

BATTLE OF THE IAC TITANS

Infrastructure as code

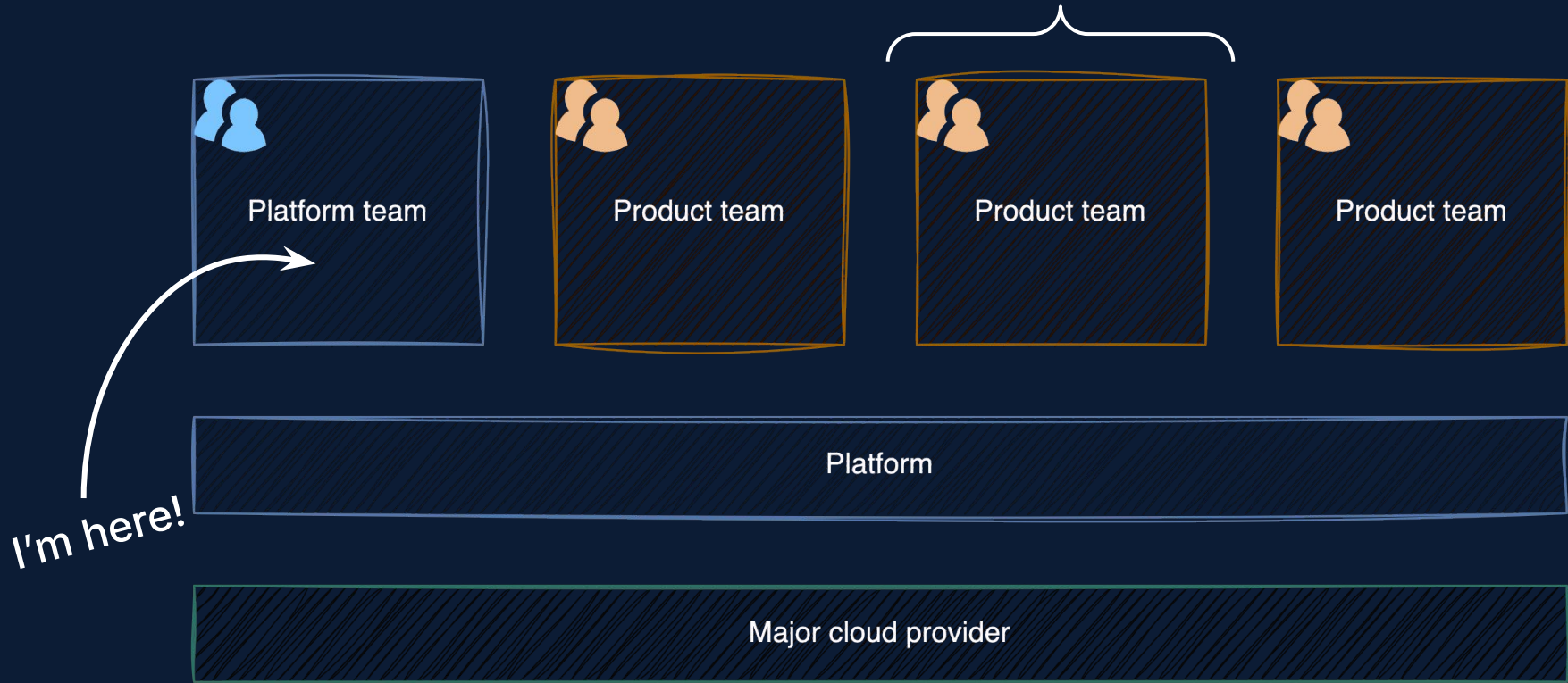
A dramatic black and white photograph of a stormy night sky. Dark, heavy clouds fill the upper portion of the frame. Two bright lightning bolts are visible, striking down from the clouds towards the horizon. The lower portion of the image shows a dark, silhouetted landscape with some distant lights and trees. The overall mood is intense and powerful.

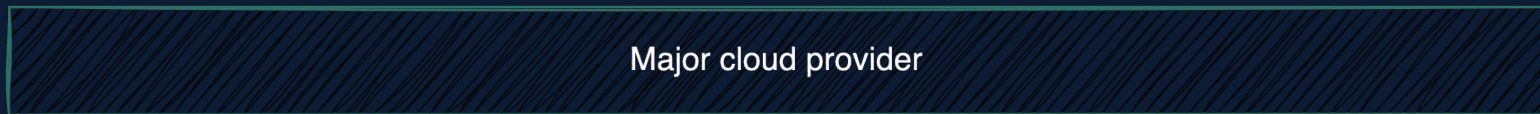
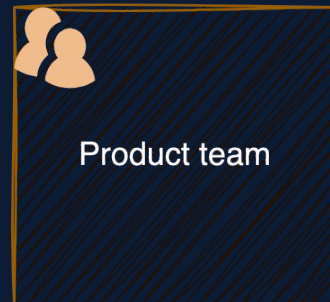
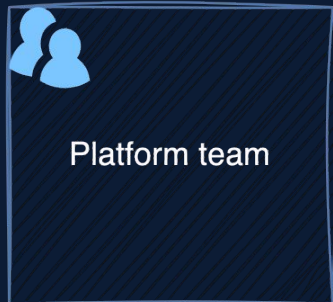
Infrastructure as code





End-to-end responsible





Let's shift gears!


```
Parameters:
  BucketPrefix:
    Type: String
    Default: "javazone2025"

Resources:
  MyBucket:
    Type: "AWS::S3::Bucket"
    Properties:
      BucketName: !Sub "${BucketPrefix}-my-bucket"

Outputs:
  BucketName:
    Value: !Ref MyBucket
```

```
$ aws cloudformation create-stack \
  --stack-name "my-stack" \
  --template-body file:///template.yml
```



2011

AWS CloudFormation

```
resource "aws_s3_bucket" "example" {  
  bucket = "battle-of-the-iac-titans"  
}  
  
module "static_website" {  
  count = 2  
  source = "../modules/static-website"  
  bucket = aws_s3_bucket.example.id  
  domain = "${each.index}.ekern.me"  
}
```

\$ terraform init
\$ terraform apply



```
Plan: 1 to add, 0 to change, 0 to destroy.
```

```
Do you want to perform these actions?
```

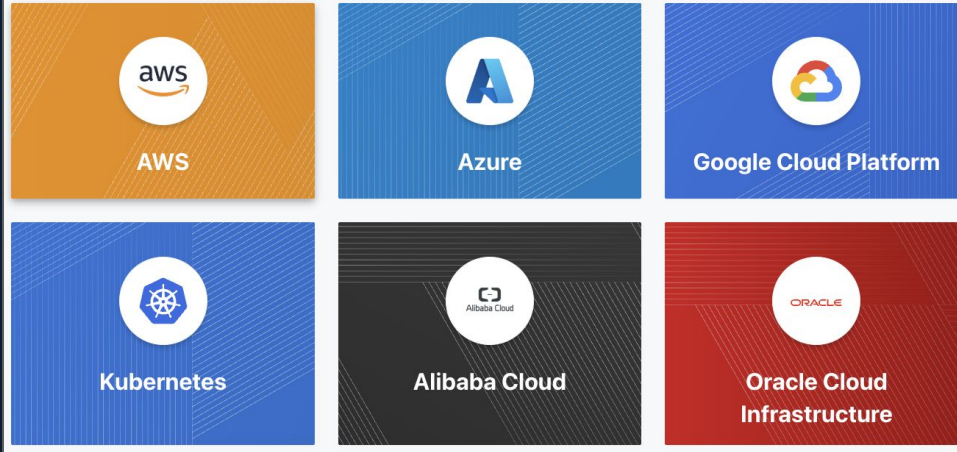
```
Terraform will perform the actions described above.  
Only 'yes' will be accepted to approve.
```

```
Enter a value: |
```



⌘ Providers

Providers are a logical abstraction of an upstream API. They are responsible for understanding API interactions and exposing resources.



Terraform Provider for Dominos Pizza



Build **passing**



TF Registry **v0.2.4**



Downloads **13668**



TypeScript

```
const exampleLambda = new lambda.Function(this, "ExampleLambda", {  
  code: lambda.Code.fromInline(  
    `exports.handler = event => ({ "message": "Hello world!" })`,  
  ),  
  handler: "index.handler",  
  runtime: lambda.Runtime.NODEJS_22_X,  
})
```

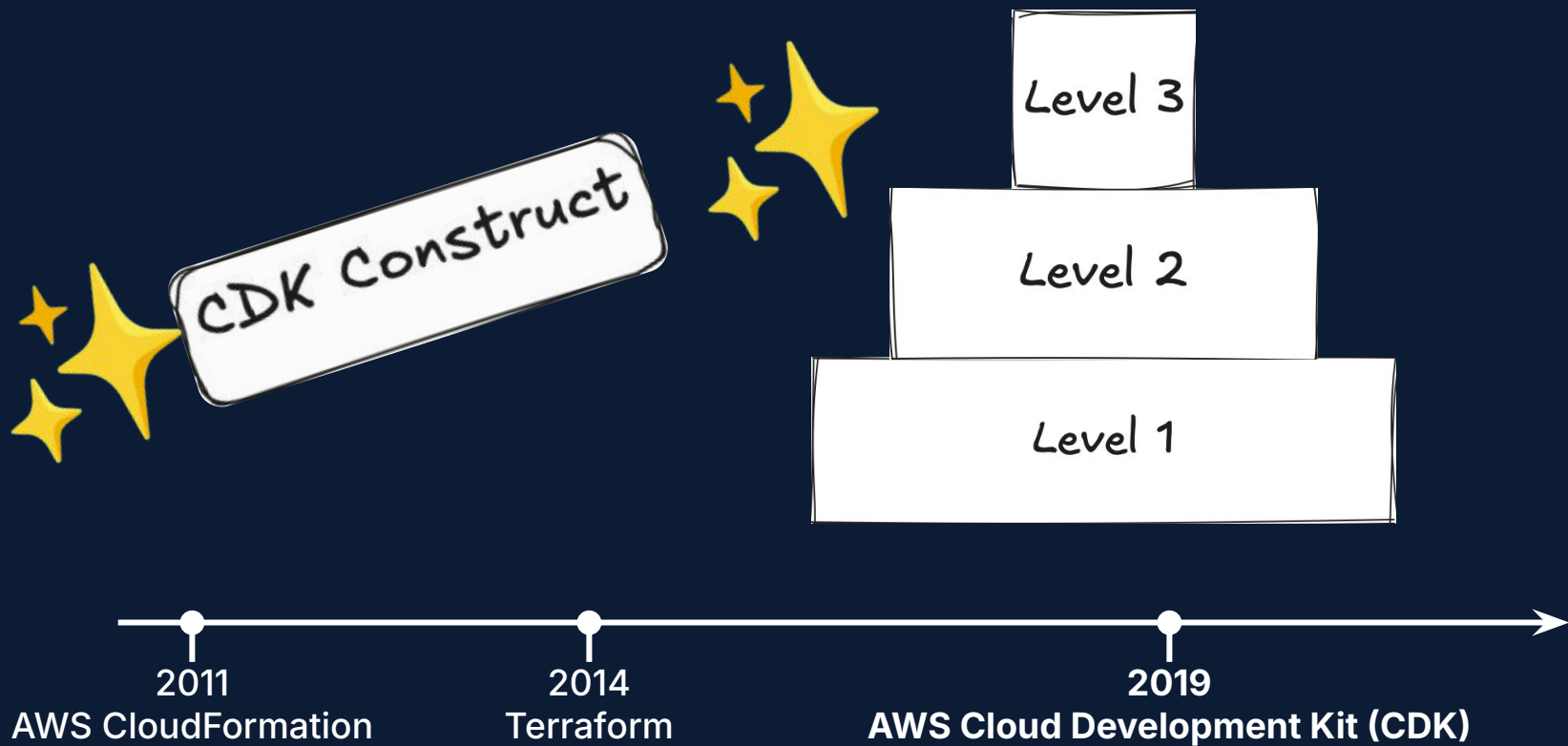
\$ cdk synth



103 lines of
CloudFormation

```
{
  "Resources": {
    "ExampleLambdaB672C330": {
      "Type": "AWS::Lambda::Function",
      "Properties": {
        "Code": {
          "ZipFile": "exports.handler = event => ({ \"message\": \"Hello world!\" })"
        },
        "Handler": "index.handler",
        "Role": {
          "Fn::GetAtt": [
            "ExampleLambdaServiceRole0AC04327",
            "Arn"
          ]
        },
        "Runtime": "nodejs22.x"
      },
      "DependsOn": [
        "ExampleLambdaServiceRole0AC04327"
      ],
      "Metadata": {
        "aws:cdk:path": "LambdaStack/ExampleLambda/Resource"
      }
    }
  }
}
```





CDK App

CDK Stage

CDK Stack

CDK Construct

CDK Construct

CDK Stack

CDK Construct

CDK Construct

2011

AWS CloudFormation

2014

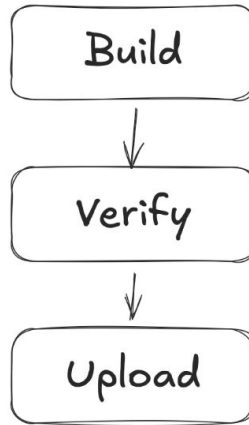
Terraform

2019

AWS Cloud Development Kit (CDK)

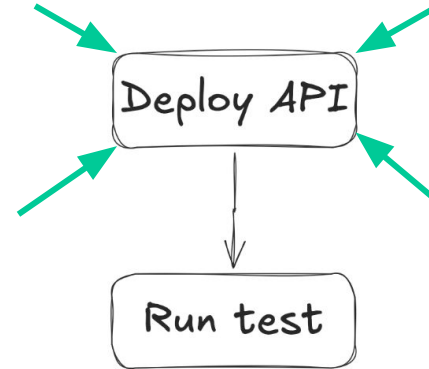
Demo

Build pipeline



Trigger

Deployment pipeline



Terraform

Strengths

- + Easy to grasp
- + Supports all popular APIs
- + Performant
- + Drift reconciliation

Weaknesses

- Limited language
- Mainly a provisioning tool
- Developer experience

AWS Cloud Development Kit (CDK)

Strengths

- + General-purpose language
- + High-level building blocks
- + Cloud development framework
- + Developer experience

Weaknesses

- Requires discipline
- Specific to AWS
- Leaky abstraction
- No drift reconciliation

```
# create s3 bucket and lambda function
# ...

resource "aws_iam_role_policy" "grant_lambda_s3_put" {
  role = aws_iam_role.lambda.id
  policy = jsonencode({
    Version = "2012-10-17"
    Statement = [
      {
        Effect    = "Allow"
        Resource  = "${aws_s3_bucket.example.arn}/*"
        Action    = [
          "s3:PutObject",
          "s3:PutObjectLegalHold",
          "s3:PutObjectRetention",
          "s3:PutObjectTagging",
          "s3:PutObjectVersionTagging",
          "s3:Abort*"
        ]
      }
    ]
  })
}
```

Terraform

```
// create s3 bucket and lambda function
// ...

bucket.grantPut(lambdaFunction)
```

AWS CDK

Suggestions



Expertise



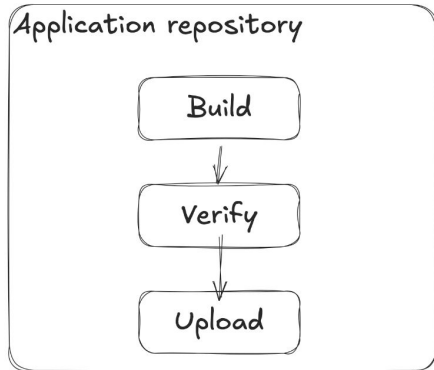
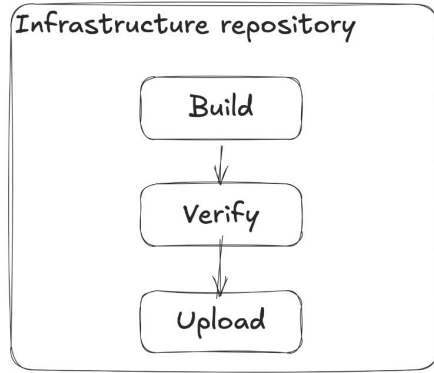
KISS



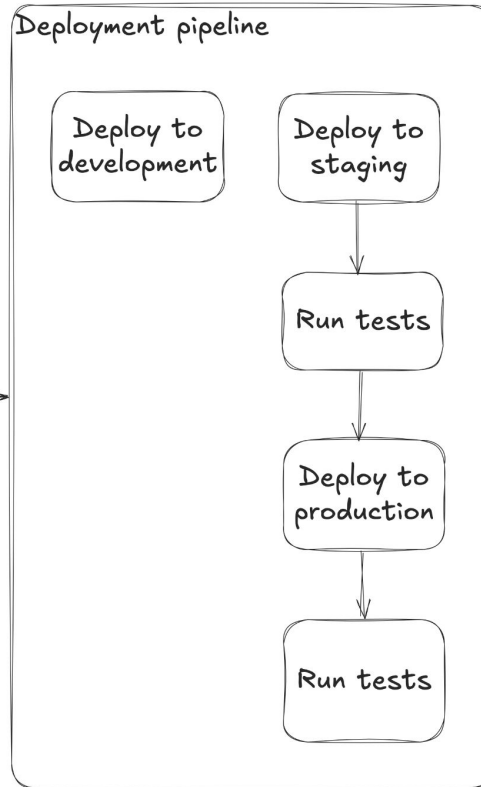
Building blocks

AWS CDK

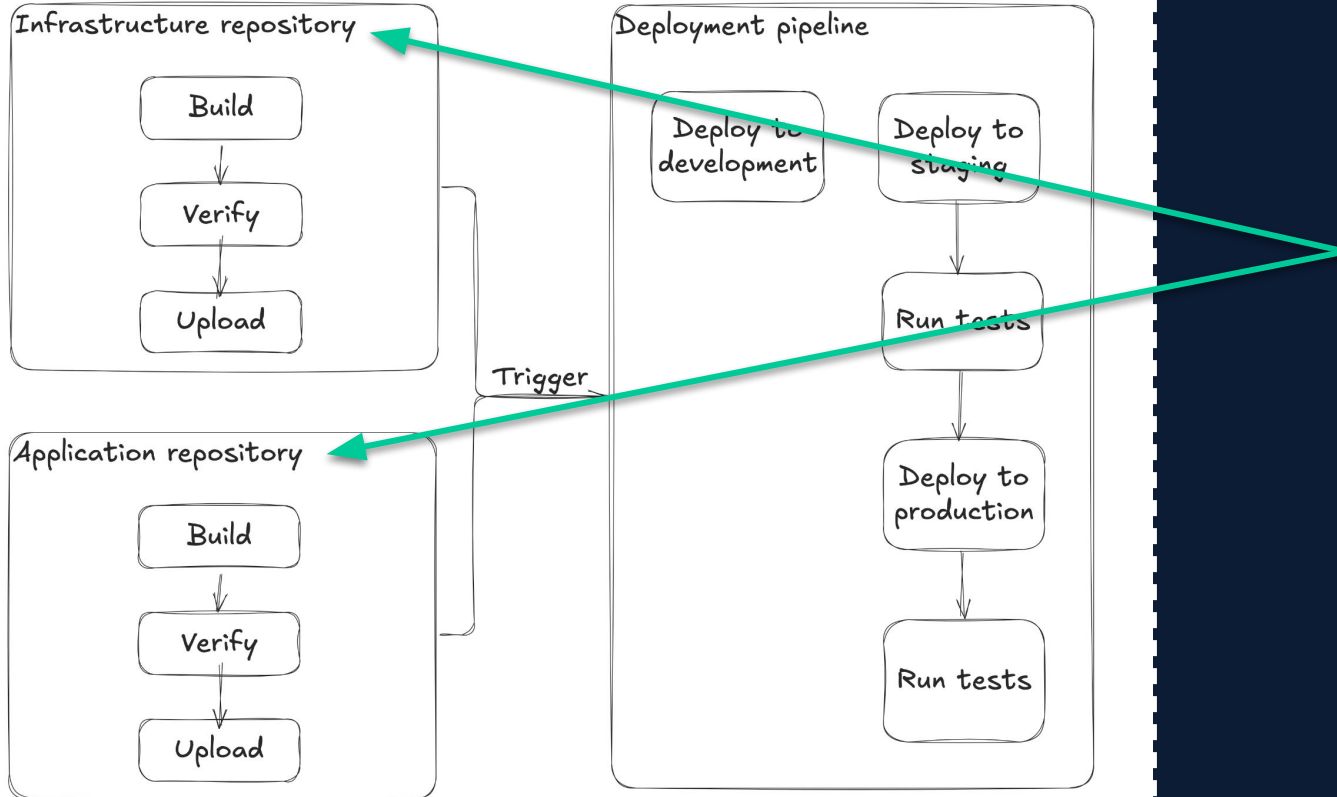
- Understand CloudFormation
- Start with TypeScript
- Use snapshot testing



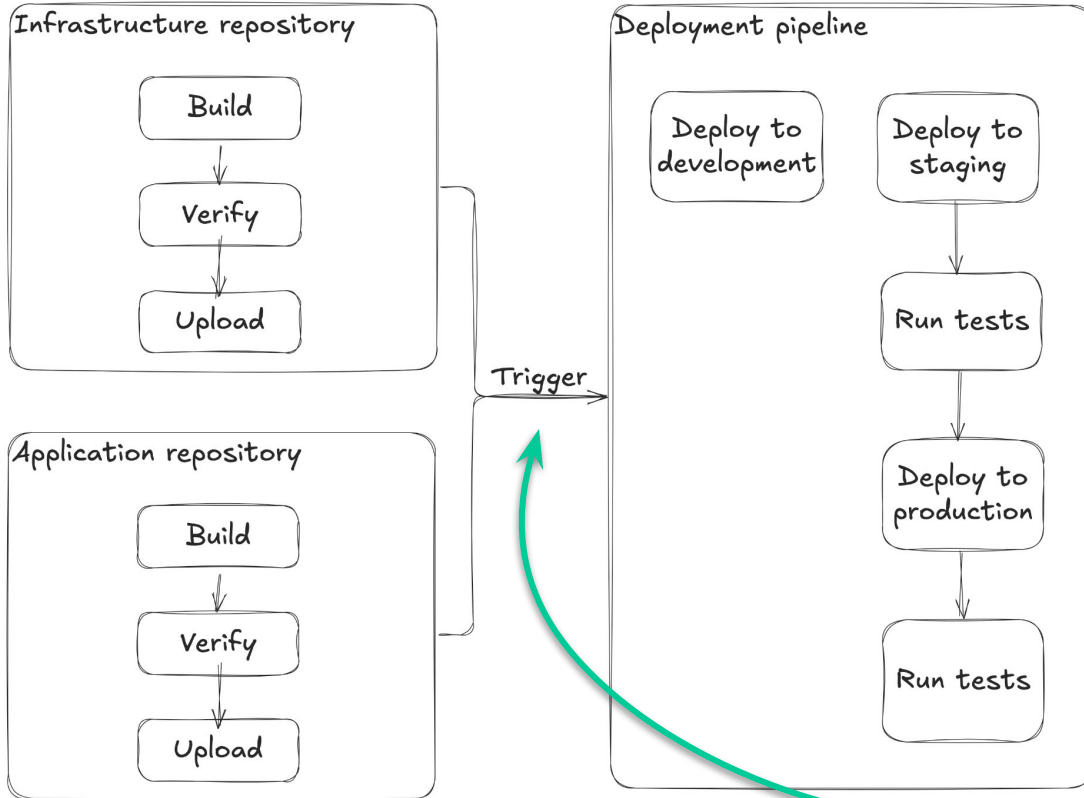
Trigger



Owned by a single team

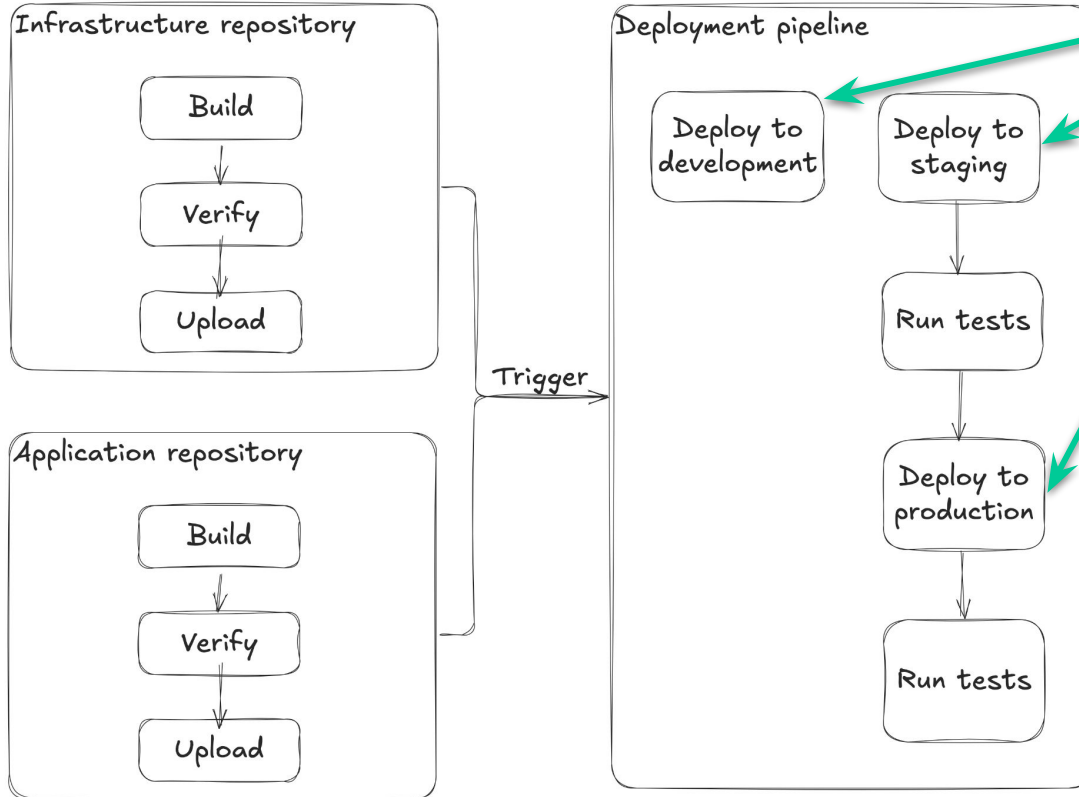


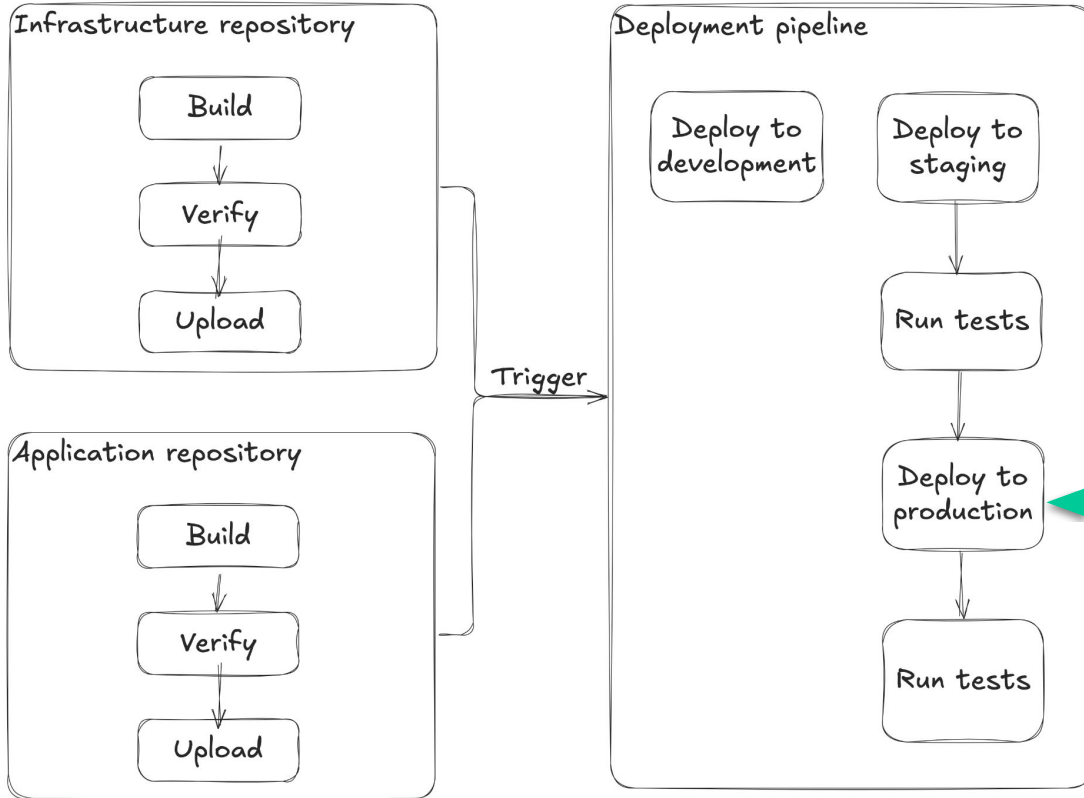
Separation of
concerns



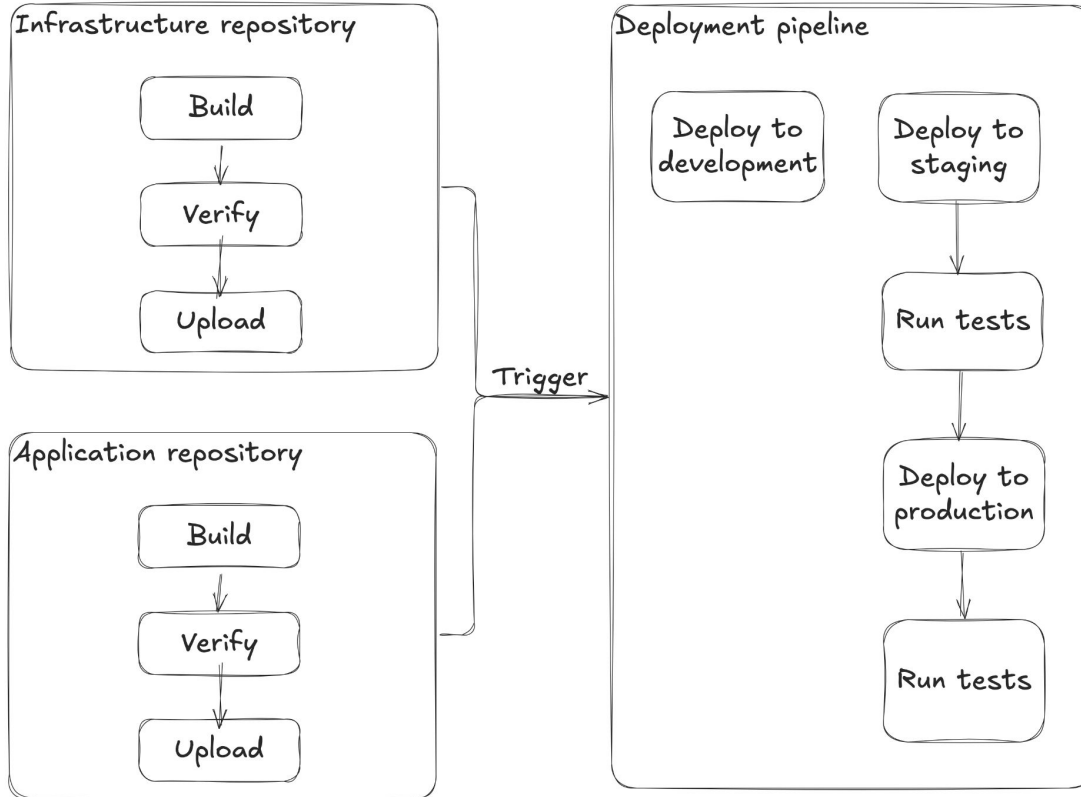
Trigger the same pipeline

Separate
environments



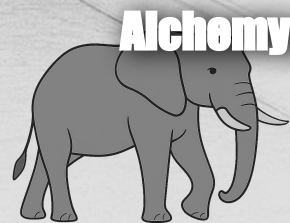
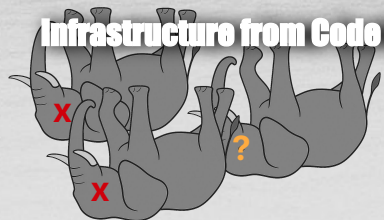
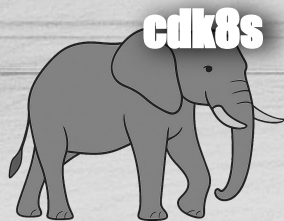
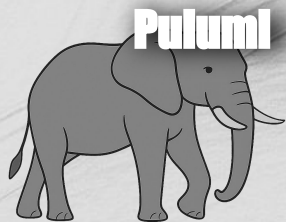
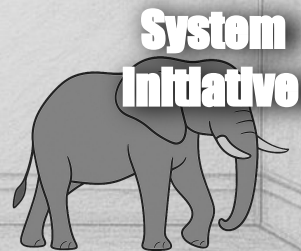
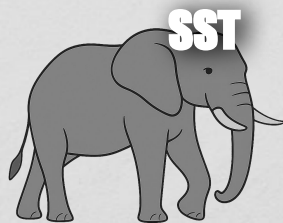
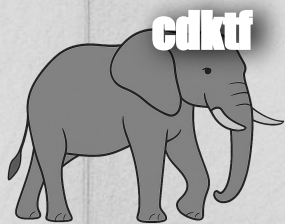


Organized into
*Shared stacks and
Single service stacks*



Continuous
Deployment





Thank you!



Erlend Ekern



ekern.me



linkedin.com/in/erlendekern



twitter.com/erlendekern

Slides and more materials

