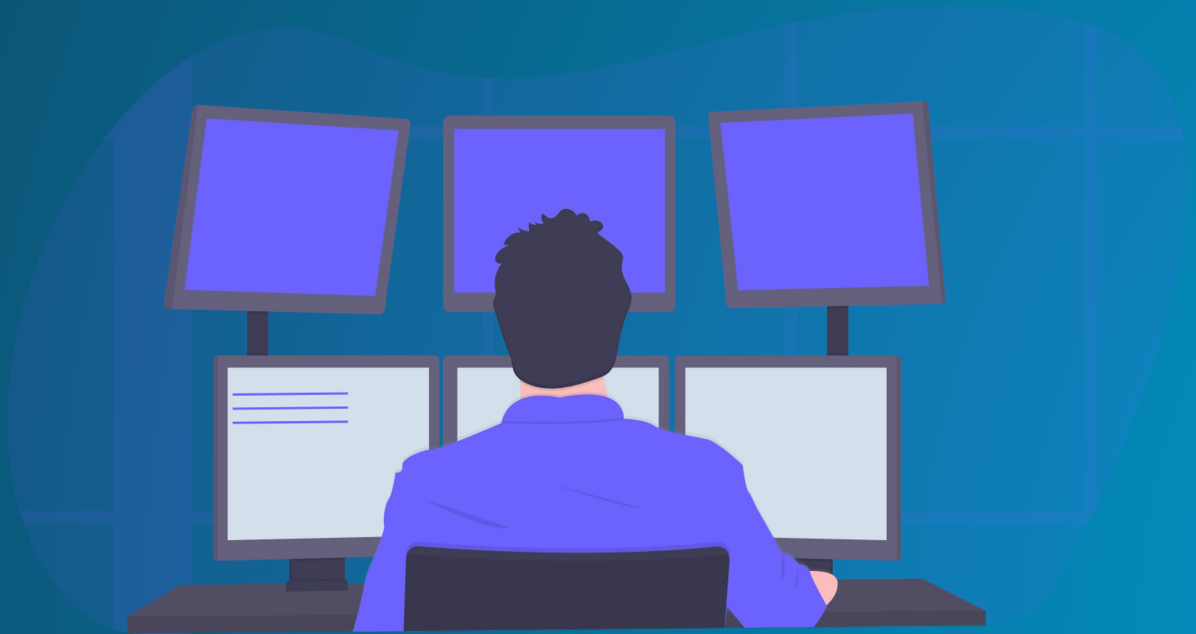
Stackable

Office Hours

2022-12-02



Stackable in a Nutshell

Founded

2020

OpenCore

b.telligent

IONOS by 1&1

Stackable Data Platform

- Open Source
- Infrastructure as Code
- Cloud native (Kubernetes)
- On-Premises, Cloud, Hybrid

Our Customers

Danske Bank

Taboola

Dentsply Sirona

T

IONOS

opencorporates

Our Team: ~20

International in Germany & Europe

Our Services

- Product Support
- Big Data Consulting
- Trainings

Network - Collaborations

KI BUNDESVERBAND

gaia-x

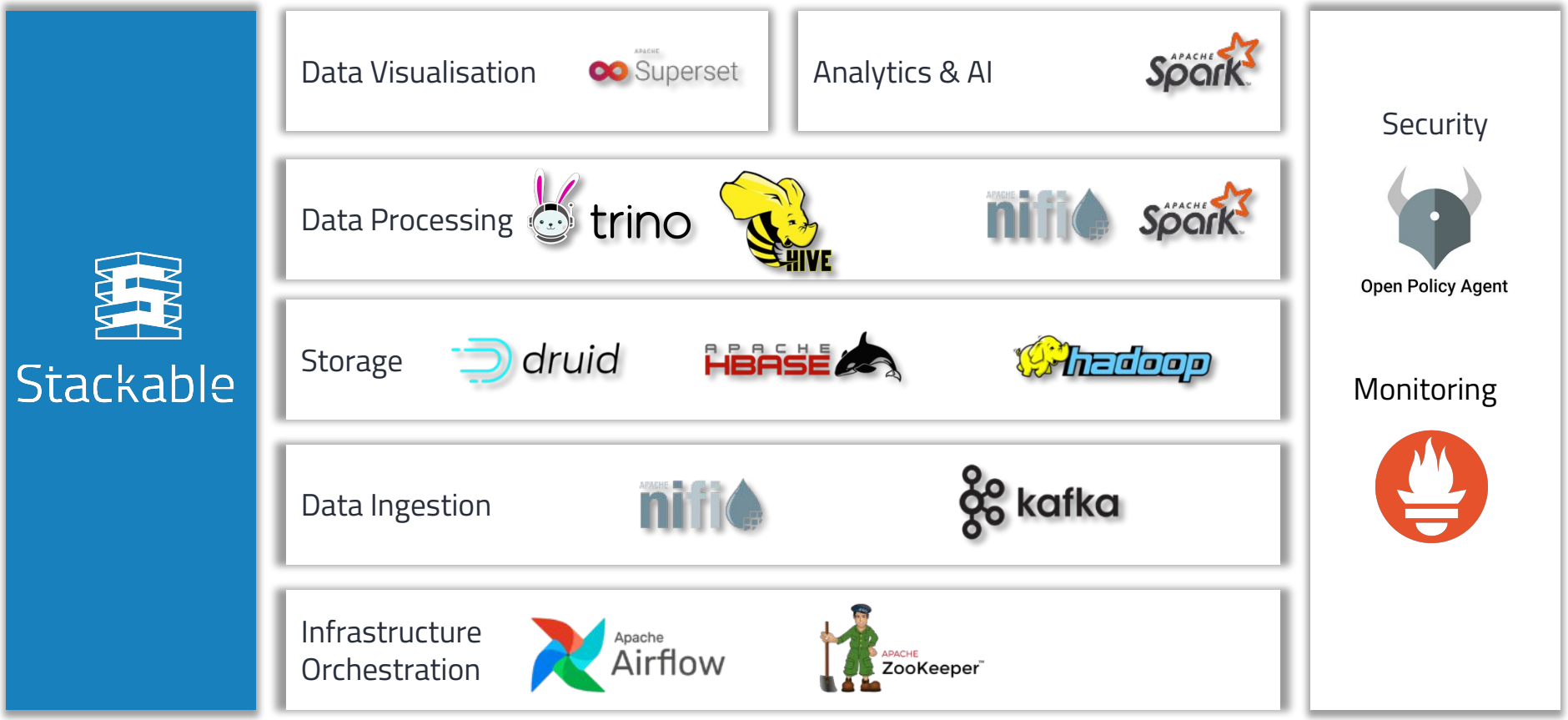
OSB Open Source Business ALLIANCE

bitkom

eco

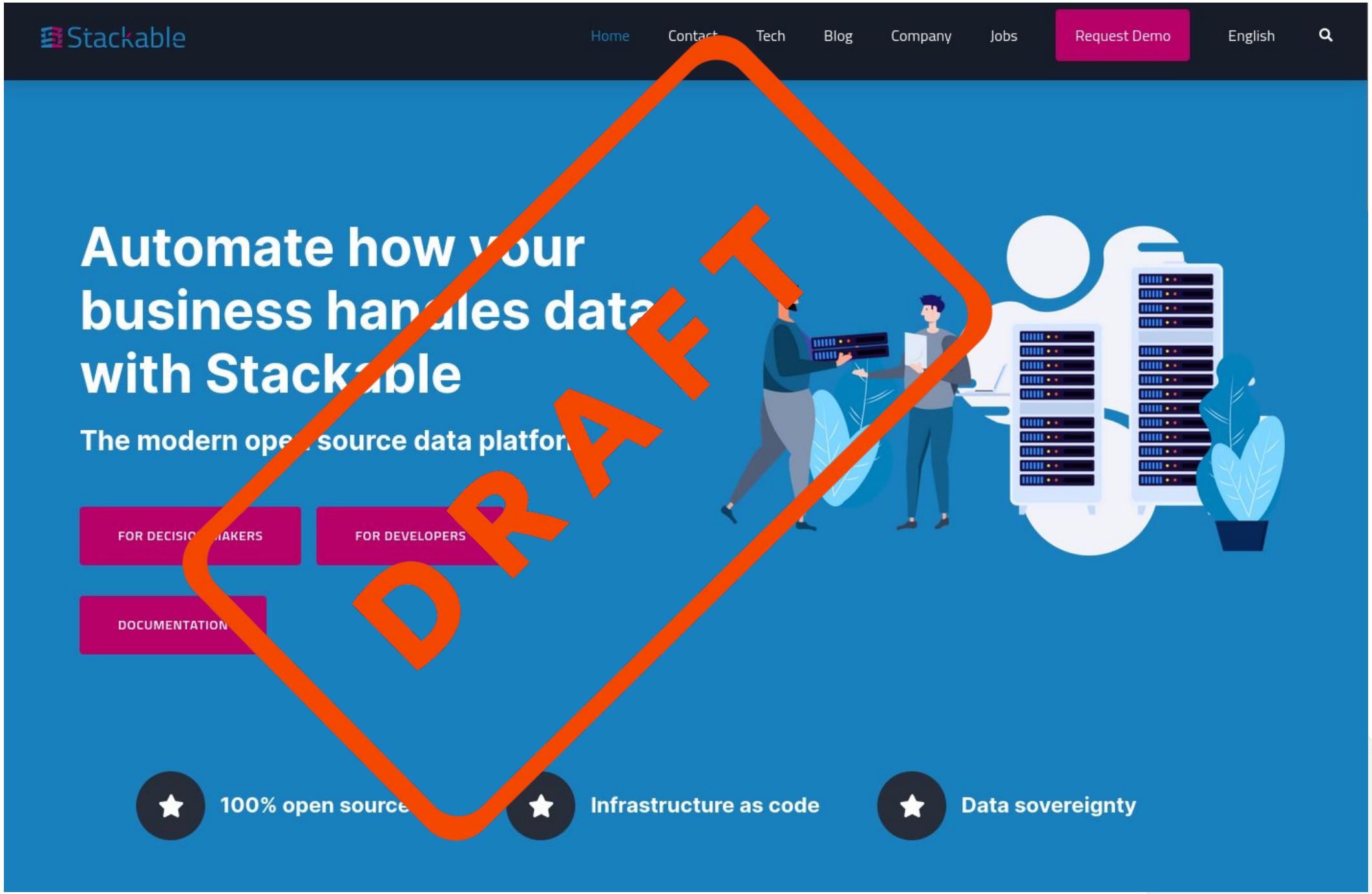
Stackable

Your free alternative Open-Source Data Platform

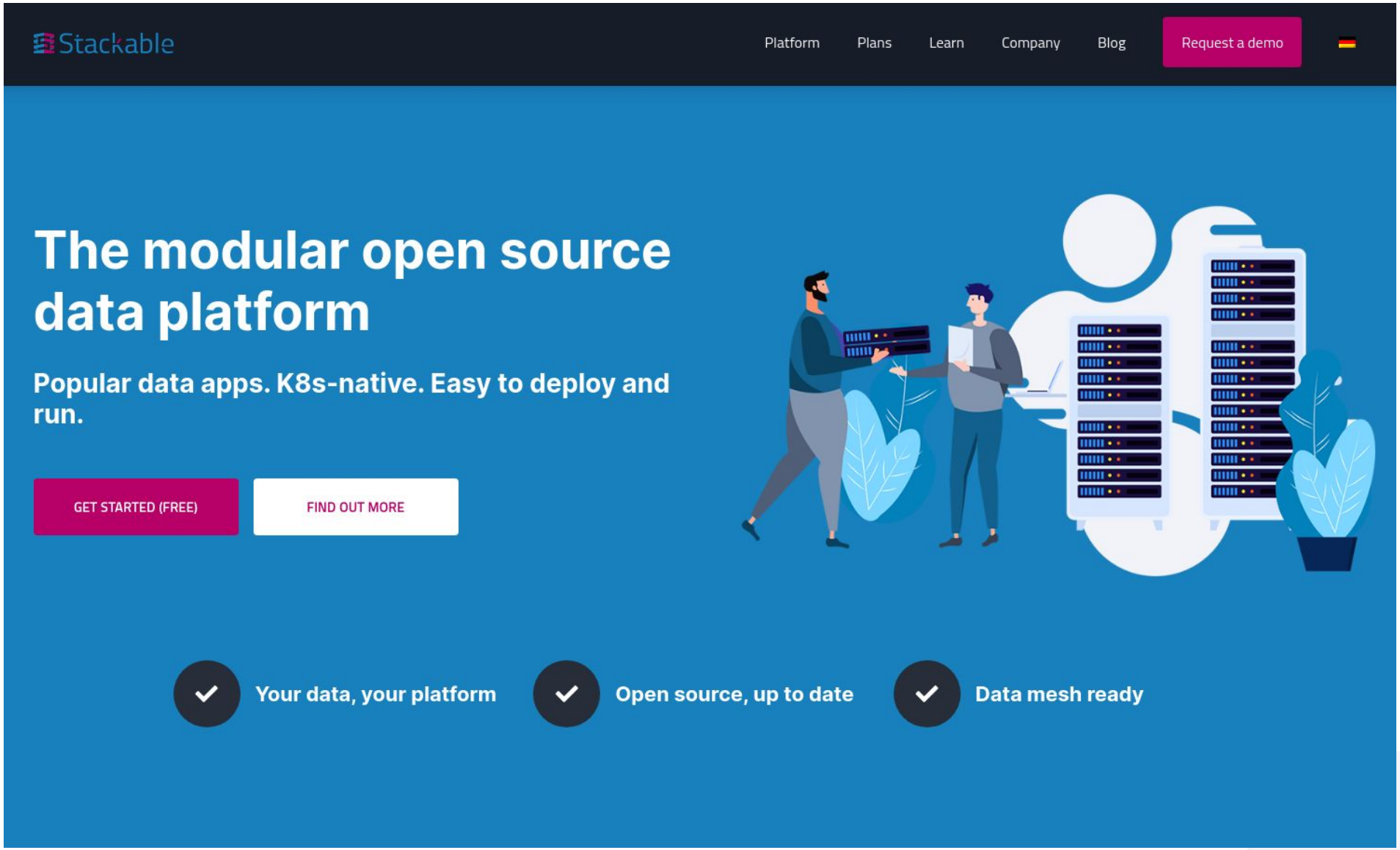


Stackable

Homepage 2.0


Stackable
Home
Contact
Tech
Blog
Company
Jobs
Request Demo
English
Automate how your business handles data with Stackable
The modern open source data platform
FOR DECISION MAKERS
FOR DEVELOPERS
DOCUMENTATION
100% open source
Infrastructure as code
Data sovereignty
DRAFT

Homepage 2.0



Stackable

Documentation

HOME GETTING STARTED CONCEPTS TUTORIALS STACKABLECTL OPERATORS CONTRIBUTE

Stackable Operator for Apache Druid

- Getting started
- Configuration
- Usage
- Concepts

0.7
nightly
0.6
0.2
0.1

Stackable Operator for Apache Druid

Stackable Operator for Apache Druid

This is an operator for Kubernetes that can manage Apache Druid clusters.

Supported Versions

The Stackable Operator for Apache Druid currently supports the following versions of Druid:

- 0.22.1
- 0.23.0

Stackable

Releases

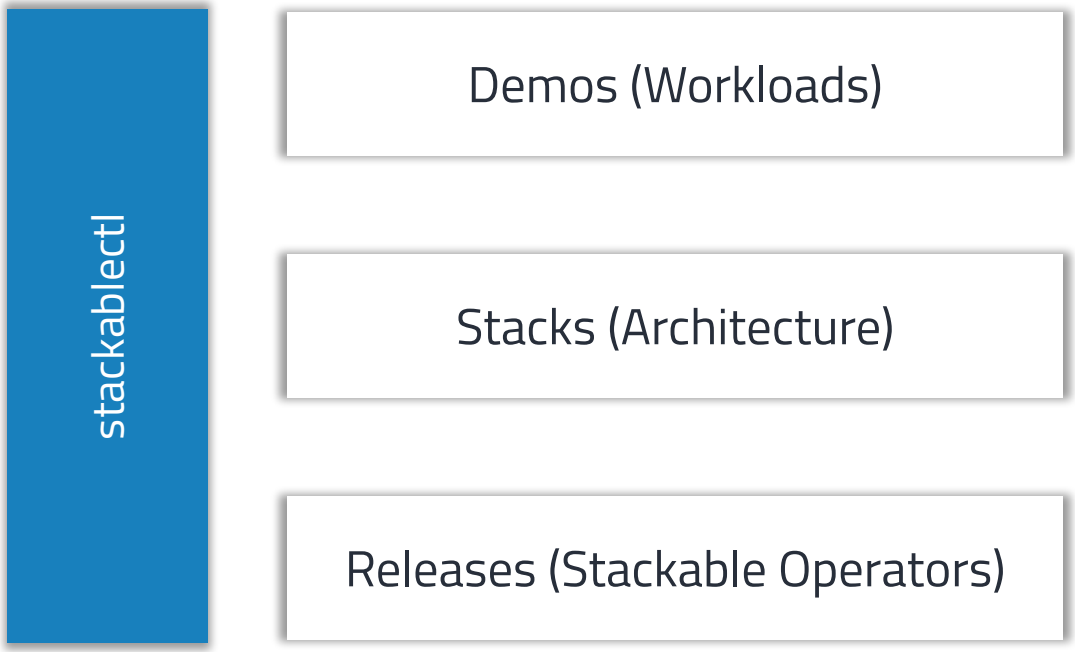
Stackable

Looking Back at Releases 22.06, 22.09 & 22.11

- stackablectl
- LDAP support for tools that support it
  - Druid
  - NiFi
  - Airflow
- Extended OpenShift support for our operators
- Kafka TLS support with secret operator
- TrinoCatalogs
- Demos!
- Updated product versions
- Resource management
- HBase Phoenix support
- …

Stackable

Demos



Stackable

Current Demos (a selection)

DEMO: DATA-LAKEHOUSE-ICEBERG-TRINO-SPARK

Data Lakehouse technology showcase

This technology demo showcases some of Stackable's latest **release 22.11 features**.

The demo contains elements of previous demos i.e.

- real-time event streaming with Apache NiFi
- Trino for SQL access and
- visual data display and analysis with Apache Superset

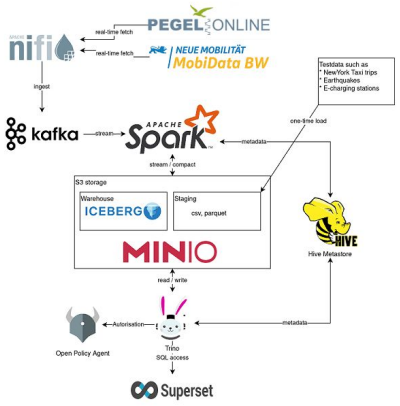
But, adding to this, the demo also includes new lakehouse features such as the integration with Apache Iceberg providing e.g. transactional consistency and full schema evolution.

The result is a powerful blueprint for a modern data stack with the Stackable Data Platform.

Other highlights of the demo:

- Apache Spark: A multi-language engine for executing data engineering, data science, and machine learning. This demo uses it to stream data from Kafka into the lakehouse.
- Open policy agent (OPA): An open source, general-purpose policy engine that unifies policy enforcement across the stack. This demo uses it as the authorizer for Trino, which decides which user is able to query which data.

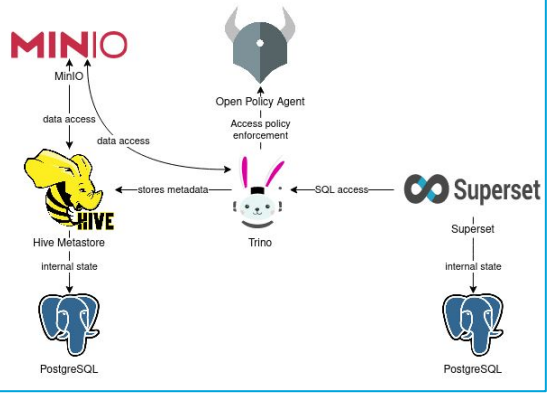
TUTORIAL



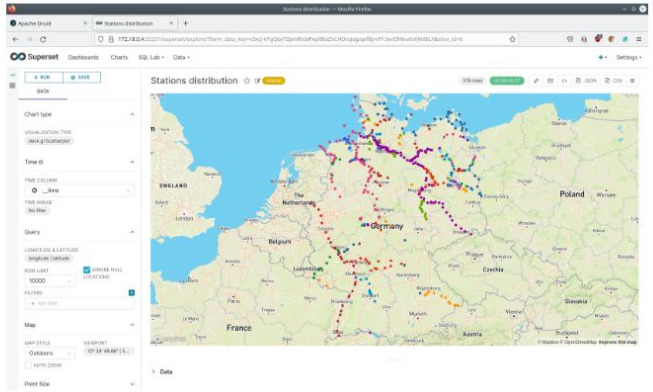
Analysis with a data lake

This Stackable Data Platform demo shows data stored in S3 for analysis up to display in the dashboard. Our Stackable operators are used to configure and roll out various components. In particular, this example shows how role-based data access can be implemented using the Open Policy Agent:

- MinIO, an S3-compatible object store, persistently stores the data for this demo.
- Hive-Metastore stores the metadata necessary to make the sample data accessible via SQL and is used by Trino in our example.
- Trino is our extremely fast, distributed SQL query engine for Big Data analytics that can be used to explore data spaces and that we use in the demo to provide SQL access to the data.
- Finally, Apache Superset we use to retrieve data from Trino via SQL queries and build dashboards on that data.
- Open Policy Agent (OPA): an open source, universal policy engine that unifies policy enforcement across the stack. In this demo, OPA authorizes which user can query which data.



Real-time display of water levels



Low water or danger of flooding – the water levels of our rivers have moved into the public interest in times of climate change.

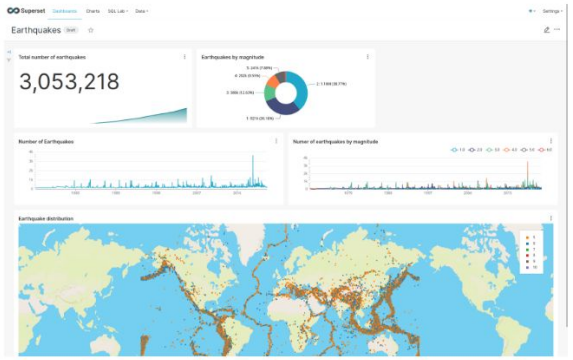
Our Stackable Data Platform demo shows the water levels of rivers in near real-time for Germany based on data from Pegel Online.

Several components of the Stackable Data Platform play together without requiring much configuration effort:

Apache Nifi and Kafka are used to fetch water level measurements from gauging stations distributed across Germany via an API from Pegel Online and store them in Apache Druid.

Druid is a scalable real-time database that can be queried using SQL. This method is used in the demo to query gauge levels via Apache Superset and visualize them in the dashboard. For permanent storage, Druid requires a so-called "deep storage", which is implemented in our example via MinIO as an S3-compatible object store, as it is available in most public and private cloud environments.

Event streaming of earthquake data



This Stackable Data Platform demo shows streamed earthquake data end-to-end up to the dashboard. It includes the following operators:

- Superset: a modern platform for data exploration and visualization. This demo uses Superset to retrieve data from Druid via SQL queries and build dashboards on that data.
- Kafka: A distributed event streaming platform for high-performance data pipelines, streaming analytics, and data integration. In this demo, Kafka is used as an event streaming platform to stream data in near real-time.
- Nifi: An easy-to-use, powerful system to process and distribute data. This demos uses it to fetch earthquake-data from the internet and ingest it into Kafka.
- Druid: A real-time database to support modern analytics applications. This demo uses Druid to ingest and store data in near real-time from Kafka and provide access to the data via SQL.
- MinIO: An S3-compatible object store. In this demo, it is used as persistent storage for Druid to store all streamed data.

Stackable

Release 22.11

Stackable Release 2022-11

Planned release date is: 2022-11-11

Release process

We will use a release branch for this release. The process can be summarized as follows:

Overview

- Development in short-lived feature branches with PRs merged to main (current practice)
- A release branch is created from main per minor version (e.g. 4.2.x)
- This branch is used for testing and verifying, with fixes being made in main and then cherry-picked to the release branch
- When the release branch is tested and ready, it is tagged and remains open for any subsequent bug fixing etc.

Process steps

- [ ] initiate the release process by creating a release branch e.g. 4.2.x and replace “nightly” etc. (as the current release script does)
- [ ] the docs version is also set, but in antora still marked as prerelease=true
- [ ] conduct a non-formal feature freeze (not technically necessary but not a bad idea either). This branch is now the release-candidate-branch, where tests are made, demos verified etc.
- [ ] test and make changes to code and docs in main and then cherry-picked into the release branch
- [ ] when testing is complete, tag the release branch
- [ ] the docs can now be marked as prerelease=false
- [ ] during the release process, document and plan what can be automated for future releases

Further bug-fixes follow the same pattern (made in main, cherry-picked, release branch re-tagged)
A platform release is then defined by the individually released operators at the time of the release plus the stackablectl releases.yaml.

Release checklists

Beginning of the release cycle

- [ ] Epic: Update products to latest versions #260
- [x] Bump Rust version

Before feature freeze

- [ ] Bump operator-rs to latest version in all operators

Stackable

Release 22.11

- Iceberg Support in Trino
- Full restart Support for NiFi
- Restart Operator
- More OpenShift Compatibility
- Many many many smaller things

Stackable

Release 23.01

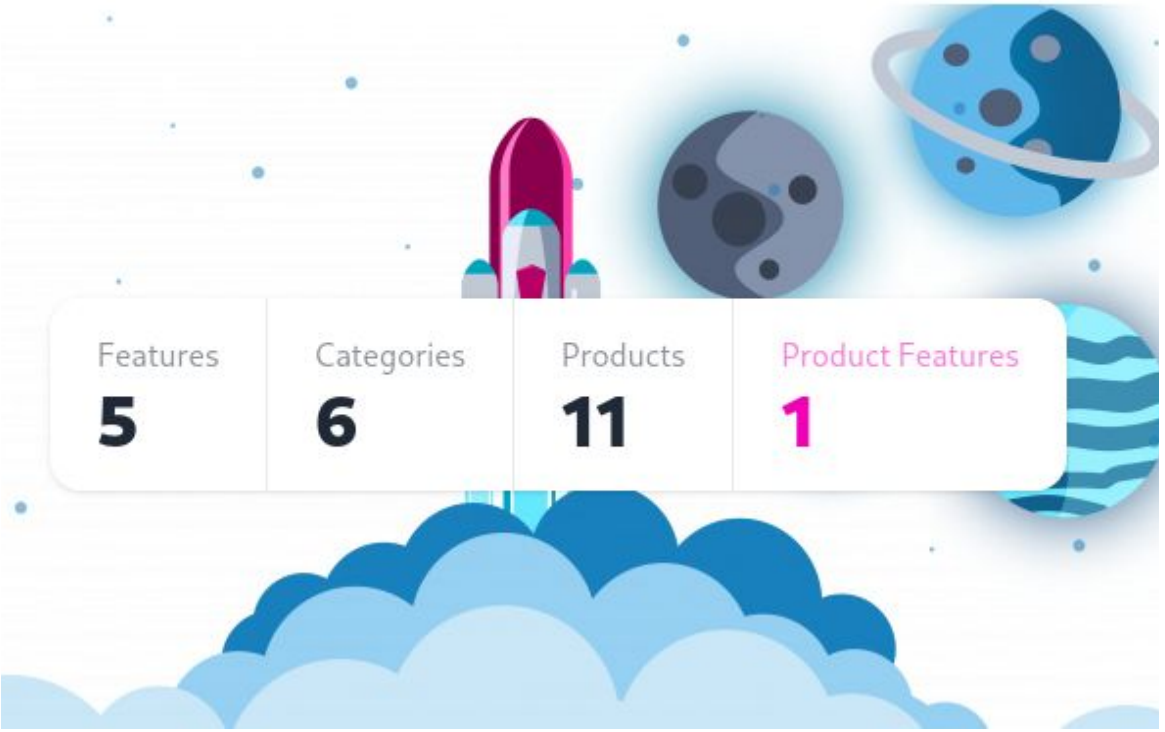
Focus Topics

- Authentication & Authorization
- Monitoring & Log Aggregation
- New demo
- OpenShift
- Product Image specification (i.e. “offline” support)

Stackable

Feature Tracker

Features	Categories	Products	Product Features
75	32	11	53



Stackable

GBIF

GBIF | Global Biodiversity Information Facility

Stackable

Benchmarks

We ran some benchmarks!

* Blog post with numbers and exact commands upcoming

Stackable

Benchmarks

Azure	Google Cloud	IONOS	GBIF
Stackable OPENSHIFT	Stackable CLOUDERA	Stackable	Stackable CLOUDERA Stackable

Benchmark Environment

- GKE 1.23.8
- Machine Type: C2 Standard 8 (see table)
- 16 Nodes (3 Master, 13 Worker)
- SDP 2022.11 / CDH 6.3.4

	Google (c2-standard-8)
CPU	8 (VCPU)
RAM	32 GB
Storage	1 Drive*

“*” More hard drives just share the available bandwidth according to the docs

Stackable

Kubernetes vs. “Bare Metal”

ms (rounded)	Workload A 50% Read / 50% Update		Workload B 95% Read / 5% Update		Workload C 100% Read		Workload D 95% Read/ 5% Insert		Workload E 5% Insert / 95% Scan		Workload F 50% Read / 50% Read Modify Write	
	CDH	**SDP**	**CDH**	**SDP**	**CDH**	**SDP**	**CDH**	**SDP**	**CDH**	**SDP**	**CDH**	**SDP**
Teil 1: Average Latency	1	1	0	0	0	0	0	0	5	5	1	1
Teil 1: 95 Percentile	1	1	1	1	1	1	1	1	10	17	1	1
Teil 1: 99 Percentile	4	4	4	3	3	3	3	4	17	27	4	3
Teil 2: Average Latency	1	1	1	1	n/a	n/a	1	1	5	6	1	1
Teil 2: 95 Percentile	1	1	1	1	n/a	n/a	1	1	11	17	2	3
Teil 2: 99 Percentile	6	4	5	4	n/a	n/a	5	4	18	27	3	6

SDP as fast or faster than CDH

SDP less than 1 ms slower

SDP over 1 ms slower

Stackable

Stackable Data Platform / Trino

2022-11-24 09:04:53 to 2022-11-24 21:20:25

Namespace default | Cluster trino

CPU request	CPU limit	Memory limit	Cluster name	Workers
10	20	50 GiB	trino	5

CPU usage

Memory usage

Internal memory usage

Network usage

Rows/s read

Logical bytes read

Running queries

Finished queries

Stackable Data Platform / MinIO

2022-11-24 09:04:46 to 2022-11-24 19:03:03

Instance: minio

Uptime

10 hours

Total S3 Traffic Inbound

403 GiB

Total S3 Traffic Outbound

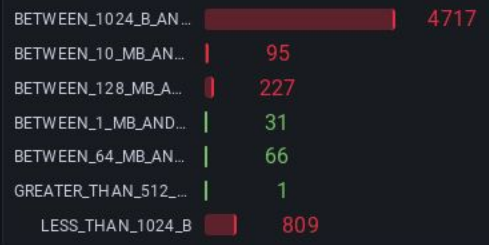
1.07 TiB

Capacity

Free	89%
Used	11%

Data Usage Growth

Object size distribution



Total Open FDs

661

Total Goroutines

10.8 K

Total Online Servers

5

Total Online Disks

4

Number of Buckets

2

Total Offline Servers

0

Total Offline Disks

1

Number of Objects

6.08 K

Time Since Last Heal Activi...

9.82 hour

Time Since Last Scan Activ...

23.8 s

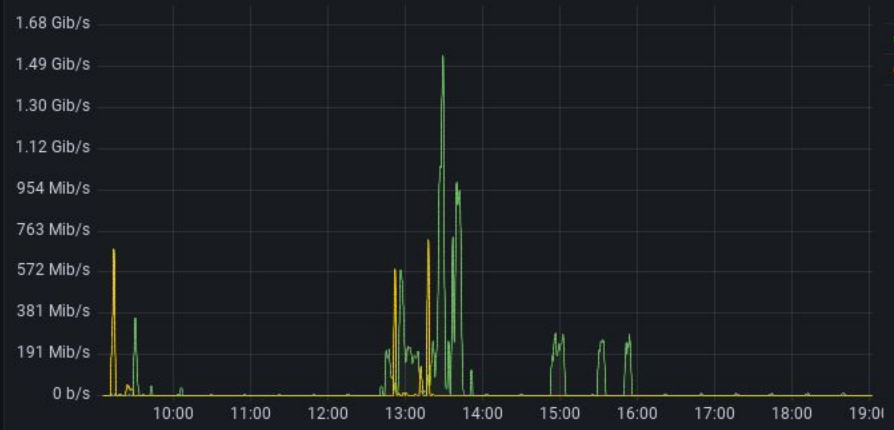
S3 API Data Received Rate

- Data Received [minio-0.minio-svc.default.svc.cluster.local:9000]
- Data Received [minio-1.minio-svc.default.svc.cluster.local:9000]
- Data Received [minio-2.minio-svc.default.svc.cluster.local:9000]

S3 API Data Sent Rate

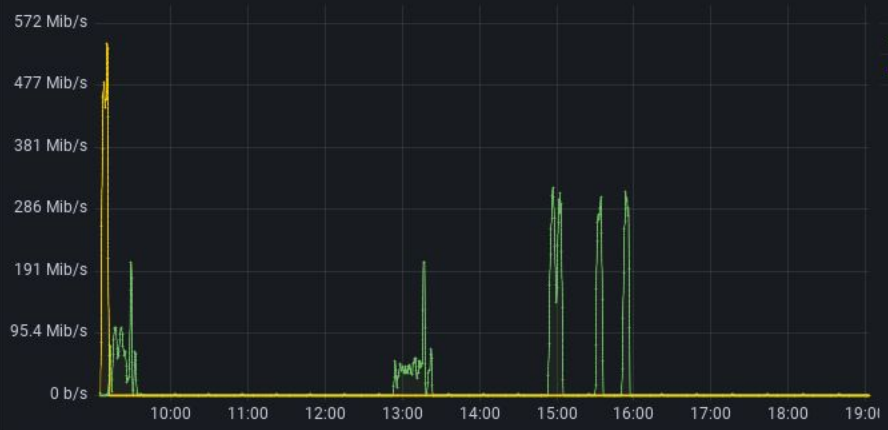
- Data Sent [minio-0.minio-svc.default.svc.cluster.local:9000]
- Data Sent [minio-1.minio-svc.default.svc.cluster.local:9000]
- Data Sent [minio-2.minio-svc.default.svc.cluster.local:9000]

Bucket Traffic Sent



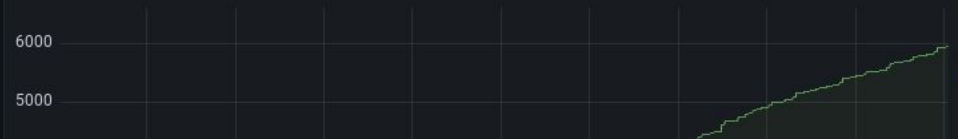
	Mean	Max
lakehouse	43.0 Mib/s	1.54 Gib/s
staging	6.54 Mib/s	724 Mib/s

Bucket Traffic Received



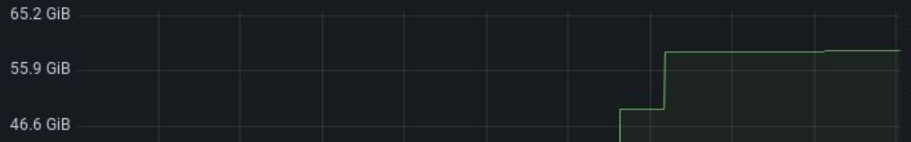
	Mean	Max
lakehouse	13.8 Mib/s	320 Mib/s
staging	4.40 Mib/s	542 Mib/s

Bucket objects



	Max	Last
lakehouse	5945	5945
staging	153	153

Bucket size



	Max	Last
lakehouse	59.3 GiB	59.3 GiB
staging	20.0 GiB	20.0 GiB

Next Office Hours:

27.1.2023

Follow us

https://www.linkedin.com/company/stackabletech/

https://twitter.com/stackabletech

https://github.com/stackabletech

https://www.xing.com/pages/stackable

Coming soon! (https://slack.stackable.de)

Subscribe to our newsletter: https://newsletter.stackable.tech/

Stackable

Thank you

Contact

Lars Francke
lars.francke@stackable.tech
+49 172 4554978

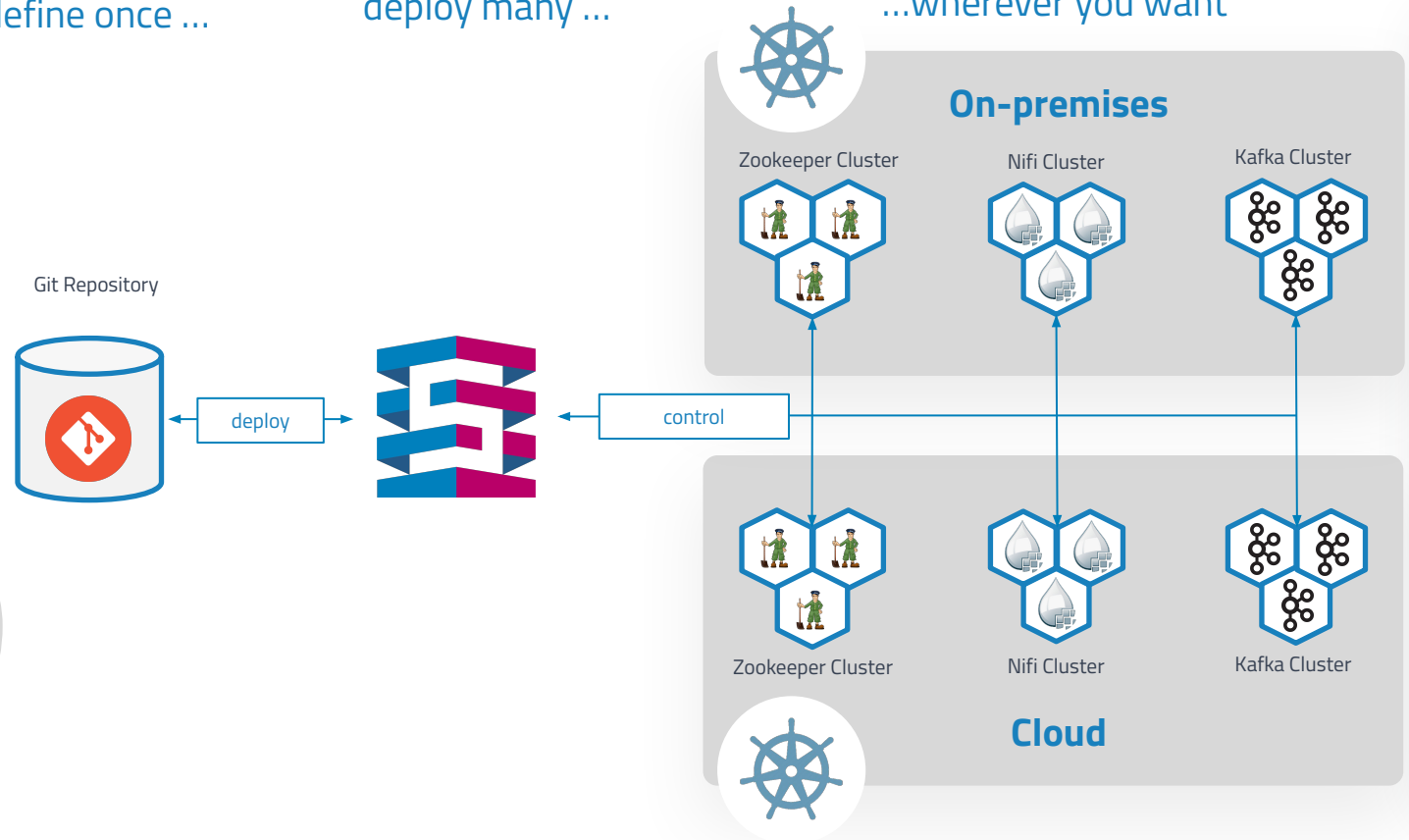
Stackable

Define your Data Platform as Code

define once ...

deploy many ...

...wherever you want



Stackable Data Platform:

A comprehensive set of software components playing together to

- deploy,
- manage,
- monitor,
- update and
- secure

up-to-date open source products using a 100% Infrastructure-as-Code approach

Stackable