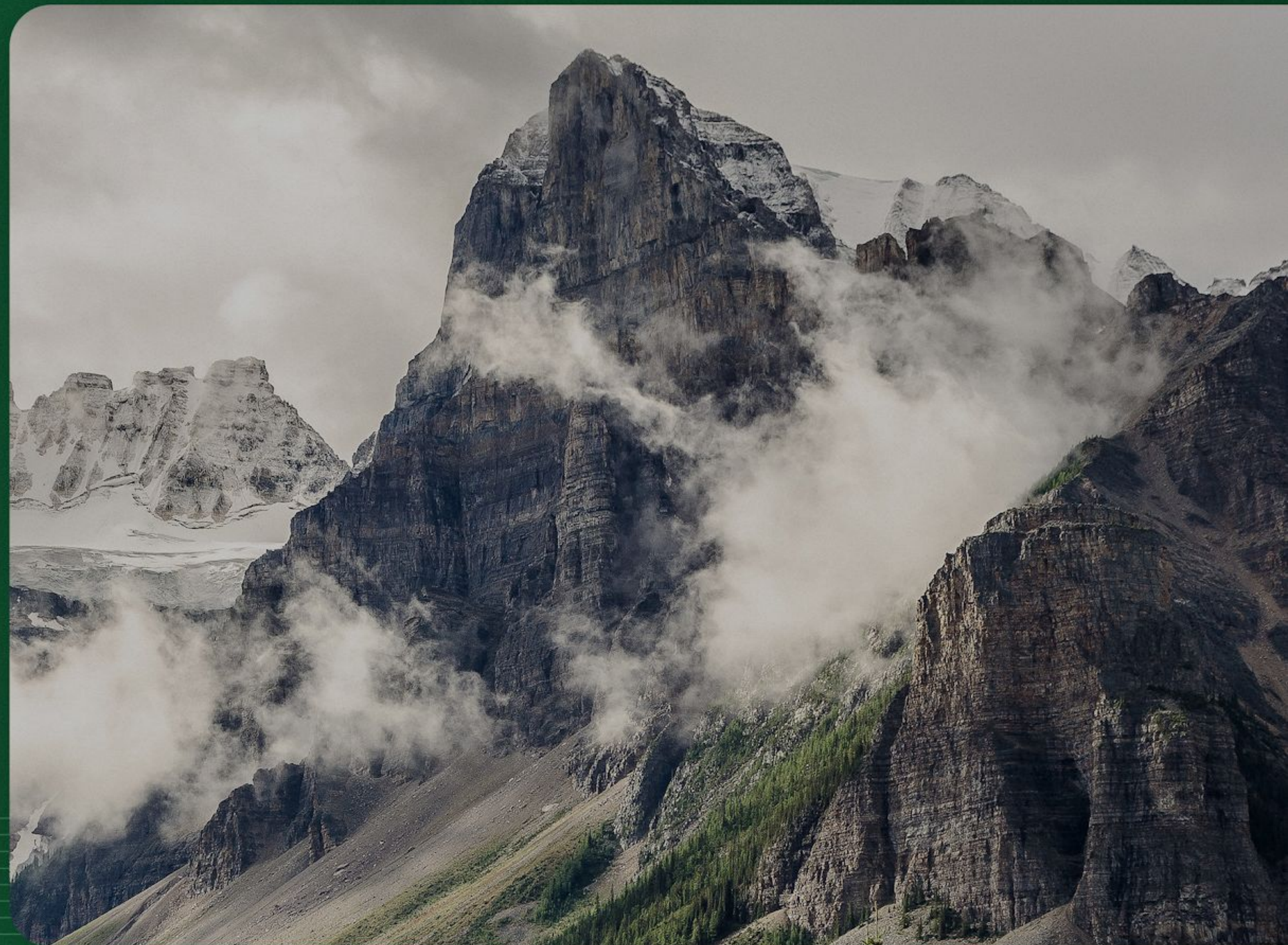

Functional Platform Engineering with Scala and Pulumi



About me

Łukasz Biały

Scala Developer Advocate
@ VirtusLab

Scala, FP, Distributed Systems,
DevOps (*apparently*)

X(witter): @lukasz_bialy

Agenda

- What is Platform Engineering?
- How can one do Platform Engineering?
- How can one do it without crying (or rather w/ crying, just a bit less)
- DEMO TIME: Pulumi + Besom + Automation API

Wait, what happened to DevOps?

Is DevOps Dying? DevOps is Dead with the Rise of Platform Engineering



Krispy Solutions · [Follow](#)

3 min read · Sep 21, 2023



1



DevOps

DevOps is dead, long live Platform Engineering!



Chris Stephenson
CTO @ Humanitec

True DevOps – “you build it, you run it” – has been the guiding methodology behind many development teams for years. However, the harsh reality is that most developers don’t like dealing with infrastructure. It isn’t just a source of significant cognitive load, but it also takes time away from their core job of writing, debugging, and running applications.

DEVOPS / PLATFORM ENGINEERING / SOFTWARE DEVELOPMENT / TECH CULTURE

DevOps Is Dead. Embrace Platform Engineering

Platforms provide golden paths, with recommended tools and best security practices built in, reducing cognitive load while preserving developer freedom.

Sep 22nd, 2022 7:10am by [Aeris Ransom](#)



Ok, what is it then?

What is platform engineering?

Platform engineering is the discipline of designing and building toolchains and workflows that enable self-service capabilities for software engineering organizations in the cloud-native era. Platform engineers provide an integrated product most often referred to as an “Internal Developer Platform” covering the operational necessities of the entire lifecycle of an application.



Luca Galante
Product @ Humanitec

Uhh, ok?

What is platform engineering, briefly?



Platform engineering involves creating and managing a shared set of tools, services, and capabilities that enable developers to build, deploy, and operate software more efficiently. The goal of platform engineering is to abstract the complexities of infrastructure and operations from developers, allowing them to focus more on coding and less on the underlying systems. This often includes providing automated workflows, continuous integration and deployment pipelines, observability tools, and security practices integrated into a common platform that teams across an organization can use.

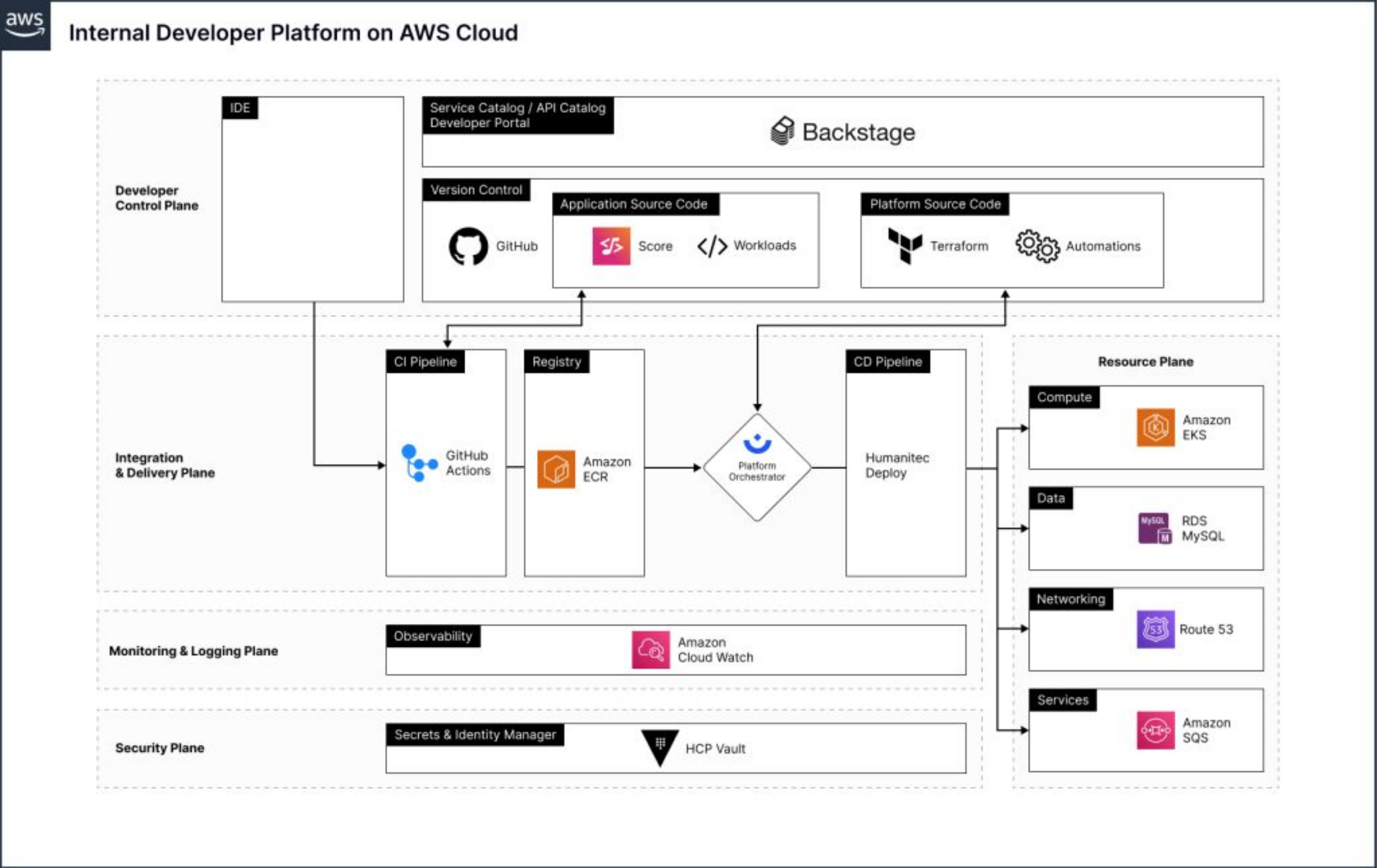


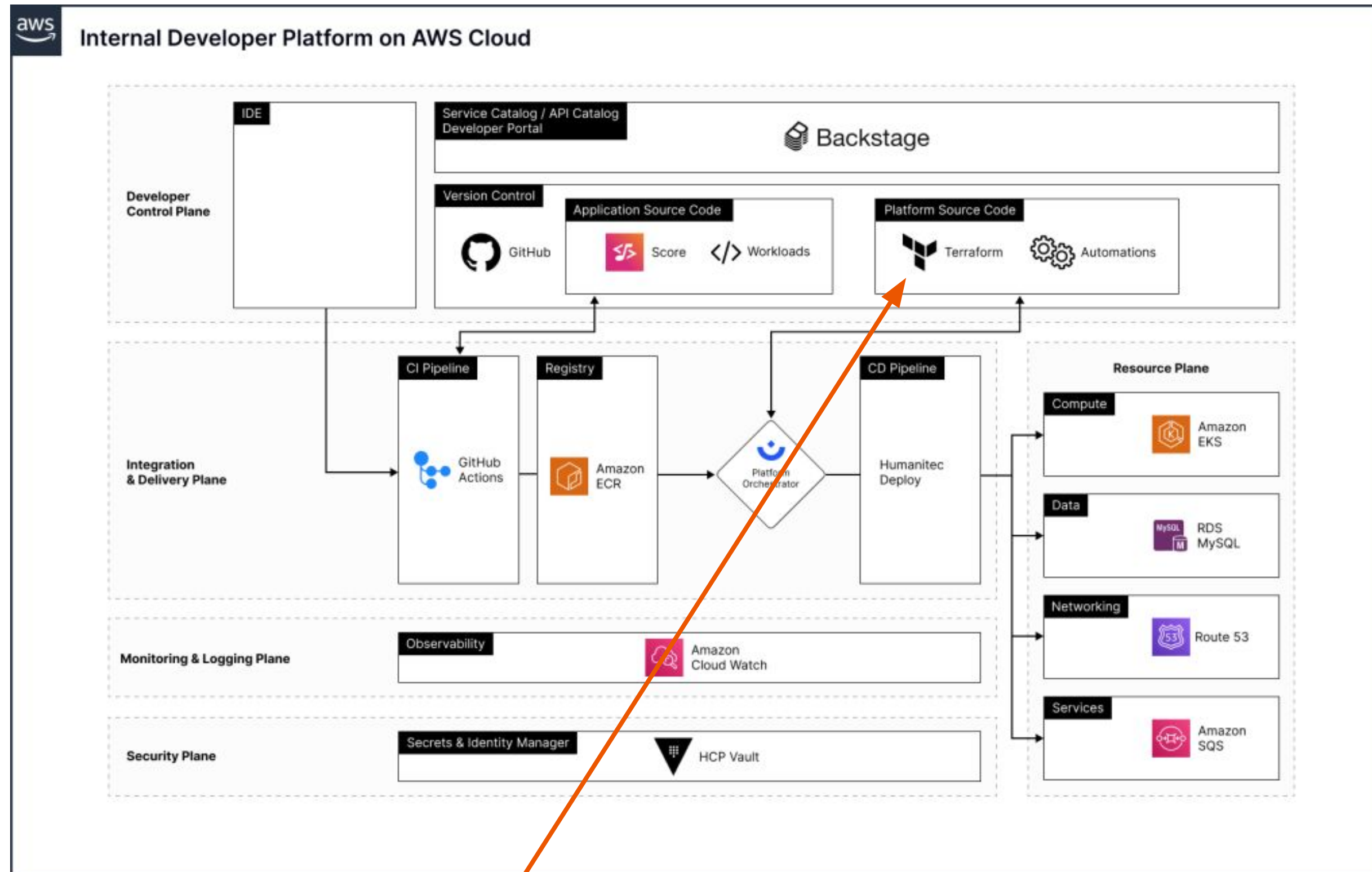
My definition



How does one practice Platform Engineering?

After a lengthy search...





wrapping terraform in CI yaml pipelines

Can we do better?



Pulumi

What's Pulumi?

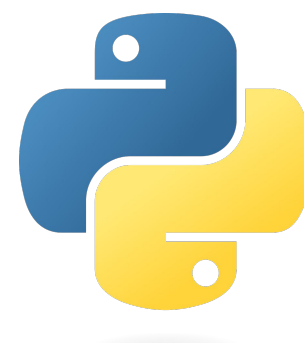
*"Using Pulumi, you author cloud programs using **your favorite language**, spanning low-level infrastructure-as-code to highly productive and modern container- and serverless-powered applications."*

- Joe Duffy, CEO of Pulumi

Favorite languages?



Pulumi



Anatomy of a Pulumi program

- Stacks
 - a Pulumi program is a blueprint
 - instance of the blueprint is a stack

- Resources & Inputs:

```
val catsBucket = aws.s3.Bucket("cats", BucketArgs(  
    acl = "public-read" // << this is an Input!  
)) : Output[s3.Bucket]
```

- Outputs:

```
Stack.exports(  
    catsUrl = catsBucket.websiteEndpoint  
)
```


A small example

```
import besom.*
import besom.api.aws

@main def main = Pulumi.run {
  val catsBucket = aws.s3.Bucket("cats", BucketArgs(
    acl = "public-read"
  ))

  val dogsBucket = aws.s3.Bucket("dogs", BucketArgs(
    acl = "public-read"
  ))

  Stack.exports(
    catsUrl = catsBucket.websiteEndpoint,
    dogsUrl = dogsBucket.websiteEndpoint
  )
}
```

**A bit more
info for your
early
morning
coffee
doomscroll:**

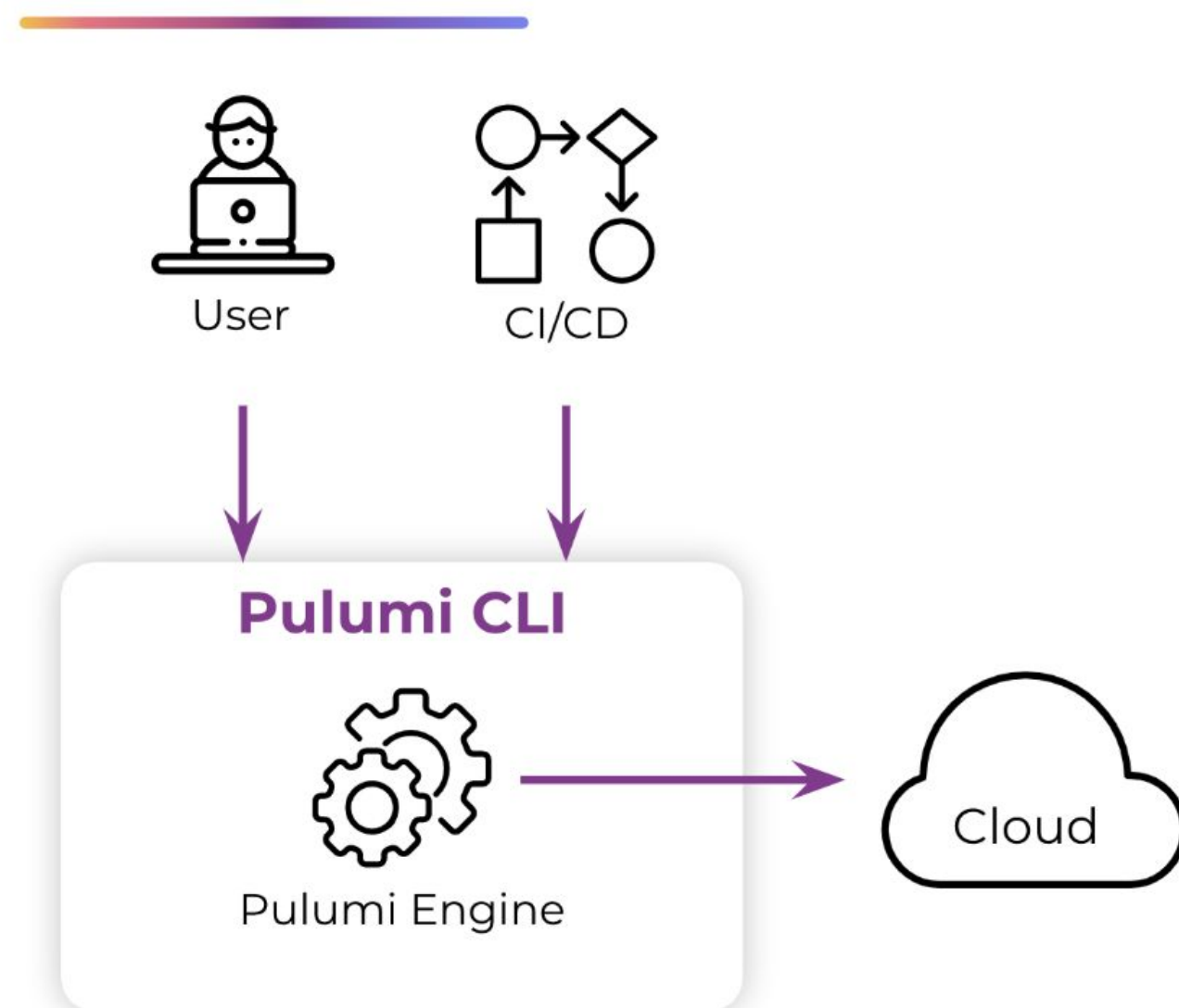


Platform Engineering according to



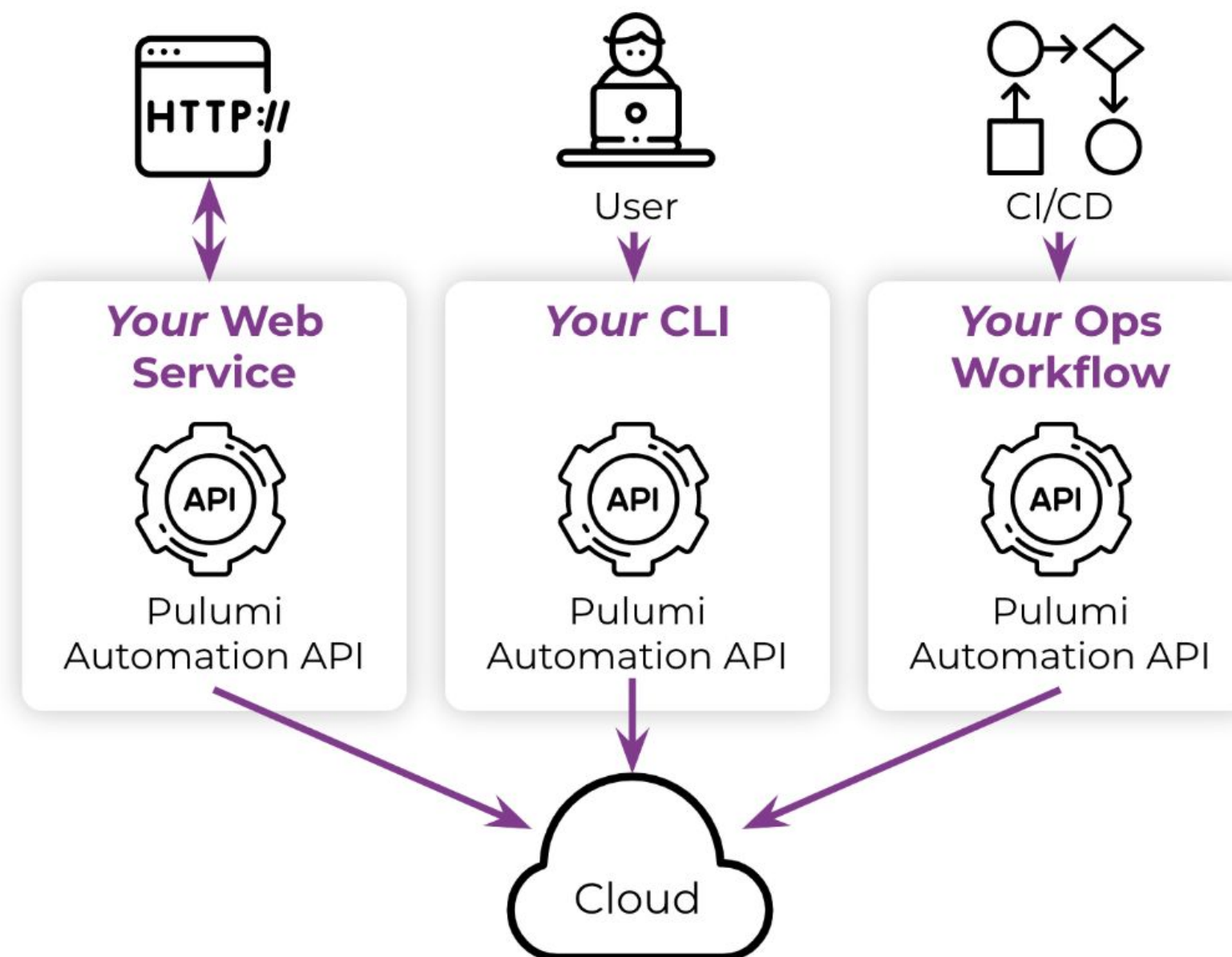
Pulumi

Deploy Your Workflow



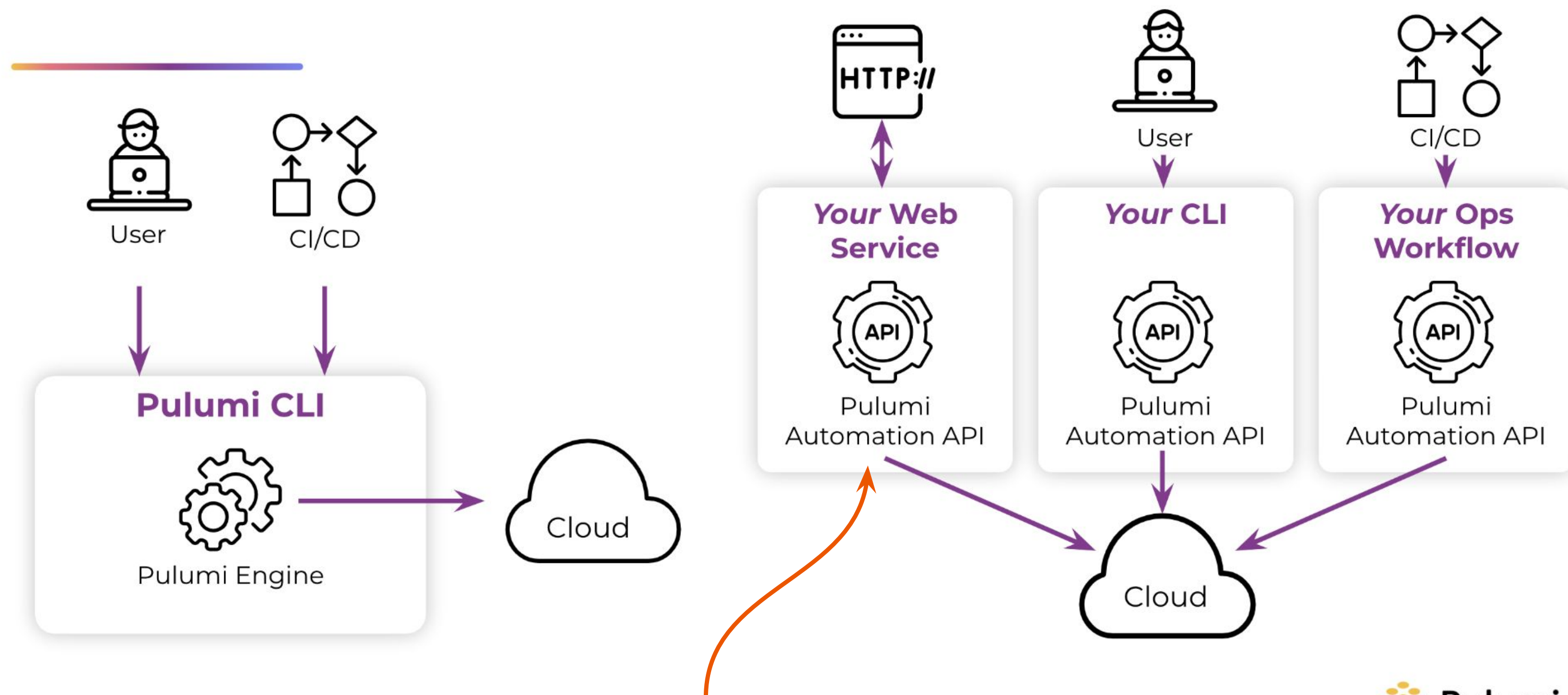
Automation API

NEW!



 Pulumi

Deploy Your Workflow



Interesting!

A practical example



Let's build a Netlify clone

- Hosting static content only
- Dynamic provisioning of cloud resources for user's website on-demand (*self-service*)
- You get a nice URL

How to draw an owl

1.



1. Draw some circles

2.



2. Draw the rest of the fluffy owl

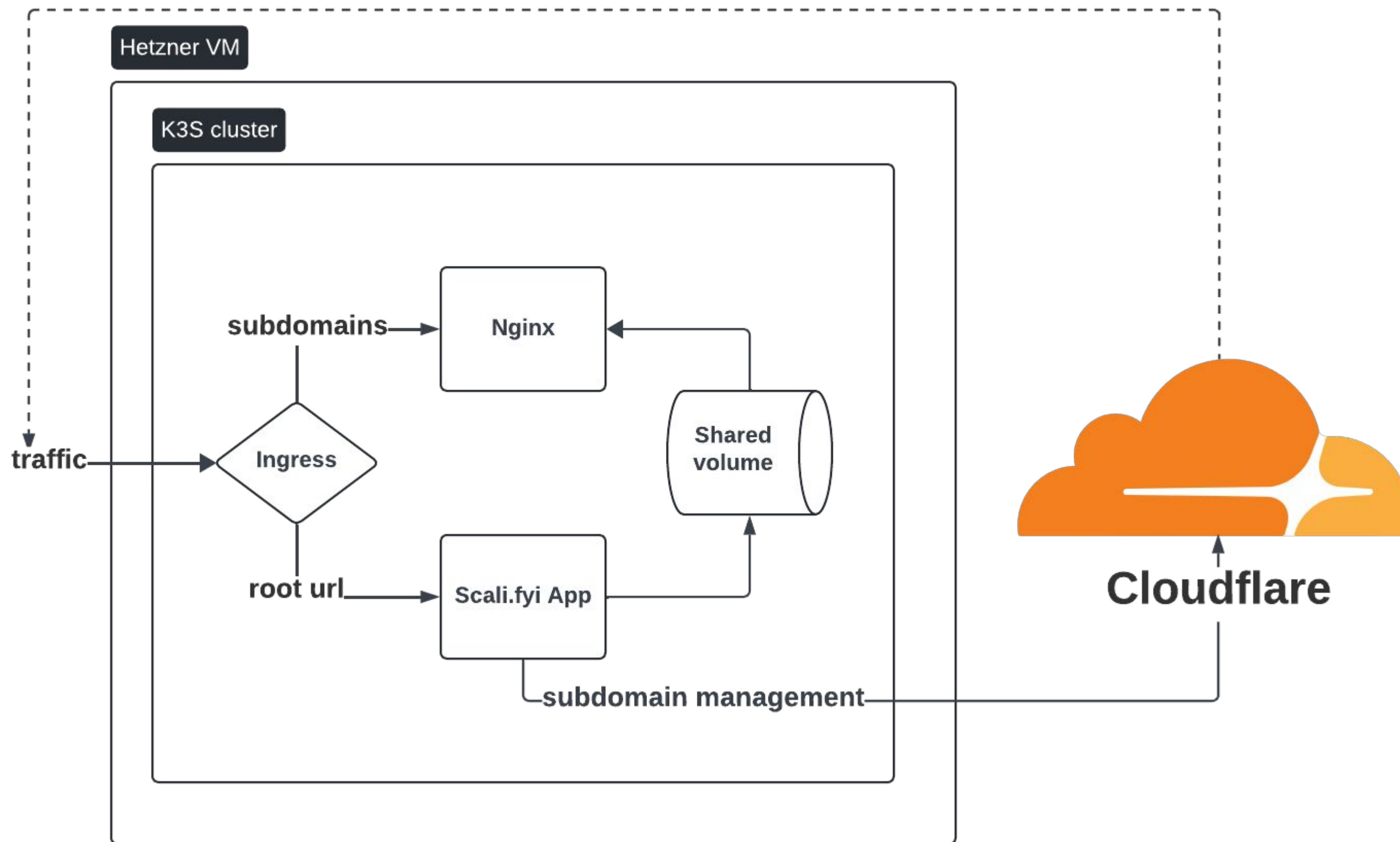


Do's and don'ts

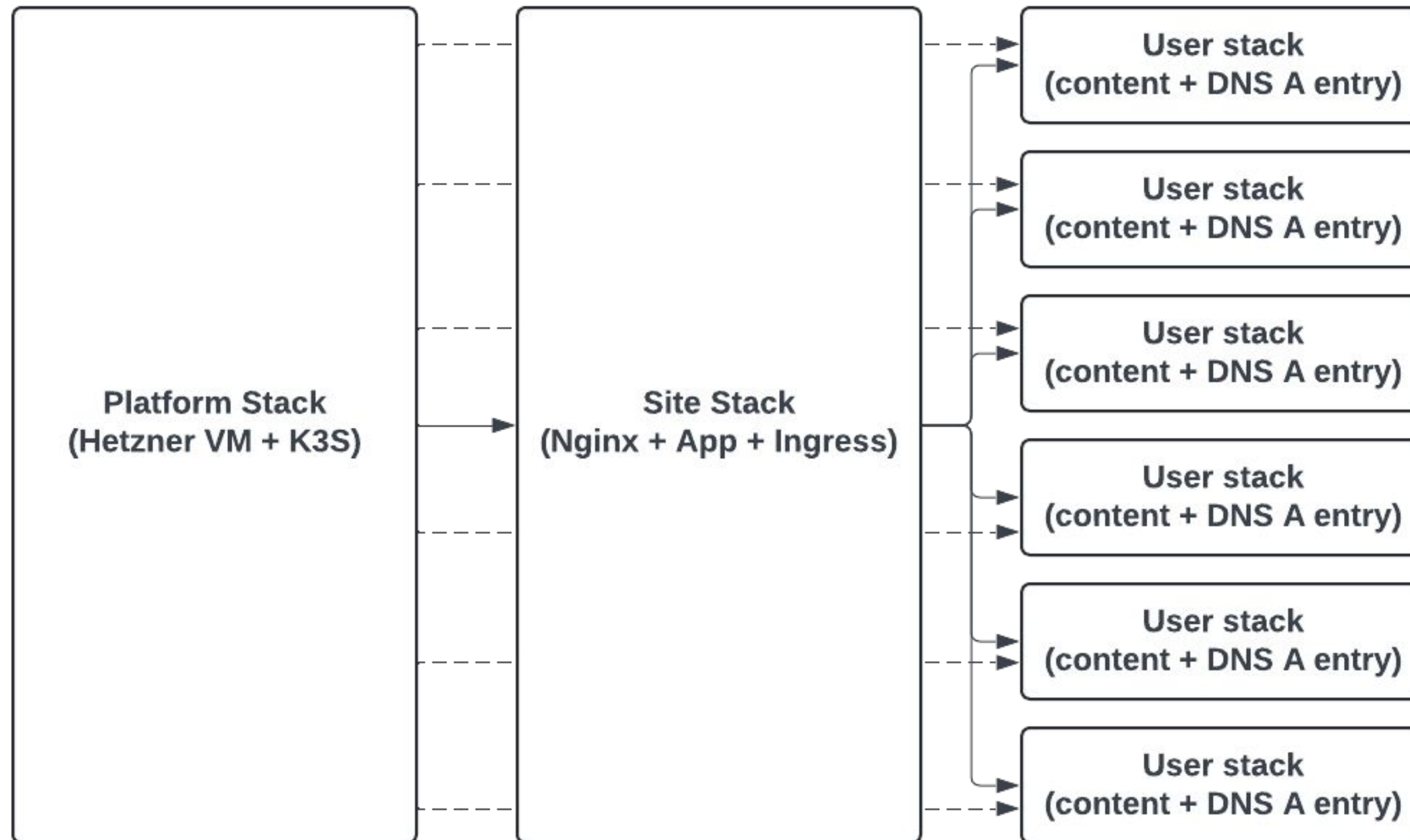
- DO NOT CLICK DEPLOY TWICE, I was too lazy to lock the button 🤡
- Put a subdomain you'd like to use, e.g.: ***monadic-burritos.scali.fyi***
- Be creative, it will crash on duplicates!
- Put any HTML into the content box, you can adjust the existing one
- Click Deploy **ONCE** and *maybe* it will work

The link





Architecture



Pulumiception

The code

Links!

- <https://virtuslab.github.io/besom/>
- <https://github.com/VirtusLab/besom>
 - <https://www.pulumi.com/>
- <https://github.com/lbialy/scali.fyi>

Thank you

