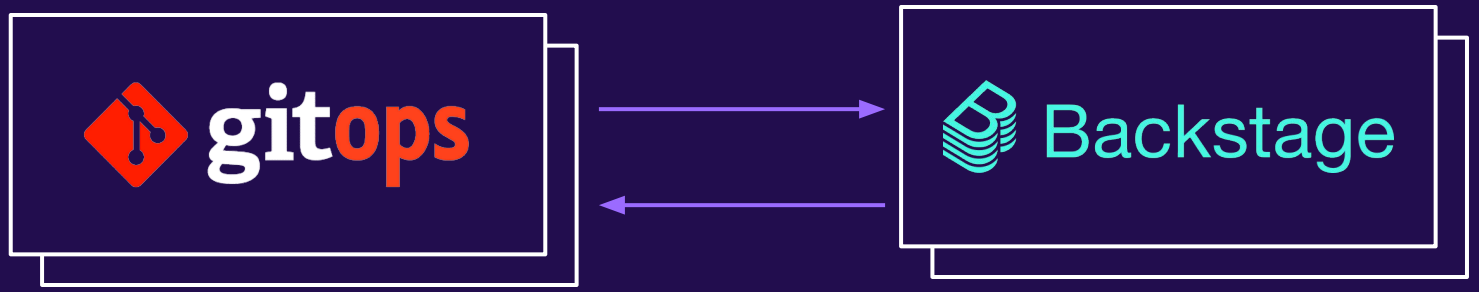
Shahar Shmaram | Ran Mansoor

How we migrated over 1000 services to Backstage using GitOps and survived to talk about it!

AppsFlyer Engineering

KubeCon + CloudNativeCon Europe 2023


gitops
Backstage

Ran Mansoor
Platform Software Engineer

Shahar Shmaram
Platform Software Engineer

This is AppsFlyer today

$40bn
Ad spend Measured per year

~14K
Customers

$300M
In funding

$300+M
ARR

65%
Global market share*

25
Offices globally

$9.8M
Fraud blocked per day

10K
Integrated technology partners

1,500+
Employees

magma · 8 EIGHT ROADS · T CAPITAL PARTNERS · Goldman Sachs · pitango · QUMRA CAPITAL · GENERAL ATLANTIC · salesforce ventures

Source: App Annie, Mobile Measurement Provider (MMP) SDKs implemented globally on iOS & Android (April 2021)

AppsFlyer Engineering

+400

AppsFlyer engineers. Operate in autonomous squads.

+1000

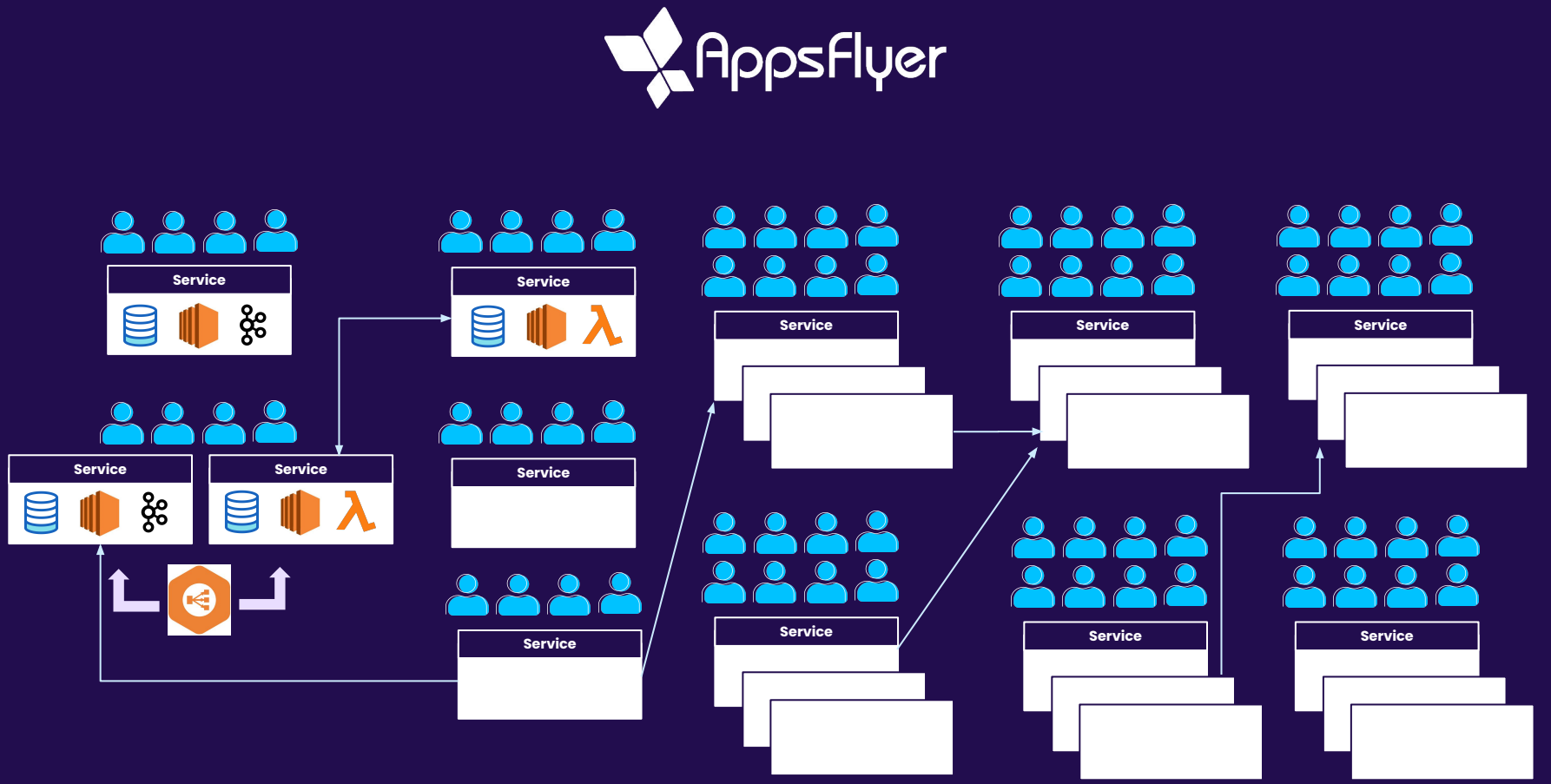
Microservices **handling over 3M events per second.**

~250K

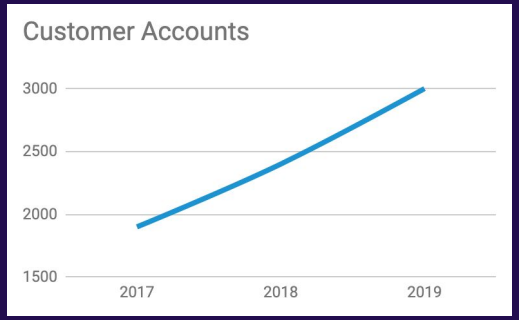
Cloud resources and dozens of SaaS integrations.

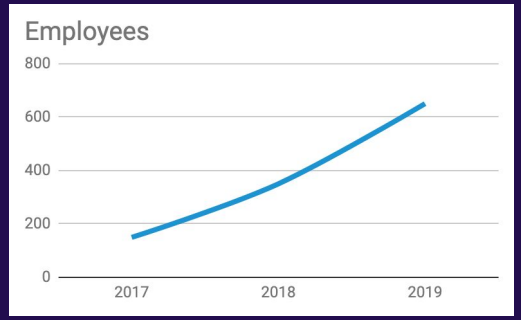
Platform Team

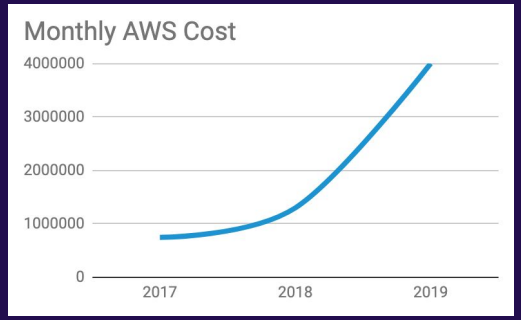
~50	6	3	+350
Engineers	Teams	Product Managers	R&D engineers

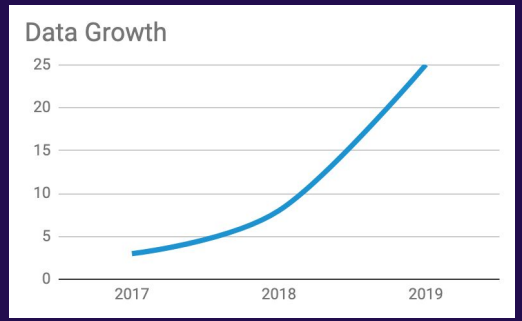

Appsflyer
Service
Service
Service
Service
Service
Service
Service
Service
Service
Service
Service
Service
Service

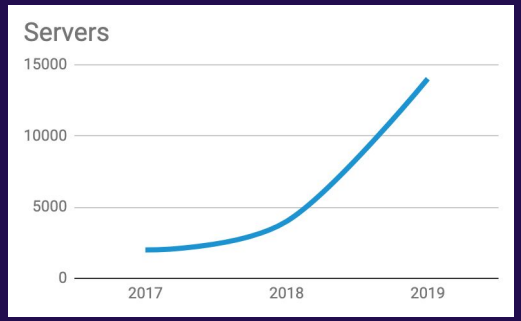
Hyper-Growth











We Won the market but...

- ✖ No alignment
- ✖ Manually managed resources
- ✖ Unknown resources dependencies and ownership
- ✖ Exploding budget
- ✖ Lack of technical documentation

Frustration

Solution Principals

Auditable

Declarative

Single Source of Truth

Community Driven

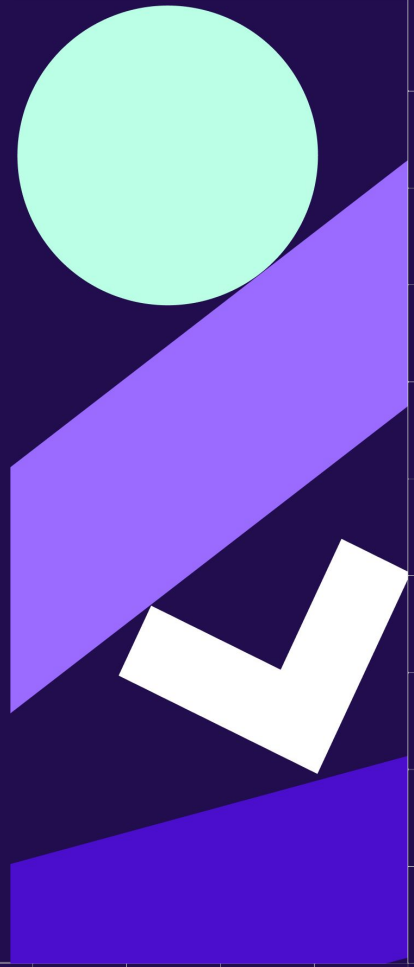
Self Serve

Visibility

Gitops

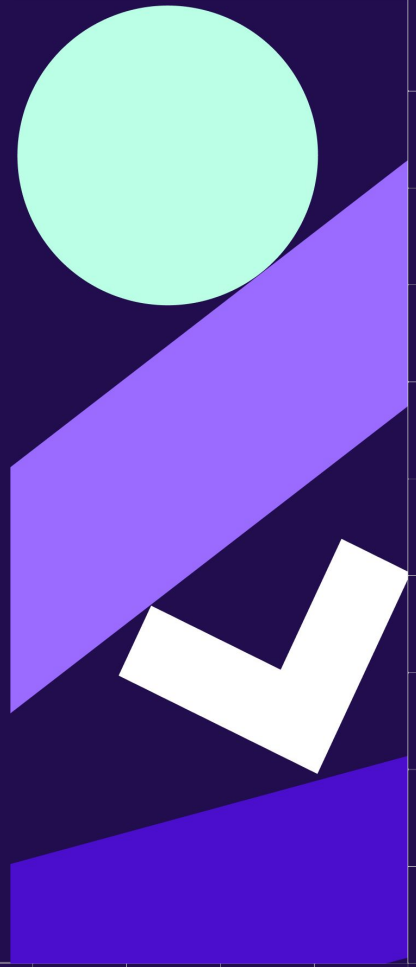
- Leverages git as the single source of truth for managing the entire software development lifecycle.

- Emphasizes the use of declarative infrastructure-as-code (IaC) and automated deployment pipelines.



Gitops - Core Principles

- Declarative
- Versioned & Immutable
- Pulled Automatically
- Continuously Reconciled



Solution Principals

Auditable

Declarative

Single Source of Truth

Community Driven

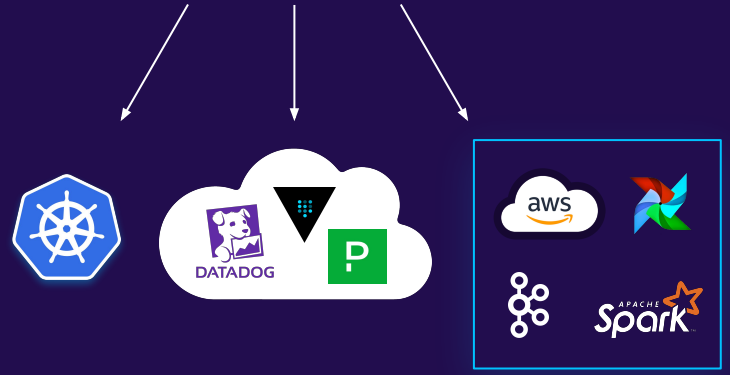
Self Serve

Visibility

Terraform

Terraform is a tool for building, changing, and versioning infrastructure safely and efficiently

Infrastructure as Code (IaC)



Execution Plans

```
Plan: 2 to add, 0 to change, 2 to destroy.
```

Solution Principals

Auditable

Declarative

Single Source of Truth

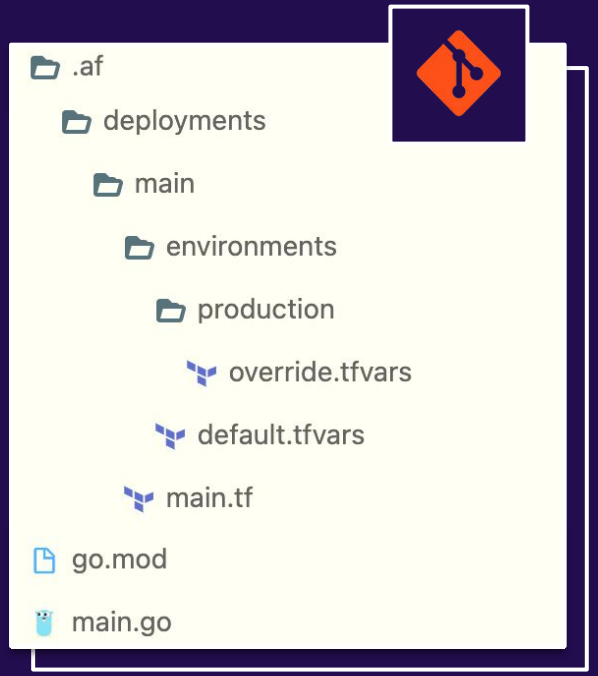
Community Driven

Self Serve

Visibility

Meta Folder

- Configuration close to application code
- .af folder in every git repository
- Contains deployment manifest and resources the application needs



Solution Principals

Auditable

Declarative

Single Source of Truth

Community Driven

Self Serve

Visibility

Developer A

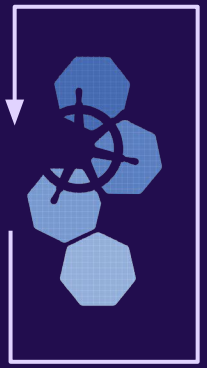
Developer A

- ❏ Commit code changes
- ❏ Declare service deployment plan

GitOps Workflow

Developer A

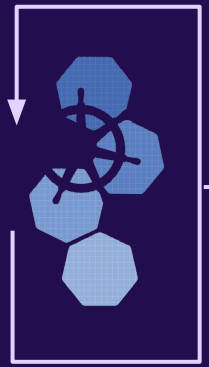
- Commit code changes
- Declare service deployment plan



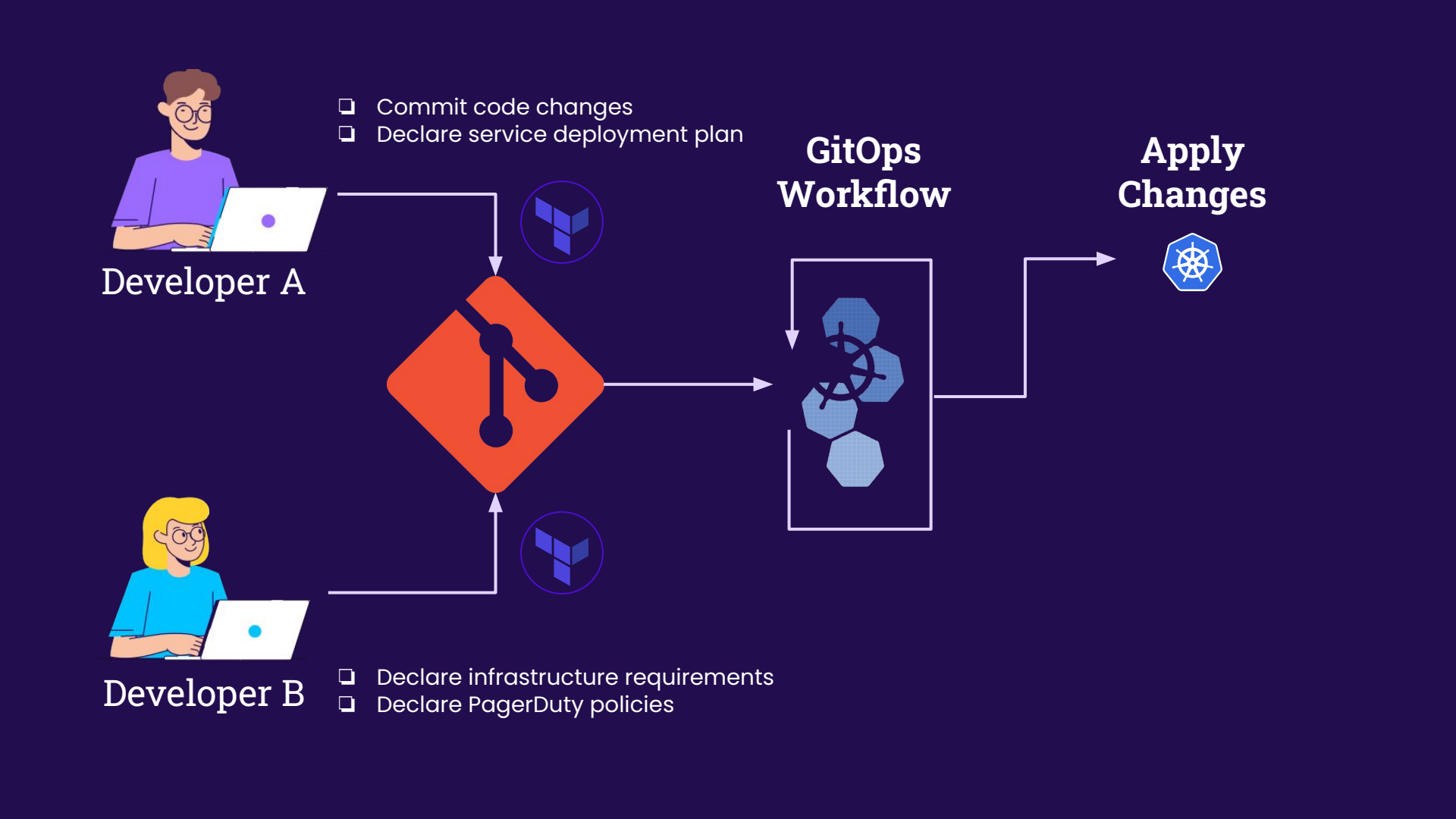
Developer A

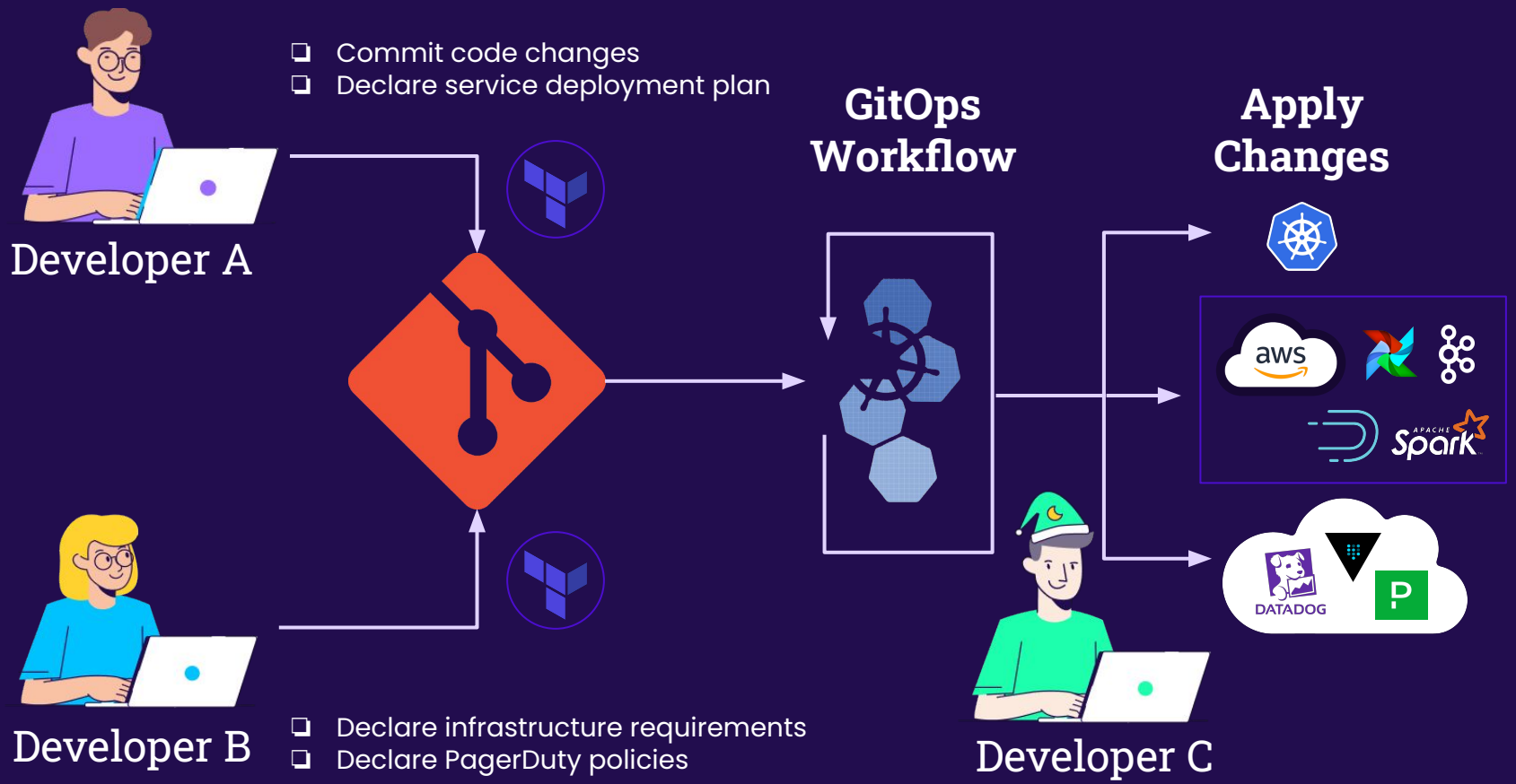
- Commit code changes
- Declare service deployment plan

GitOps Workflow



Apply Changes


Developer A
Commit code changes
Declare service deployment plan
GitOps Workflow
Apply Changes
Developer B
Declare infrastructure requirements
Declare PagerDuty policies


Developer A
❑ Commit code changes
❑ Declare service deployment plan
GitOps Workflow
Apply Changes
aws
Spark
DATADOG
Developer B
❑ Declare infrastructure requirements
❑ Declare PagerDuty policies
Developer C

Solution Principals

Auditable

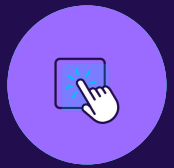
Declarative

Single Source of Truth

Terraform - git

Community Driven

kubebuilder flux



Self Serve

Visibility

- ✓ Managed resources
- ✗ Where can I find all my teams' services?
- ✗ Who is the owner of a service?
- ✗ Who is consuming from my Kafka cluster?
- ✗ Who depends on my resource?


No Visibility

Backstage

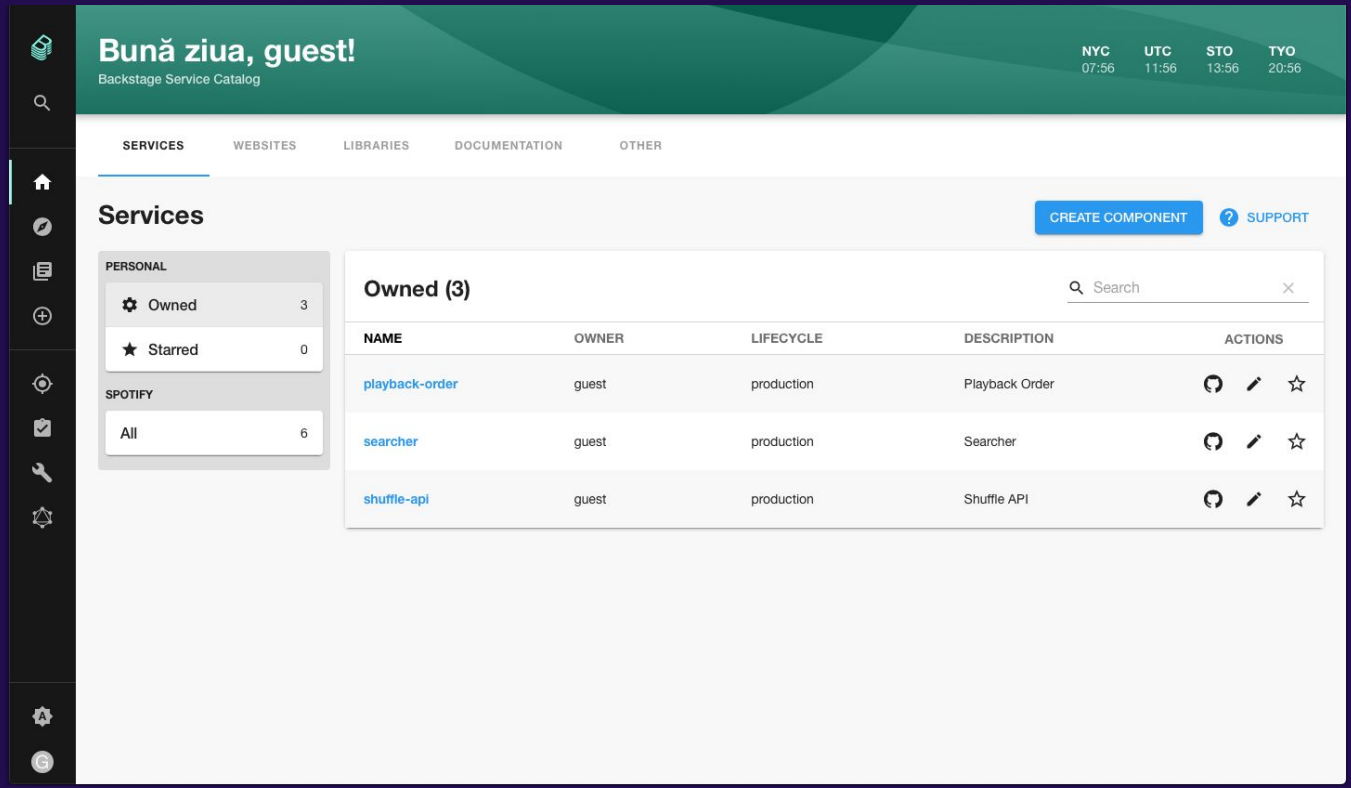
An open platform for building developer portals

Core Features

Easy To Use Software Catalog

- Manage all the software in one place
- a uniform overview
- Knowledge on all resources with metadata using YAML files

Software Catalog



Software Catalog

```yaml
apiVersion: backstage.io/v1alpha1
kind: Component
metadata:
  name: ib-test
  description: |
    A service for testing Backstage functionality. Configured for GitHub Actions and
    Sentry.
  annotations:
    github.com/project-slug: roadiehq/sample-service
    # The Sentry organization is stored in the app-config.yaml of the Backstage instance.
    sentry.io/project-slug: sample-service
    backstage.io/techdocs-ref: dir:./
    pagerduty.com/integration-key: 1cf15c07f6f440e0a8d65b7ce80be834
    snyk.io/org-name: dtuite
    snyk.io/project-ids: b6ab231c-a019-4def-a148-4a10a79d6302
    lighthouse.com/website-url: https://api.development.env.tryflux.com/lighthouse-audit
spec:
  type: service
  owner: engineering
  lifecycle: experimental
  providesApis:
    - sample-service
```

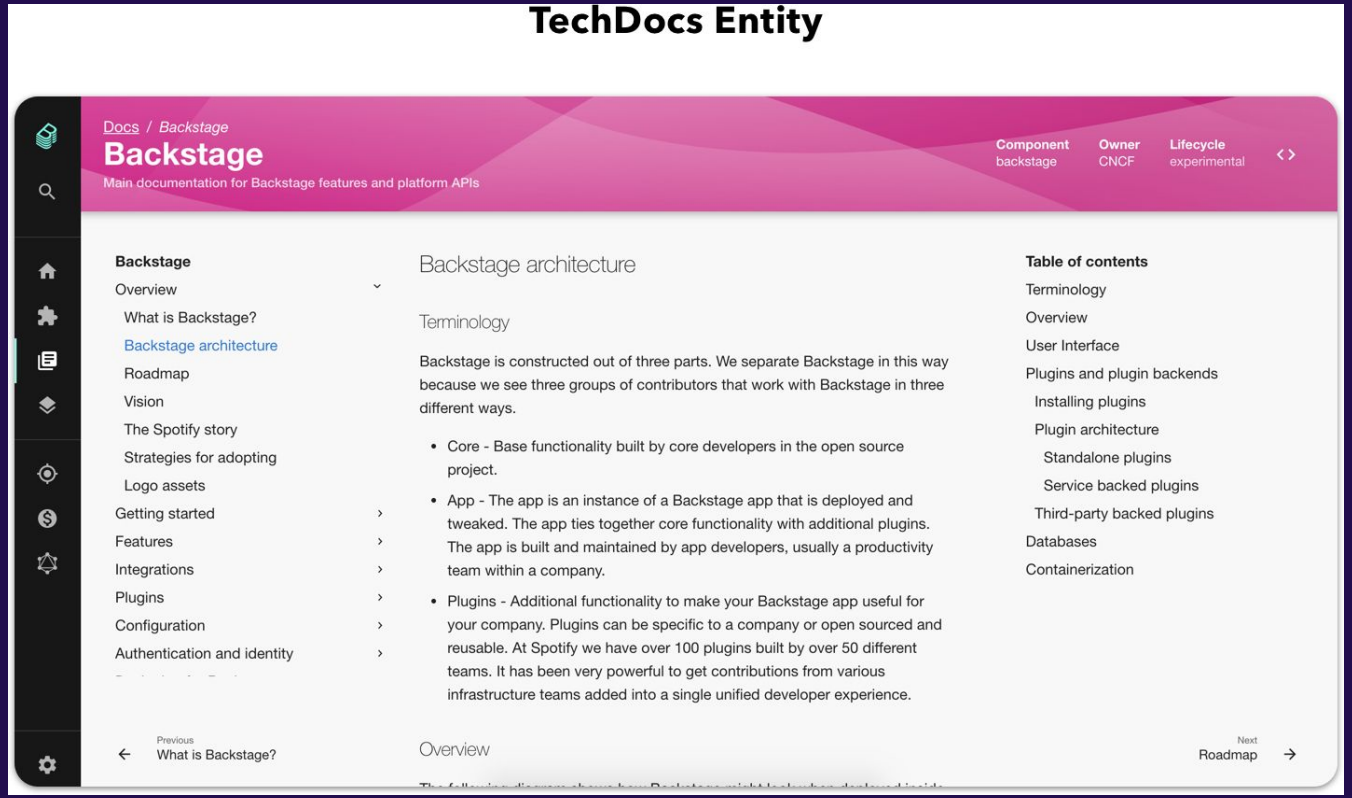
Core Features

Documentation as code

- Easy to write - markdown files
- Maintainable
- Can easily be added as part of the code
- Easy to find - docs close to infra

Documentation as code

TechDocs Entity

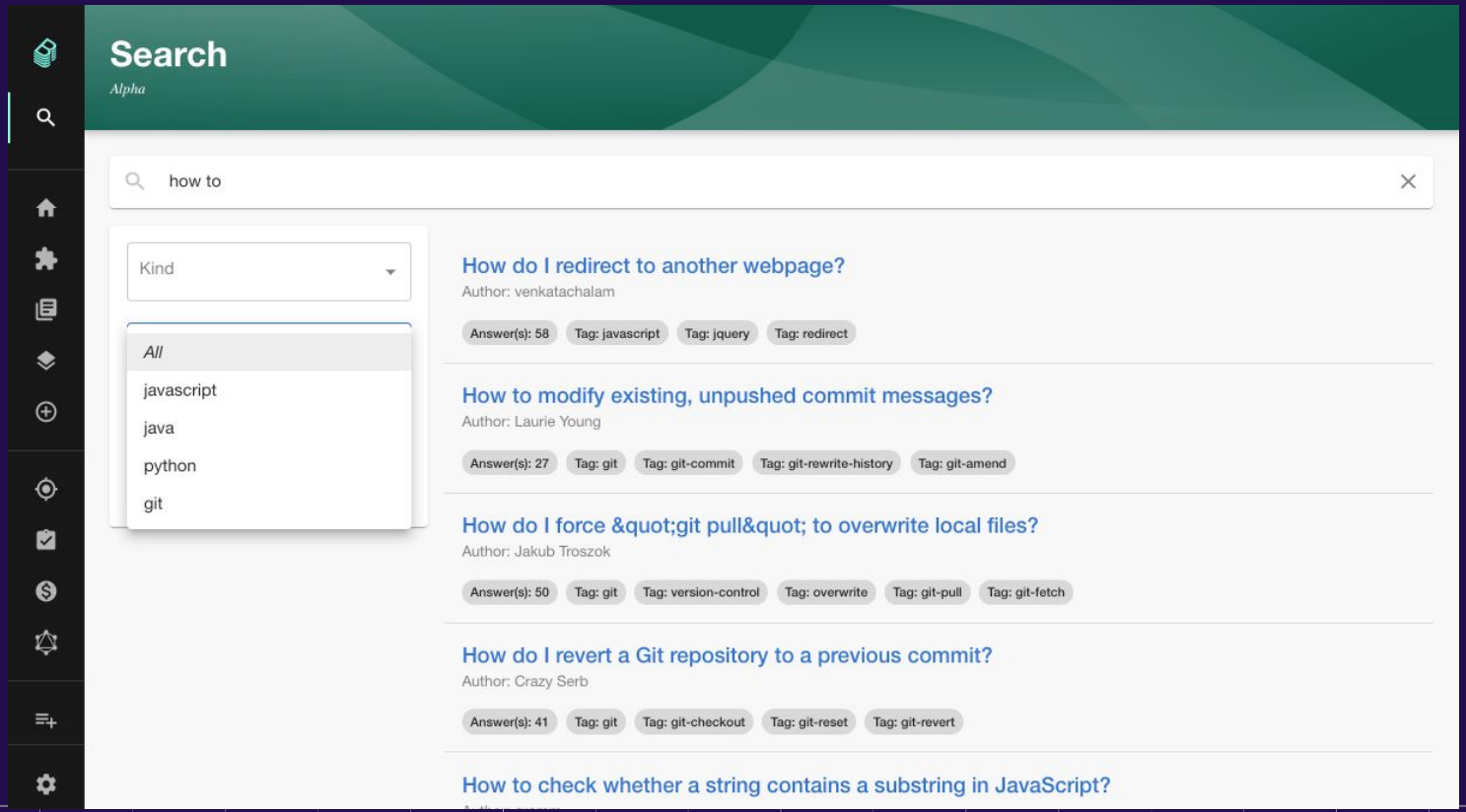


Core Features

Powerful Search

- Everything can be found
- Index all entities and properties
- Customizable

Powerful Search

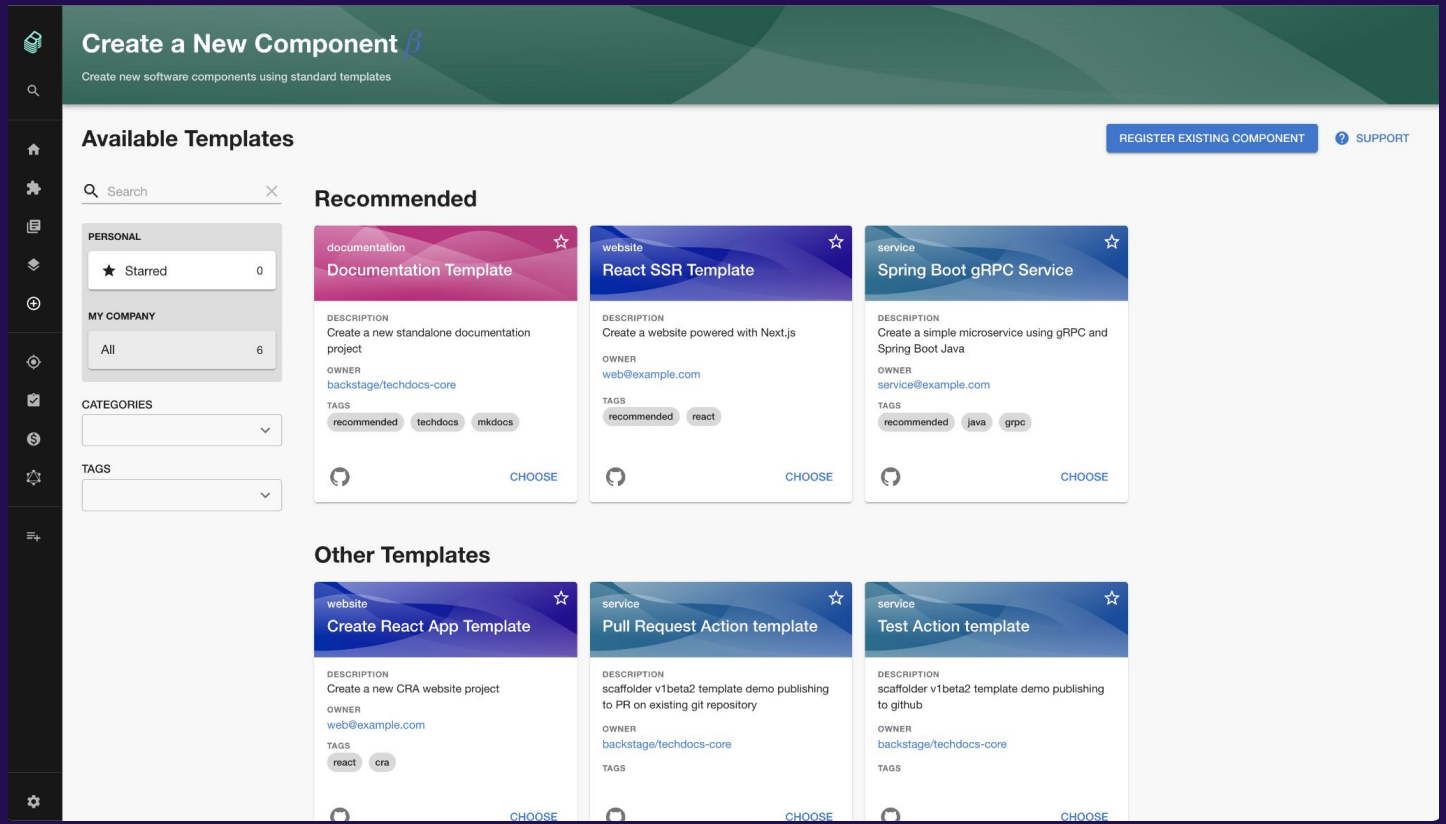


Core Features

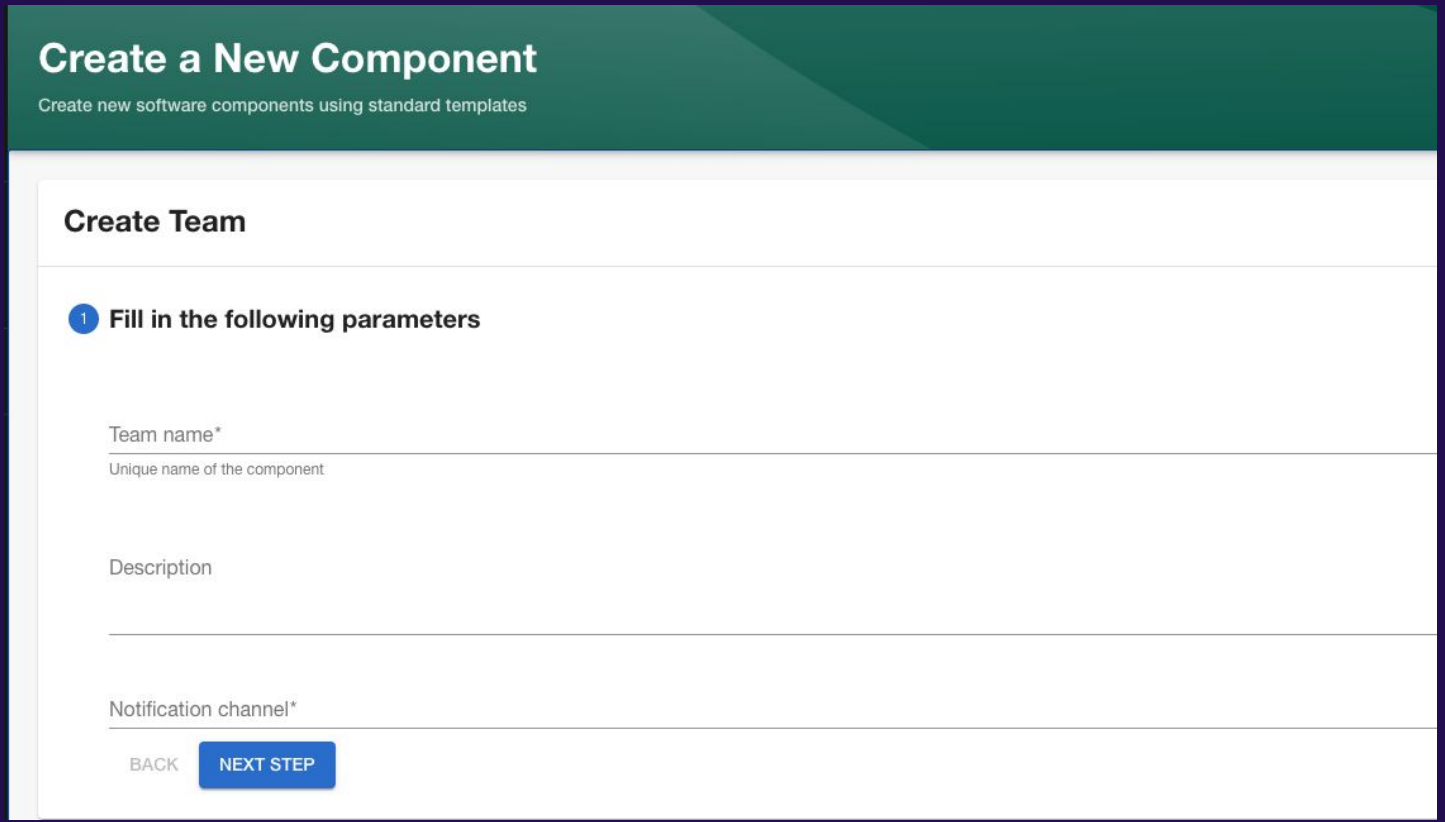
Automated Software Templates

- “get started” guide
- Easy to create using YAML file translated to code
- Customizable

Software Templates



Software Templates

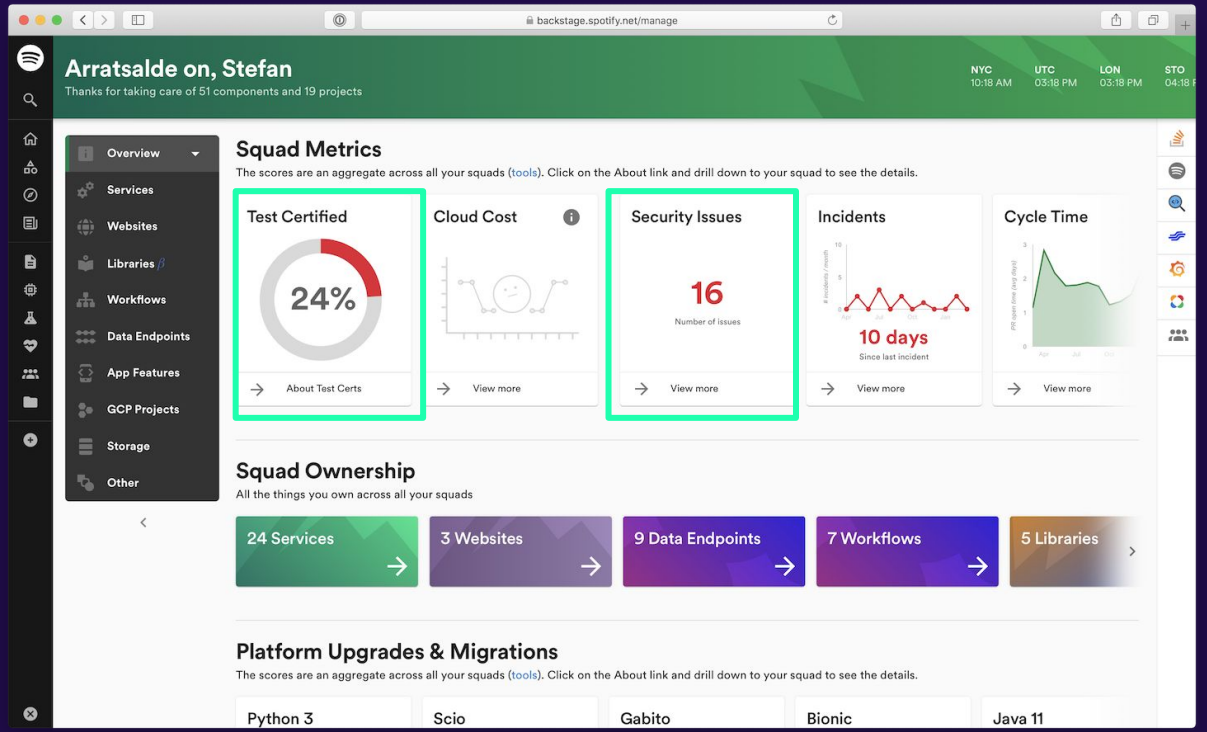


Core Features

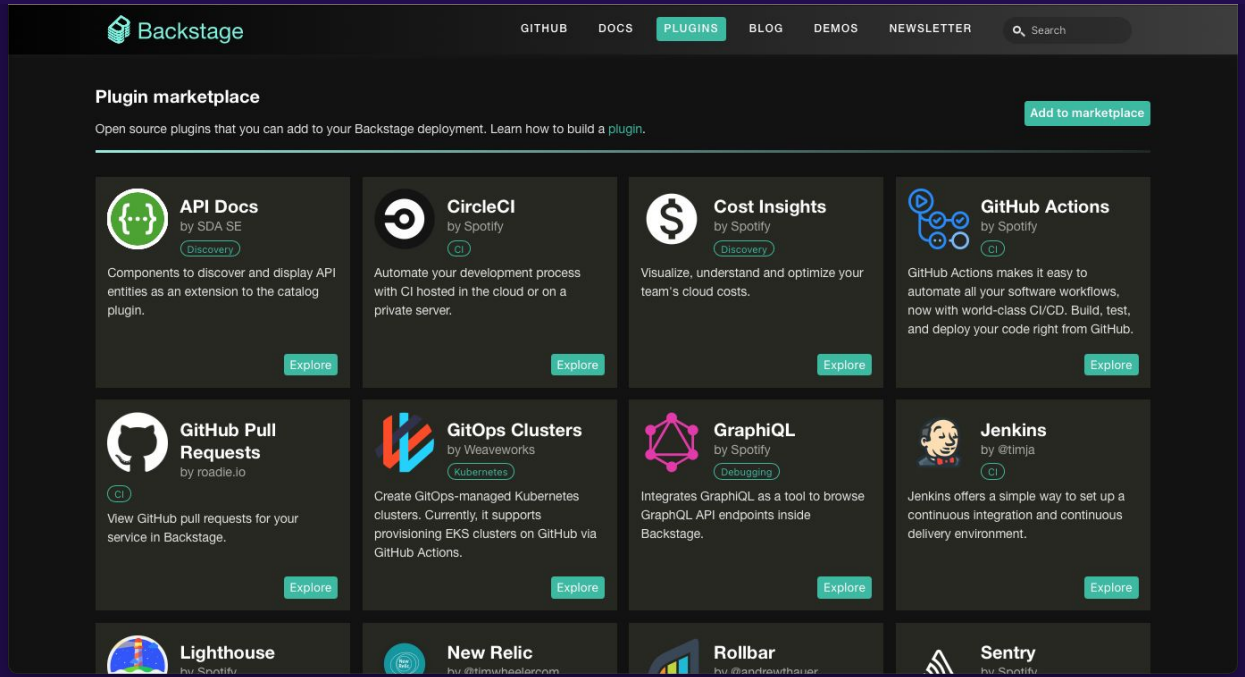
Self contained Plugins

- Functional building blocks
- Allows to expose capabilities
- Isolated code
- Marketplace

Plugins

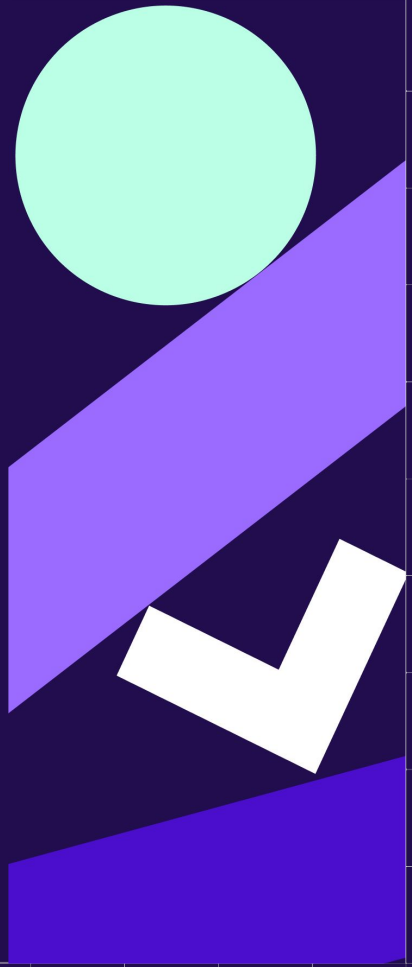

backstage.spotify.net/manage
Arratsalde on, Stefan
Thanks for taking care of 51 components and 19 projects
NYC 10:18 AM
UTC 03:18 PM
LON 03:18 PM
STO 04:18
Overview
Services
Websites
Libraries
Workflows
Data Endpoints
App Features
GCP Projects
Storage
Other
Squad Metrics
The scores are an aggregate across all your squads (tools). Click on the About link and drill down to your squad to see the details.
Test Certified
24%
About Test Certs
Cloud Cost
View more
Security Issues
16
Number of issues
View more
Incidents
10 days
Since last incident
View more
Cycle Time
View more
Squad Ownership
All the things you own across all your squads
24 Services
3 Websites
9 Data Endpoints
7 Workflows
5 Libraries
Platform Upgrades & Migrations
The scores are an aggregate across all your squads (tools). Click on the About link and drill down to your squad to see the details.
Python 3
Scio
Gabito
Bionic
Java 11

Plugins

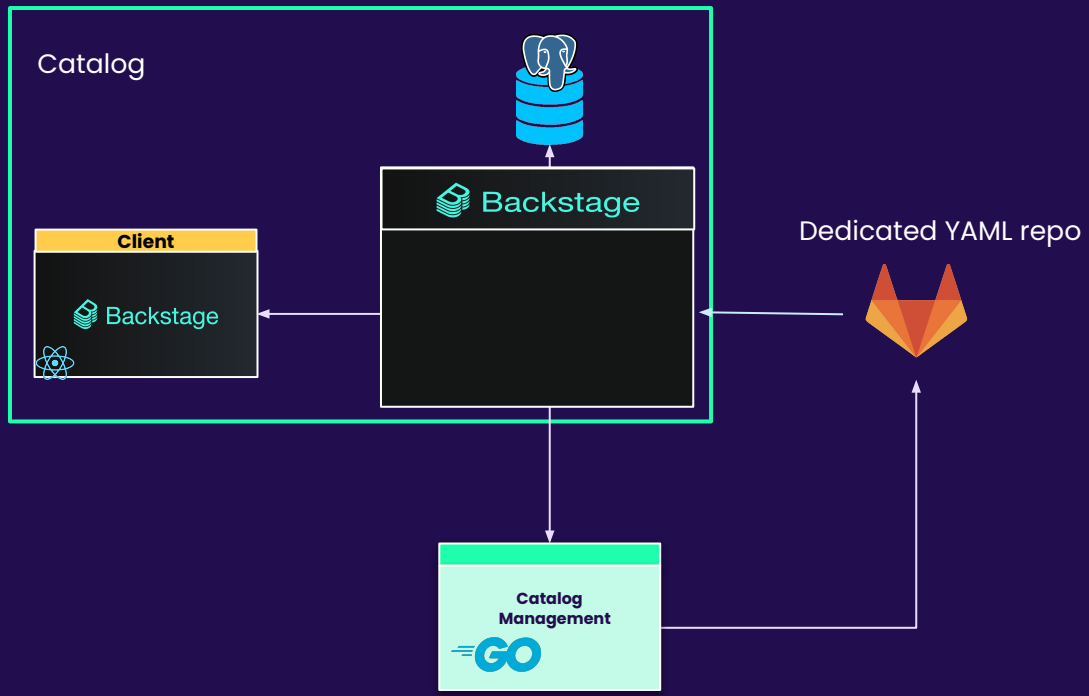

Backstage
GITHUB
DOCS
PLUGINS
BLOG
DEMOS
NEWSLETTER
Search
Plugin marketplace
Open source plugins that you can add to your Backstage deployment. Learn how to build a plugin.
Add to marketplace
API Docs
by SDA SE
Discovery
Components to discover and display API entities as an extension to the catalog plugin.
Explore
CircleCI
by Spotify
CI
Automate your development process with CI hosted in the cloud or on a private server.
Explore
Cost Insights
by Spotify
Discovery
Visualize, understand and optimize your team's cloud costs.
Explore
GitHub Actions
by Spotify
CI
GitHub Actions makes it easy to automate all your software workflows, now with world-class CI/CD. Build, test, and deploy your code right from GitHub.
Explore
GitHub Pull Requests
by roadie.io
CI
View GitHub pull requests for your service in Backstage.
Explore
GitOps Clusters
by Weaveworks
Kubernetes
Create GitOps-managed Kubernetes clusters. Currently, it supports provisioning EKS clusters on GitHub via GitHub Actions.
Explore
GraphiQL
by Spotify
Debugging
Integrates GraphiQL as a tool to browse GraphQL API endpoints inside Backstage.
Explore
Jenkins
by @timja
CI
Jenkins offers a simple way to set up a continuous integration and continuous delivery environment.
Explore
Lighthouse
New Relic
Rollbar
Sentry

One stop shop for all

- One system to externalize all platform capabilities
- Resources standardization
- Distributed plugin development done by platform team owners;
  - Rapid development
  - Accountability
  - Ownership



Backstage Architecture in AppsFlyer



Solution Principals

Auditable

Declarative

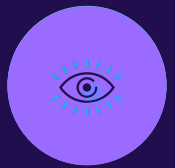
Single Source of Truth

Terraform - git

Community Driven

kubebuilder flux

Self Serve



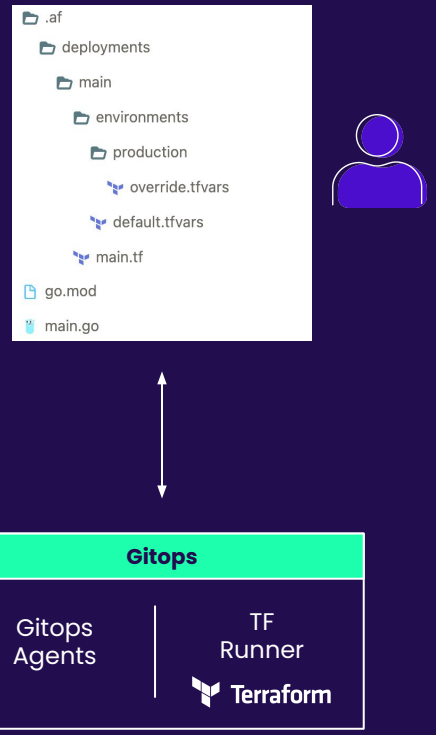
Visibility

All resources needs to be managed using TF and appear in Backstage

Resources in TF

All resources managed by TF in .af

- Service Deployment Manifests
- Service Resources (PG Policies, DD Monitors)
- Infrastructure (RDS)



All resources should be managed using TF and appear in Backstage

Resources in Backstage

Service

Lambda

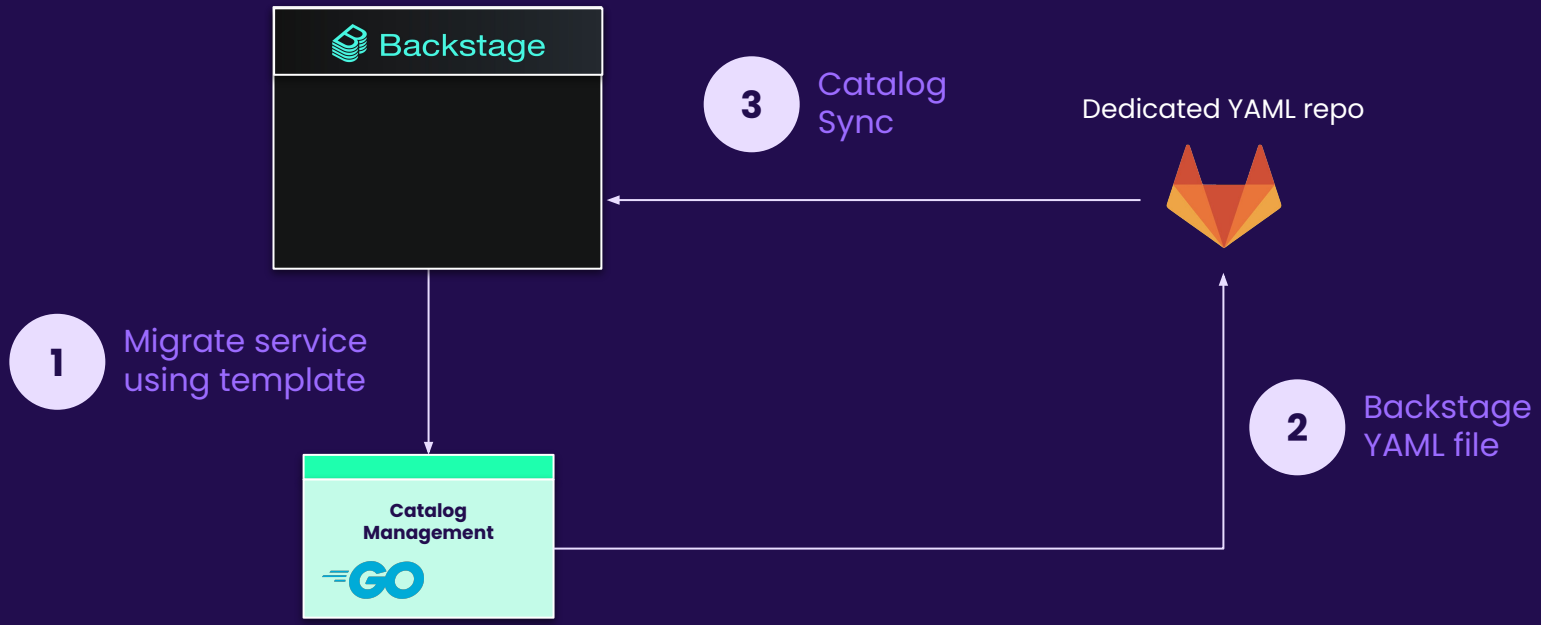
S3 Bucket

Pagerduty Service

Datadog Monitor

EC2

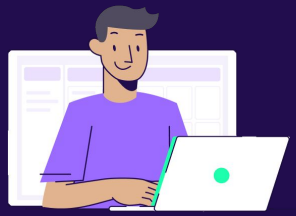
Migrating Service to Backstage



All resources should be **managed using TF** and **appear in Backstage**

Is it enough??


gitops
Terraform
Backstage
YAML



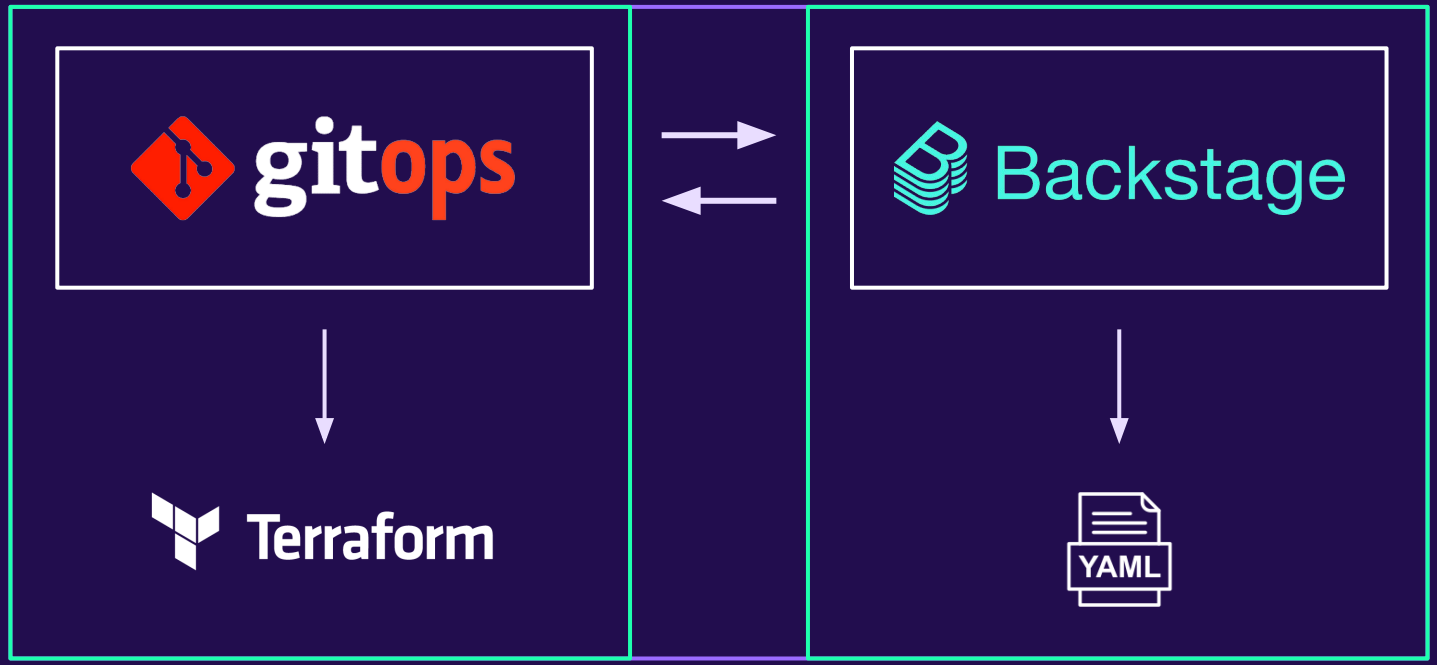
Commit lamda.tf

Creates new lamda in catalog

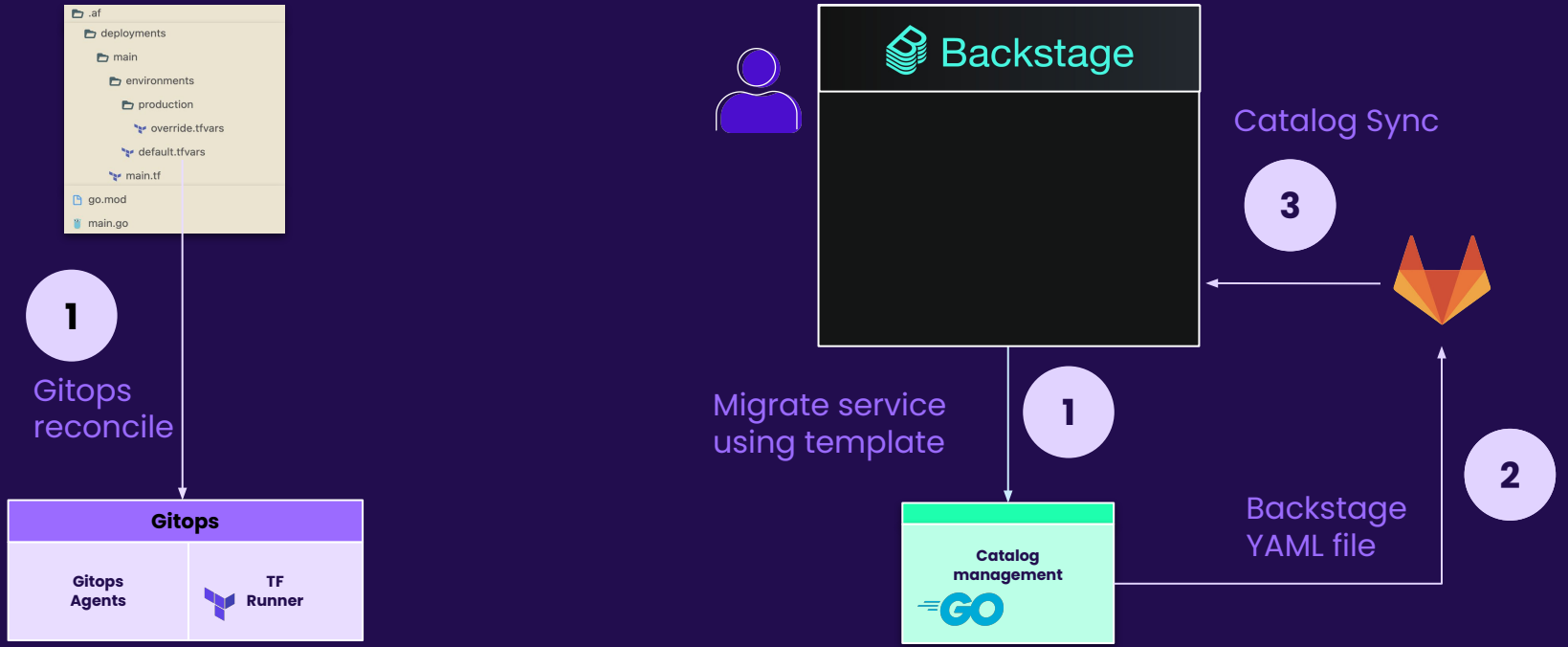
Backstage

Gitops Flow

All resources should be managed using TF, appear in Backstage, and always sync


gitops
Backstage
Terraform
YAML

Migrate service using Backstage



What is a service in TF?

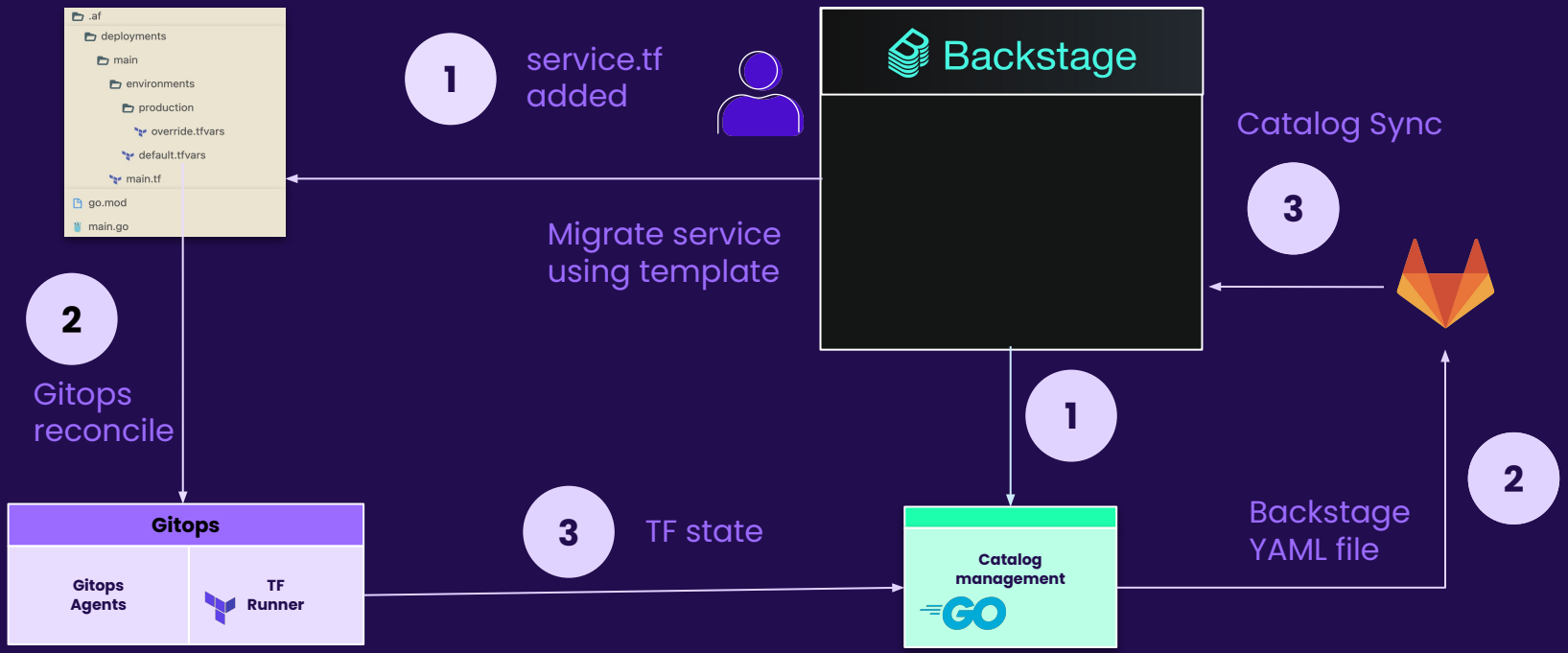
~~Service == Deployment~~

- Service can be described before deployment
- Contains metadata such as:
  - Language
  - Description
  - PagerDuty Service
  - Owner
  - etc...

service.tf 254 bytes

```
module "service" {
  source = "git::https://gitlab.appsflyer.com/devops/terraform/modules/service?ref=v1.0.0"
  service_name = "af-service"
  language = "go"
  description = "an Appsflyer service"
  pagerDutyServiceID = "abc"
  owner = "Platform Team"
}
```

Migrate service using Backstage



01

Migrate service
Create the TF file in the user repository

02

Gitops validation
Uses a web socket to listen to the gitops process until it's done and creates the YAML file

03

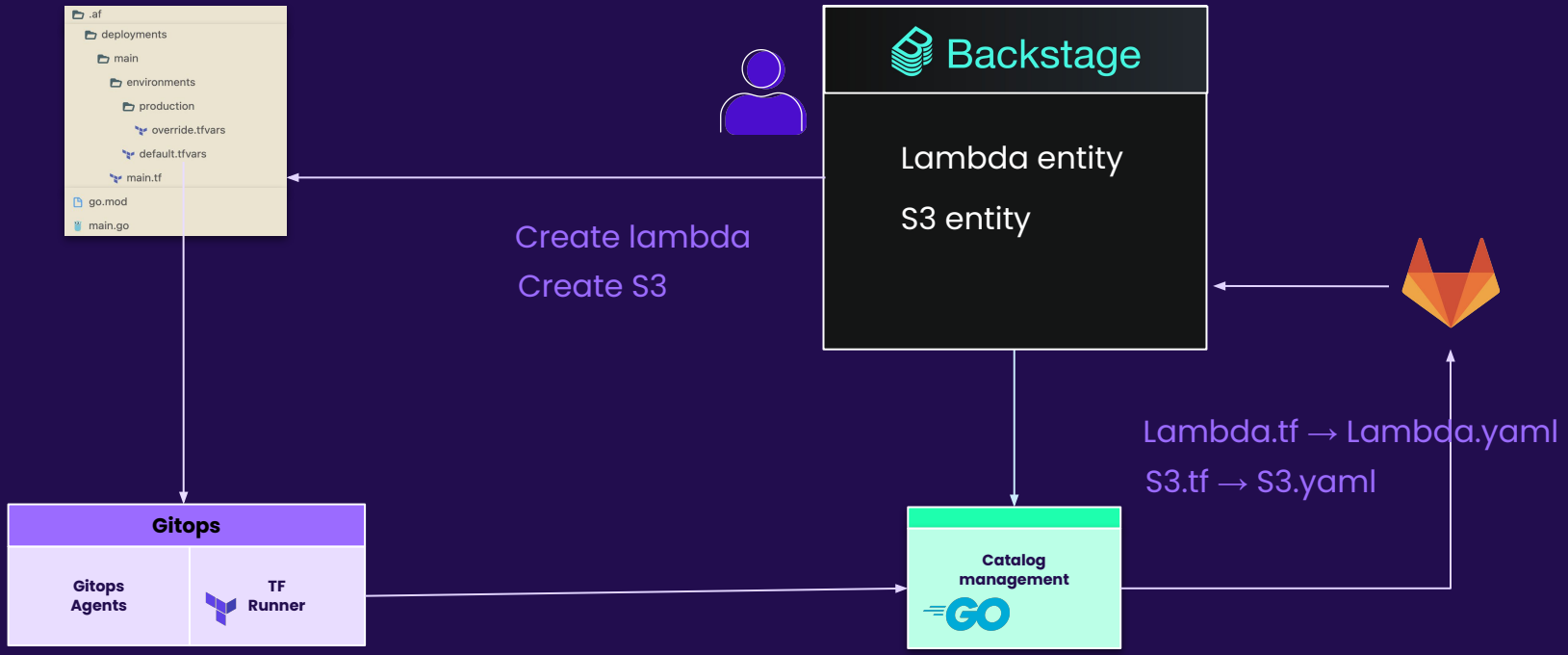
Backstage validation
Wait until Backstage synced the new created YAML file into the catalog

Migration Stats

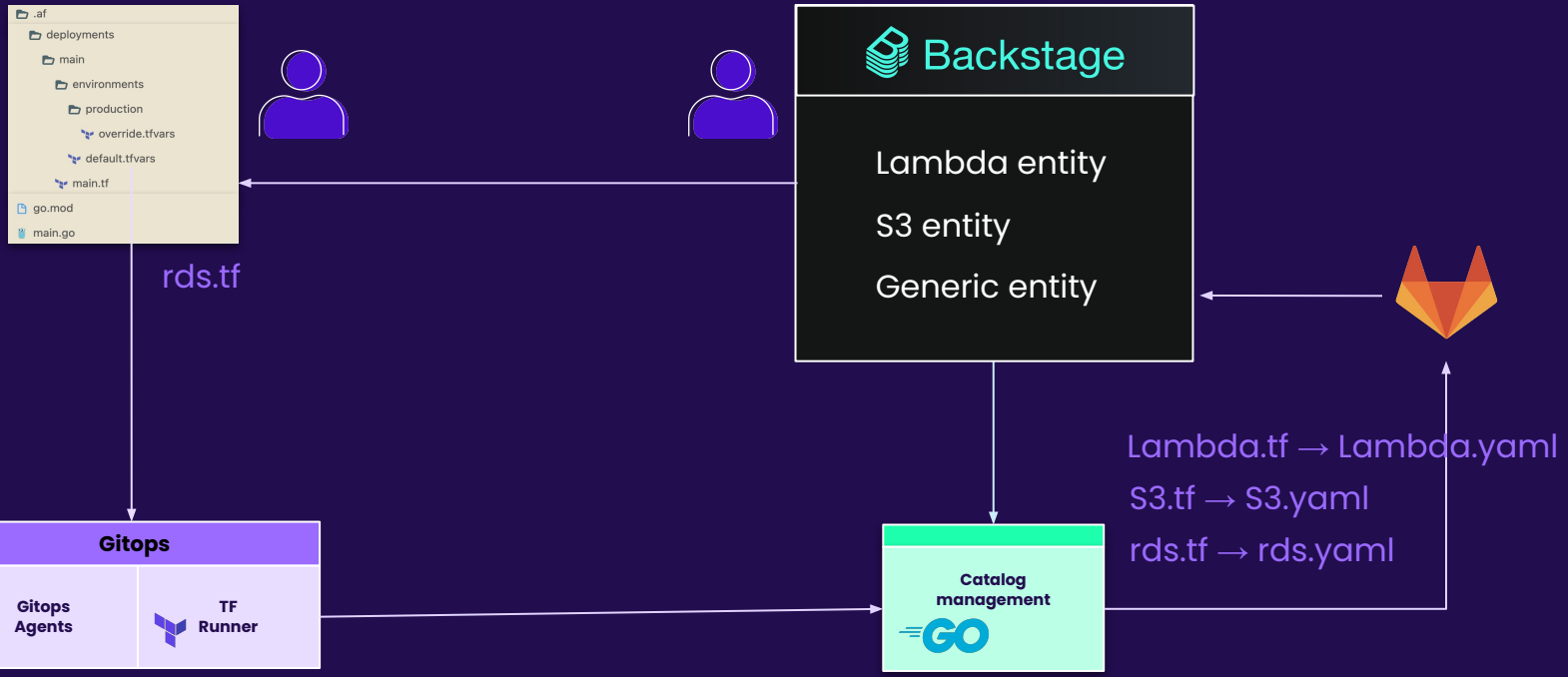
+1,000	+45	~5
Services	Teams	Months


WE LIKE IT, WE REALLY LIKE IT
WE WANT MORE!

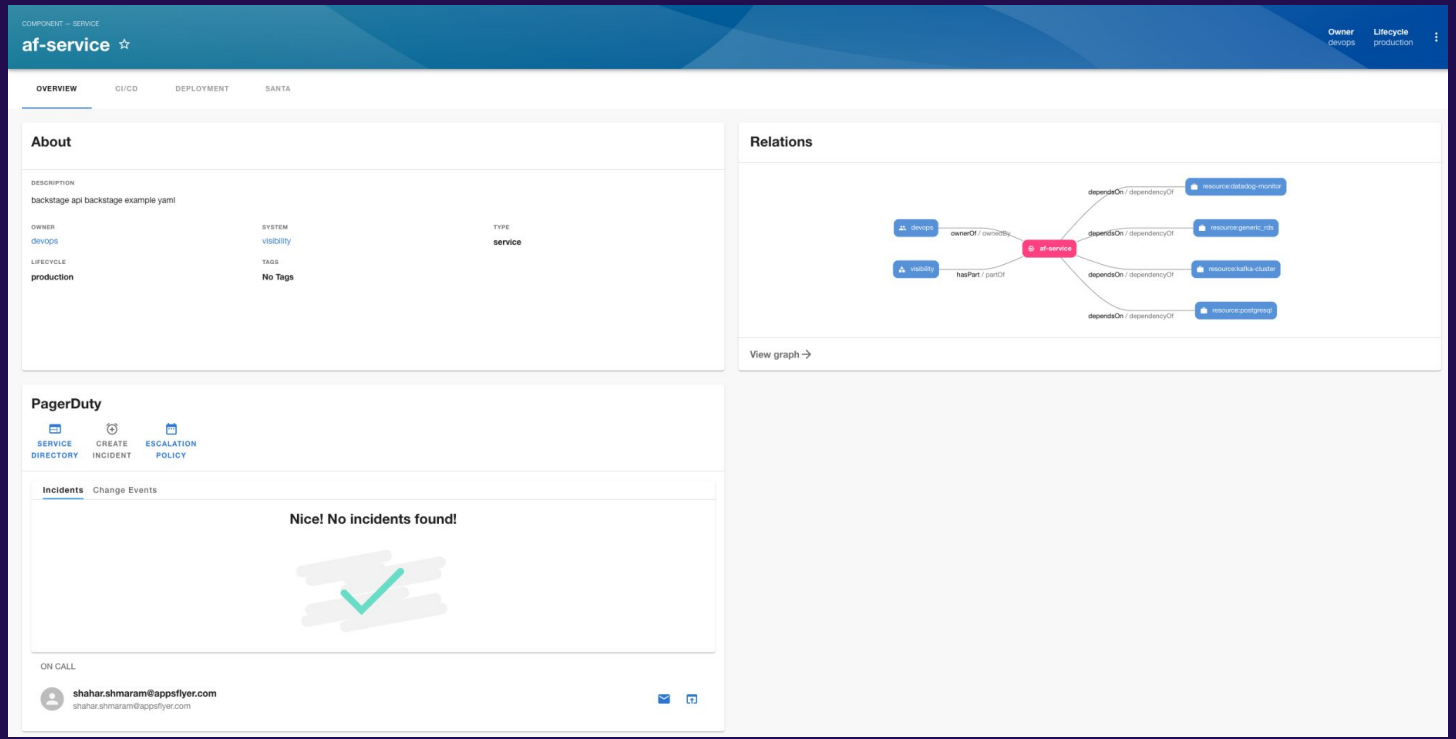
Support additional resources using Backstage



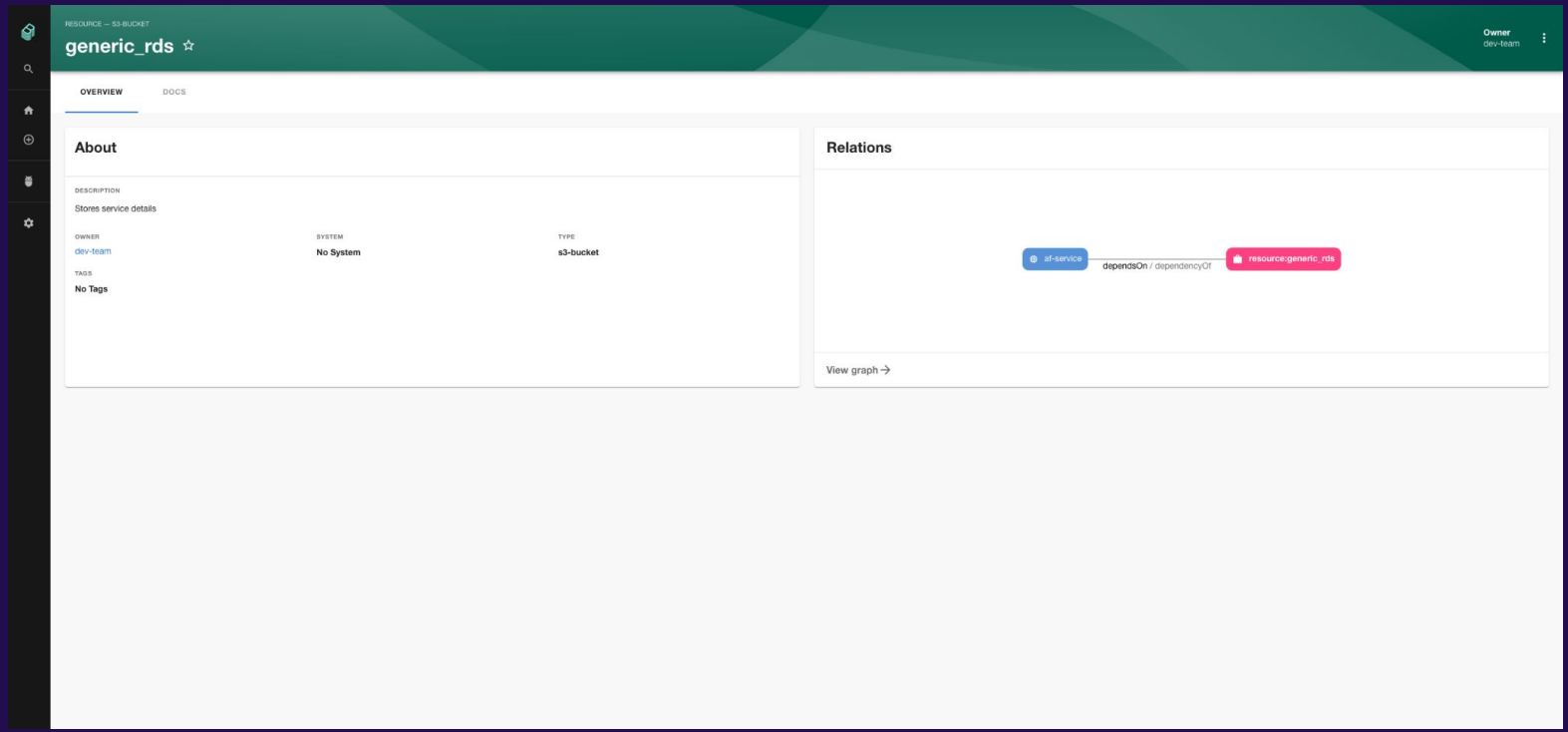
Support new TF resources



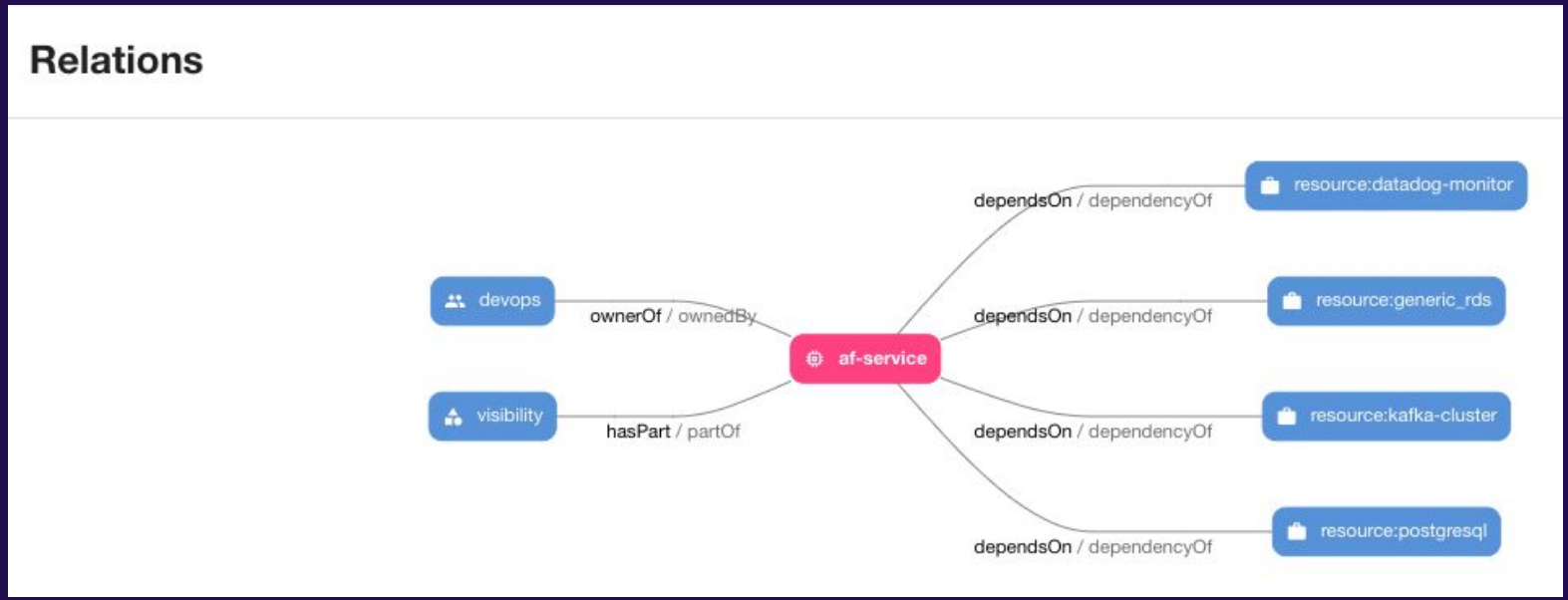
Non generic entity (service)



Generic entity (RDS)

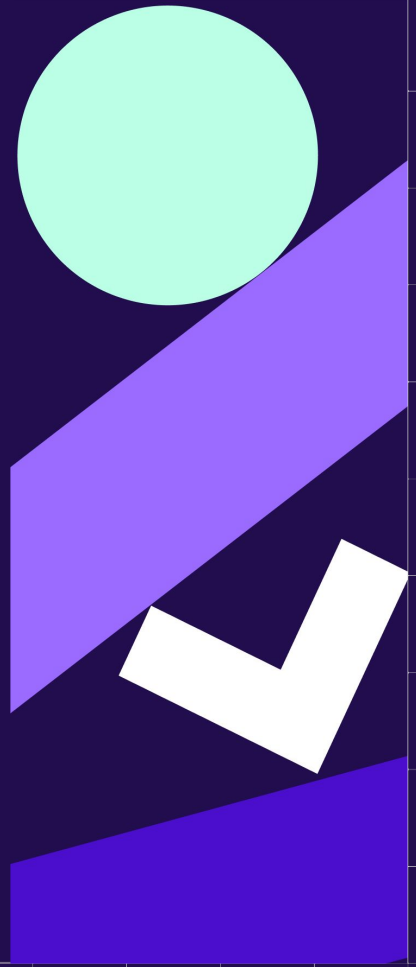


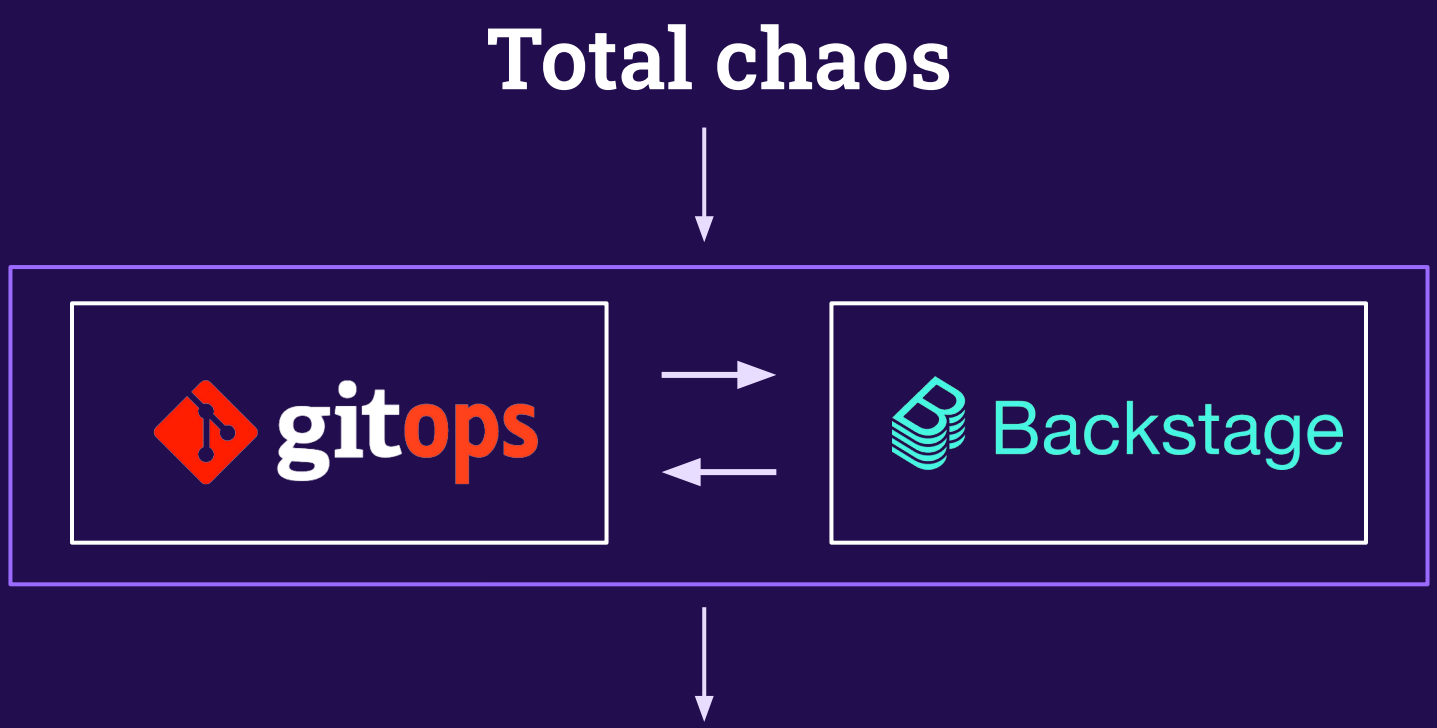
Entity representation



Our Main takeoffs:

- Change is hard
- Developer experience
- Always manage your resources




Total chaos
gitops
Backstage
Managed resources

Thank you

Q&A

Feedback form