TL;DR

The interest in Backstage keeps growing, with many folks joining the Backstage Kubecon sessions, visiting the Backstage booth, and joining BackstageCon.

Spotify released a set of commercial plugins for Backstage, including the following:

Spotify Commercial Plugins for Backstage

Soundcheck: See how well your components measure up to engineering best practices and standards at a glance.

RBAC: Manage access and protect your data in Backstage flexibly and easily. Role-based access control (RBAC) should be as easy to manage as possible — because the simpler it is to do, the better protected your data will be.

Skill Exchange: Share expertise across your org by creating an internal marketplace for on-the-job learning opportunities and promoting a culture of self-led growth. Talent, knowledge, and experience can often get trapped inside organizational silos.

Pulse: Find out how your R&D teams really feel. Pulse is an R&D survey framework that lets you track key productivity and satisfaction metrics, analyze the anonymized response data to discover trends and visualize those insights (all within Backstage).

Insights: Surveys are great for capturing how your people feel. But you also want to know how they're actually behaving. The Insights plugin is your window into Backstage usage at your organization.

Core Technical talks

What is This Tech Insights Anyways? - Jussi Hallila & Irma Solakovic, Roadie Portside Ballroom (Room 260)

- https://www.youtube.com/watch?v=Im0n6lNiI_4

The Evolution of Backstage Backends - Patrik Oldsberg & Johan Haals, Spotify

- https://www.youtube.com/watch?v=ZXF7uxrEBVY

Measuring Developer Happiness: Analytics for Backstage - Eric Peterson, Spotify

- https://www.youtube.com/watch?v=DtGV2zR2z6Y

Other technical talks

Leading Architectural Change with Backstage - Andy Hoffman, Caribou Portside Ballroom (Room 260)

- https://www.youtube.com/watch?v=6Ss1e-9X_JY

A Picture is Worth a Thousand Words: Bring Data Visualization in Backstage - Olivier Liechti, Avalia Systems

- https://www.youtube.com/watch?v=riOXPNyp0e0

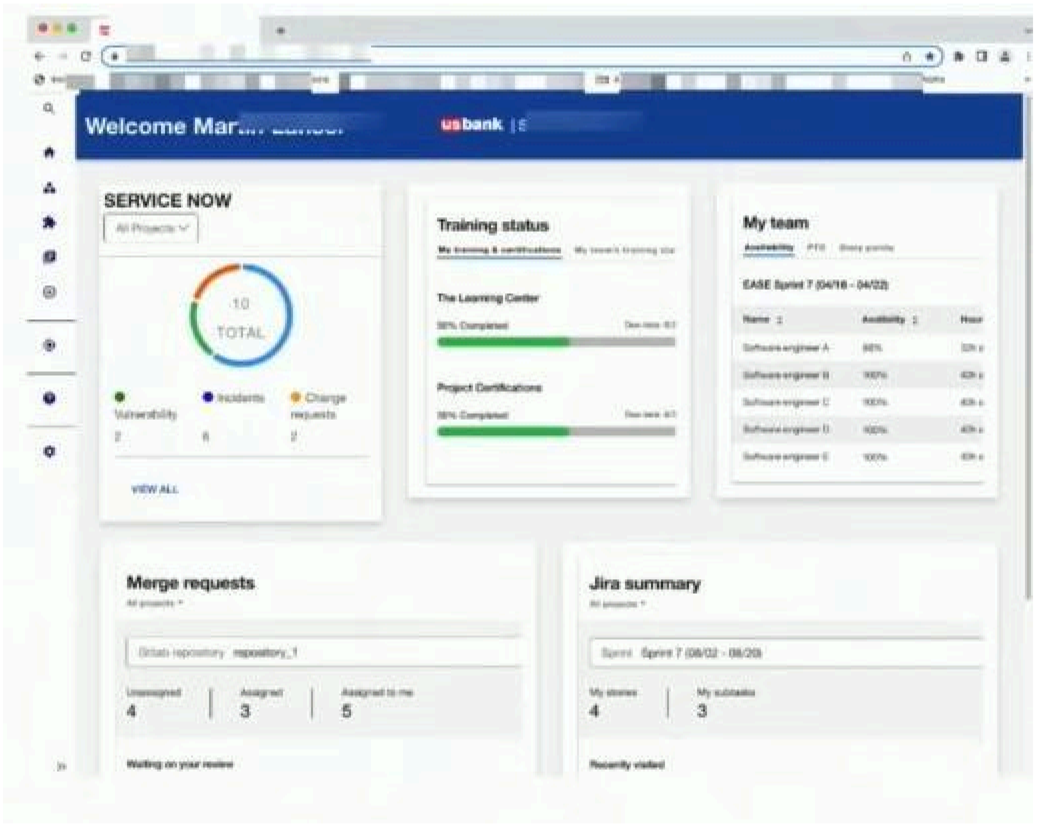
Slides published

Spotify - Building for Developer Happiness
Vmware - Providing a Curated experience for Developers with Backstage
Spotify - The evolution of Backstage Backends
Telus - Contributing a Backstage Plugin
Caribou - Leading the Architectural Change with Backstage
Roadie - What is tech insights anyways
Spotify - Measuring Developer Happiness
HP - How we scaled Catalog ingestion to hundreds of thousands of entities
Avalia - Bring Data Visualization to Backstage
DAZN - What we learned during our journey using Backstage

Adopters Feedback Talks summaries

[Enabling Developer Experience Journey in U.S. Bank - Ravikumar Tadikonda & Poonam Garg, U.S. Bank Portside Ballroom (Room 260)]

- 5th largest bank in the US
- US, Canada, Europe
- 3 years back started the journey
- Engineers are what we care about the most
    - Common engineering challenges
        - Day-to-day multiple systems
            - Integrate ERPs, APIs, and other systems
        - Processes
        - Self-service
        - Onboarding team member
        - Friction
        - Time to onboard
        - Compliance
- Developer experience
    - Priority #1 - Create and manage
        - Creating new and managing existing projects through automation and simplified guided processes.
        - Using software templates
        - Dashboard with
            - Tickets in -jira, Service now
            - Source control, Gitlab
            - Access Management
            - Vulnerabilities and risks


Welcome Mar[redacted]

usbank | [redacted]

SERVICE NOW

All Projects

10 TOTAL

● Vulnerability 2 ● Incidents 6 ● Change requests 2

VIEW ALL

Training status

My training & certifications | My team's training [illegible]

The Learning Center

58% Completed

Project Certifications

58% Completed

My team

Availability | PTO | Story points

EASE Sprint 7 (04/18 - 04/22)

Name	Availability	Hour
Software engineer A	80%	[illegible]
Software engineer B	100%	[illegible]
Software engineer C	100%	[illegible]
Software engineer D	100%	[illegible]
Software engineer E	100%	[illegible]

Merge requests

All projects

Gitlab repository repository_1

Unassigned 4 | Assigned 3 | Assigned to me 5

Waiting on your review

Jira summary

All projects

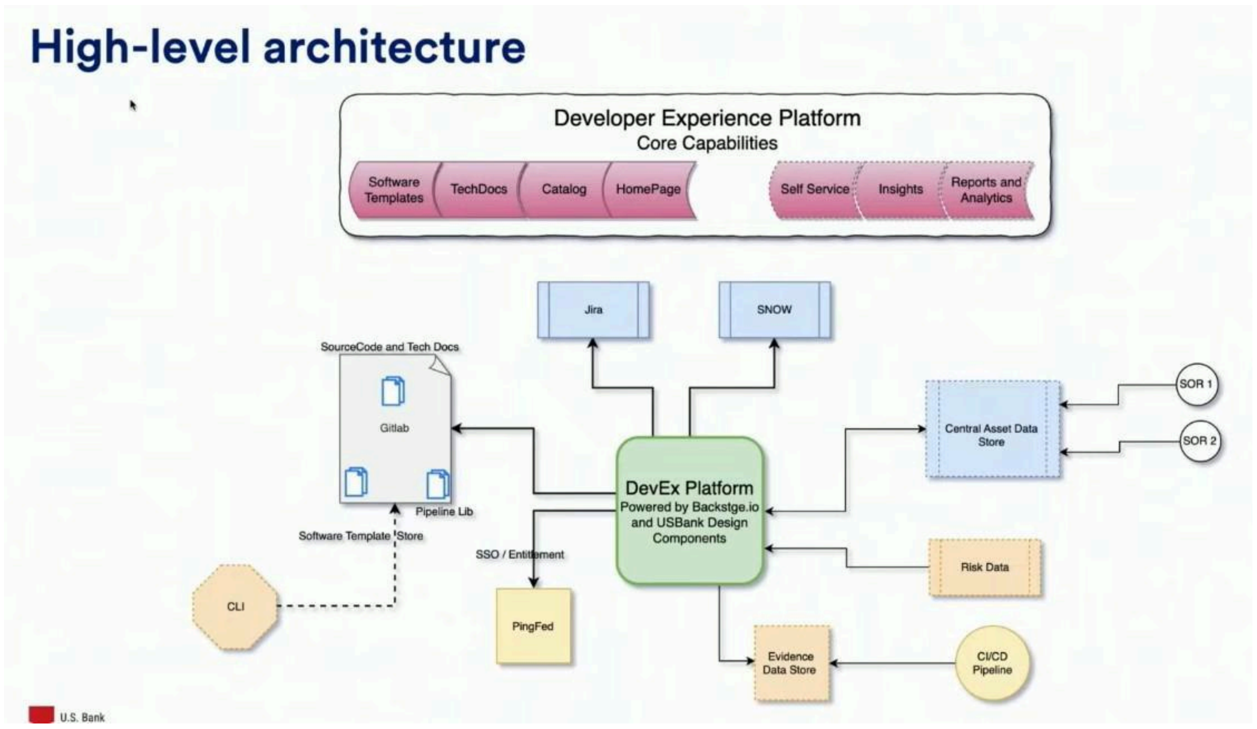
Sprint Sprint 7 (08/02 - 08/20)

My stories 4 | My subtasks 3

Recently visited

    ○ Priority #2 - Discover and reuse
        ■ Facilitate the discovery of existing solutions, encouraging re-use and collaboration to reduce redundancy and eliminate silos
        ■ Software Catalog
    ○ Priority #3 - Insight and Control
        ■ Gain insight into projects and reduce risks with standardized and automated controls
        ■ Relationships and asset graphs
        ■ Documentation
        ■ Vulnerabilities
        ■ Pipeline status
    ○ Personas identified
        ■ Engineers
        ■ Auditors
        ■ Managers
        ■ Compliance and risk control owners
            ● Create new and manage existing controls
● Team supporting Backstage
    ○ Started with 1 person
    ○ Currently 8 people and expanding

    - Following the inner source model so that any team can contribute to it
- How to implement based on complexity?
    - Scaffolder
    - Tech docs
    - Software Catalog
- Architecture

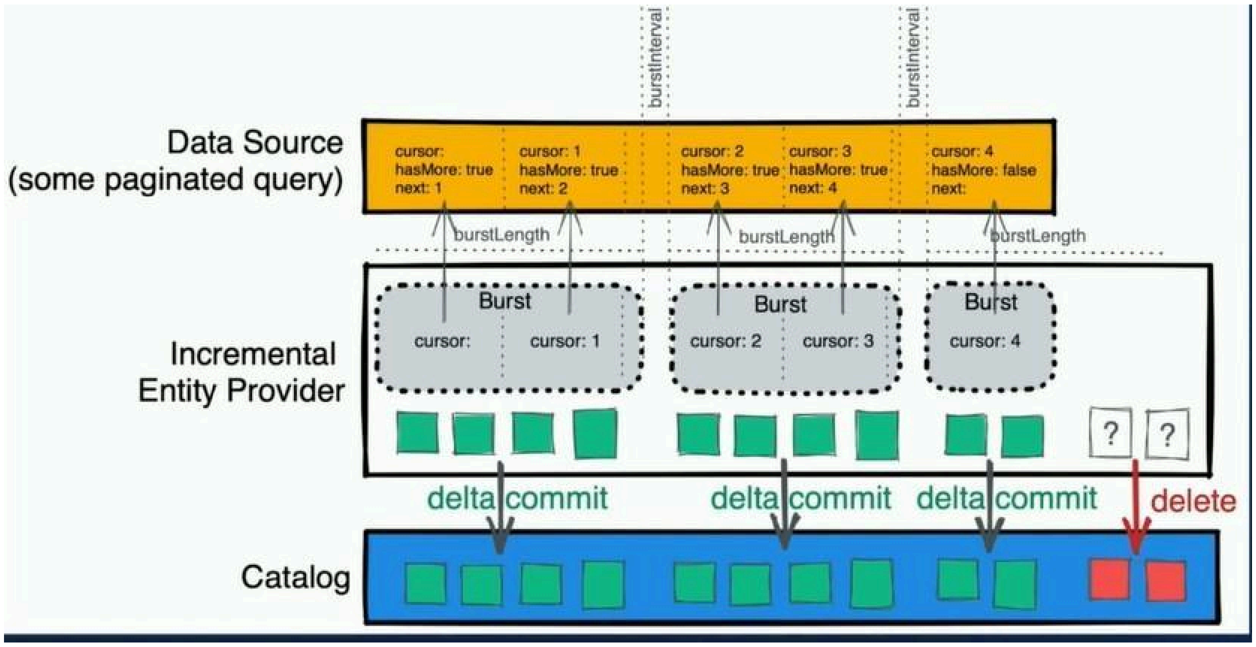


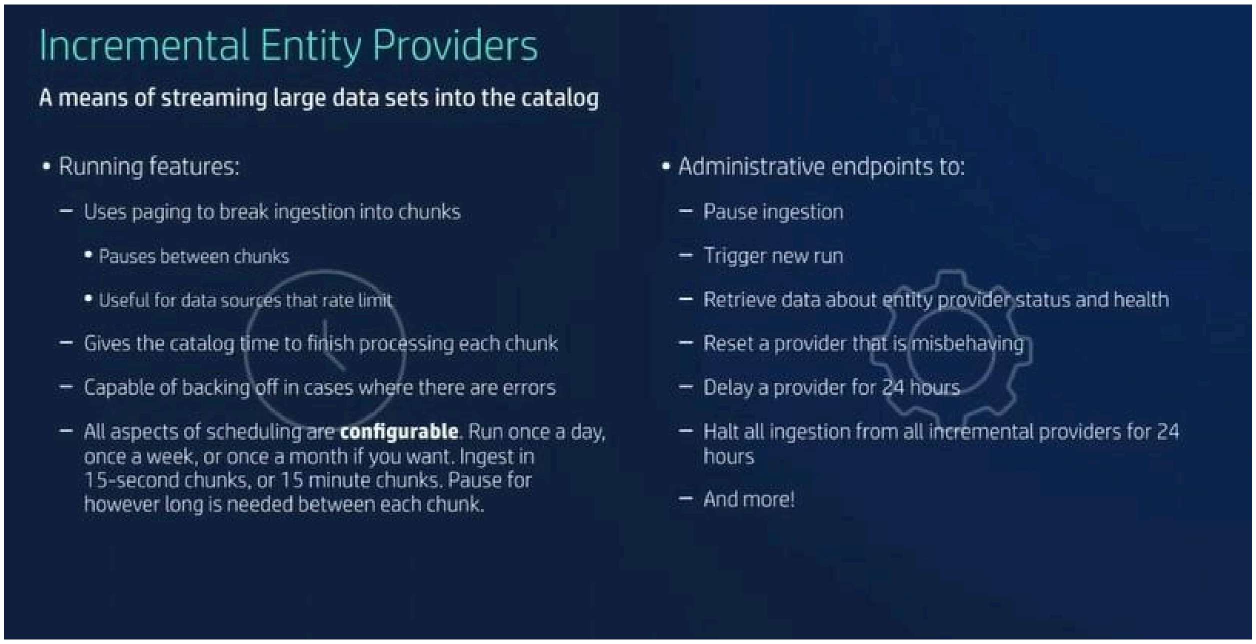
[How We Scaled Catalog Ingestion to Hundreds of Thousands of Entities - Damon Kaswell, HP Inc. Portside Ballroom (Room 260)]()

https://www.youtube.com/watch?v=5qHyZntKXRU

- Backstage is used as the Developer Portal for all the software in HP
    - APIs
    - Machine learning data
    - User software for personal computers
    - Drivers
    - Firmware
    - Web applications
    - Mobile applications

- Ingesting 200.000 entities from a single source into the Catalog
- Problem
  - Ingestion at scale is slow
  - Design patterns that didn't work
    - Catalog processors emitting locations read by other Catalog processors
      - Ingests whenever
      - No scheduling control
      - Locations pollution the Catalog
      - Entities do eventually show up in the catalog, but it takes a very long time
    - Entity providers emitting locations that are still read by Catalog Processors
      - Ingest locations when you want. Entities? Still, not so much
      - If you use an entity provider to ingest a large data source directly, it's a bad time
      - Locations still polluting the catalog
      - Entities do eventually show in the catalog, but it takes a very long time
    - Catalog processors vs. Entity providers
      - Catalog Processor
        - Pros
          - Simple to implement
          - Mature
          - One-stop shop for entity ingestion
        - Cons
          - Leaves behind orphans
          - Creates long delays in entity ingestion
          - Better for entity "side effects" - emitting relations
      - Entity provider
        - Pros
          - Dedicated strictly to ingestion
          - More flexibility in scheduling
          - Able to clean up orphans
        - Cons
          - Cannot emit relations at all
          - Only two options for ingestion "full mutation" or "delta mutation"
          - Delta mutation dependent on external integration e.g. webhooks
          - Full mutation incapable of handling very large data sets triggering eros such as "JavaScript heap out of memory"
- Solution: Incremental entity providers

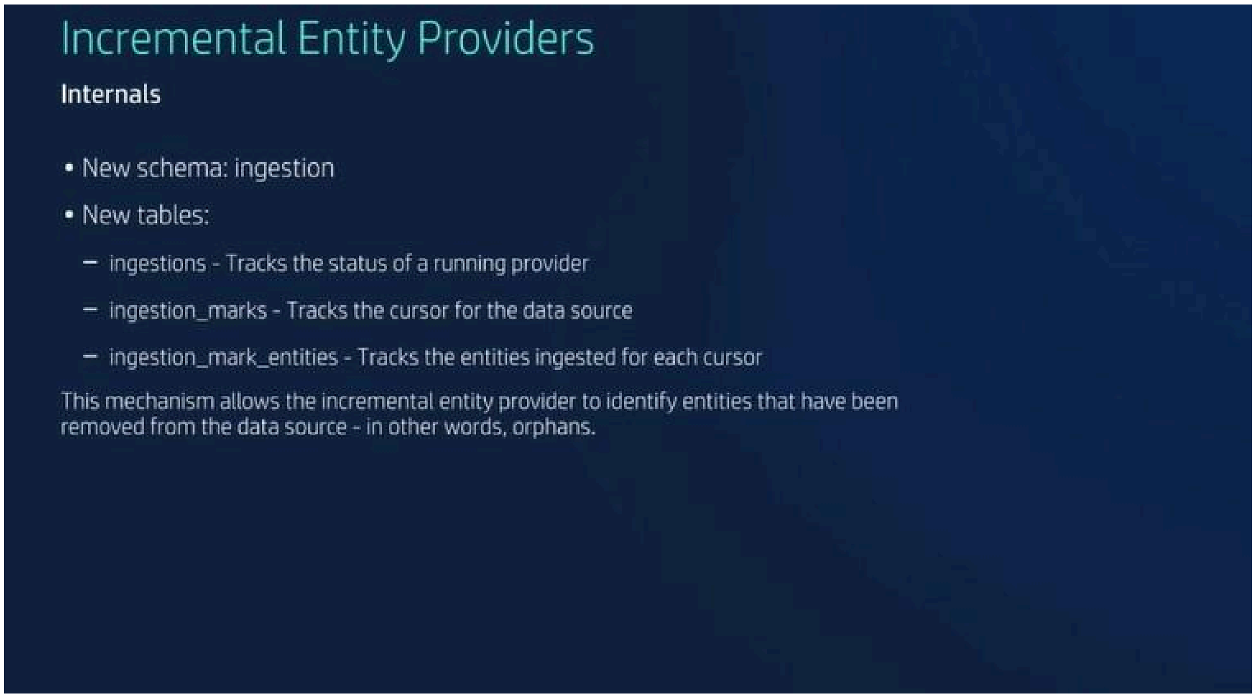




Incremental Entity Providers

A means of streaming large data sets into the catalog

- Running features:
  - Uses paging to break ingestion into chunks
    - Pauses between chunks
    - Useful for data sources that rate limit
  - Gives the catalog time to finish processing each chunk
  - Capable of backing off in cases where there are errors
  - All aspects of scheduling are **configurable**. Run once a day, once a week, or once a month if you want. Ingest in 15-second chunks, or 15 minute chunks. Pause for however long is needed between each chunk.
- Administrative endpoints to:
  - Pause ingestion
  - Trigger new run
  - Retrieve data about entity provider status and health
  - Reset a provider that is misbehaving
  - Delay a provider for 24 hours
  - Halt all ingestion from all incremental providers for 24 hours
  - And more!



Incremental Entity Providers

Internals

- New schema: ingestion
- New tables:
  - ingestions - Tracks the status of a running provider
  - ingestion_marks - Tracks the cursor for the data source
  - ingestion_mark_entities - Tracks the entities ingested for each cursor

This mechanism allows the incremental entity provider to identify entities that have been removed from the data source - in other words, orphans.

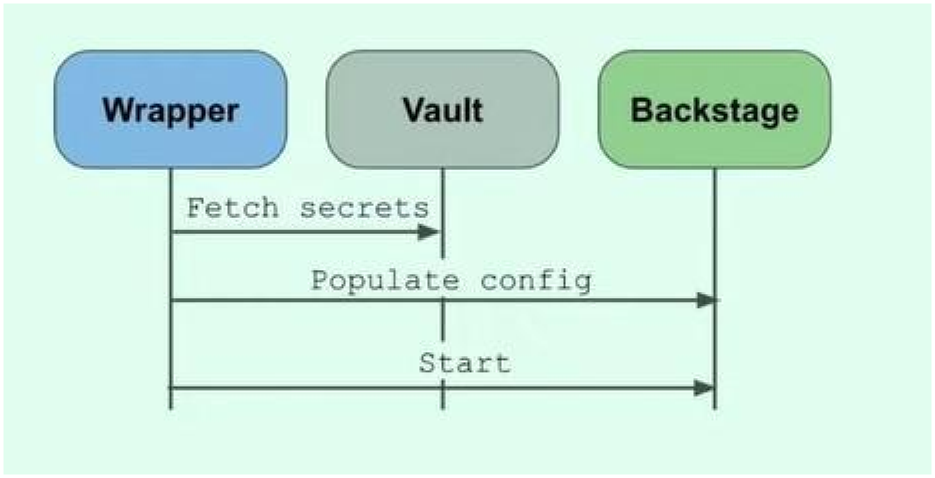
[What We Learned During Our Journey in Using Backstage at DAZN - Marco Crivellaro, DAZN Portside Ballroom (Room 260)]()

https://www.youtube.com/watch?v=tvogOVhAXI8

- What is DAZN
  - Live sports streaming OTT
  - 11 countries' exclusive rights
  - 180+ countries' global rights
  - DAZN engineering
    - 700 engineers
    - 70 teams
    - Distributed in several countries

- Questions that led to Backstage
    - Which team owns a given service?
    - What does such a service do?
    - Where can we find the code?
    - Where is it deployed to?
    - What are its dependencies?
- Technical learnings
  - Start small
  - The number of plugins is overwhelming

        - Start with what matters the most and build from there
        - Implementing all the plugins that they would think of was a mistake because it raised more questions rather than answering them
        - They started with the Software Catalog, then tech docs and recently, software templates
        - Use what you know and keep it simple
        - Deployed in ECS Fargate
        - Azure AD OIDC
        - RDS PostgreSQL
        - Hashicorp Vault for secrets



- Backstage is an aggregator, not the single source of truth
- Operations
    - Monthly release upgrades
        - Upgrade helper
        - Changelog
    - Stay in touch with the community
        - Community sessions
        - Discord
- Cultural learnings
    - Contributors must feel welcome
        - Make it easy for internal teams to contribute to your internal instance
        - Contributors with domain knowledge are key to success
        - Make your voice heard
            - Engage with the internal engineering team
                - Blogposts
                - New features announcements
                - Demos
                - Contributions shout outs
                - Townhall

            - Give to the community

[Building an Internal Developer Platform with Backstage Plugins - Taras Mankovski, The Frontside Software, Inc Portside Ballroom (Room 260)]

- Presented a PoC plugin that allows downloading a CLI to interact with Backstage to avoid context switching, given the developer works from the IDE
- Backstage App use cases
    - Includes
        - Discovery
        - Templating
    - Lacks
        - Environments
            - Run the Kubernetes plugin from the CLI
        - Secrets
            - Change to other platforms to manage secrets
        - Releases
        - Logs
        - Heroku like DX on IDPs
            - Self serve Databases
            - Push to build workflow
            - Automatic rollback
            - Autoscaling
            - Clean Web UI
            - Full-featured CLI