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

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 : GitC

2021 (Oc

.0 published

CLOUD NATIVE
COMPUTING FOUNDATION

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

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 CLOUD NATIVE COMPUTING FOUNDATION

2021 (October) : GitOps principles v1.0.0 published

GitOps Principles

v1.0.0

2021-10-08
HTTPS://OPENGITOPS.DEV/
BLOG/1.0-ANNOUNCEMENT/

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

IaC ←

1 Declarative

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.

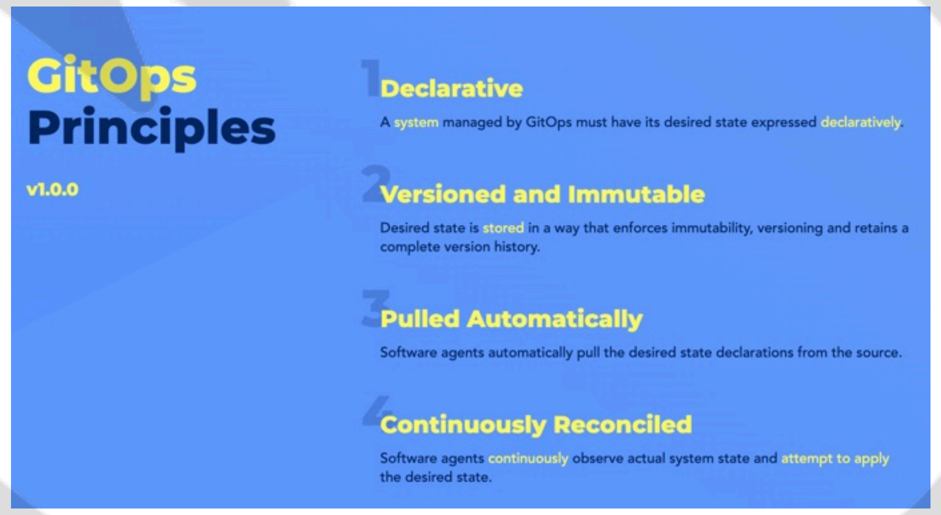
IMPLEMENTING GITOPS - PULL & PUSH PIPELINES

GitOps - “Push Pipeline”

“Core Infra”
Networking
IAM
DB
Kubernetes Cluster

GitOps - “Pull Pipeline”

kubernetes



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.

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

"devops"
push
INFRA REPO
triggers
BUILD PIPELINE
CI/CD Engine
deploys
observes
ENVIRONMENT
Networking
Database
Kubernetes Cluster
etc.
amazon web services
Google Cloud Platform
GitHub Actions
Pulumi
Declarative IaC Tool
RECONCILER LOOP
Declarative IaC State Definition

CALLISTA

IMPLEMENTATION : TRUE GITOPS - “PULL PIPELINE”

CI (Cont. Integration) ⬅️

"dev"
push
APPLICATION CODE REPO
triggers
BUILD PIPELINE
CI/CD Engine
GitHub Actions
pushes container images
IMAGE REGISTRY
RECONCILER LOOP
updates
APPLICATION INFRA REPO
Declarative IaC
observes / fetches
writes (optionally)
argo
flux
KUBERNETES CLUSTER
AGENT (OPERATOR)
deploys
observes
DEPLOYMENT

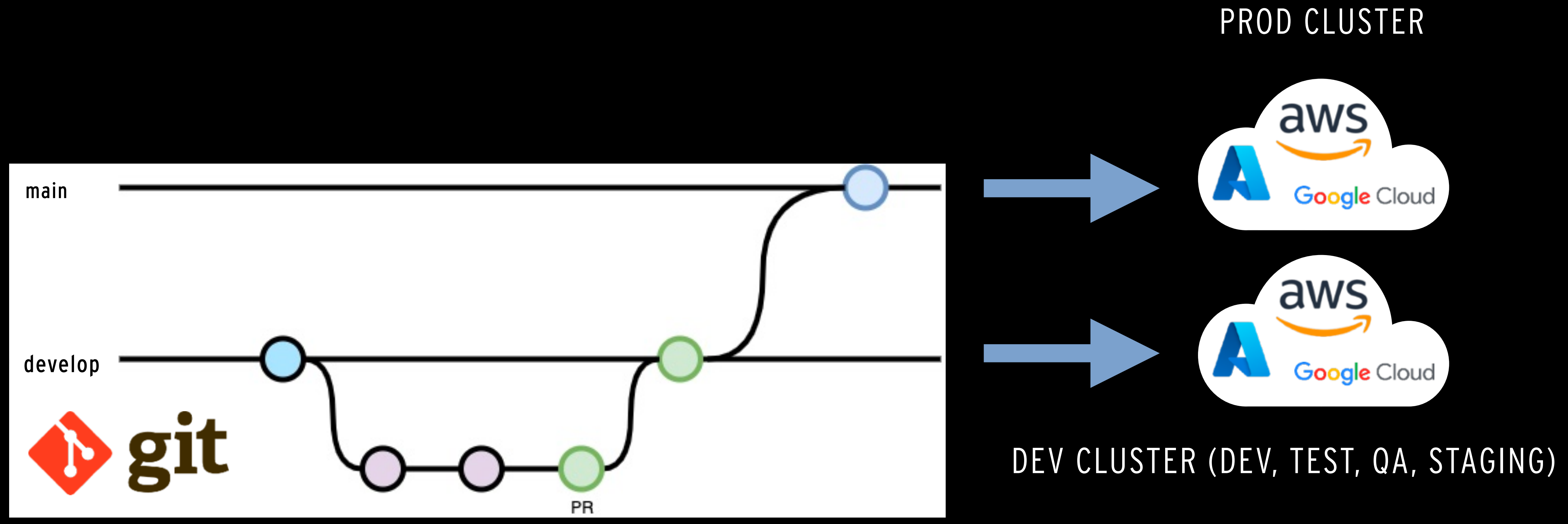
➡️ CD (Cont. Delivery)

Purpose: Application Cont. Deployment

CALLISTA

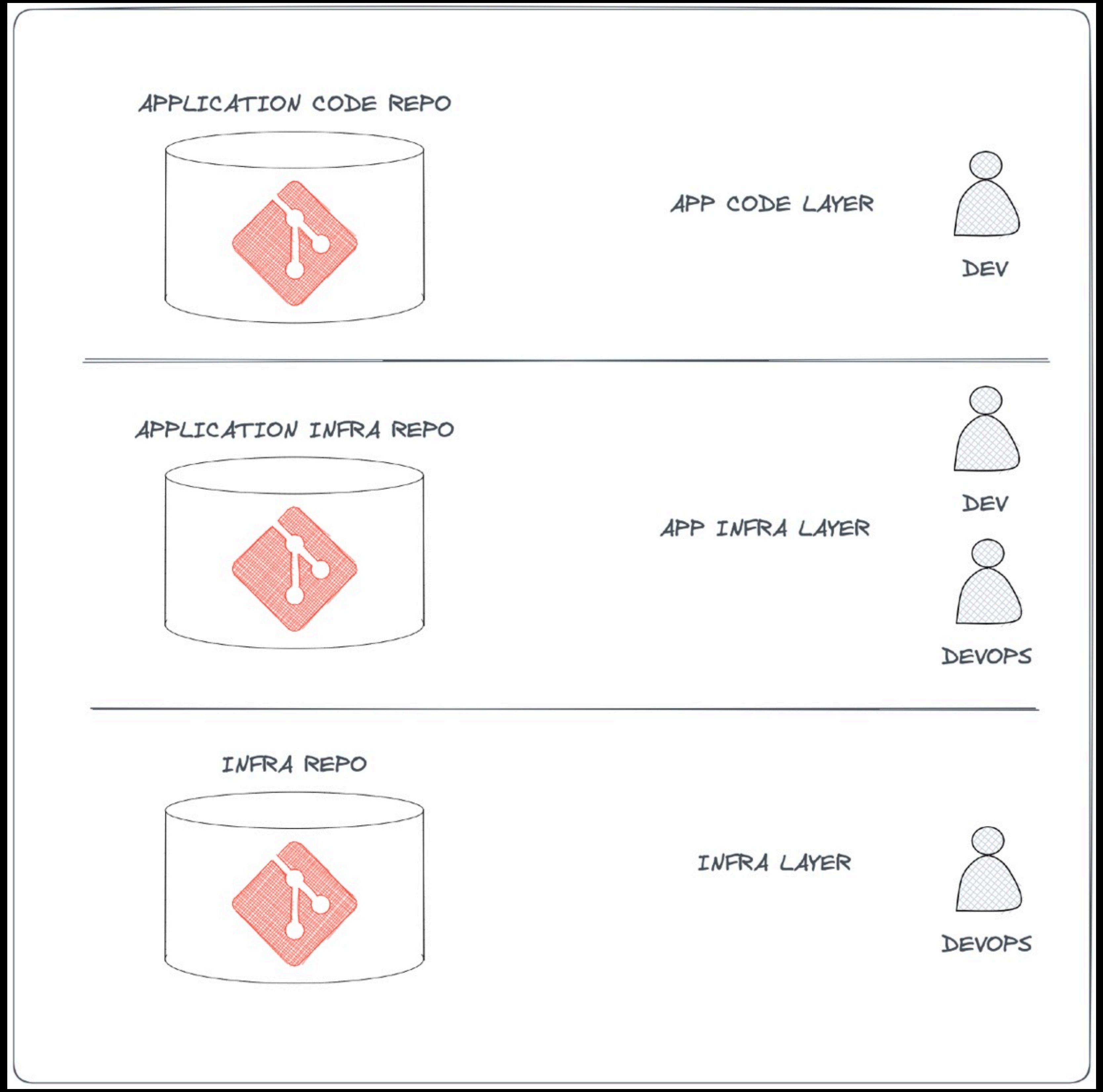
IMPLEMENTING GITOPS - "OPS BY PULL REQUEST"

- 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

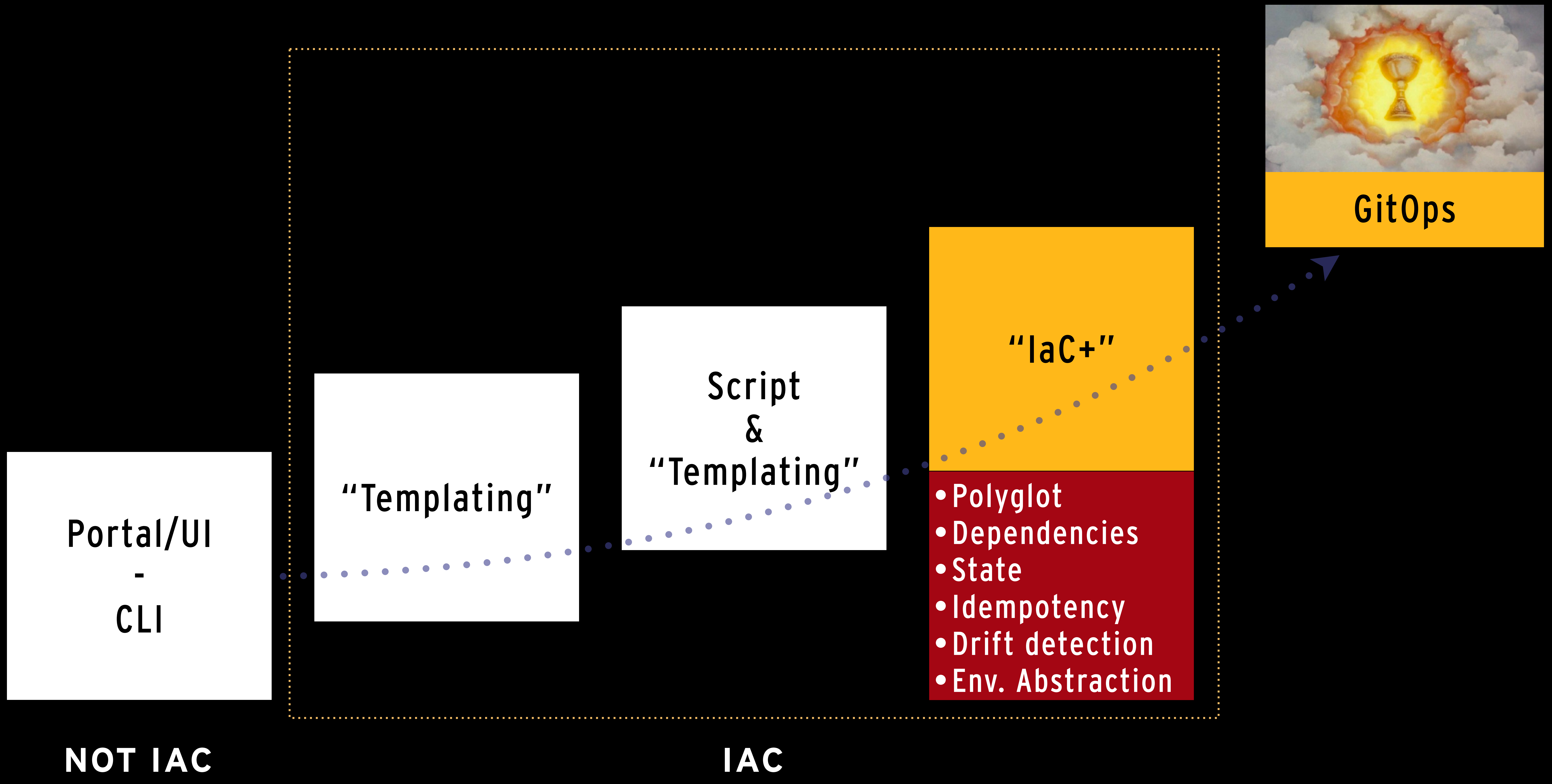
AGENDA

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

IAC TOOLS FOR CLOUD

Single cloud / Vendor Proprietary	Multi cloud / Multi purpose
AWS CloudFormation AWS CDK	Pulumi Terraform
Azure Resource Manager (ARM) Azure Bicep	Crossplane
Google Cloud Deployment Manager	Ansible
. . .	. . .

PROVISIONING CLOUD INFRA - A STEPWISE PROGRESSION

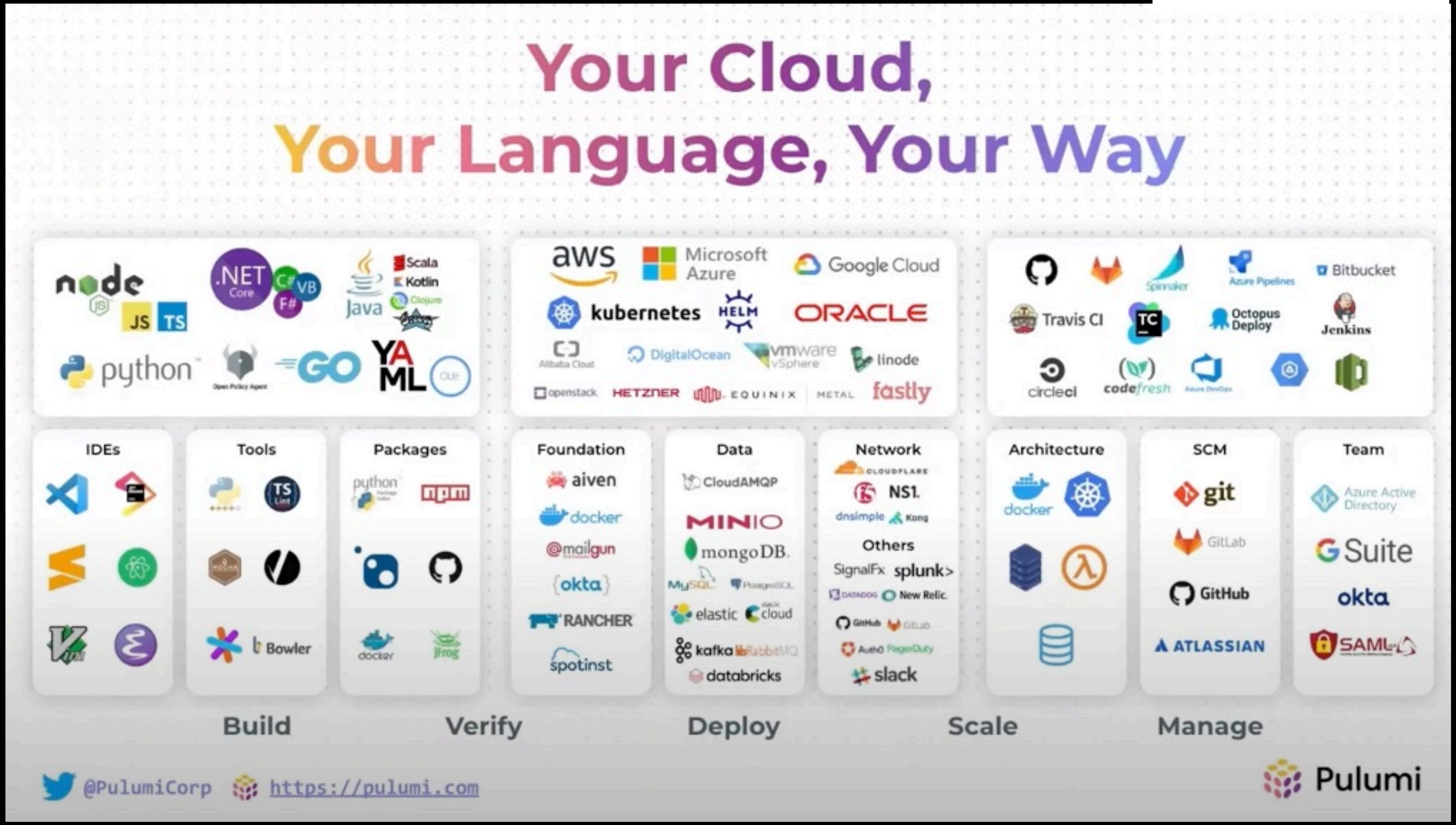


CALLISTA

PULUMI - INTRO

- Infrastructure as Code tool for creating, deploying, and managing infrastructure
  - “modern” & “traditional”
- Supports 70+ providers
- Open source
- Used by
  - Atlassian
  - Meta
  - Merzedes-Benz
  - …

v1.0 : 2019-09

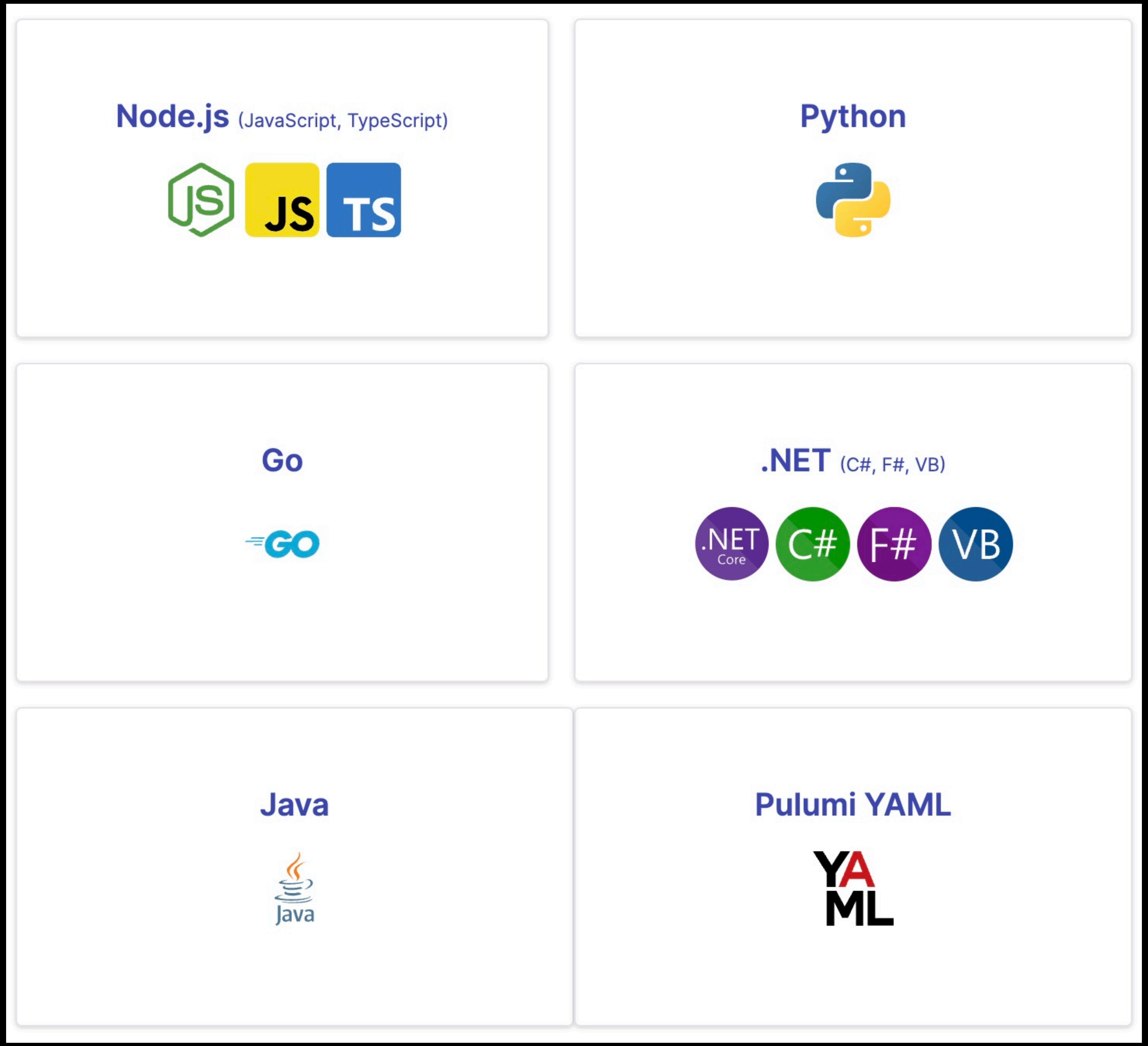


Your Cloud,
Your Language, Your Way
node JS TS .NET Core C# VB F# Java Scala Kotlin Clojure python Open Policy Agent GO YAML CUE
aws Microsoft Azure Google Cloud kubernetes HELM ORACLE Alibaba Cloud DigitalOcean vmware vSphere linode openstack HETZNER EQUINIX METAL fastly
Bitbucket Spinnaker Azure Pipelines Travis CI TC Octopus Deploy Jenkins circleci codefresh Azure DevOps
IDEs
Tools
Bowler
Packages
python npm docker JFrog
Foundation
aiven docker mailgun okta RANCHER spotinst
Data
CloudAMQP MINIO mongoDB MySQL PostgreSQL elastic cloud kafka RabbitMQ databricks
Network
CLOUDFLARE NS1 dnsimple Kong Others SignalFx splunk> DATADOG New Relic GitHub GitLab Auth0 PagerDuty slack
Architecture
docker
SCM
git GitLab GitHub ATLASSIAN
Team
Azure Active Directory G Suite okta SAML
Build
Verify
Deploy
Scale
Manage
@PulumiCorp https://pulumi.com
Pulumi

CALLISTA

PULUMI GOODNESS

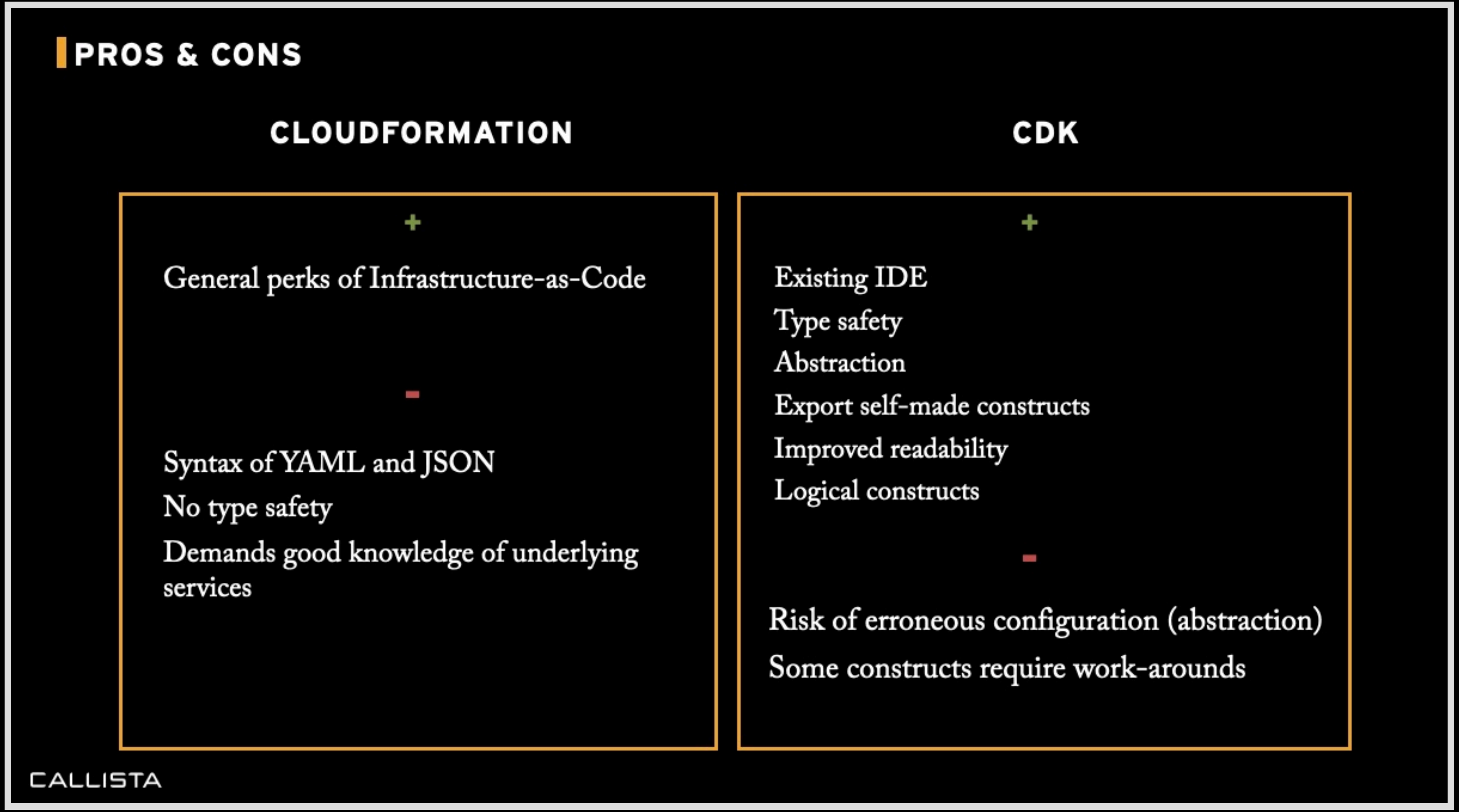
- Multi-cloud
- “Infrastructure as Software”
- “Native providers”
  - Same day access to new features
    - » https://www.pulumi.com/blog/pulumiup-native-providers/
- Has properties to support “a GitOps style workflow”:
  - Can observe infrastructure & detect drift
  - Idempotent
- Environment abstraction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

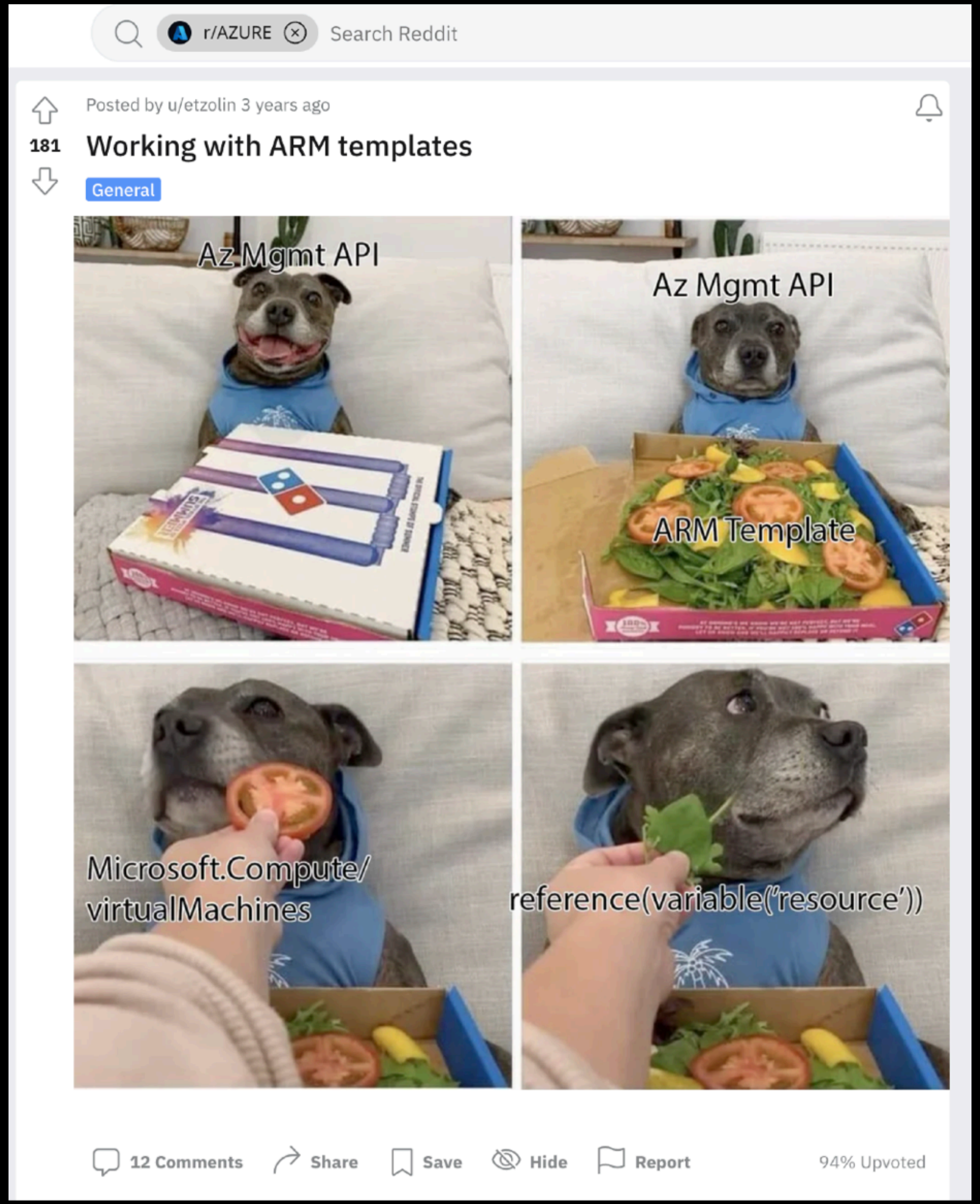
CLOUDFORMATION	CDK
+	+
General perks of Infrastructure-as-Code	Existing IDE Type safety Abstraction Export self-made constructs Improved readability Logical constructs
-	-
Syntax of YAML and JSON No type safety Demands good knowledge of underlying services	Risk of erroneous configuration (abstraction)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

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


r/AZURE Search Reddit

Posted by u/etzolin 3 years ago

181 Working with ARM templates

General

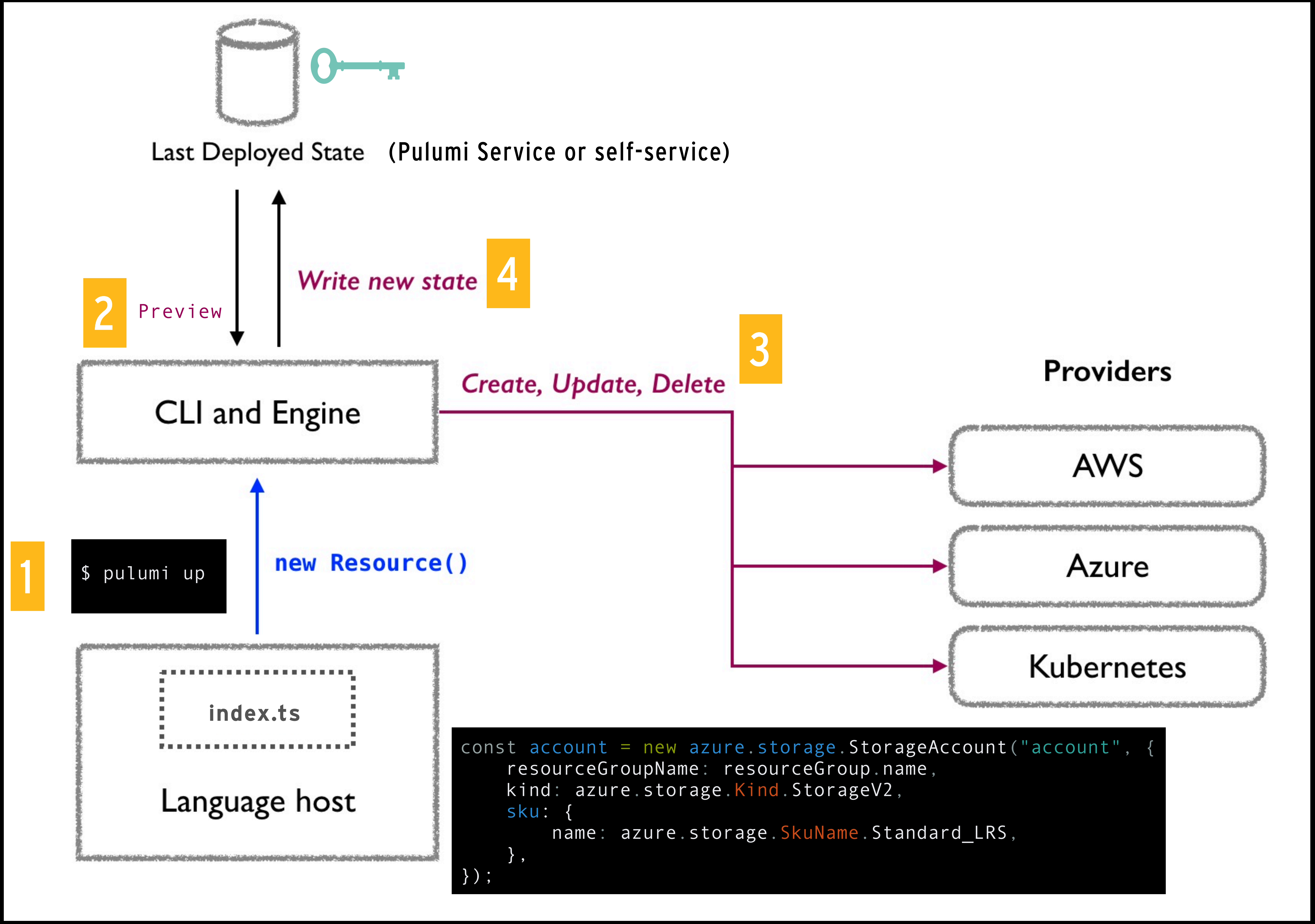
12 Comments Share Save Hide Report 94% Upvoted

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

Pulumi Typescript

CALLISTA

PULUMI ARCHITECTURE



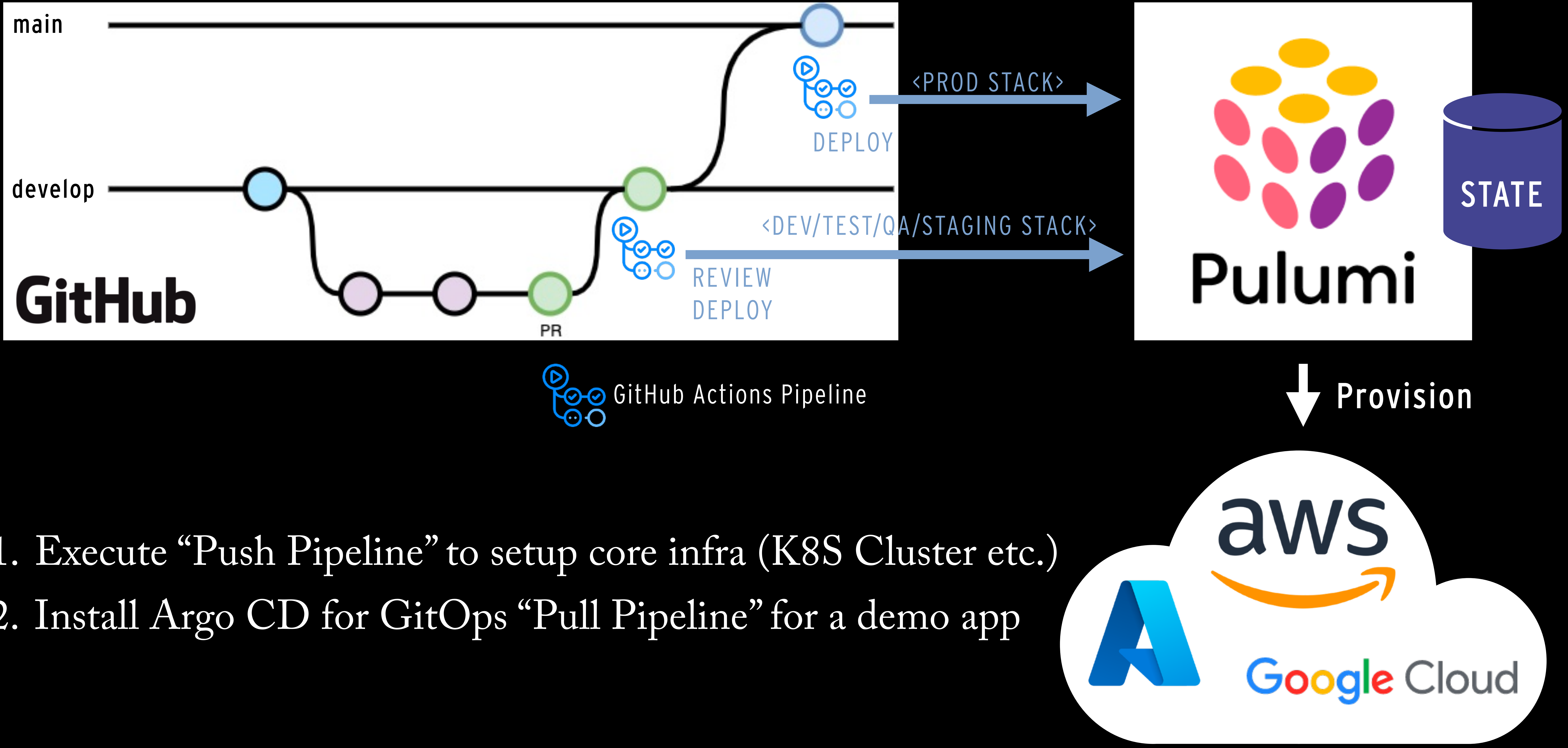
```
const account = new azure.storage.StorageAccount("account", {
    resourceGroupName: resourceGroup.name,
    kind: azure.storage.Kind.StorageV2,
    sku: {
        name: azure.storage.SkuName.Standard_LRS,
    },
});
```

CALLISTA

AGENDA

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

DEMO - STEP 1 - INFRA SETUP



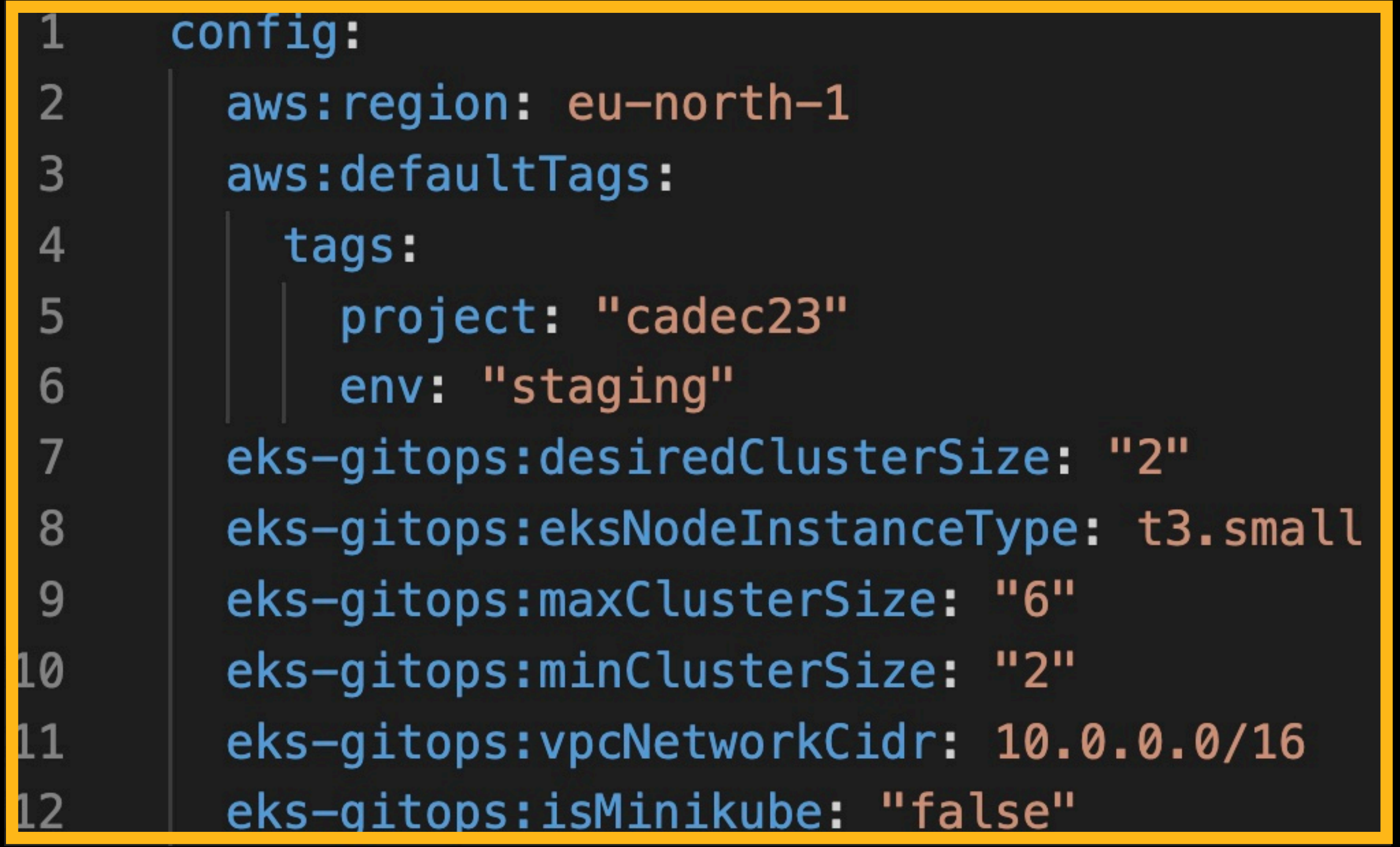
1. Execute "Push Pipeline" to setup core infra (K8S Cluster etc.)
2. Install Argo CD for GitOps "Pull Pipeline" for a demo app

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

"dev"
push
APPLICATION CODE REPO
triggers
BUILD PIPELINE
ci/cd engine
GitHub Actions
pushes container images
updates
IMAGE REGISTRY
APPLICATION INFRA REPO
Declarative IaC
observes
writes (optionally)
KUBERNETES CLUSTER
AGENT (OPERATOR)
deploys
observes
DEPLOYMENT
APP: HTTP GET "/"
argo
amazon web services

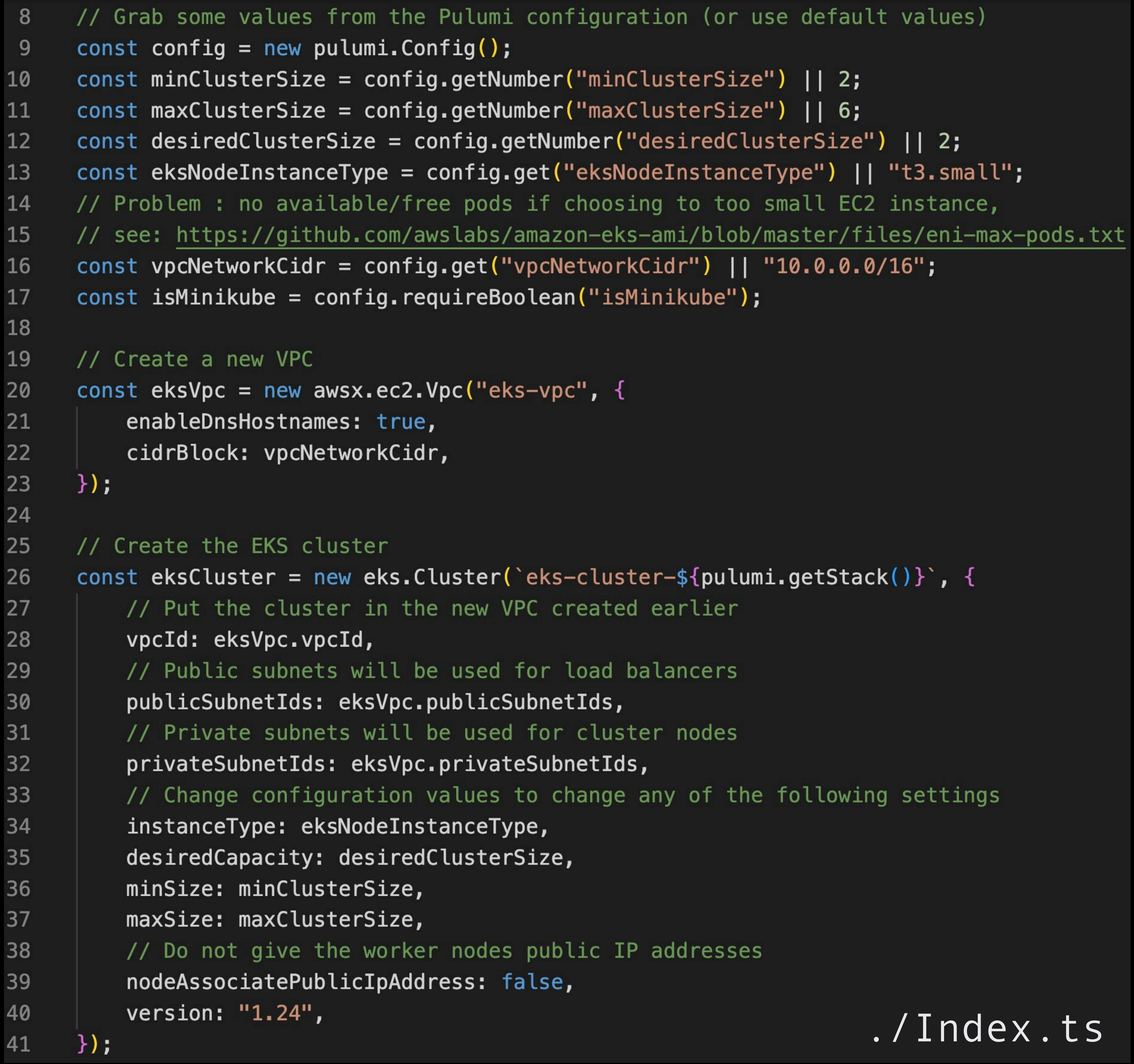
CALLISTA

PULUMI PROGRAM



```yaml
config:
  aws:region: eu-north-1
  aws:defaultTags:
    tags:
      project: "cadec23"
      env: "staging"
  eks-gitops:desiredClusterSize: "2"
  eks-gitops:eksNodeInstanceType: t3.small
  eks-gitops:maxClusterSize: "6"
  eks-gitops:minClusterSize: "2"
  eks-gitops:vpcNetworkCidr: 10.0.0.0/16
  eks-gitops:isMinikube: "false"
```

./Pulumi.staging.yaml

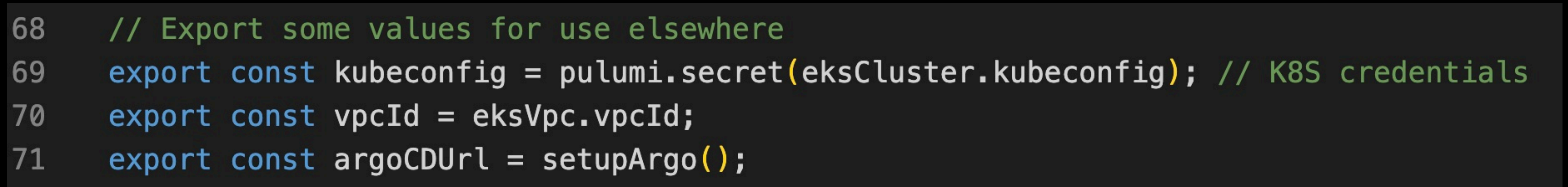


```typescript
// Grab some values from the Pulumi configuration (or use default values)
const config = new pulumi.Config();
const minClusterSize = config.getNumber("minClusterSize") || 2;
const maxClusterSize = config.getNumber("maxClusterSize") || 6;
const desiredClusterSize = config.getNumber("desiredClusterSize") || 2;
const eksNodeInstanceType = config.get("eksNodeInstanceType") || "t3.small";
// Problem : no available/free pods if choosing to too small EC2 instance,
// see: https://github.com/awslabs/amazon-eks-ami/blob/master/files/eni-max-pods.txt
const vpcNetworkCidr = config.get("vpcNetworkCidr") || "10.0.0.0/16";
const isMinikube = config.requireBoolean("isMinikube");

// Create a new VPC
const eksVpc = new awsx.ec2.Vpc("eks-vpc", {
    enableDnsHostnames: true,
    cidrBlock: vpcNetworkCidr,
});

// Create the EKS cluster
const eksCluster = new eks.Cluster(`eks-cluster-${pulumi.getStack()}`, {
    // Put the cluster in the new VPC created earlier
    vpcId: eksVpc.vpcId,
    // Public subnets will be used for load balancers
    publicSubnetIds: eksVpc.publicSubnetIds,
    // Private subnets will be used for cluster nodes
    privateSubnetIds: eksVpc.privateSubnetIds,
    // Change configuration values to change any of the following settings
    instanceType: eksNodeInstanceType,
    desiredCapacity: desiredClusterSize,
    minSize: minClusterSize,
    maxSize: maxClusterSize,
    // Do not give the worker nodes public IP addresses
    nodeAssociatePublicIpAddress: false,
    version: "1.24",
});
```

./Index.ts



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

Pulumi

CALLISTA

github.com/andtell/cloud-iac-demo-infra/actions/runs/3940026354

andtell / cloud-iac-demo-infra Public

EKS - Provision #34

Cancel workflow

Manually triggered now
andtell 736e043

Status
In progress

Total duration
–

Artifacts
–

Jobs
provision-eks-with-pulumi

aws_eks_provision.yml
on: workflow_dispatch

provision-eks-with-pulumi 12s
Deploying to aws-eks-staging

install-and-run-argocd-on-eks

Applications / argoapp-demo-app
APPLICATION DETAILS NETWORK
v2.5.7+
APP DETAILS
APP DIFF
SYNC
SYNC STATUS
HISTORY AND ROLLBACK
DELETE
REFRESH
Log out
120%

APP HEALTH
Healthy

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

LAST SYNC RESULT MORE
Sync OK
To 6288ed8
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

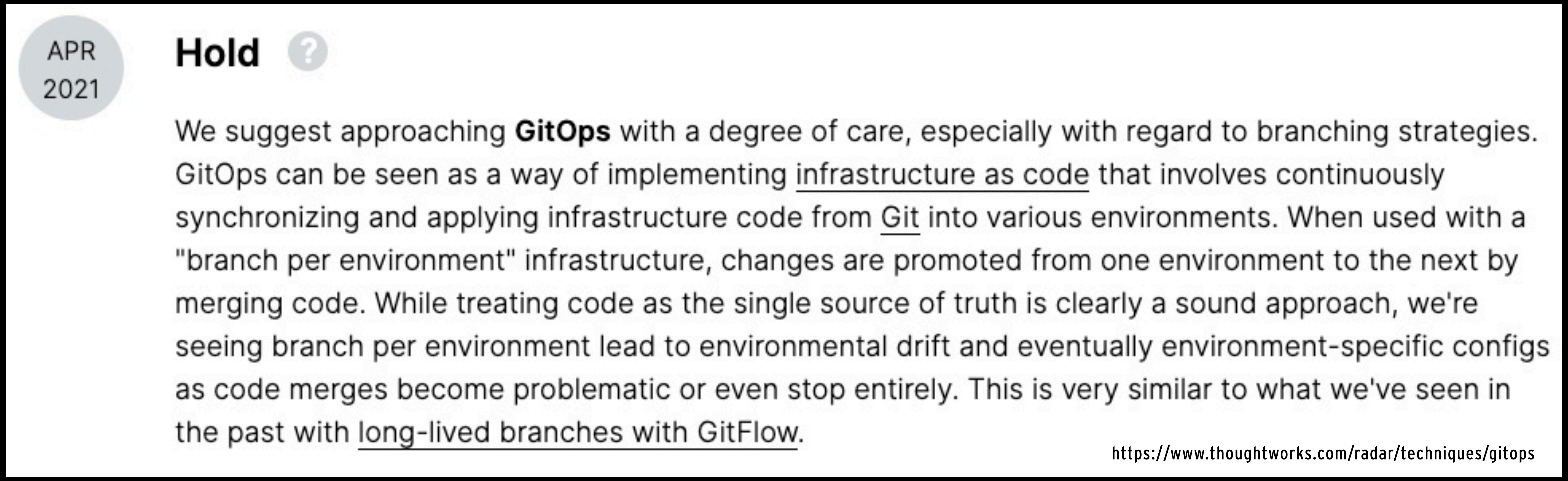
ae52820e275c0434b8cf92b8...
cadec-demo-service
svc
3 hours
cadec-demo-54d68fbf46-stm...
pod
12 minutes
running
1/1
cadec-demo-54d68fbf46-ttkcd
pod
12 minutes
running
1/1

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

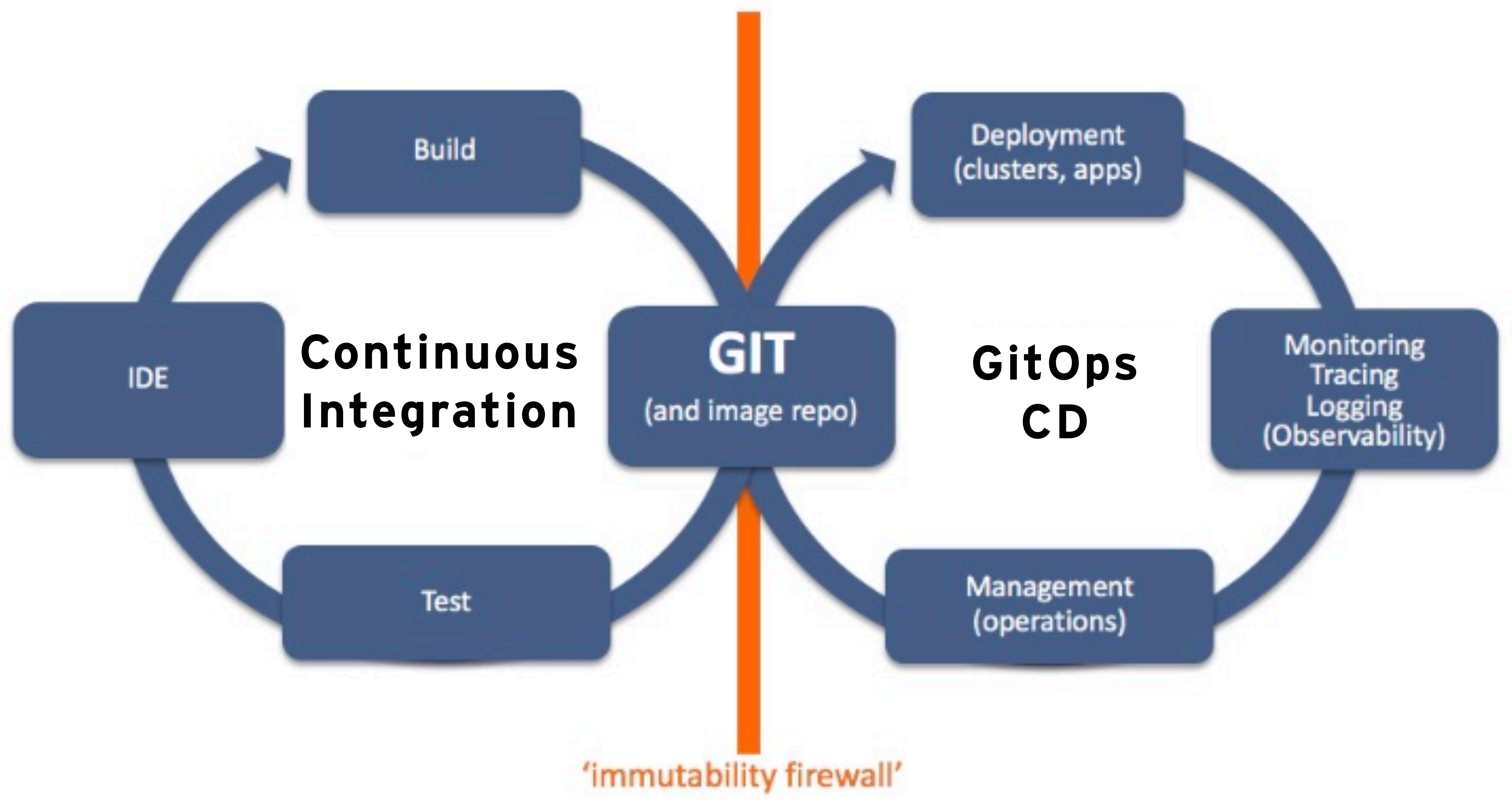
Hold

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

CALLISTA

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

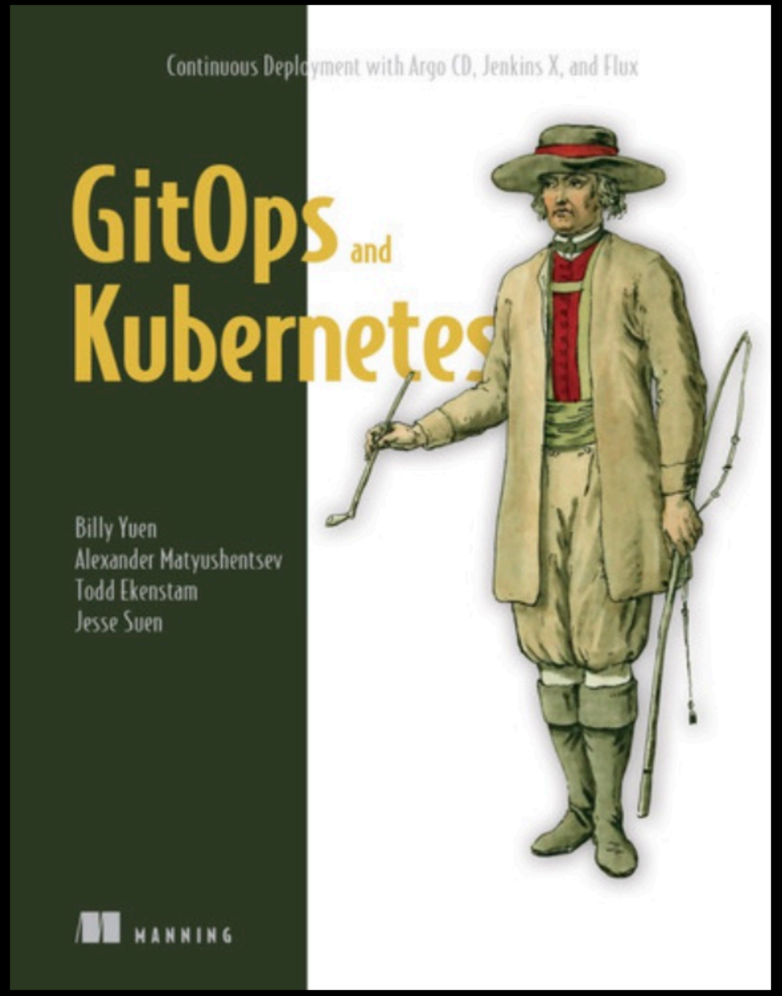
CLOSING THOUGHTS

- 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

CALLISTA

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

THANKS FOR LISTENING!

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