 GMT+0100)
Author: andtell <andtell@users.noreply.github.com> -
Comment: Bumping image version to [1b838802](#)

+ - 120%



| AGENDA

- GitOps
 - Definition
 - Implementation
- Infrastructure as Code (IaC) - Pulumi
- Demo
- **Wrap Up**

WRAP UP : CHALLENGES

- Branching scheme (don't do “branch per env”)
- The foundational infra setup requires a competent IaC tool to work “GitOps style” - e.g. Pulumi
- Separation of concerns in config is not always clear
- PR-workflow might introduce a bottleneck
- Culture: developer-centric (may not sit well with operations engineers)

APR
2021

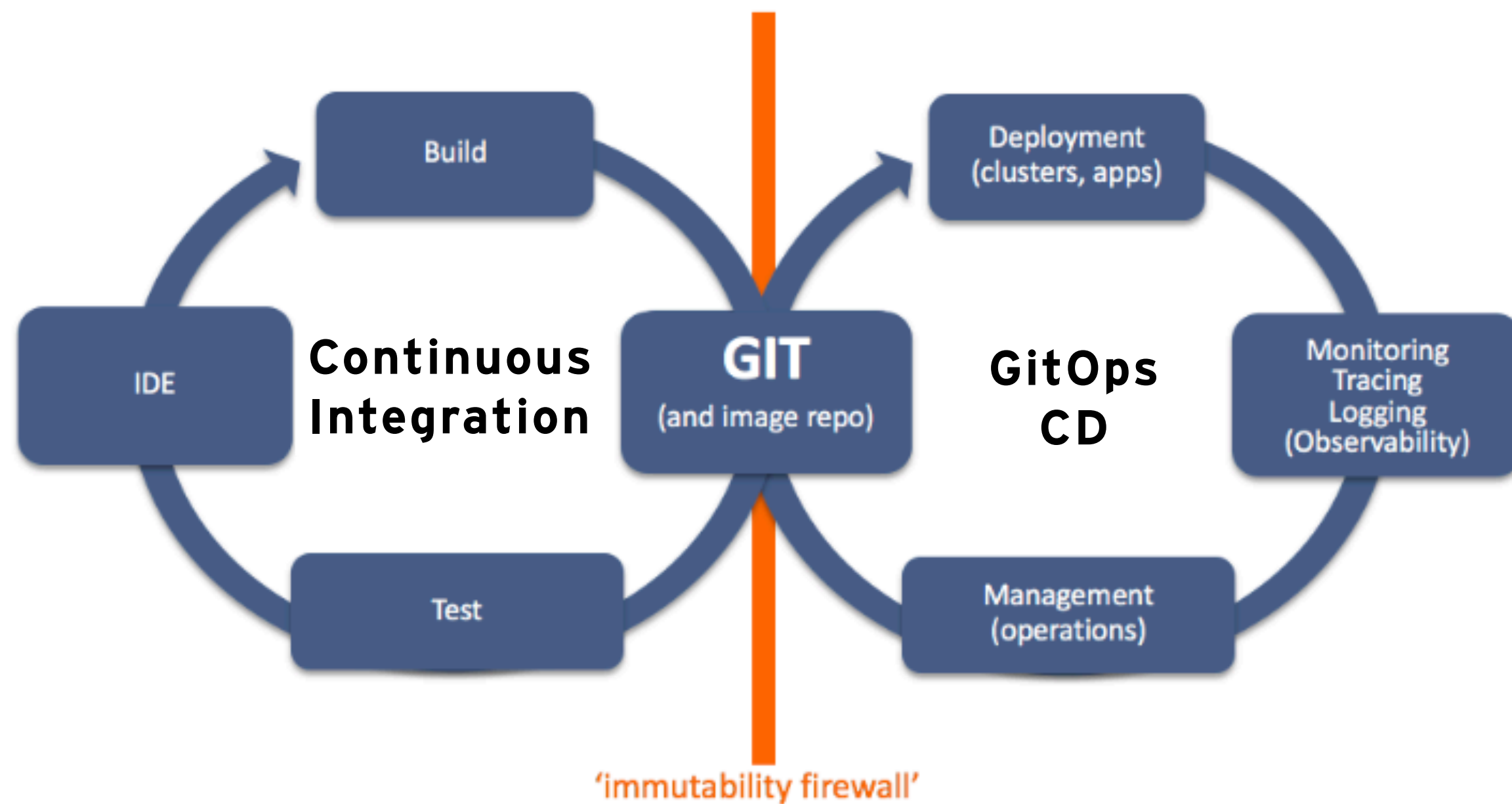
Hold ?

We suggest approaching **GitOps** with a degree of care, especially with regard to branching strategies. GitOps can be seen as a way of implementing infrastructure as code that involves continuously synchronizing and applying infrastructure code from Git into various environments. When used with a "branch per environment" infrastructure, changes are promoted from one environment to the next by merging code. While treating code as the single source of truth is clearly a sound approach, we're seeing branch per environment lead to environmental drift and eventually environment-specific configs as code merges become problematic or even stop entirely. This is very similar to what we've seen in the past with long-lived branches with GitFlow.

<https://www.thoughtworks.com/radar/techniques/gitops>



WRAP UP : WHAT DID WE GET WITH GITOPS?



Git as the single source of truth of a system's desired state

ALL intended operations are performed as git push (possibly with pull request), for all environments

ALL diffs between Git and observed state are automatically reconciled

ALL changes are observable, verifiable and auditable

Immutable Infrastructure

Automation!
No more Cowboy Ops
`$ git revert HEAD`

... at least on K8S 🙄



`$ git log --graph --abbrev-commit`

Modified from source: <https://www.weave.works/blog/what-is-gitops-really>

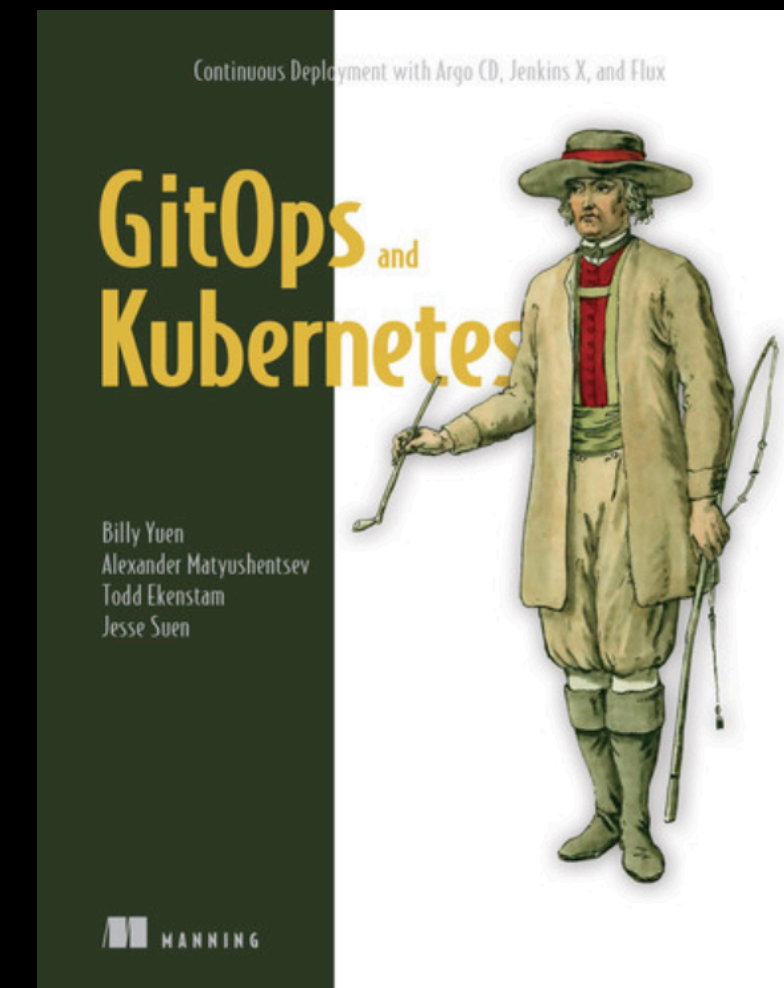
|| CLOSING THOUGHTS

- GitOps is 👍
 - Low hanging fruit - a twist on usage of already existing tools / established processes
 - Beneficial even if not applying all principles (i.e. in scenarios outside K8S with no “agent”)
 - “Infrastructure As Code done right”
- Declarative IaC + programming language is 👍
 - The code should clearly express the **desired state**, over-engineering will lead to poor visibility



WHERE TO GO FROM HERE

- GitOps for non-Kubernetes runtimes e.g. Serverless
- IaC test automation (unit-, integration-, E2E)
- Pulumi deep-dive
- Argo CD vs Flux CD
- Policy as Code
- Managing secrets
- Other providers (Azure, Google)
- ...



[HTTPS://WWW.MANNING.COM/BOOKS/GITOPS-AND-KUBERNETES](https://www.manning.com/books/gitops-and-kubernetes)

THANKS FOR LISTENING!



**kubectl
apply**



**git
push**

andreas.tell@callistaenterprise.se
<https://www.linkedin.com/in/andreastell>