Backstage

Happy developers make happy code

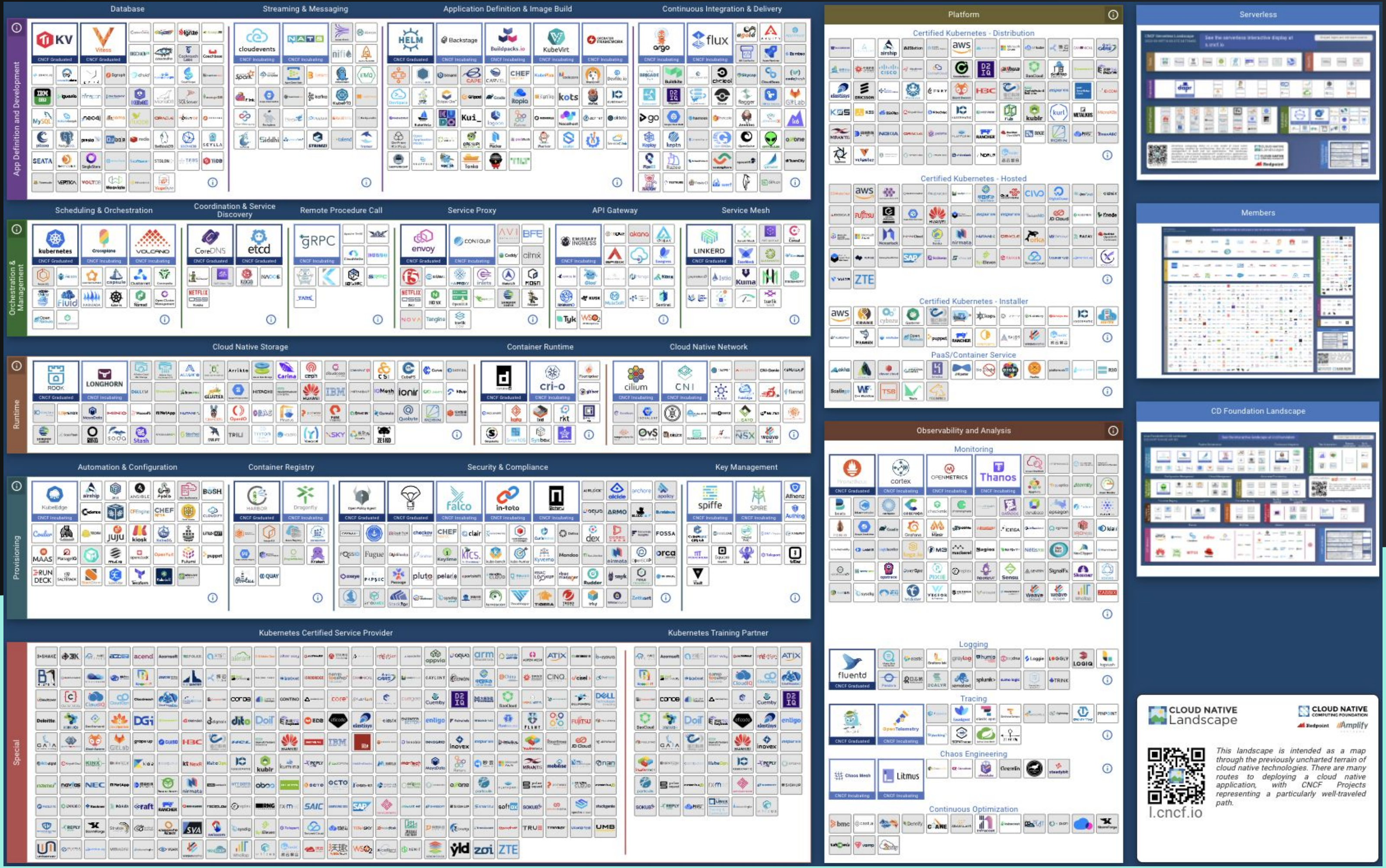
Backstage is an open platform for building developer portals.

Created at

Spotify®

Donated to

CLOUD NATIVE
COMPUTING FOUNDATION


Database
Streaming & Messaging
Application Definition & Image Build
Continuous Integration & Delivery
App Definition and Development
Scheduling & Orchestration
Coordination & Service Discovery
Remote Procedure Call
Service Proxy
API Gateway
Service Mesh
Orchestration & Management
Cloud Native Storage
Container Runtime
Cloud Native Network
Runtime
Automation & Configuration
Container Registry
Security & Compliance
Key Management
Provisioning
Kubernetes Certified Service Provider
Kubernetes Training Partner
Special
Platform
Certified Kubernetes - Distribution
Certified Kubernetes - Hosted
Certified Kubernetes - Installer
PaaS/Container Service
Observability and Analysis
Monitoring
Logging
Tracing
Chaos Engineering
Continuous Optimization
Serverless
Members
CD Foundation Landscape
CLOUD NATIVE Landscape
CLOUD NATIVE COMPUTING FOUNDATION
Redpoint
Amplify
l.cncf.io

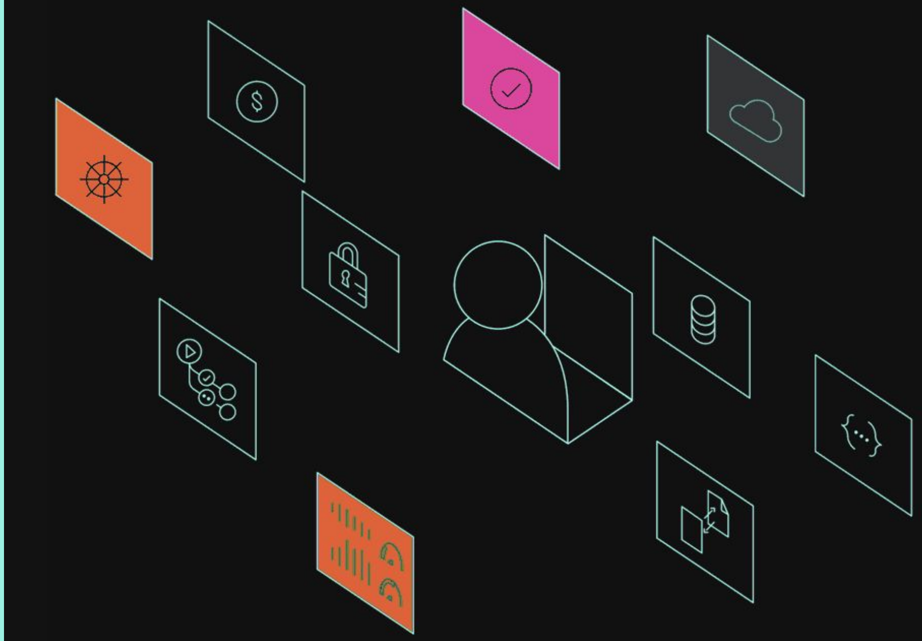
This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.

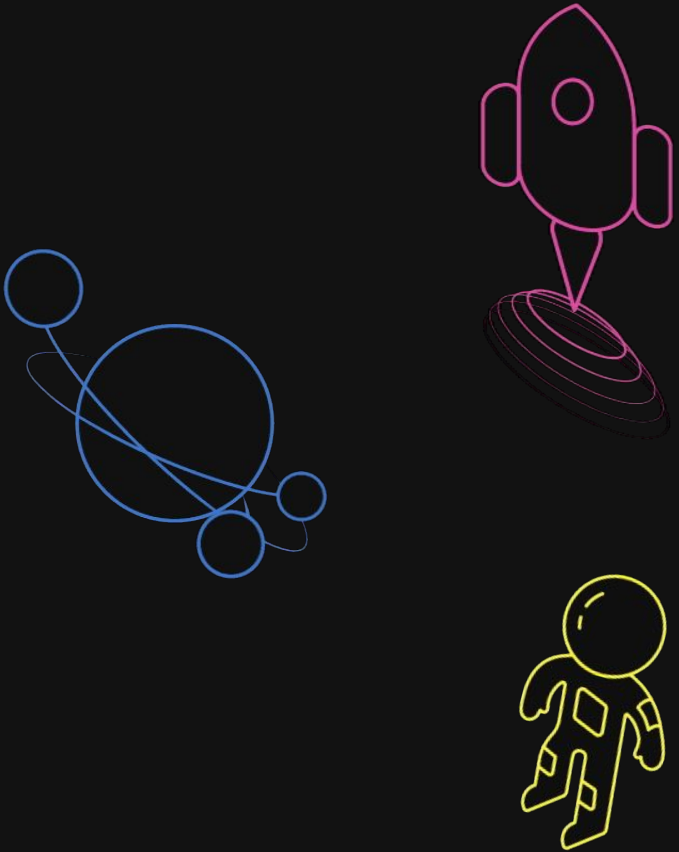
A developer portal = one frontend for your entire infrastructure

Unifies all your tooling, services, apps, data, docs with a single, consistent UI

Makes sense of everything in your ecosystem, regardless of how and where individual components are running

Let developers focus on what they do best (leading to much less activity in #aaargh Slack channel)





Backstage lets any developer:

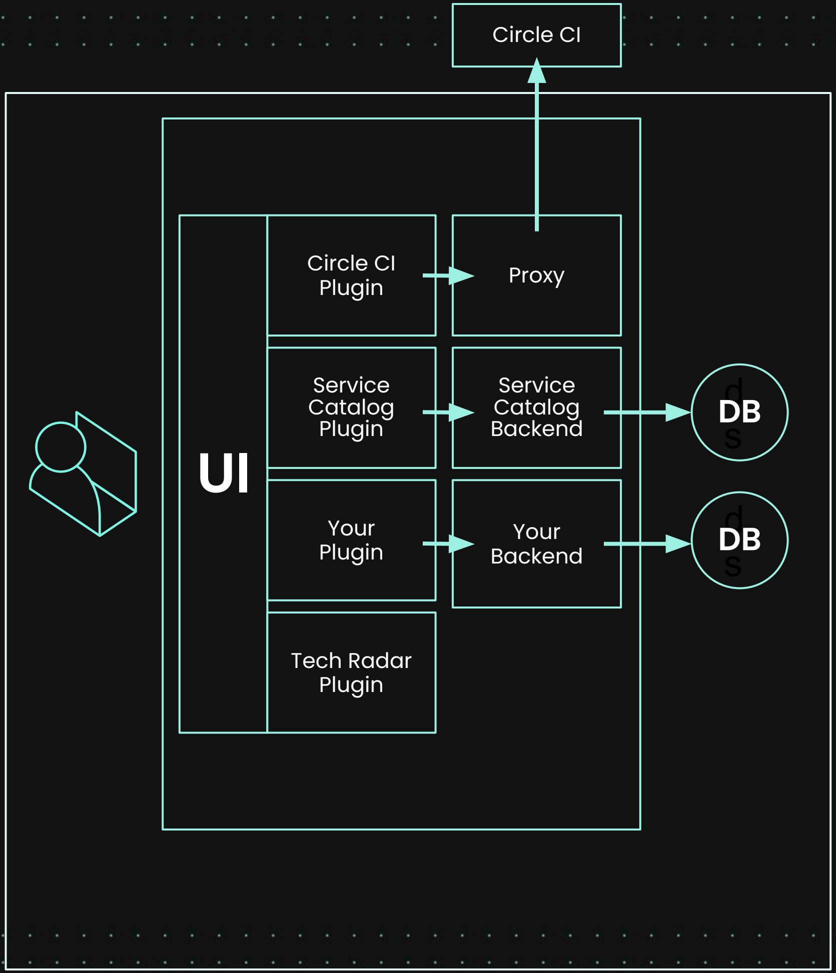
- ❑ **Create** new software in seconds, aligned to your best practices
- ❑ **Manage** all the software they own in one centralized location
- ❑ **Explore** the entire software ecosystem, enabling collaboration across your org

Backstage has a customizable, extensible plugin architecture

❑ Built with modern technologies and common frameworks.

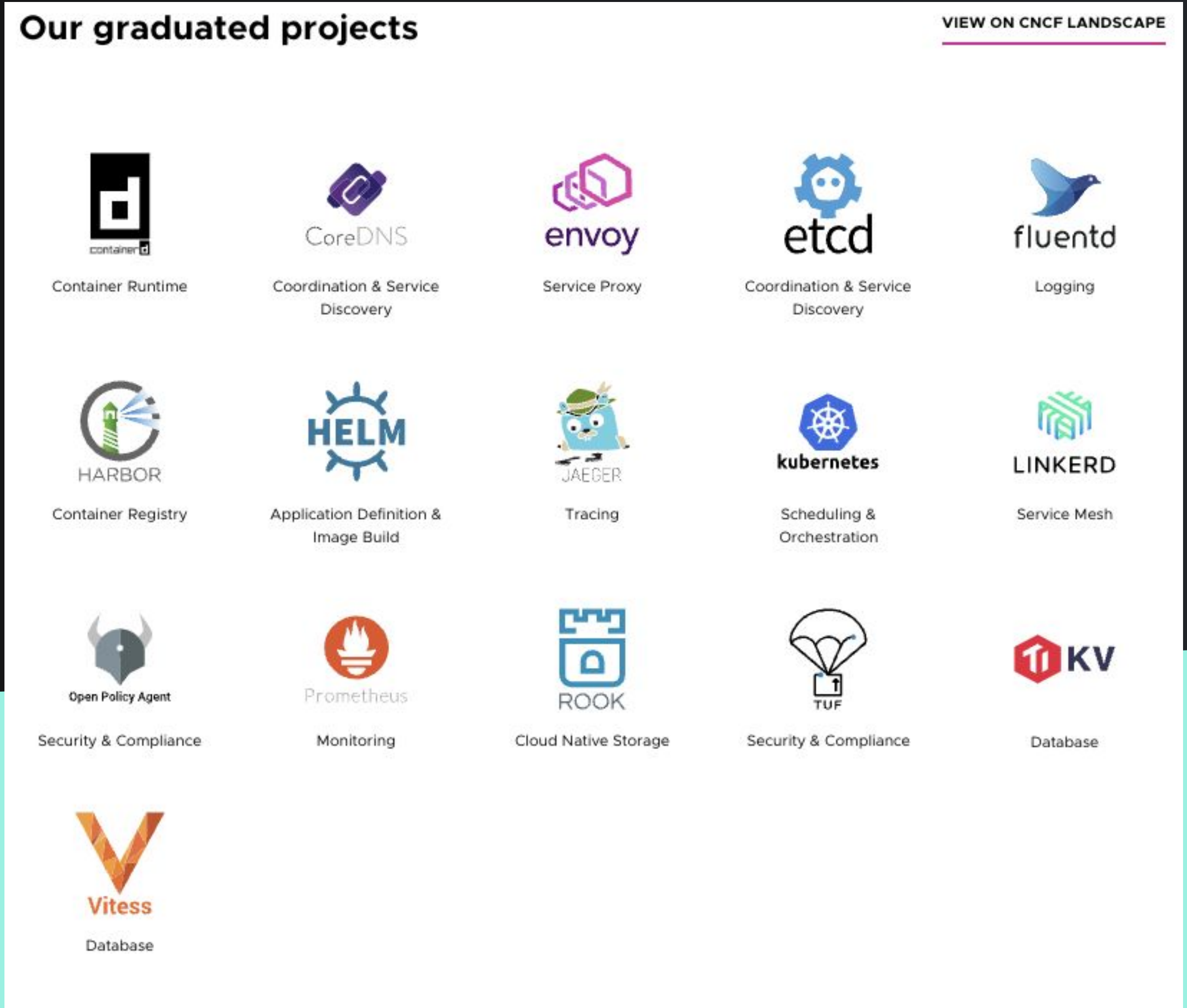
❑ Makes it easy to develop for and contribute to your dev portal.

❑ Cloud-agnostic and vendor-neutral.



A true platform, inside and out

- ❑ A small team maintains the core features of the Backstage app
- ❑ Different platform teams build and maintain all the other plugins
- ❑ Feature teams use the plugins to build and maintain their software (and provide feedback directly to the plugin owners)



Our graduated projects

VIEW ON CNCF LANDSCAPE

containerd
Container Runtime

CoreDNS
Coordination & Service Discovery

envoy
Service Proxy

etcd
Coordination & Service Discovery

fluentd
Logging

HARBOR
Container Registry

HELM
Application Definition & Image Build

JAEGER
Tracing

kubernetes
Scheduling & Orchestration

LINKERD
Service Mesh

Open Policy Agent
Security & Compliance

Prometheus
Monitoring

ROOK
Cloud Native Storage

TUF
Security & Compliance

TiKV
Database

Vitess
Database

Backstage is owned and led by a diverse, rapidly growing open source community

16k+
Stars on GitHub

2k+
Project forks

500+
Contributors (+10 per week)

60+
Open source plugins

Backstage has been adopted by 100+ companies across industries

NETFLIX

Roku

PELOTON

Expedia

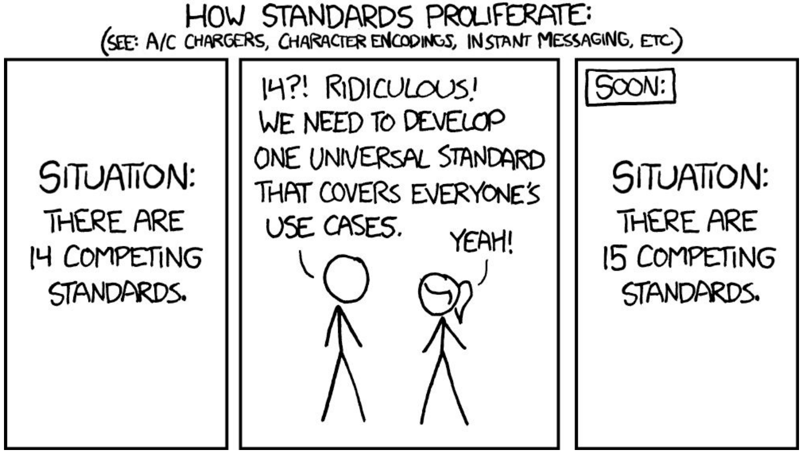
American Airlines

TELUS

TOYOTA

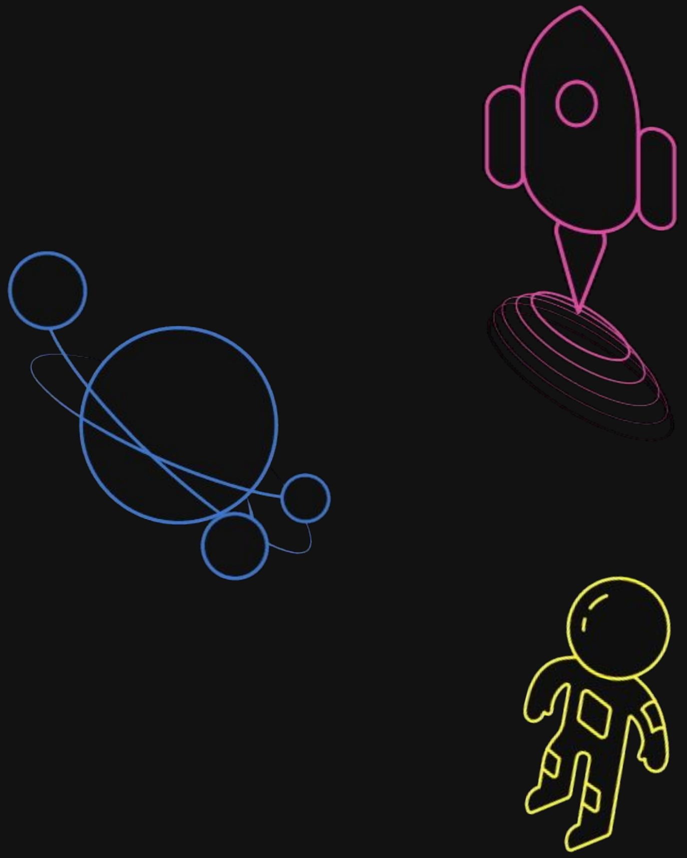
Core Philosophy

- ❑ **Backstage is the interface**
  - ❑ Aggregator, rarely the source of truth
- ❑ **Autonomy**
  - ❑ We build it together; no big central platform team
  - ❑ The team closest to the problem owns the plugin
- ❑ **Ownership**
  - ❑ There should be a single point of contact for any software
  - ❑ These owners are empowered and responsible for metadata


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

Quick Demo

http://demo.backstage.io



Backstage lets any developer:

❑ **Create** new software in seconds, aligned to your best practices

❑ **Manage** all the software they own in one centralized location

❑ **Explore** the entire software ecosystem, enabling collaboration across your org

Encouraging Consistency:
Software Templates

❑ Encode your org’s technical best practices

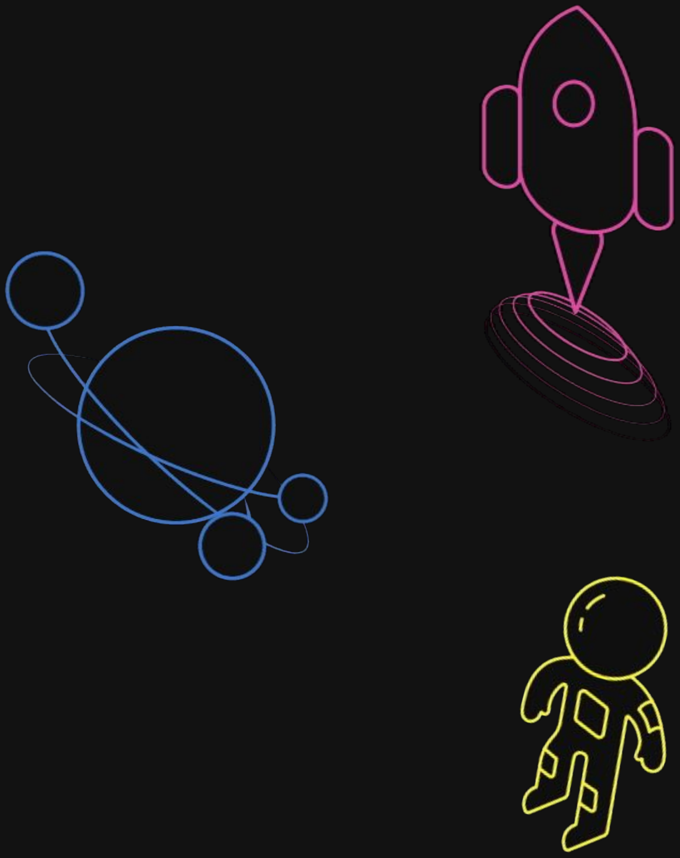
❑ Also automatically register new software

❑ Hand-crafted yaml not required

Scaffolder Demo

- Log into gcp console as ben
- Demo
  - Run it - mention nginx
  - Show template code
  - Show repo
  - Show CI/CD, scroll around it
  - Click on the workflow run to show what's it doing
  - Click on kubernetes tab to show deploying to kubernetes
  - Show that BS isn't replacing your existing stuff. Can still look at GCP console, kubectl, github, etc.
- Demo with Grafana dashboard
  - Use other template
  - Show template code, show create.ts under backend/src/sc/ac/gra - "grafana:dashboard:create", difference in the action
  - Show new "create graphs in grafana" step and new Grafana link
  - Show dashboard, no-data until it deploys, and then magic

Demo



Backstage lets any developer:

❑ **Create** new software in seconds, aligned to your best practices

❑ **Manage** all the software they own in one centralized location

❑ **Explore** the entire software ecosystem, enabling collaboration across your org

The Software Model

- ❑ **Core entity kinds**
  - ❑ Domain, System, Component, API, Resource, User, Group
- ❑ **For each kind, many types**
  - ❑ Service, Website, Library, GraphQL, ML Model…
- ❑ **Relationship graph**
  - ❑ ownedBy, providesApi, memberOf, dependencyOf…
- ❑ **Annotations**

Getting Declarative: catalog-info.yaml

❑ Lives with the source code

❑ Uses the Kubernetes object model

❑ Extensible

❑ Optional, there are other ways

Example: catalog-info.yaml

```yaml
apiVersion: backstage.io/v1alpha1
kind: Component
metadata:
 name: petstore
 description: The Petstore is an example service that provides an OpenAPI spec.
 links:
   - url: https://github.com/swagger-api/swagger-petstore
     title: GitHub Repo
     icon: github
spec:
 type: service
 lifecycle: experimental
 owner: team-c
 providesApis:
   - petstore-api
```

Check us out at the Project Pavillion (Booth #2)