



# Using Argo CD and Terraform

GitOps Con EU 2025

Kostis Kapelonis

# Kostis Kapelonis



Developer Advocate (Octopus Deploy)

Argo CD, Argo Rollouts

[kostis.kapelonis@octopus.com](mailto:kostis.kapelonis@octopus.com)







# Agenda



- 1 Argo CD And Terraform
- 2 Approach 1 - DNS all the things
- 3 Approach 2 - Git provider
- 4 Approach 3 - Inject K8s resources
- 5 Approach 4 - GitOps Bridge
- 6 The future?
- 7 Conclusion



# The problem



# Terraform/OpenTofu

- Describe your infra in HCL
- Use terraform apply in CI
- Env0, Scalr, Spacelift, TF cloud, Atlantis
- Teams often abuse TF for everything (when you have a hammer...)

```
terraform {  
  required_providers {  
    aws = {  
      source  = "hashicorp/aws"  
      version = "~> 4.16"  
    }  
  }  
  
  required_version = ">= 1.2.0"  
}  
  
provider "aws" {  
  region = "us-west-2"  
}  
  
resource "aws_instance" "app_server" {  
  ami           = "ami-830c94e3"  
  instance_type = "t2.micro"  
  
  tags = {  
    Name = "ExampleAppServerInstance"  
  }  
}
```



# Status Quo



DB, Load balancer, VM, Gateway, Storage, DNS, Key/Value store, RBAC, etc



# New Challenger



Kubernetes

DB, Load balancer, VM, Gateway, Storage, DNS, Key/Value store, RBAC, etc

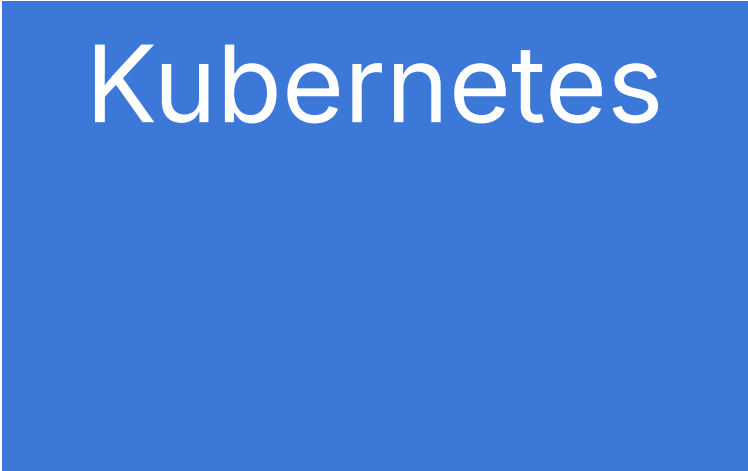
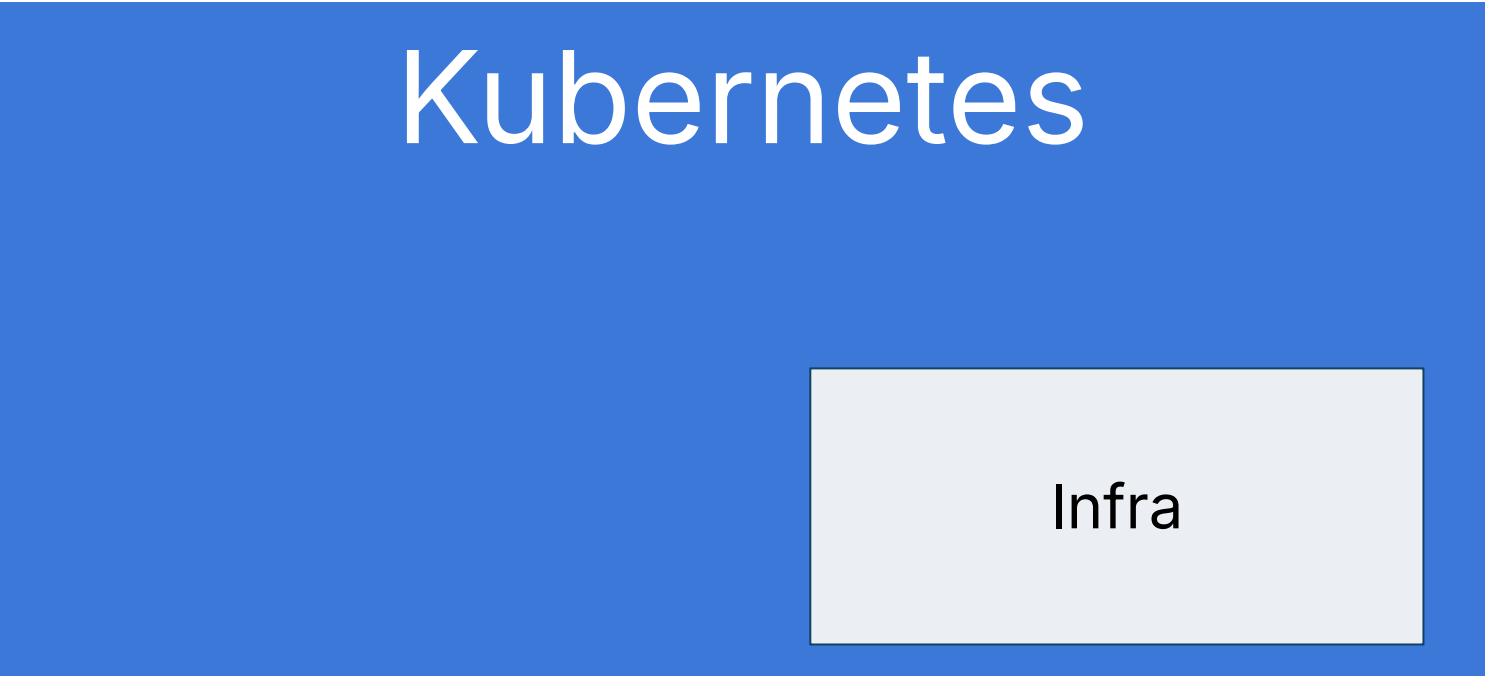


# New Challenger



DB, Load balancer, VM, Gateway, Storage, DNS, Key/Value store, RBAC, **Kubernetes**, etc

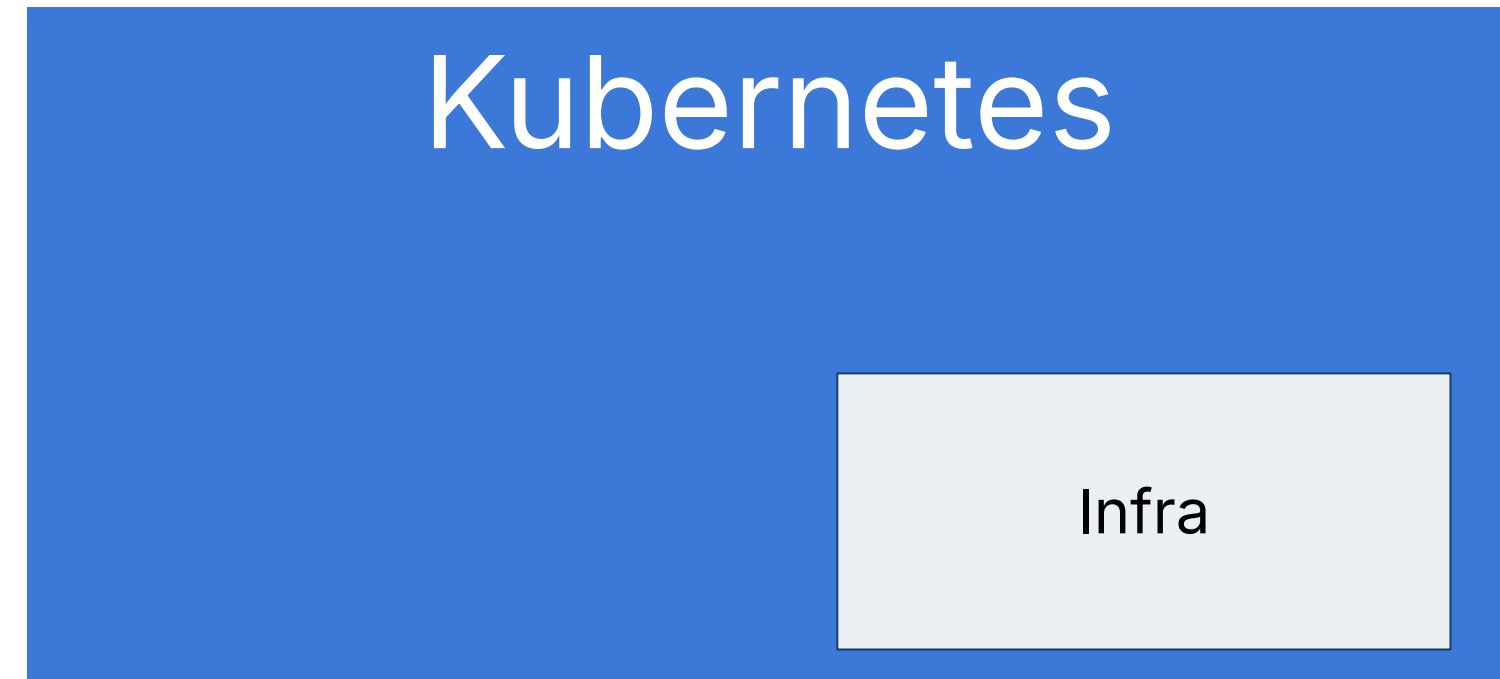
3 choices



## 3 choices



1) Most teams are here



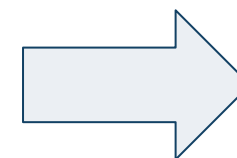
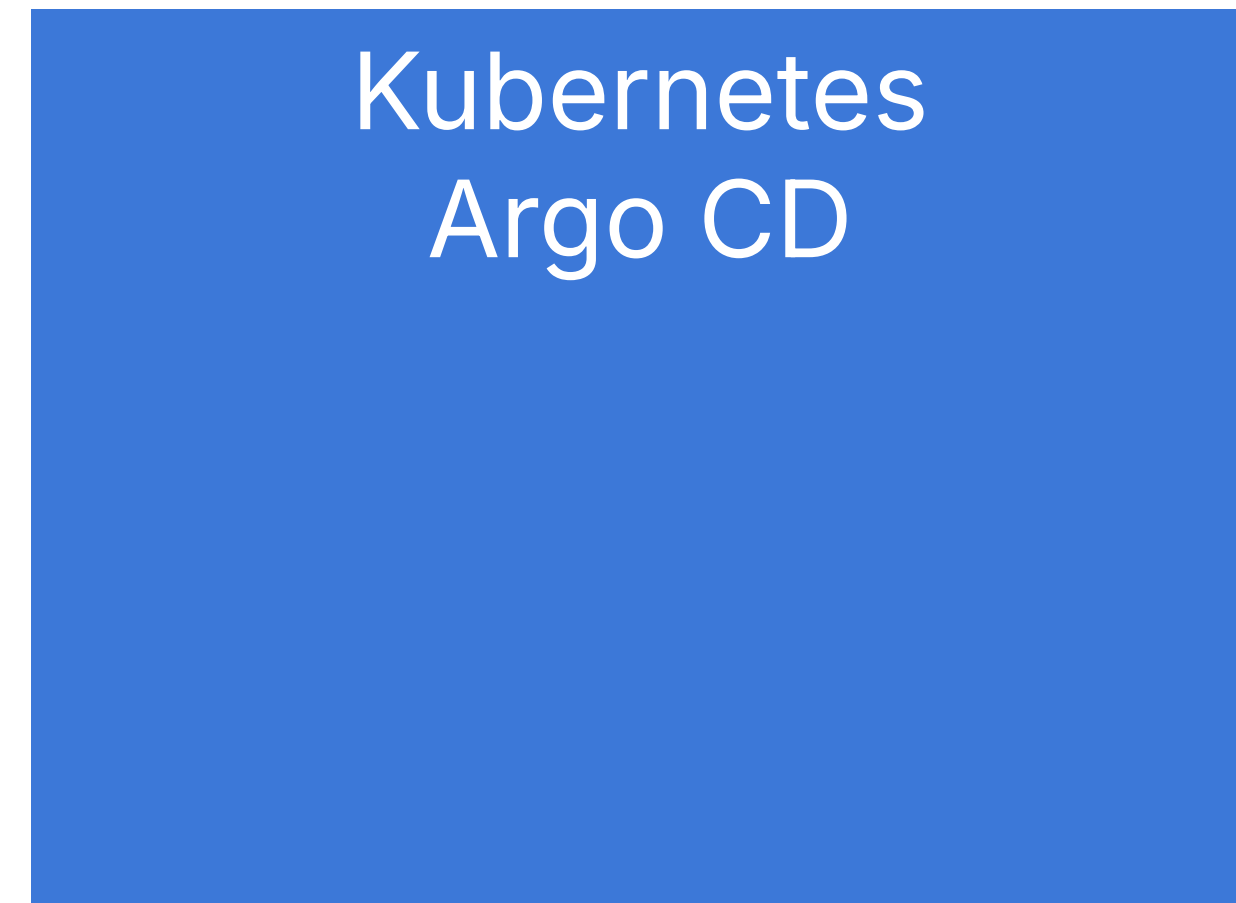
3) Bleeding edge



2) You should be here



# The problem



How to pass information?  
(secrets/settings/values/conf)



## People often ask

- My Argo CD application requires a secret that is handled by Terraform. How do I pass it over?
- My Helm chart needs a value that only Terraform knows. How do I expose it to Helm?
- My Kubernetes application needs a secret/role/cert that Terraform creates. How do I handle this scenario?





 **r/kubernetes** • 2 mo. ago  
guettli

...

## Do you manage Cloud Resources with Kubernetes or Terraform?

Do you manage Cloud Resources with Kubernetes or Terraform/OpenTofu?

Afaik there are:

- AWS Controllers for Kubernetes
- Azure Service Operator
- Google Config Connector

Does it make sense to use these CRDs instead of Terraform/OpenTofu?

What are the benefits/drawbacks?


↑ 12

↓

💬 22

🔒

🔗 Share

 **r/kubernetes** • 3 days ago  
ReverendRou

...

## ArgoCD as part of Terraform deployment?

I'm trying to figure out the best way to get my EKS cluster up and running. I've got my Terraform repo deploying my EKS cluster and VPC. Ive also got my GitOps Repo, with all of my applications and kustomize overlays.

My question is this: What is the general advice with what I should bootstrap with the Terraform and what should be kept out of it? I've been considering using a helm provider in Terraform to install a few vital components, such as metrics server, karpenter, and ArgoCD.

With ArgoCD, and Terraform, I can have them deploy the cluster and Argo using some root Applications which reference all my applications in the GitOps repo, and then it will effectively deploy the rest of my infrastructure. So having ArgoCD and a few App of Apps applications within the Terragorm

↑ 2

↓

💬 12

🔒

🔗 Share

 **r/kubernetes** • 1 yr. ago  
Tarzion


📁 ...

## Terraform and ArgoCD

Hi,

Is there any way to define the argocd\_application resource in Terraform by simply referring to an external repository containing the Helm chart rather than defining all the elements inside the Terraform manifests file?

I am trying to find a way to decouple the ArgoCD application from the Terraform code so that the same Helm chart can be used outside of Terraform as well plus it avoids the need to keep updating Terraform code for any changes inside the Helm chart configuration.

 **r/kubernetes** • 5 mo. ago  
lulzmachine 👤 Top 1% Commenter

...


## How to deal with Terraform-generated values and GitOps (ArgoCD)?

EDIT: please comment with your experiences of what you are doing, and what went well or badly for you. Thank you

Hello! We're running ArgoCD for a lot of user-land applications already, but are now looking into running infrastructure-type applications with ArgoCD as well, and are looking into how to join the worlds of terraform and Git/OpsArgoCD. Seems like there are many ways to solve the problem.

Basically: we use terraform to create our AWS-resources like IAM roles, S3 buckets, RDS databases etc. We have a "cluster\_infra\_bootstrap"-terraform module that sets up something like ~20 different resources like loki, grafana, nginx, external-secrets and others. What is the best way to transfer them to the GitOps world?

The variants we've tried so far:

 **r/kubernetes** • 1 yr. ago  
ImpressionHorror6535

...

## What do you automate with ArgoCD and terraform


Recently I started to use ArgoCD application of applications structure - I named the project "the godfather of applications" lol - and I use it for everything applications with kustomize, deploying helm charts and keeping terraform values.yaml, and cronjobs. And it works like a charm.

One thing that I couldn't do it using ArgoCD is creating secrets, like TLS certificates secret and Docker private repository Auth json for each new namespace.

So today I automated the creation of TLS, Docker secret using terraform. Terraform is connected to my laptop access. So when I want to create new namespace I create it via ArgoCD, add the terraform variables, and I manually trigger terraform script.

What do you think about it? I think there's Hashicorp Vault or something but I felt overwhelmed by their docs.


And you what do you use for automation in your kubernetes environment?

 **r/kubernetes** • 3 days ago  
Cloud--Man

...

## Argo CD Setup with Terraform on EKS Clusters

I have an EKS cluster that I use for labs, which is deployed and destroyed using Terraform. I want to configure Argo CD on this cluster, but I would like the setup to be automated using Terraform. This way, I won't have to manually configure Argo CD every time I recreate the cluster. Can anyone point me in the right direction? Thanks!

 **r/kubernetes** • 5 yr. ago  
Estanho

📁 ...

## Terraform output to kubernetes

So I'm using Terraform to build our infrastructure (GCP based so GKE, static IP etc).


Now what I wanted to do was create some kubernetes resources that depend on things created on Terraform, such as that static IP that I could use on an Ingress object via the " `kubernetes.io/ingress.global-static-ip-name` " annotation.

I was hoping maybe outputting those things to a ConfigMap from Terraform using the kubernetes provider. But apparently that's not very useful since I could only find resources using those on pod definitions. So apparently I can't use ConfigMap values in labels or annotations.

If that's really the case, what would be the best way to do this connection? I don't want to have my k8s resources defined inside Terraform because I also want to use some custom GCP resources and I don't want to have to deal with maintaining those on Terraform.

I was thinking on maybe just grabbing the Terraform outputs and using them to interpolate my kubernetes files, but that would require some (simple but still) custom tooling and I was hoping there was some standard for that process. I was hoping I could use Kustomize for that but apparently it's not supposed to be used like that.

Are there any suggestions for how to deal with that?

 **r/kubernetes** • 6 yr. ago  
kevinjqiu

📁 ...

## Anyone using terraform-helm-provider to deploy software on Kubernetes?

Hi all,

We've been using [helmfile](#) for deploying software on our Kubernetes cluster and it has been working wonderfully, except that our infrastructure provisioning was in terraform. This creates a disconnect between infrastructure provisioning and software deployment. One such disadvantage is that for infra provisioned/managed by terraform, we have to duplicate values in our helm values files. e.g., we create our kafka cluster on bare vms using terraform, and for software that requires kafka hostnames, we have to hardcode the kafka hostnames in helmfile, as opposed to using data resource in terraform to get those values dynamically out of the terraform state file.

Terraform does have a [helm provider](#), which looks promising. I'm not a fan of HCL to be honest, but if it means lower overhead and complexity, I'm willing to roll. I'm just wondering if anybody is using terraform helm provider and what your experience with it has been like? Any major caveats? From the first look of it, secret management seems to be non-existent. With helmfile, it's integrated with helm-secret pretty well.

Thanks!





# Approach 1: Use DNS



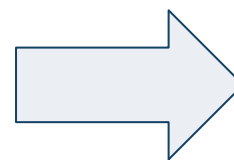
# The problem



```
app_mode = development

[paypal]
paypal_url = https://development.paypal.example.com
paypal_cert=/secrets/ssl/paypal.crt

[mysql]
db_host=prod-rds.us-east.acm.com
db_port=3386
db_user=billing
db_pass=toomanysecrets
```



Hostname/Port

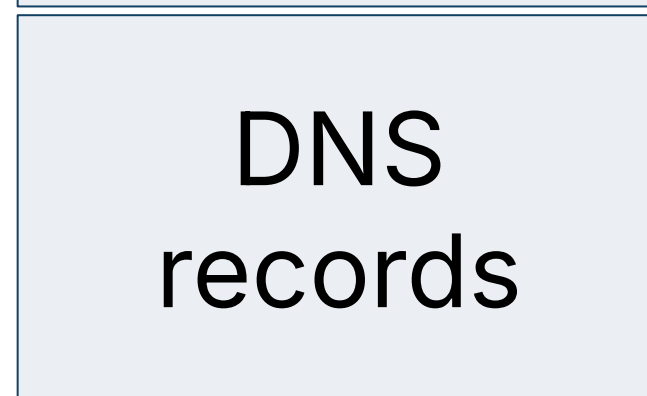


# Examples

- You want to create a Database with Terraform and use the hostname/port in a Kubernetes application
- You want to create a Loadbalancer/Ingress with Terraform and pass hostname to a Kubernetes application
- You want to create a unique name for each Kubernetes cluster and pass it to the application
- You have dependencies between applications (even across different clusters)



# The Solution



```
app_mode = development

[paypal]
paypal_url = https://development.paypal.example.com
paypal_cert=/secrets/ssl/paypal.crt

[mysql]
db_host=db.prod.acme.com
db_port=3386
db_user=billing
db_pass=toomanysecrets
```

[db.prod.acme.com](https://db.prod.acme.com)  
[db.stg.acme.com](https://db.stg.acme.com)  
[db.qa.acme.com](https://db.qa.acme.com)



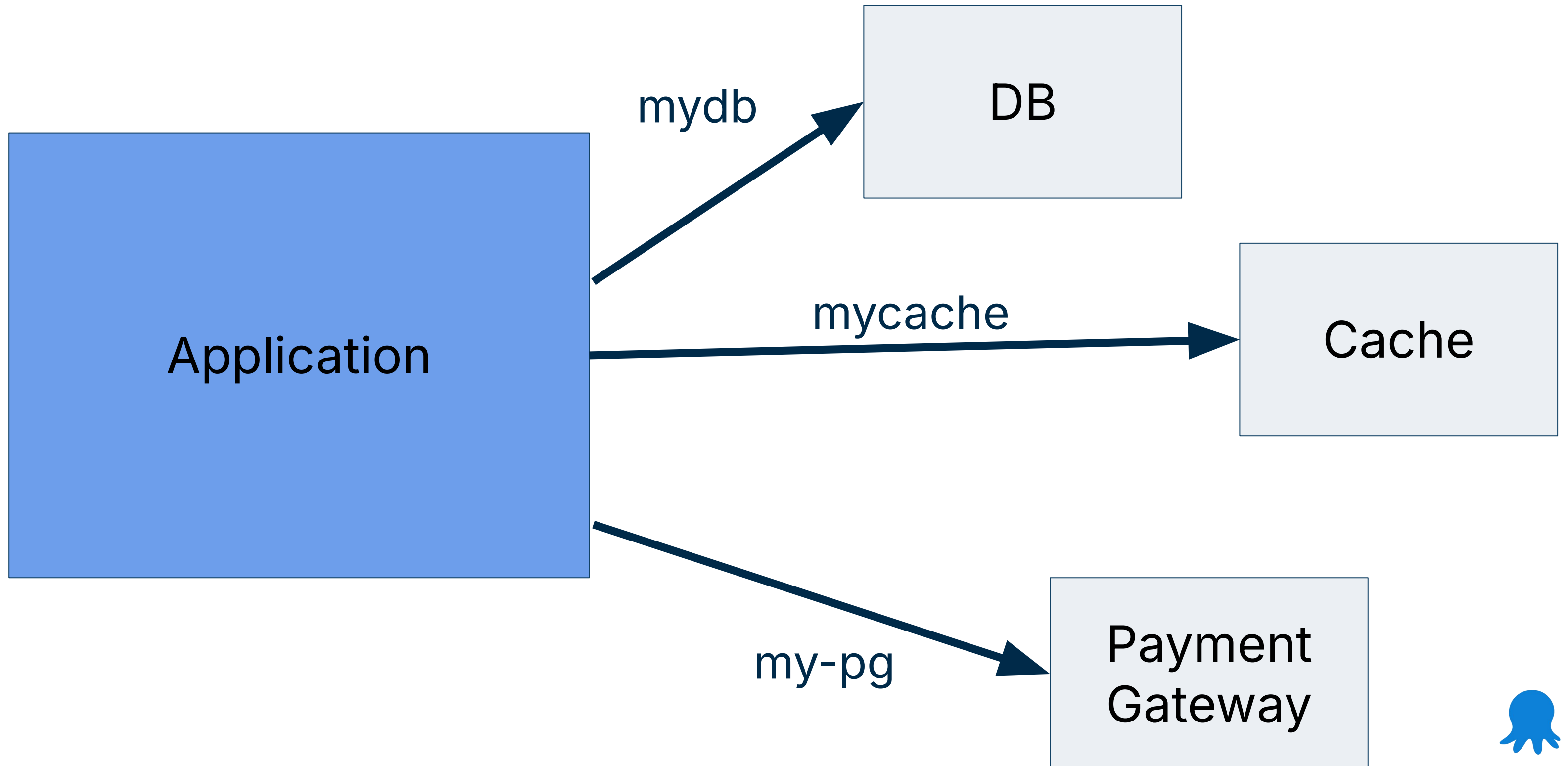


# General application guidelines

- Applications should have external conf (instead of looking on their own for configuration) - <https://12factor.net/config>
- Applications shouldn't know they run inside K8s or care about terraform
- Minimize external dependencies only to the absolute essentials (db, cache, queue, backend)



# Apps should know nothing about Terraform or K8s



# What does “mydb” hostname resolve to..

- **mydb** → Database running in the Same cluster using a DB operator
- **mydb** → ExternalName to another K8s cluster
- **mydb** → RDS instance with global Route53 record
- **mydb** → /etc/hosts for legacy db
- **mydb** → External Name → Route53 record → active/passive DB in other regions

The application does not care which method is used!





# Work **WITH** your Developers





**POD**

**Application**

**Internal  
Pod info**

**Get info**

**Vault**

**Consul**

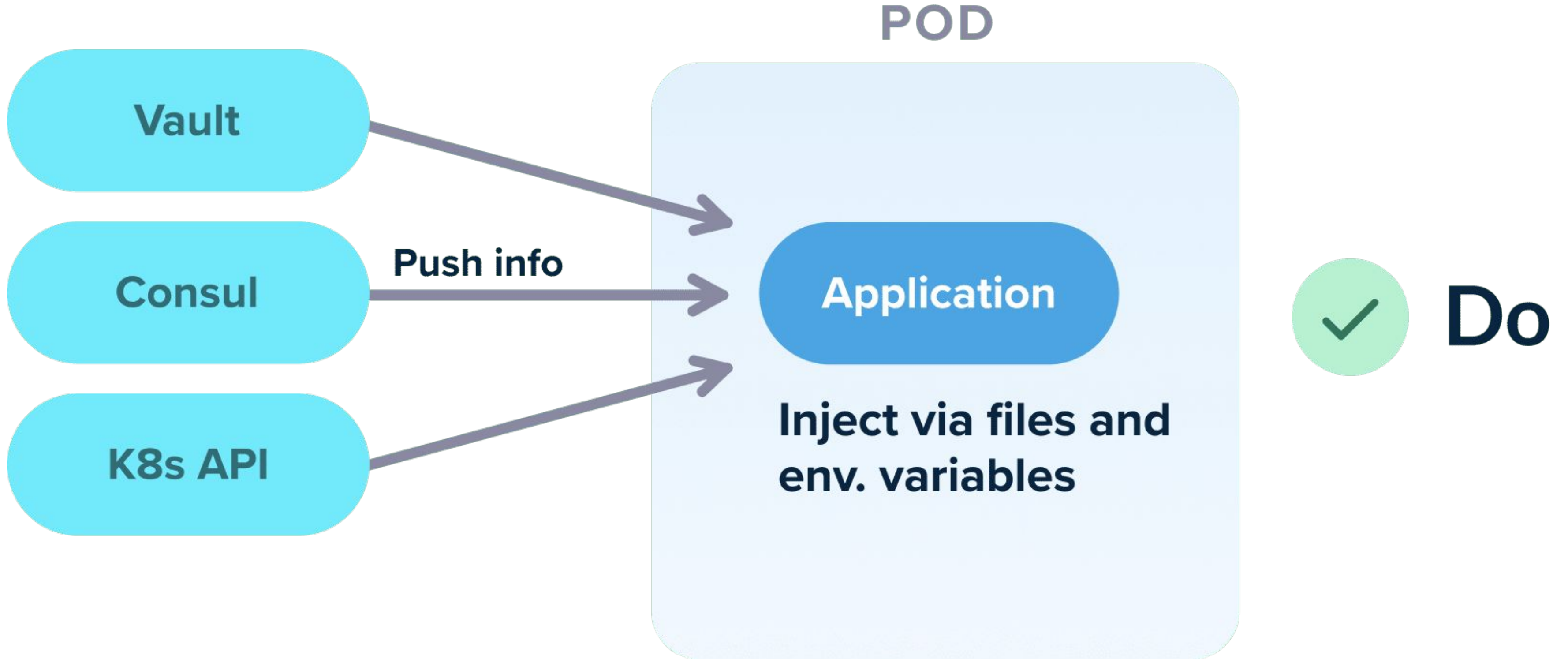
**K8s API**



**Don't**







# Approach 1 - Use DNS or other abstractions

- ++ Super simple - **nothing** to transfer from Terraform to K8s
  - + Works great for developers - easy local debugging
  - ✗ You have no single source of truth (configuration is in two places)
  - ✗ Works mostly with hostnames and other networking configuration
  - ✗ You still need to account for secrets
  - ✗ Application might need source code changes




# **Approach 2: Pass info via Git**



Pass via Git



Providers / integrations / github / Version 6.6.0 ▾ Latest Version

github  Overview Documentati

GITHUB DOCUMENTATION

Filter

github provider

Resources

- github\_actions\_environment\_secret
- github\_actions\_environment\_variable
- github\_actions\_organization\_oidc\_subject\_claim\_customization\_template


## github\_repository

This resource allows you to create and manage repositories within your GitHub organization or personal account.

**Note**

When used with GitHub App authentication, even GET requests must have the `contents:write` permission. Without it, the following arguments will be ignored, leading to unexpected behavior and confusing diffs: `allow_merge_commit`, `allow_squash_merge`, `allow_rebase_merge`, `merge_commit_title`, `merge_commit_message`, `squash_merge_commit_title` and `squash_merge_commit_message`.

Providers / integrations / github / Version 6.6.0 ▾ Latest Version

github  Overview Documenta

GITHUB DOCUMENTATION

Filter

- project
- github\_organization\_ruleset
- github\_organization\_security\_manager

## github\_repository\_file

This resource allows you to create and manage files within a GitHub repository.

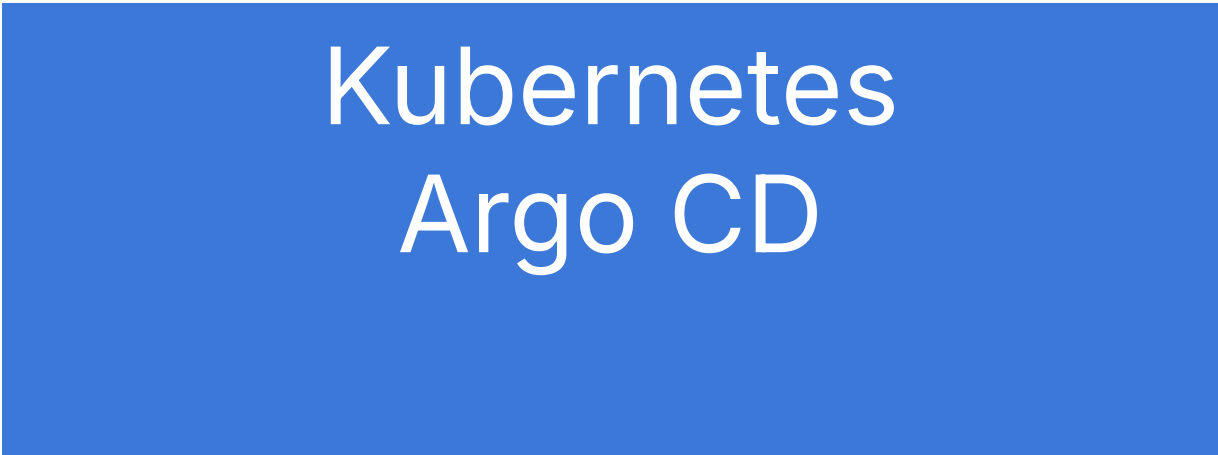
### Example Usage

<https://registry.terraform.io/providers/integrations/github/latest/docs>





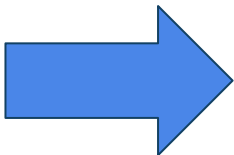
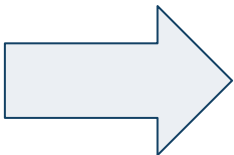
# Pass via Git



Create repo with K8s

Manifests

Create file (configmap)



Sync files (created by  
Terraform)



# Pass via Git

```
apiVersion: v1
kind: ServiceAccount
metadata:
  annotations:
    azure.workload.identity/client-id: "5cee0ff9-0208-4555-93b8-b37eb6f239a9"
  name: "workload-identity-sa"
  namespace: "default"
```

Write Service Account file   **git**  Sync Service Account file



## Approach 2 - Pass info via Git

- + Very flexible - can pass any kind of configuration
- + Single source of truth
- + Argo CD is isolated from what Terraform does
- ✗ More moving parts, you also need guardrails
- ✗ You still need to account for secrets (can't commit them to Git)
- ✗ Might have performance/security issues and bottlenecks



# **Approach 3: Inject K8s resources**



# Kubernetes Provider

The Kubernetes (K8S) provider is used to interact with the resources supported by Kubernetes. The provider needs to be configured with the proper credentials before it can be used.

Use the navigation to the left to read about the available resources.

## Example Usage

```
provider "kubernetes" {  
  config_path    = "~/.kube/config"  
  config_context = "my-context"  
}  
  
resource "kubernetes_namespace" "example" {  
  metadata {  
    name = "my-first-namespace"  
  }  
}
```

Copy

## Example Usage - Chart Repository

```
resource "helm_release" "example" {  
  name      = "my-redis-release"  
  repository = "https://charts.bitnami.com/bitnami"  
  chart     = "redis"  
  version   = "6.0.1"  
  
  set = [  
    {  
      name = "cluster.enabled"  
      value = "true"  
    },  
    {  
      name = "metrics.enabled"  
      value = "true"  
    }  
  ]  
}
```

Copy

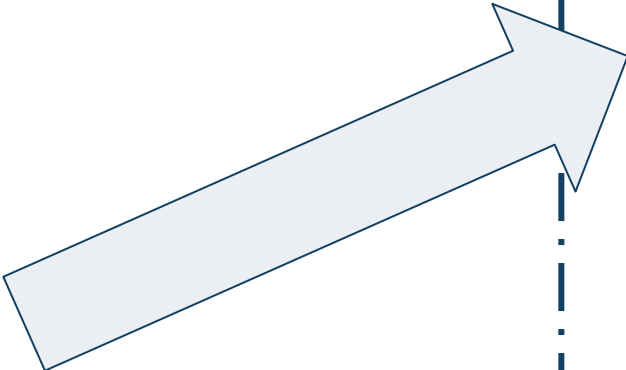
<https://registry.terraform.io/providers/hashicorp/kubernetes/latest/docs>



# Inject outside of Git



Inject Resource with  
Helm/K8 provider



ConfigMap



Deployment

Service



kubernetes



git





## Approach 3 - Inject resources with K8s/Helm provider

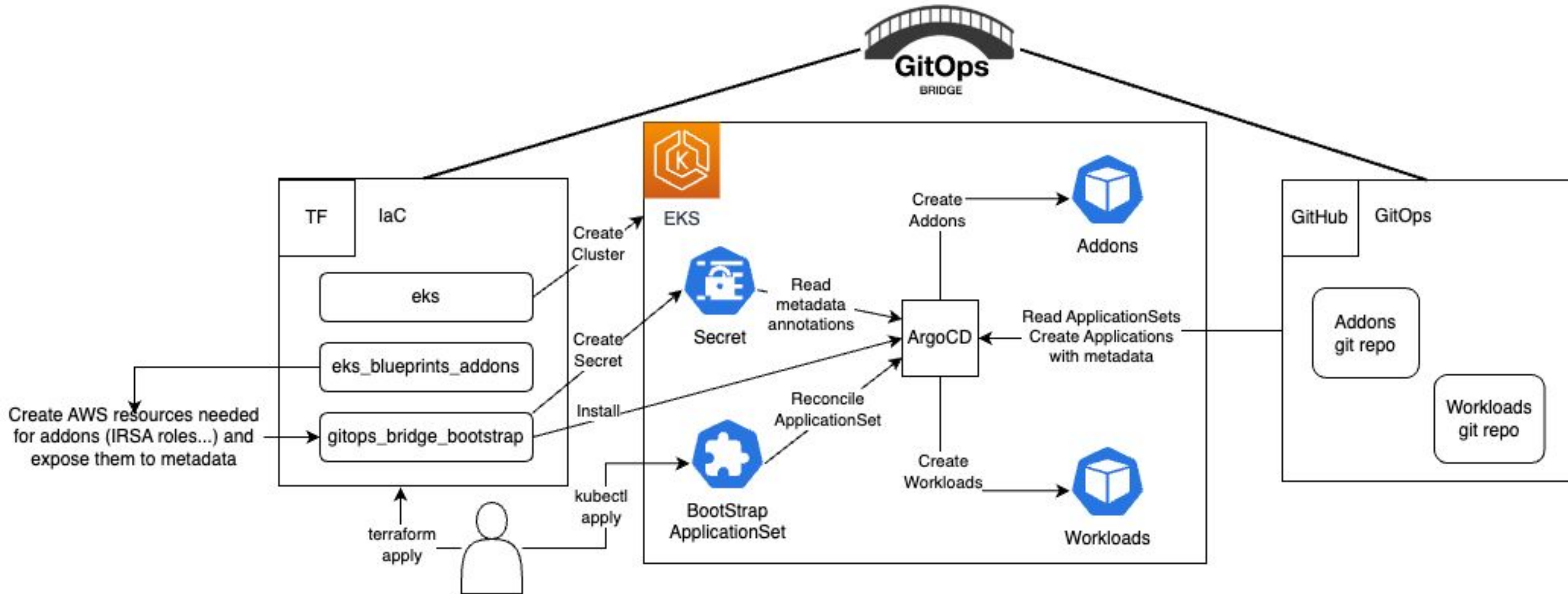
- + + Very flexible - can pass any kind of configuration
- + Can pass secrets directly in the cluster
- ✗ Argo CD controls only part of the application
- ✗ Not GitOps, and no single source of truth
- ✗ Complex Dependency ordering
- ✗ Terraform was never designed for application deployments





# Approach 4: GitOps Bridge





<https://github.com/gitops-bridge-dev/gitops-bridge>



# Pass via Cluster labels

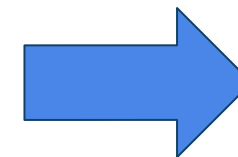
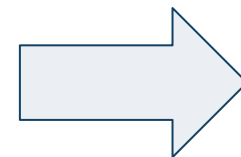


Create K8s cluster and other  
resources

Cluster metadata



**kubernetes**



Argo CD

Sync ApplicationSets and  
get information from Cluster  
labels



```
apiVersion: argoproj.io/v1alpha1
kind: ApplicationSet
metadata:
  name: guestbook
  namespace: argocd
spec:
  goTemplate: true
  goTemplateOptions: ["missingkey=error"]
  generators:
  - clusters: {} # Automatically use all clusters defined within Argo CD
  template:
    metadata:
      name: '{{.name}}-guestbook' # 'name' field of the Secret
    spec:
      project: "my-project"
      source:
        repoURL: https://github.com/argoproj/argocd-example-apps/
        targetRevision: HEAD
        path: guestbook
      destination:
        server: '{{.server}}' # 'server' field of the secret
        namespace: guestbook
```

<https://argo-cd.readthedocs.io/en/stable/operator-manual/applicationset/Generators-Cluster/>





```

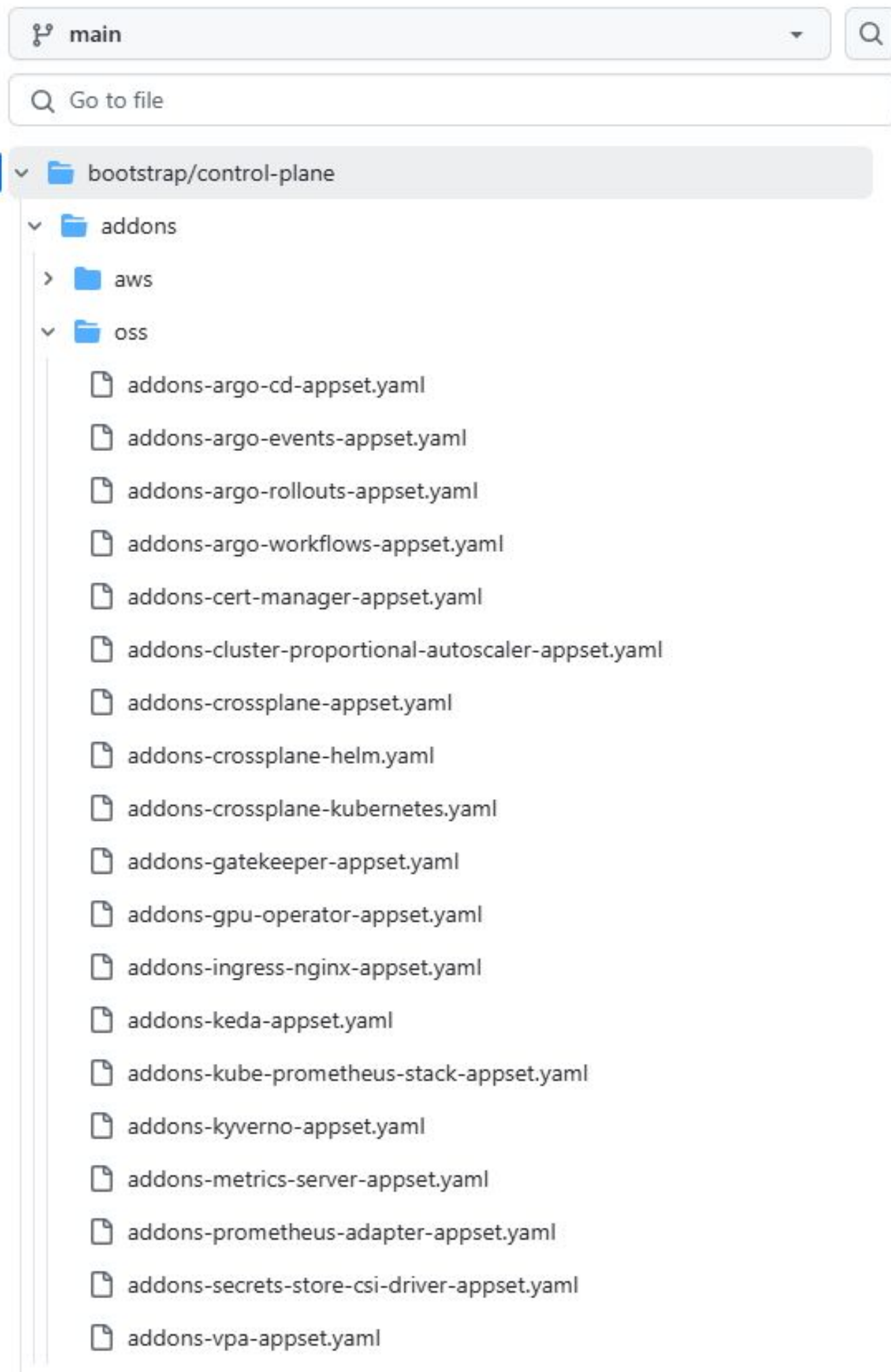
template:
  metadata:
    name: addon-{{name}}-{{values.addonChart}}
  spec:
    project: default
    sources:
      - repoURL: '{{metadata.annotations.addons_repo_url}}'
        targetRevision: '{{metadata.annotations.addons_repo_revision}}'
        ref: values
      - chart: '{{values.addonChart}}'
        repoURL: '{{values.addonChartRepository}}'
        targetRevision: '{{values.addonChartVersion}}'
        helm:
          releaseName: '{{values.addonChart}}'
          ignoreMissingValueFiles: true
          valueFiles:
            - $values/{{metadata.annotations.addons_repo_basepath}}environments/default/addons/{{values.addonChart}}/values.yaml
            - $values/{{metadata.annotations.addons_repo_basepath}}environments/{{metadata.labels.environment}}/addons/{{values.addonChart}}/values.yaml
            - $values/{{metadata.annotations.addons_repo_basepath}}clusters/{{name}}/addons/{{values.addonChart}}/values.yaml
    values: |
      clusterName: {{metadata.annotations.aws_cluster_name}}
      serviceAccount:
        name: {{metadata.annotations.aws_cloudwatch_metrics_service_account}}
        annotations:
          eks.amazonaws.com/role-arn:
            {{metadata.annotations.aws_cloudwatch_metrics_iam_role_arn}}

```

Create a service Account in Argo CD using Data known to Terraform







[https://github.com/gitops-bridge-dev/  
gitops-bridge-argocd-control-plane-te  
mplate](https://github.com/gitops-bridge-dev/gitops-bridge-argocd-control-plane-template)



## Approach 3 - Inject resources with K8s/Helm provider

- + Very flexible - can pass any kind of configuration
- + Premade templates
- ✗ Requires ApplicationSets
- ✗ Not GitOps, and no single source of truth
- ✗ Use Helm Overrides
- ✗ You still need to account for secrets



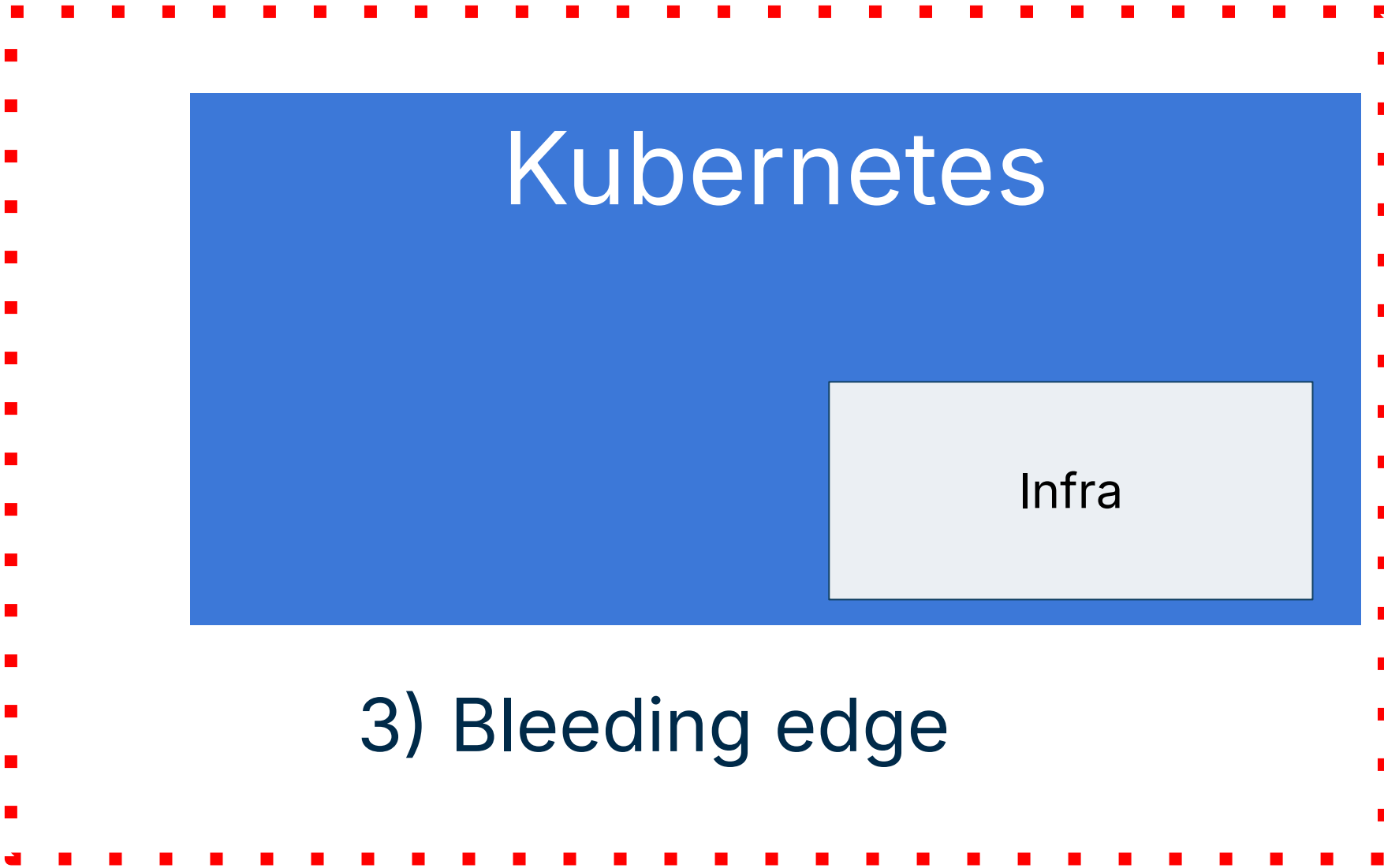
# The Future?



3 choices



1) Most teams are here



3) Bleeding edge




2) You should be here







Thread # argo-cd

 **João Pedro Alcantara** Dec 13th, 2024 at 3:03 PM  
Is it possible to use ArgoCD to deploy Infraestructure using Terraform? Is there any plugin for it?

Thread # argo-cd

 **Radoslav Rachev** Jun 13th, 2023 at 3:43 PM  
I am wondering for a way to allow argocd to use some terraform outputs as variable when creating the helm templates for application.  
Does someone configured something similar?

Thread # argo-cd

 **8bitlost** Dec 15th, 2023 at 5:48 PM  
Hi all had a query helpful if anyone can share there views on this:- flux I believe has a terraform controller that helps GitOpsify terraform . Does Argocd also have some kind of functionality where we can use GitOps for terraform ? (edited)


Thread # argo-cd

 **Kai Hendry** Aug 15th, 2022 at 8:15 AM  
Can anyone vouch for `terraform apply` from ArgoCD?


Thread # argo-cd

 **Lewis Cowles** Nov 13th, 2024 at 9:35 PM  
Would it be a terrible idea, to try to run Argo, outside of Kubernetes, to manage non-kubernetes things?

Thread # argo-cd

 **Chris Cutajar** May 6th, 2022 at 1:49 PM  
Hey folks, has anyone has a configuration setup or know of something whereby ArgoCD triggers a Terraform job?


Thread # argo-cd

 **Ishai Strauss** May 22nd, 2023 at 11:34 AM  
We'd like to use gitops for our terraform and found this project: <https://github.com/weaveworks/tf-controller>  
Anyone know if there is anything similar for argocd?





weaveworks/tf-controller	
A GitOps Terraform controller for Kubernetes	
Stars	Language
780	Go
Added by GitHub	

Thread # argo-cd

 **João Pedro Alcantara** Dec 13th, 2024 at 3:03 PM  
Is it possible to use ArgoCD to deploy Infraestructure using Terraform? Is there any plugin for it?

Thread # argo-cd

 **DevOpsGuy** Aug 18th, 2023 at 4:24 AM  
Hi All, We implemented ArgoCd recently and would like to manage the Terraform resources IaC using the ArgoCD.  
Upon research some third party vendors are providing the terraform controller to achieve that.  
  
Anyone any idea when that feature is going to be available on ArgoCD itself to manage the terraform resources ?

 **Yordis Prieto** Jun 9th, 2023 at 4:35 AM  
Hey there, I am new into DevOps, and I am learning as I go. I recently put together some deployment of the cert-manager and vault using terraform helm\_release, among other apps.  
  
I am wondering that if I should move the helm deployments into ArgoCD instead. Or if there is anything I should take into consideration before I use ArgoCD for the Helm releases.  
  
Thanks in advance!





# Crossplane



# Define any resource in K8s

Works for Google, Azure, AWS, Alibaba. You can also write your own provider

```
apiVersion: s3.aws.upbound.io/v1beta1
kind: Bucket
metadata:
  name: my-example-bucket
spec:
  forProvider:
    region: us-east-2
  providerConfigRef:
    name: default
```



# Define any resource in K8s (even DB migrations)



```
apiVersion: db.atlasgo.io/v1alpha1
kind: AtlasSchema
metadata:
  name: myapp
spec:
  # Load the URL of the target database from a Kubernetes secret.
  urlFrom:
    secretKeyRef:
      key: url
      name: mysql-credentials
  # Define the desired schema of the target database. This can be defined in either
  # plain SQL like this example or in Atlas HCL.
  schema:
    sql: |
      create table users (
        id int not null auto_increment,
        name varchar(255) not null,
        email varchar(255) unique not null,
        short_bio varchar(255) not null,
        primary key (id)
      );
```





Argo CD



Crossplane



atlas



Apps

Infra

DBs



# Conclusion



Solution	Effort/Complexity	Follows GitOps	Pros	Cons
Use DNS	Low	No single source of truth	Super Simple to use	Does not cover all scenarios
Use Git provider	High	Yes	Pure GitOps, Argo CD is isolated	Does not work with secrets
Inject Resources	Medium	No	Very powerful	Complex, Hacky, breaks GitOps and possibly Argo CD
GitOps bridge	Medium	No	Many existing templates available	Has specific requirements (e.g. applicationsets)

Understand [crossplane.io/](https://crossplane.io/)





# Thank you

Contact

[kostis.kapelonis@octopus.com](mailto:kostis.kapelonis@octopus.com)

# Learn about K8s DNS and external services

## `type: ExternalName`

Services of type `ExternalName` map a Service to a DNS name, not to a typical selector such as `my-service` or `cassandra`. You specify these Services with the `spec.externalName` parameter.

This Service definition, for example, maps the `my-service` Service in the `prod` namespace to `my.database.example.com`:

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
  namespace: prod
spec:
  type: ExternalName
  externalName: my.database.example.com
```

