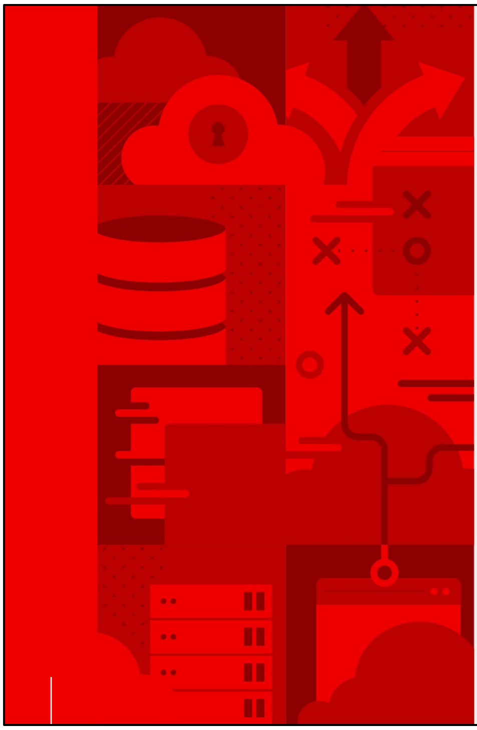
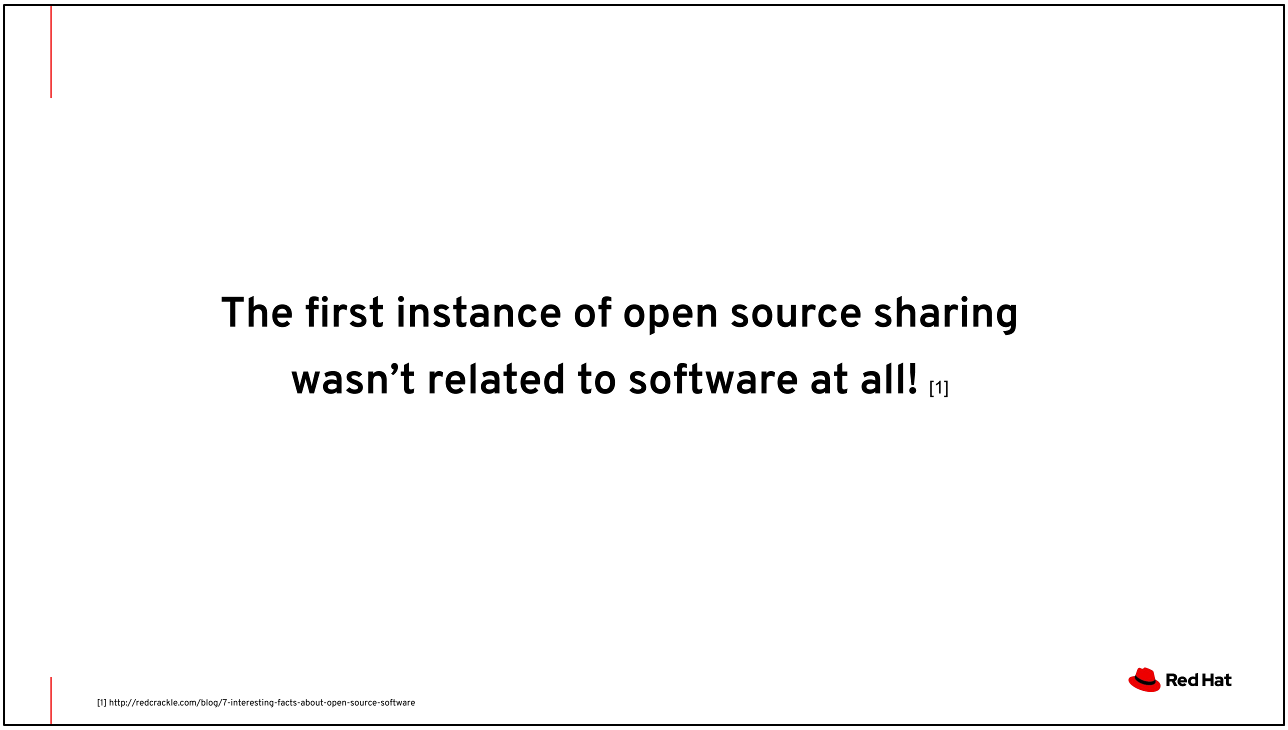


Open Source Use Cases

Rebecca Simmonds rsimmond@redhat.com

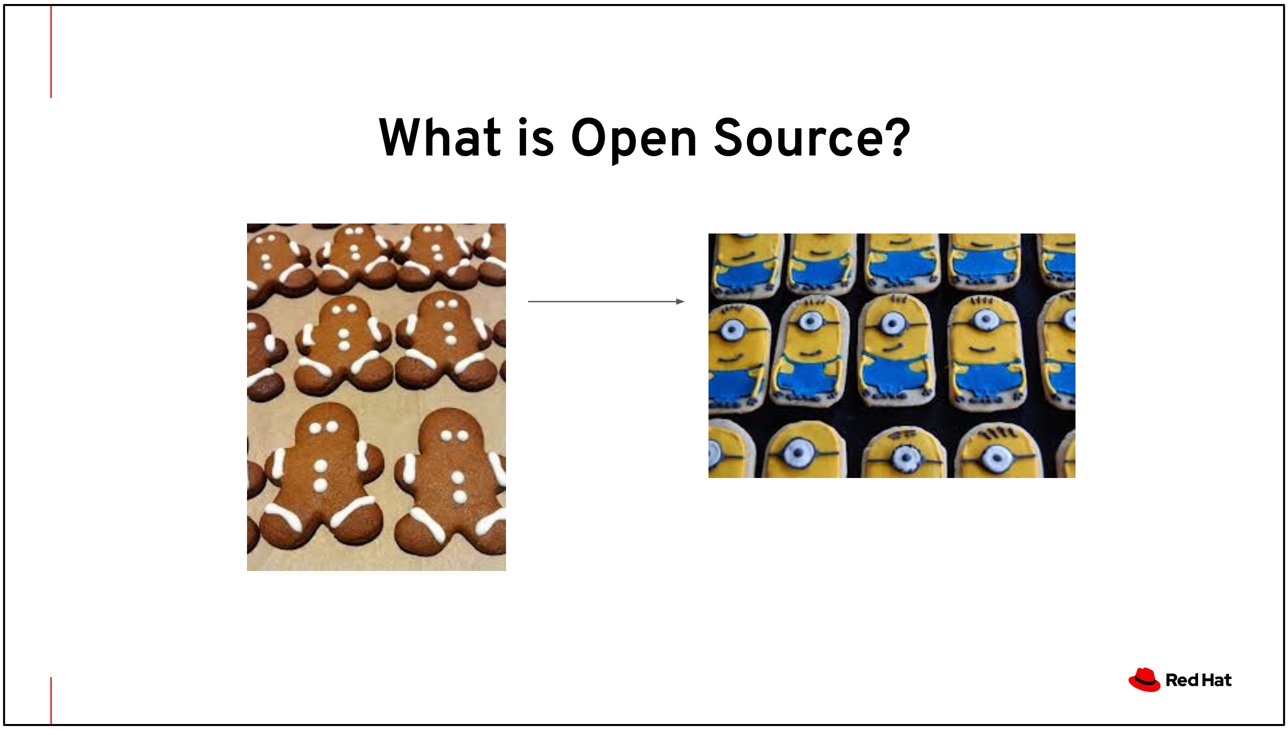
Rui Vieira rui@redhat.com



The first instance of open source sharing wasn’t related to software at all! [1]

[1] http://redcrackle.com/blog/7-interesting-facts-about-open-source-software

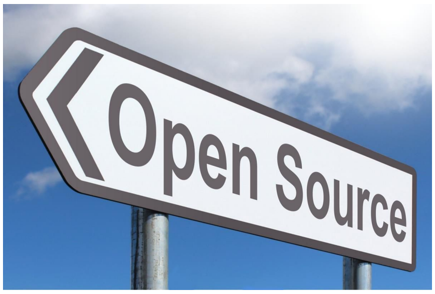
- The first instance of open source sharing dates back to even before the first computer was developed.
- In 1911, revolutionary automaker Henry Ford was instrumental in launching the Motor Vehicle Manufacturers Association.
- This association launched an open source initiative that witnessed major US auto manufacturers sharing technology patents openly without seeking any monetary benefits in return.



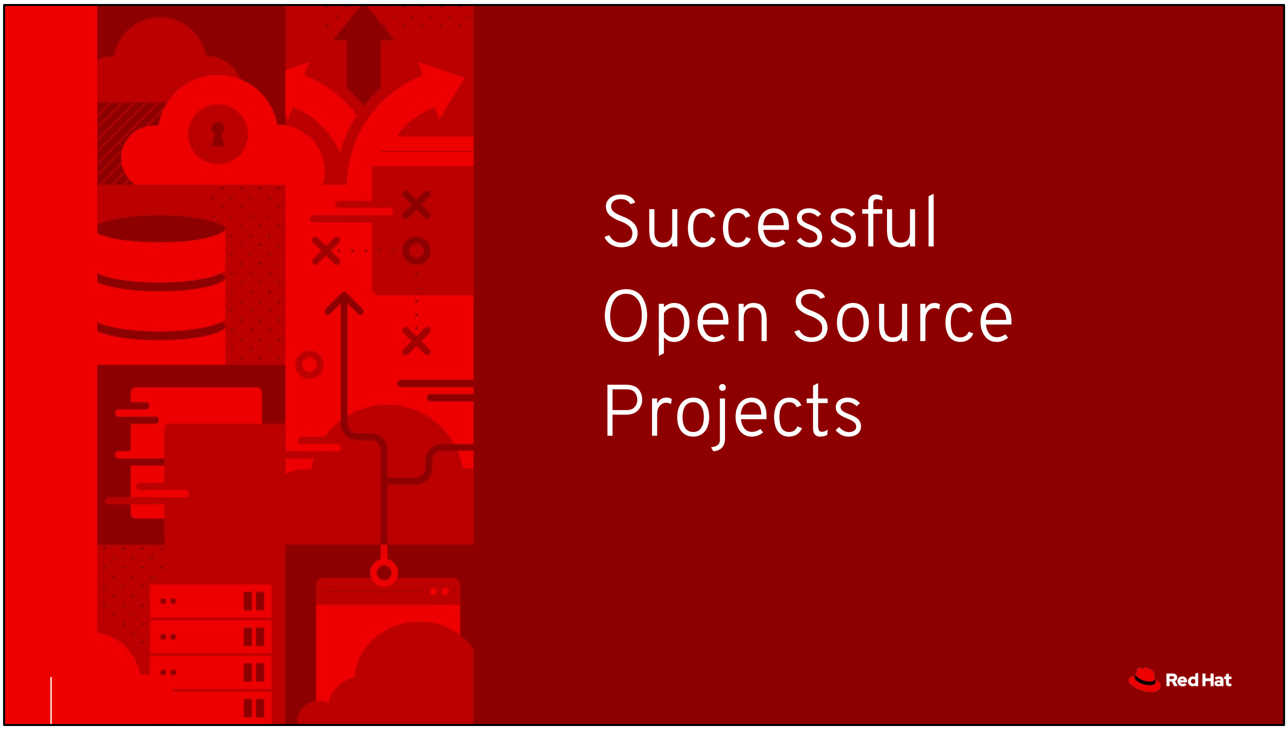
What is Open Source?

- You want to make gingerbread people so you put a recipe together and bake the first set of cookies however i want to bake minions so i use the ginger people recipe and then i just add on the different ingredients/techniques to the recipe for my cookies.
- You can do this with open source code so you are never reinventing the wheel and you are reusing code already out there etc.

The Power of Open



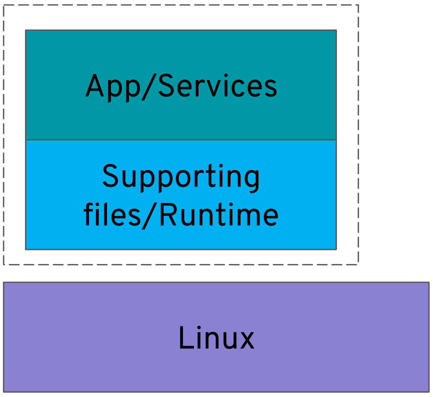
1. Collaboration
2. Feature selection
3. Application direction
4. Community



Successful Open Source Projects

- Going to speak a little about different open source projects *e.g.* rad.io Linux and Spark
- This will start with a stack explanation

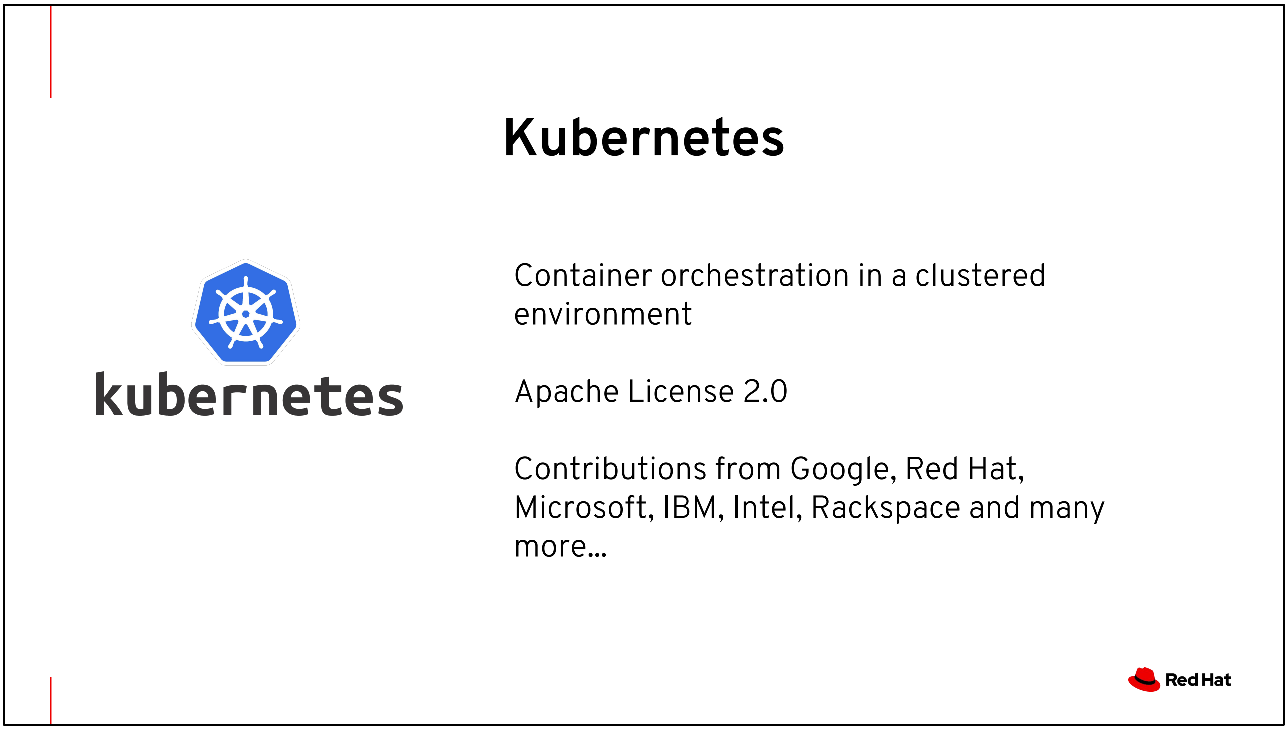
Containers



App/Services

Supporting files/Runtime

Linux



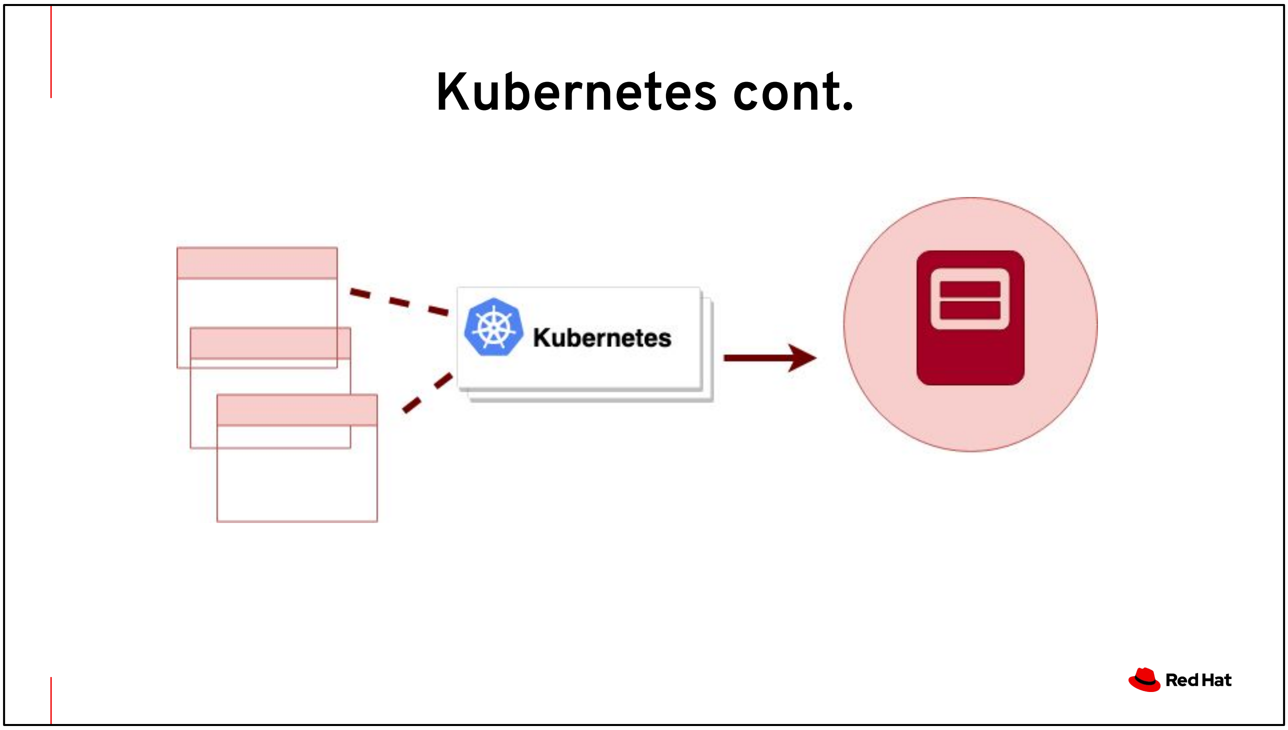
Kubernetes

Container orchestration in a clustered environment

Apache License 2.0

Contributions from Google, Red Hat, Microsoft, IBM, Intel, Rackspace and many more...

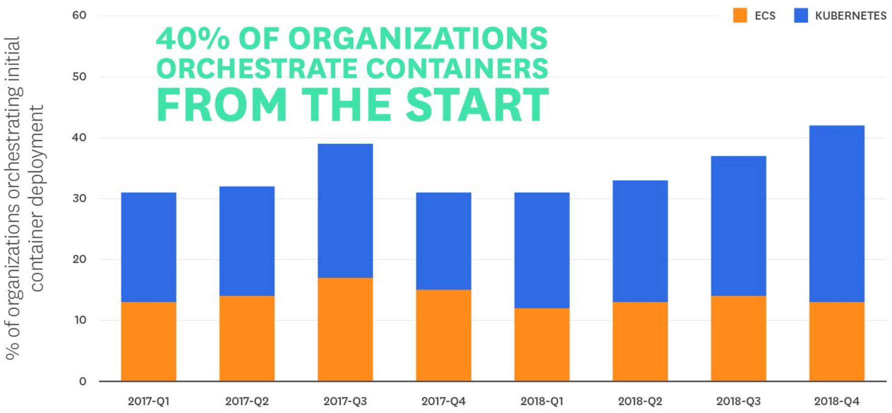
- Apache licence 2.0:Is strongly backed by community and allows you to freely use/modify and distribute projects.



Kubernetes cont.

- Groups containers to make an application into logical units for easy management and discovery.
- Released by Google but used worldwide now. Has a conference Kubecon which has over 4000 attendees.

Orchestration Usage at Initial Container Rollout



40% OF ORGANIZATIONS ORCHESTRATE CONTAINERS FROM THE START

Source: Datadog

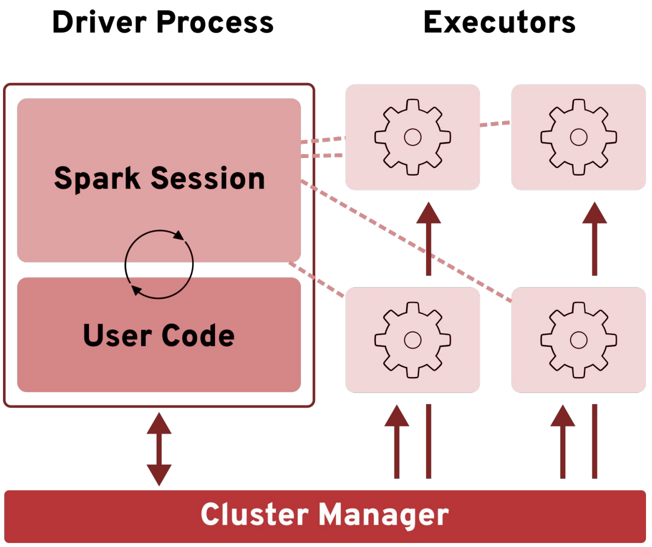
Red Hat

Openshift

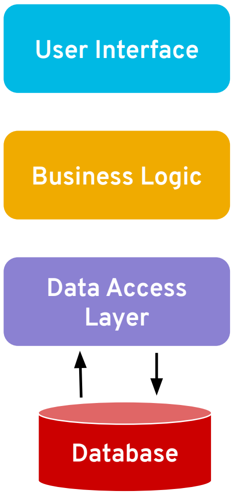
Red Hat
OpenShift

Kubernetes Enterprise Distribution

- Container security
- Application delivery and lifecycle
- Validated integrations
- Autoscaling


Driver Process
Executors
Spark Session
User Code
Cluster Manager

Monolithic Architecture



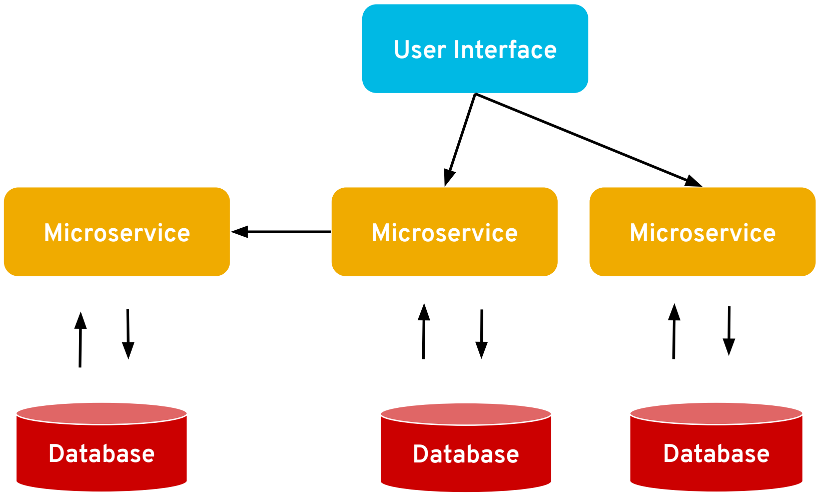
User Interface

Business Logic

Data Access Layer

Database

Microservices Architecture



User Interface

Microservice

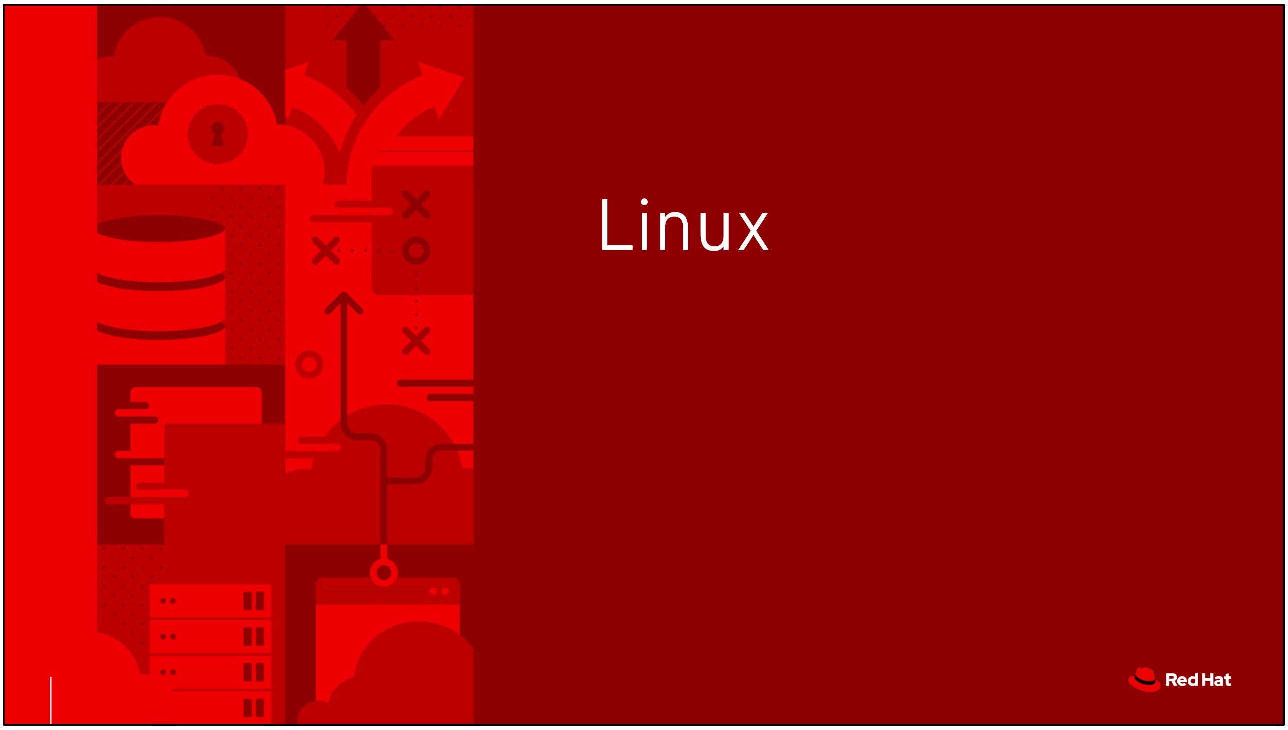
Microservice

Microservice

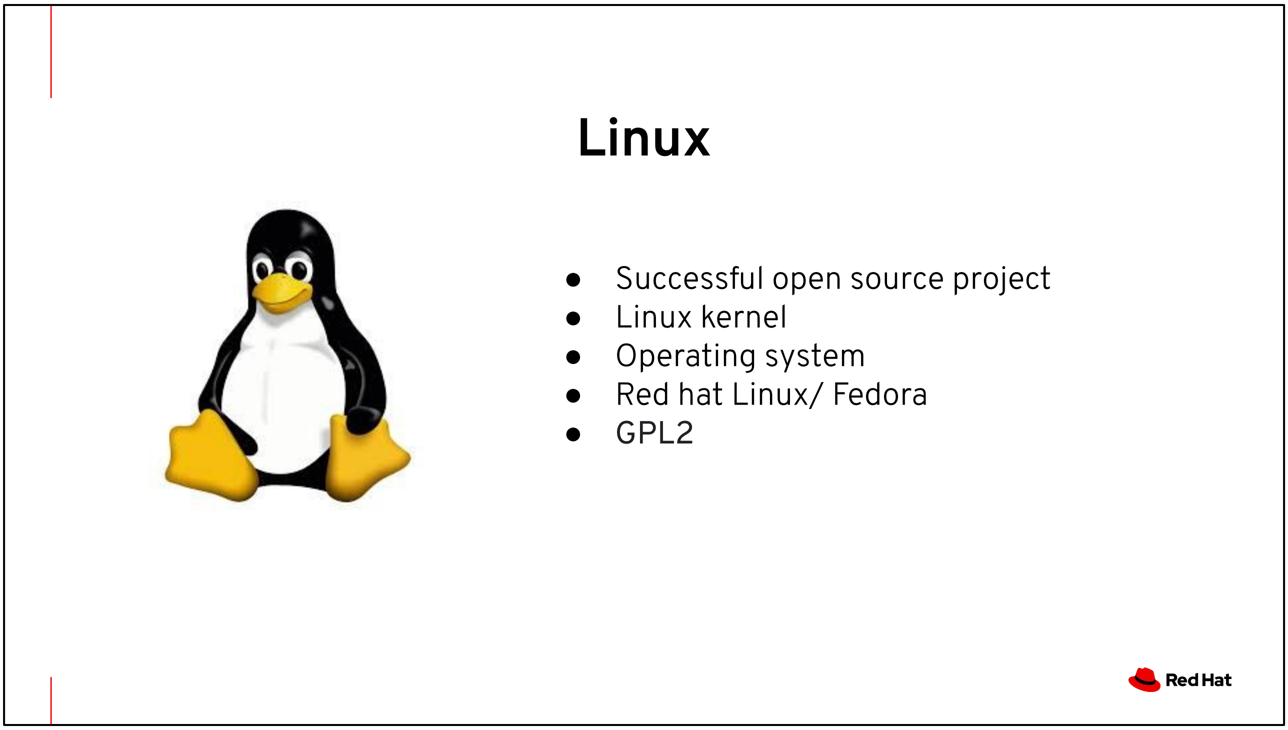
Database

Database

Database



Linux



Linux

- Successful open source project
- Linux kernel
- Operating system
- Red hat Linux/ Fedora
- GPL2

- Linux: The linux kernel was released in 1991 by Linus Torvalds.
- It was/is freely modifiable source code.
- This had mainly been restricted to colleges and universities and followed from the open source project GNU.
- It continues to this day being a popular open source operating system examples are Red Hat Linux and Fedora.
- GPL2: widely used free s/w licence which guarantees end users freedom to run/study/share and modify the s/w

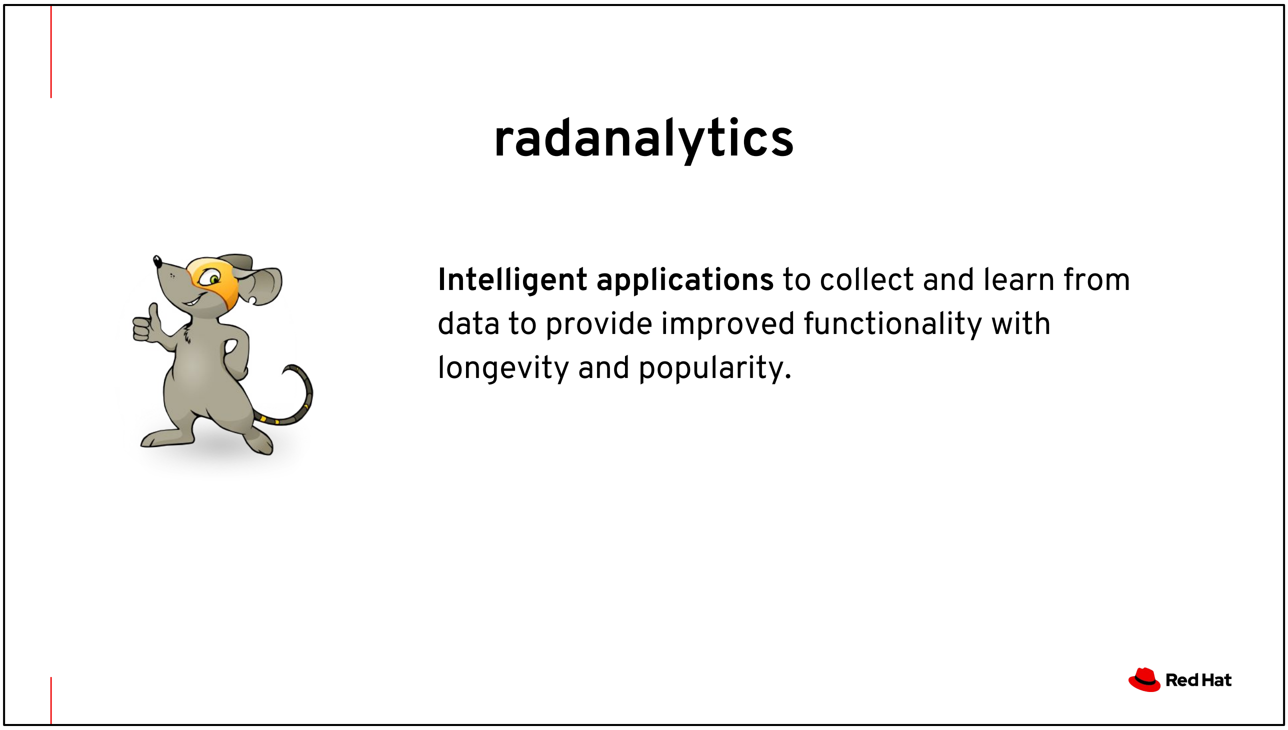
radanalytics

radanalytics

https://radanalytics.io/

Build **intelligent applications** for the cloud

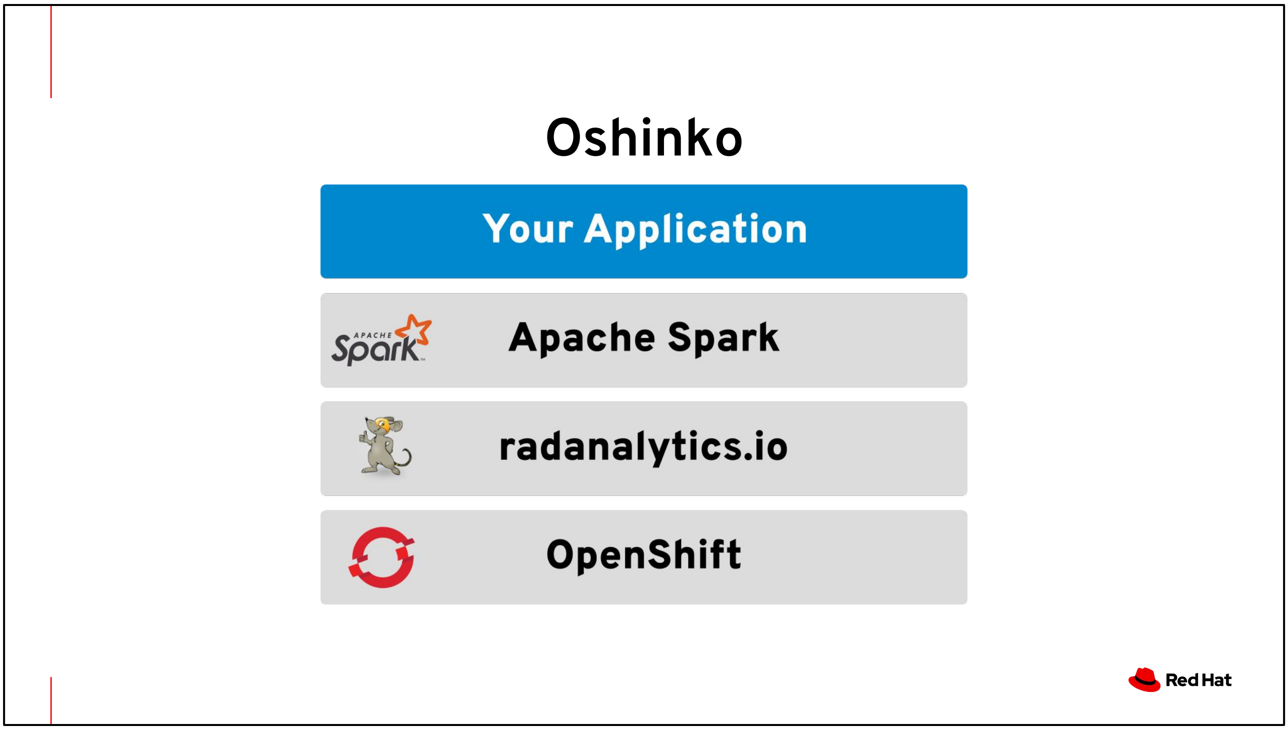
Learning resource



radanalytics

Intelligent applications to collect and learn from data to provide improved functionality with longevity and popularity.

- There is a current focus on apache spark within a project in rad.io called oshinko however this could be extended to use different tools for data processing or ML models. . .



Oshinko

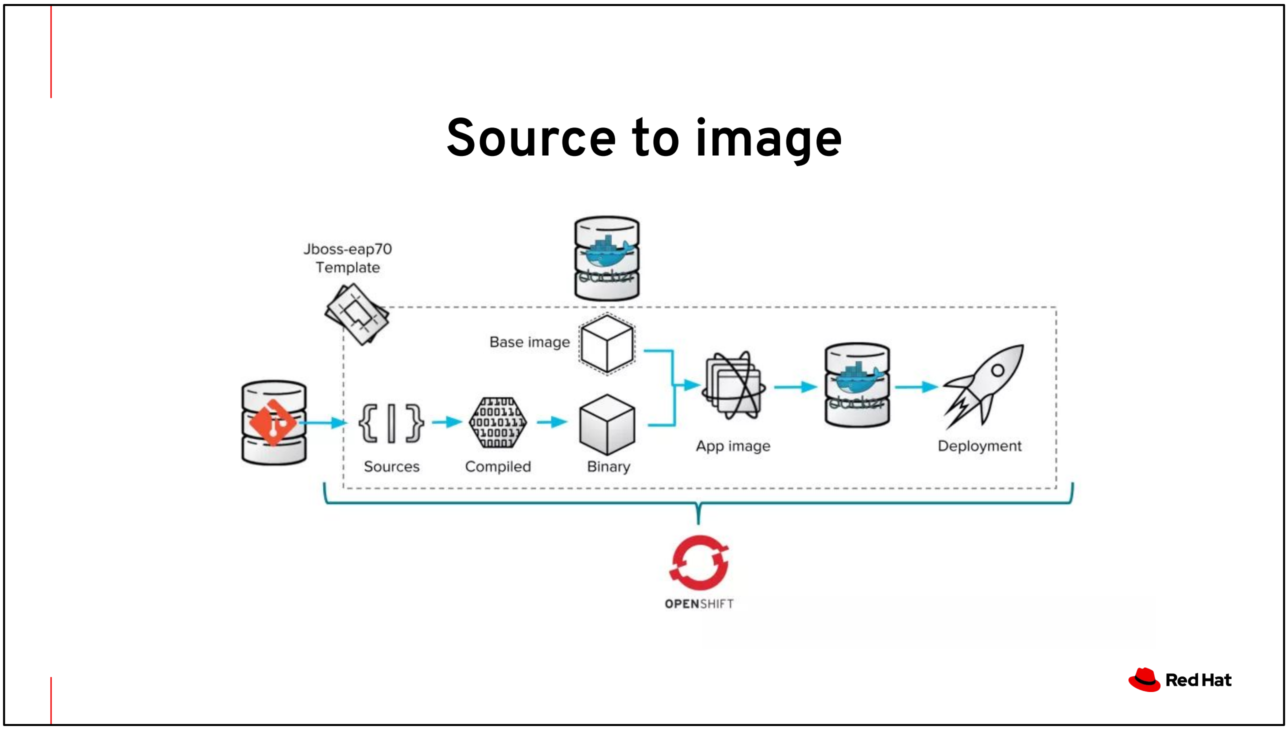
Your Application

Apache Spark

radanalytics.io

OpenShift

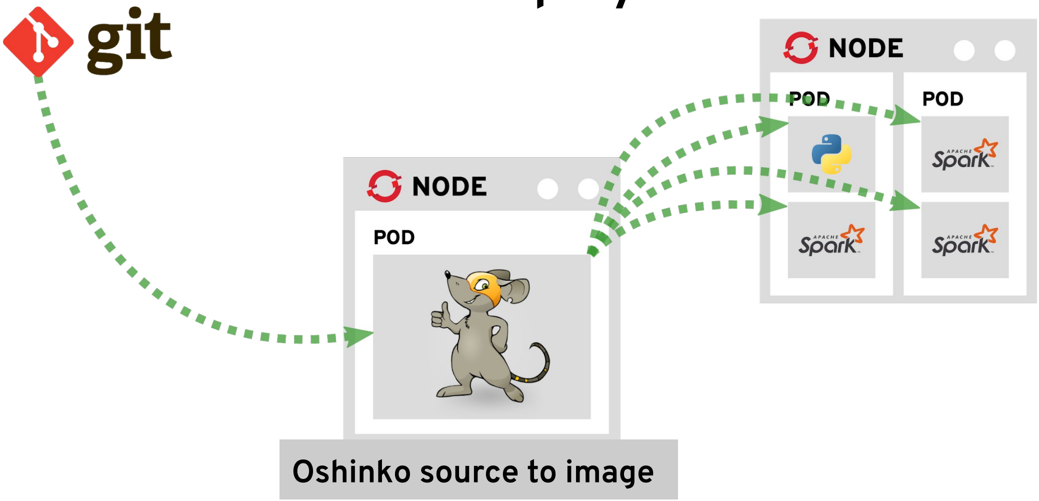
- Top level namespace describing different projects focused on apache spark deployment in openshift



Source to image

- S2i is providing images with reasonable defaults but are easily modifiable.
- It allows users to build containerized apps by simply supplying source code.
- S2i builds docker images using this source code.
- Describe diagram

Oshinko Deployment



Open Source Community

What is Community?

- Community is the people who support the project
  - Software engineers
  - Users
- They help to feature set
- Additions to the software itself

Setting up a Community

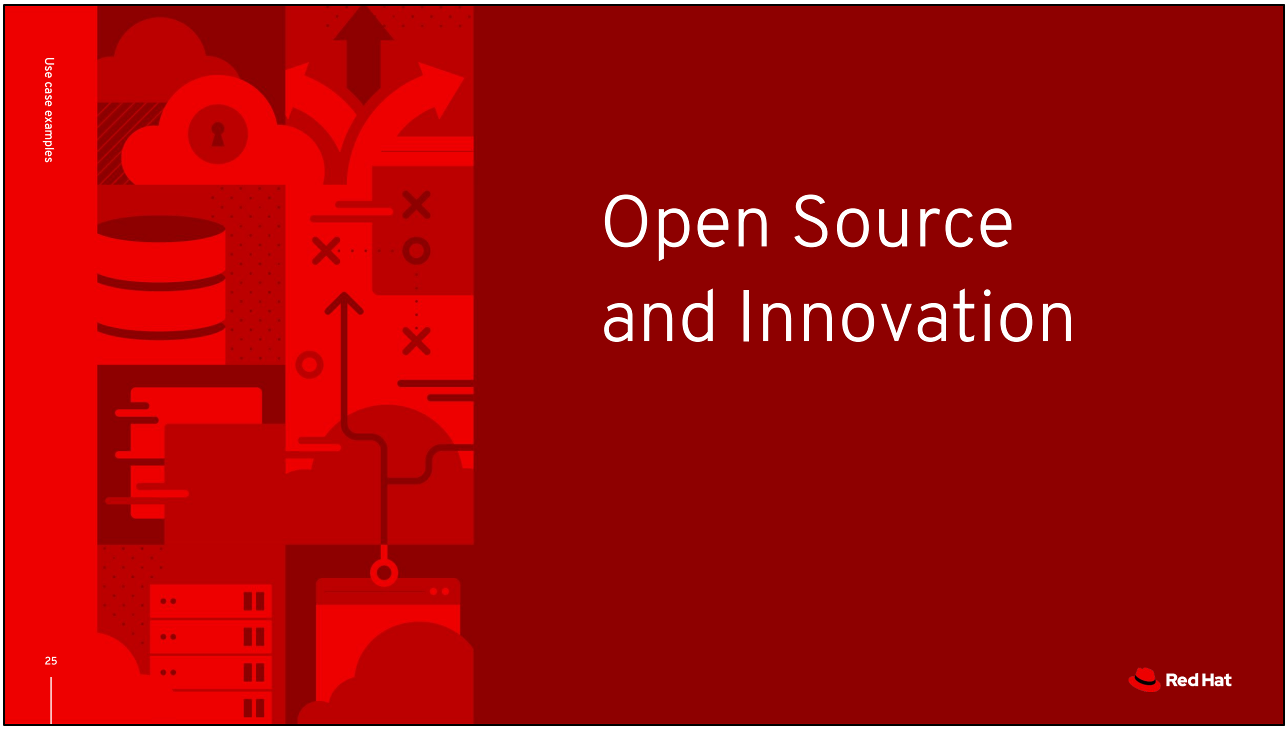
- Do you want a large community?
- Selective community, small but focused?
- How will the project be structured? - will you support growth yourself?

Decide this *before* making an open source a project

Red Hat

Example Communities

- Linux and Apache Spark
    - One person’s hobby
    - Grew quickly with interest
- Linux containers
    - Google and Redhat backed
    - Large community - world wide

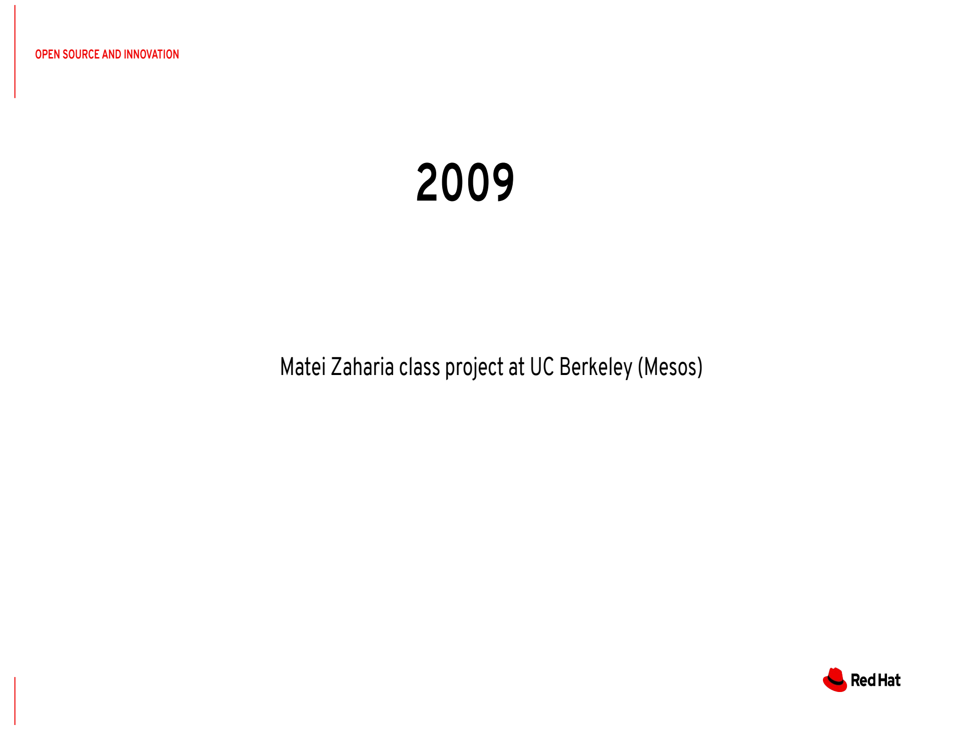


Use case examples

Open Source and Innovation

Red Hat

- Two specific use cases, built on Open Source technologies, to create AI and Machine Learning powered scalable applications on the cloud.
    - radanalytics.io, a distributed recommendation engine
    - reference architecture for end-to-end machine learning workflows, OpenDataHub.



OPEN SOURCE AND INNOVATION

2009

Matei Zaharia class project at UC Berkeley (Mesos)

Red Hat

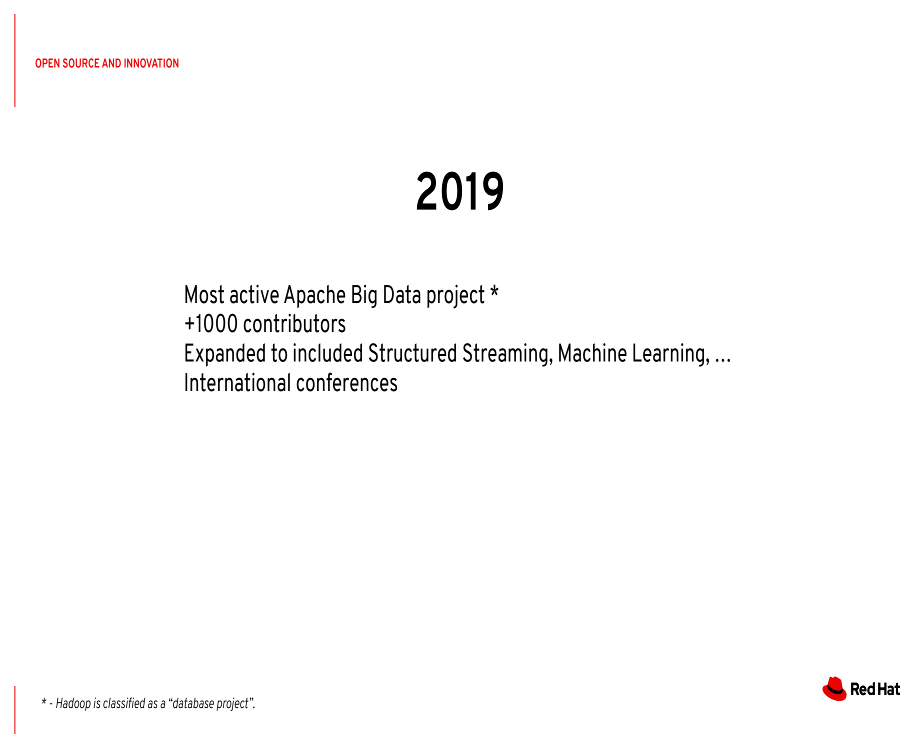
- 2009, in a class project at UC Berkeley, Matei Zaharia had the idea to build a simple cluster management framework, which would be open to different cluster computing systems.
- One he built it, he wondered what he could build on top of it.

OPEN SOURCE AND INNOVATION

2009

Matei Zaharia class project at UC Berkeley (Mesos)

- So he built Spark.



OPEN SOURCE AND INNOVATION

2019

Most active Apache Big Data project *
+1000 contributors
Expanded to included Structured Streaming, Machine Learning, ...
International conferences

** - Hadoop is classified as a "database project".*

- Fast forward 10 years.

Normally research projects get abandoned after a paper is published.

What was different?

There are many components. And if you look back, you can always revise history.

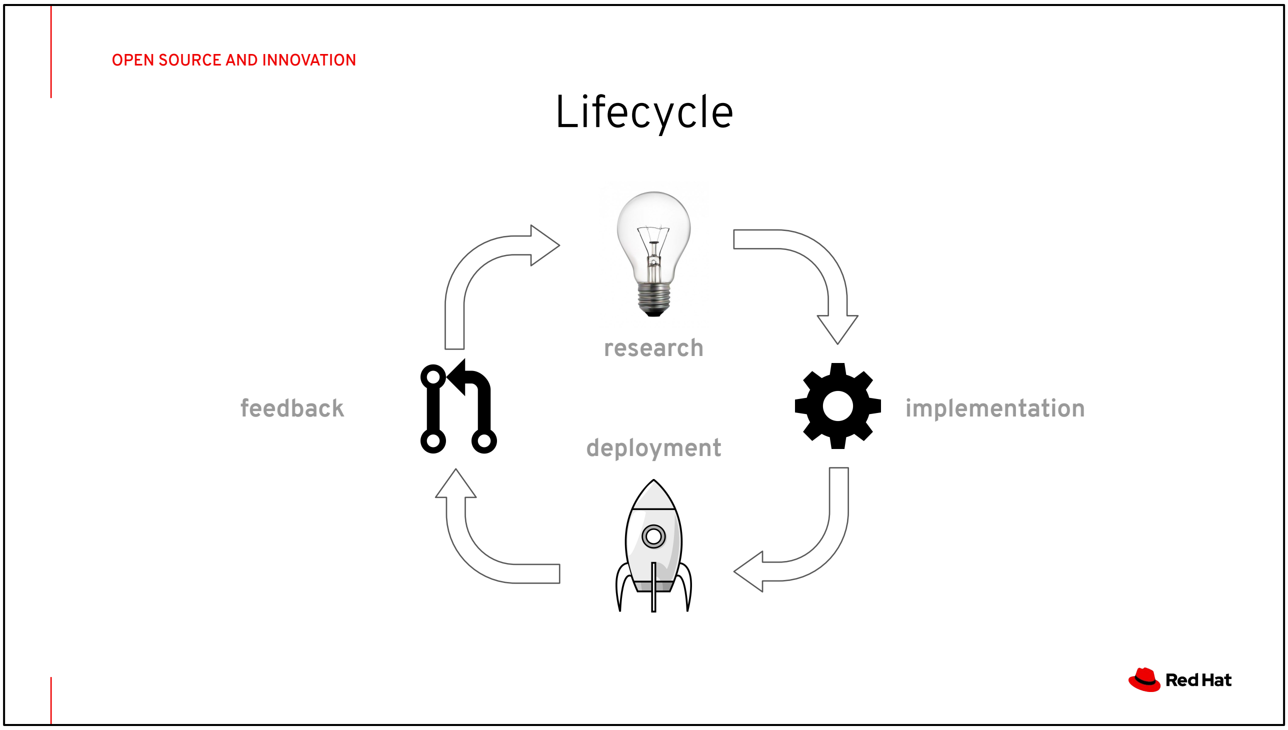
Especially if you had success.

First of all, we had a fantastic group of students.

Matei, the creator of Spark and others who did Mesos. And then another great group of **different students** who contributed and built different modules on top of Spark, and made what Spark it is today, which is really a platform. So, that's one: **the students**.

The other one was a **great collaboration with the industry**. We are seeing first hand what the problems are, challenges, so you're pretty anchored in reality.

- Matei in an interview



OPEN SOURCE AND INNOVATION

Lifecycle

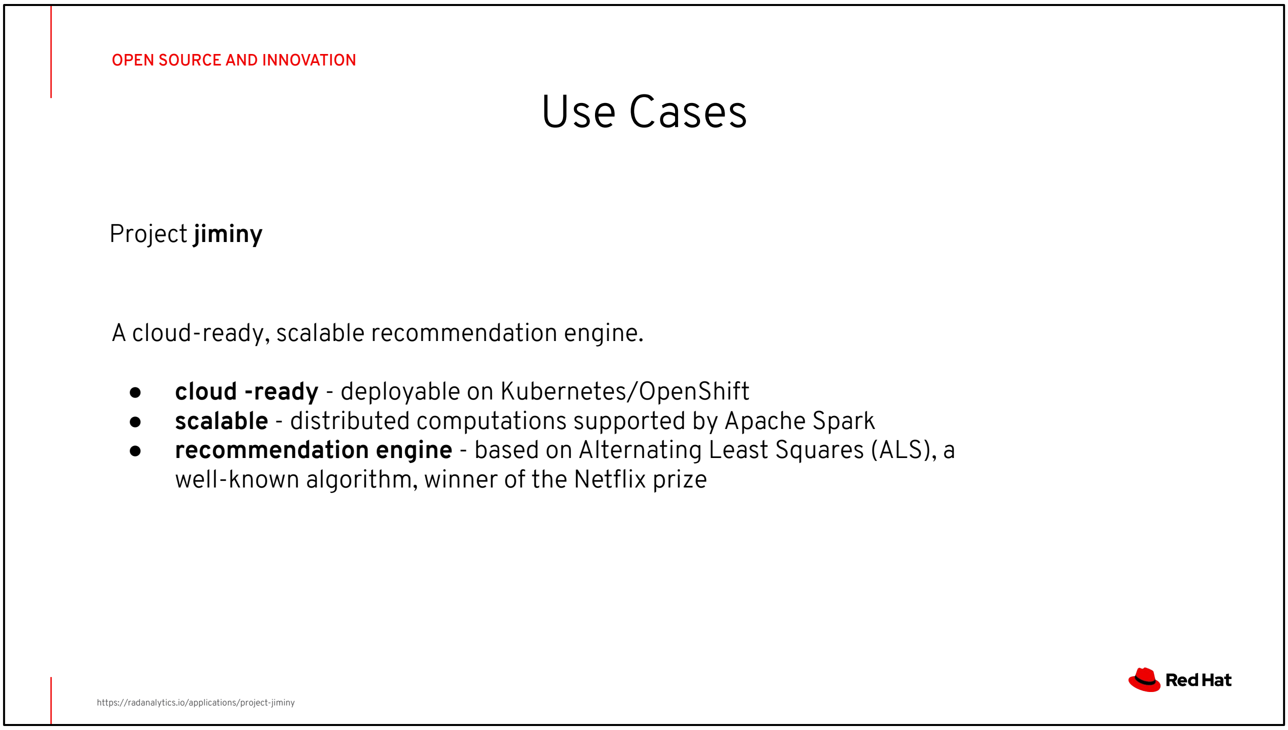
research

implementation

deployment

feedback

- Main strengths of community projects: foster innovation
- The typical lifecycle of a radanalytics project
    - start with an idea or a problem we were trying to solve within the scope of scalable intelligent applications
    - *e.g.* an architectural solution or some useful tool.
- release to the community, through the project's git repositories.
- Implementation used in different scenarios (real-world deployments, production or a teaching material)
- Feedback from the community
    - comments
    - improvements
    - bug reports
- Peer-reviewed -> project's codebase
- Repeat cycle
    - merge contributions



OPEN SOURCE AND INNOVATION

Use Cases

Project **jiminy**

A cloud-ready, scalable recommendation engine.

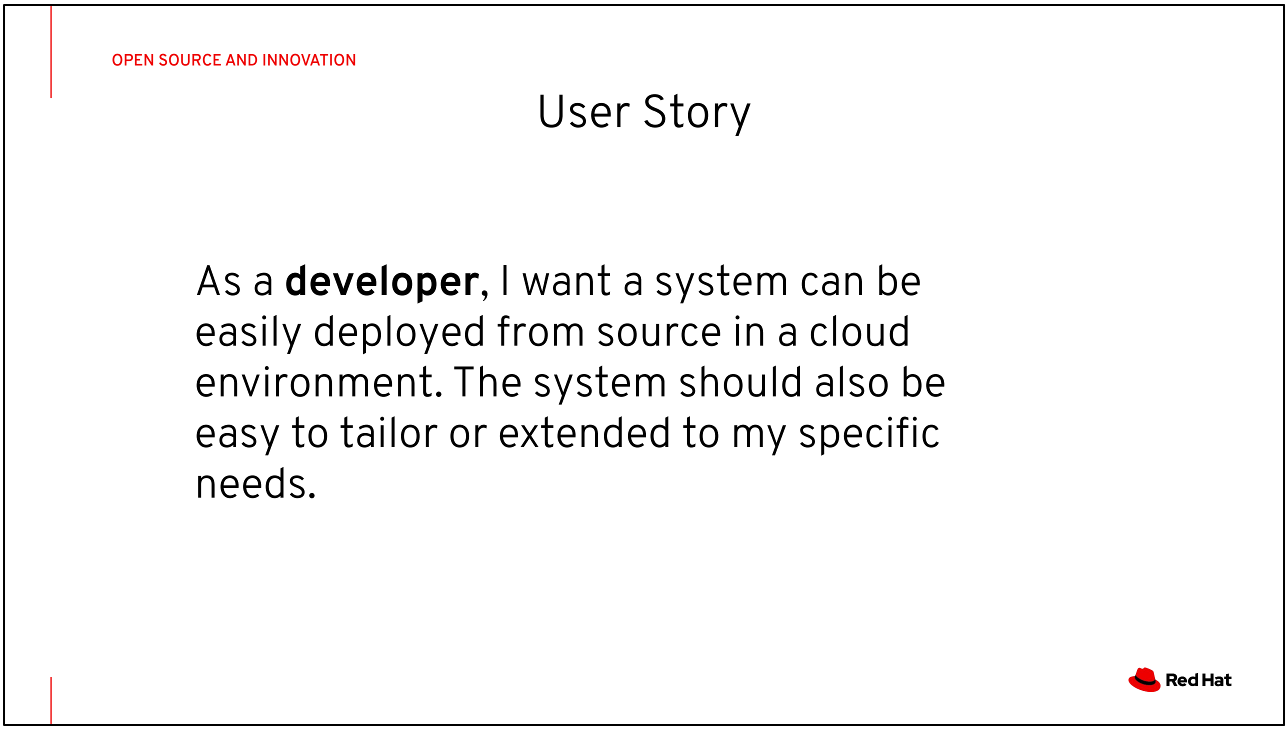
- **cloud -ready** - deployable on Kubernetes/OpenShift
- **scalable** - distributed computations supported by Apache Spark
- **recommendation engine** - based on Alternating Least Squares (ALS), a well-known algorithm, winner of the Netflix prize

https://radanalytics.io/applications/project-jiminy

Red Hat

- Several subprojects
  - Tooling
  - Architectural examples
- Project jiminy, a cloud-ready scalable recommendation engine tutorial
  - recommendation engine: a class of predictive models which can take pairs of users and products and predict an affinity, or a rating if you prefer between them. To do this, an algorithm for collaborative filtering, namely Alternating Least Squares (or ALS for short) is used
  - Cloud-ready: able to be deployed unmodified on K8s or OpenShift
  - Scalable: distributed computation with Apache Spark
    - increasing computational demands -> add more nodes to a cluster

- ALS: synergies between science, technological innovation and the software industry
- Netflix competition: ALS won. 10% increase in accuracy.
- Open Innovation
  - R&D open to everyone
  - Done under the public eye

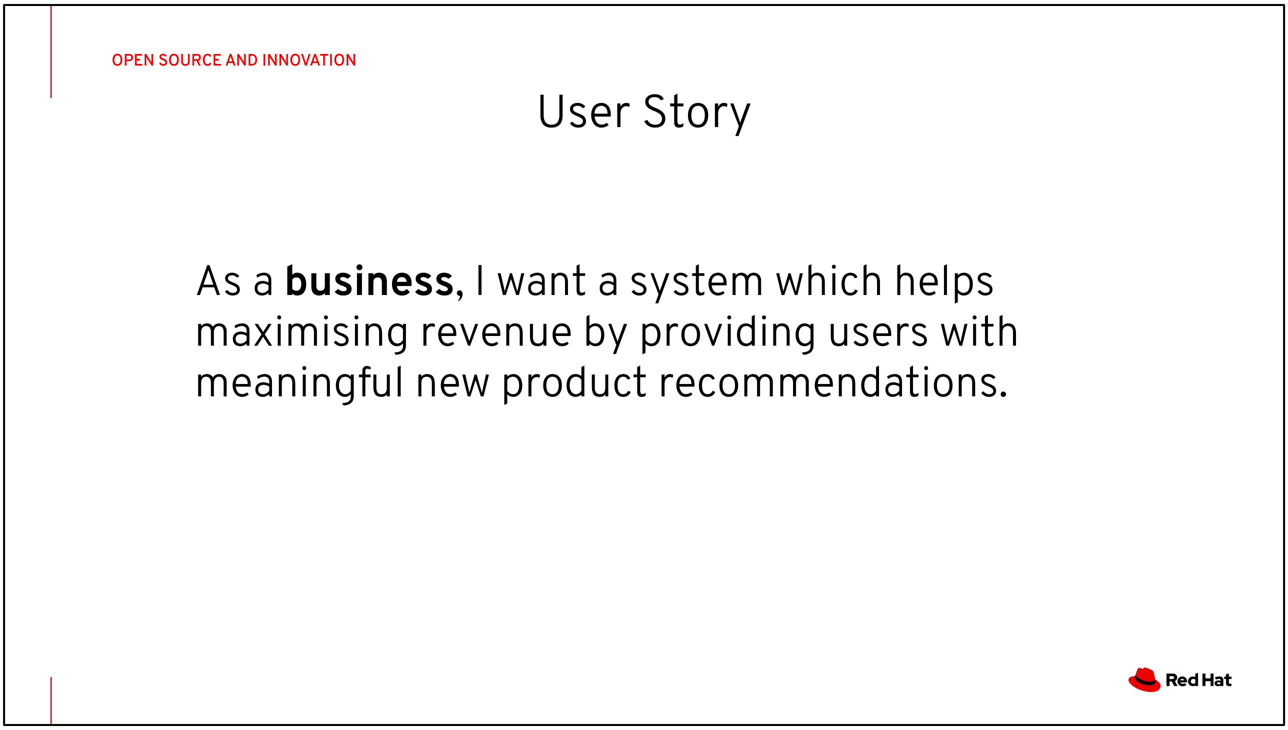


OPEN SOURCE AND INNOVATION

User Story

As a **developer**, I want a system can be easily deployed from source in a cloud environment. The system should also be easy to tailor or extended to my specific needs.

- Motivation for jiminy?
- Targeted personas:
  - Developers
    - off-the-shelf solution for a relatively complex system
    - open source => open to modification and tailoring to specific needs
      - *e.g.* changing explicit ratings to implicit ratings in the predictive model
      - customize the user interface



OPEN SOURCE AND INNOVATION

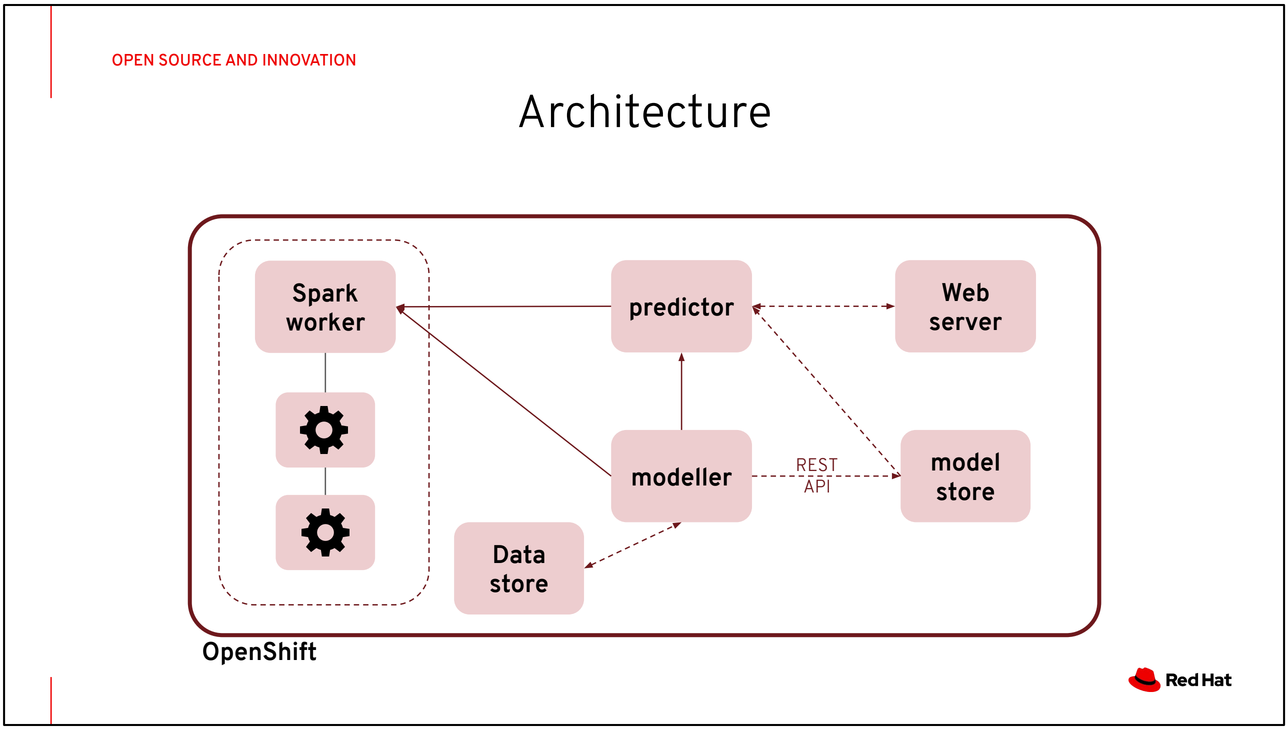
User Story

As a **business**, I want a system which helps maximising revenue by providing users with meaningful new product recommendations.

Red Hat

User Story

As an **data scientist,** I want a system which is flexible enough to let me focus on the recommendation algorithms.



OPEN SOURCE AND INNOVATION

Architecture

Spark worker

predictor

Web server

modeller

REST API

model store

Data store

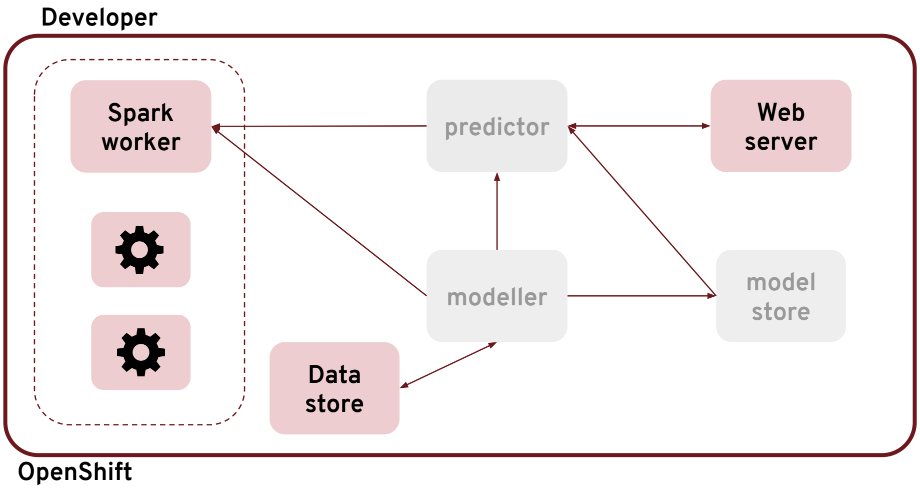
OpenShift

- How different personas contribute/use?
- microservice based architecture
    - set of components
    - clear separation concerns
    - communicating via well defined APIs
    - typically a REST interface

- data store: manages the historical ratings data
- Modeller: model training
- model store: model provisioning, versioning and storage
- Predictor: use trained model to perform predictions
- web server: connects user requests with the rest of the system
- computations are decoupled by delegating them to a Spark cluster

OPEN SOURCE AND INNOVATION

Architecture



Developer

Spark worker

predictor

Web server

modeller

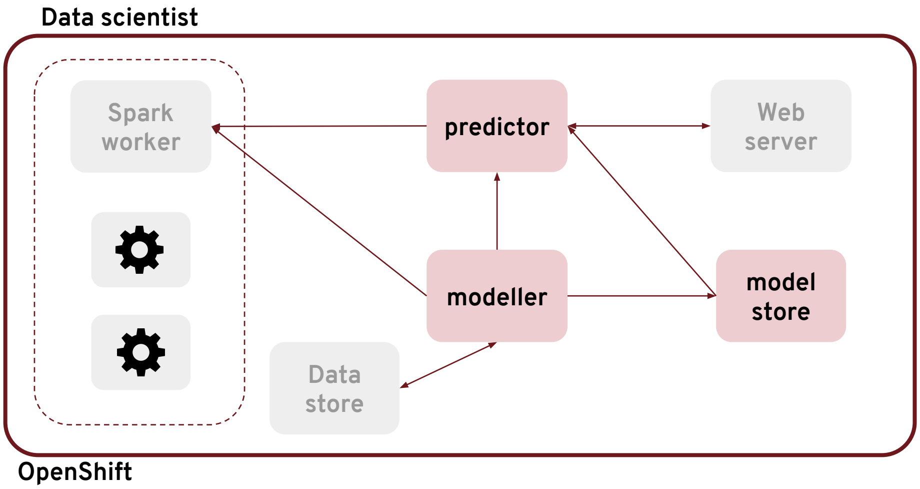
model store

Data store

OpenShift

- different personas focus on different areas
  - Developers: UI or data storage

Architecture



Data scientist

Spark worker

predictor

Web server

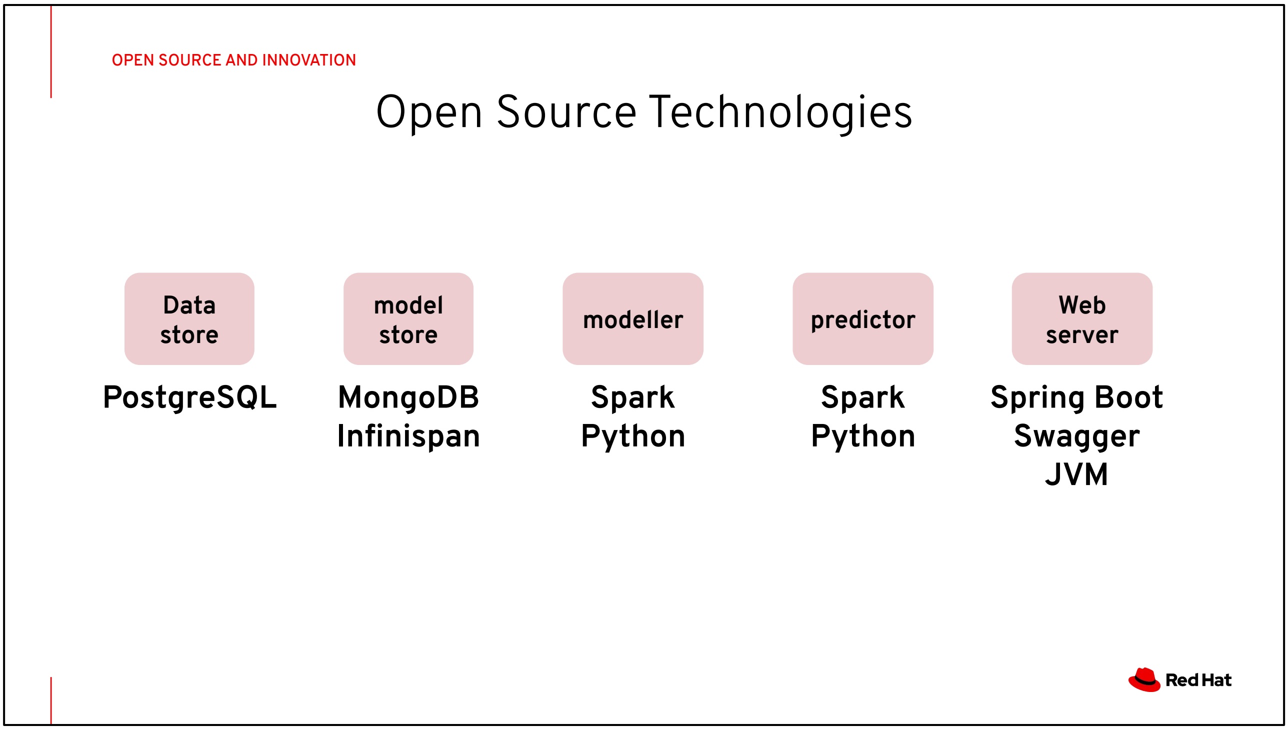
modeller

model store

Data store

OpenShift

- data scientists:
  - modelling



OPEN SOURCE AND INNOVATION

Open Source Technologies

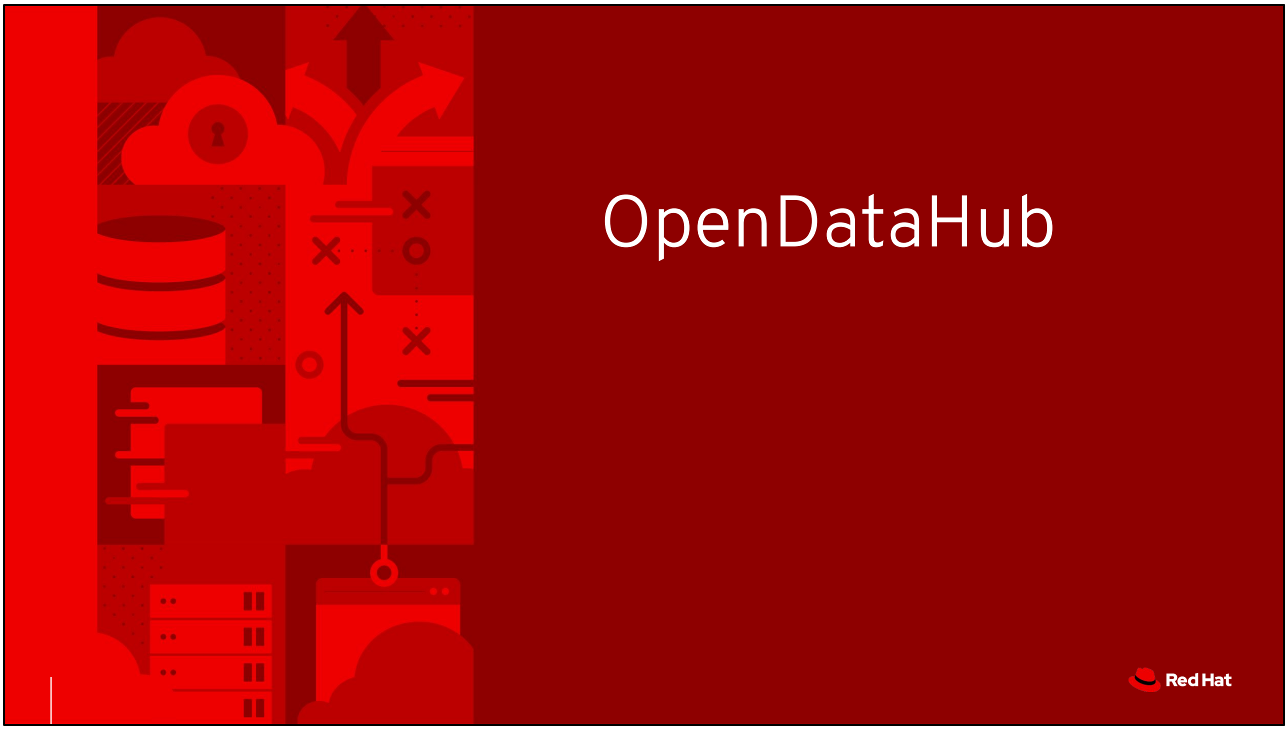
Data store	model store	modeller	predictor	Web server
PostgreSQL	MongoDB Infinispan	Spark Python	Spark Python	Spring Boot Swagger JVM

- enable parallel development
- polyglot development.
  - Data scientist -> Python
  - UI engineers -> preferred stack
- modules could be refactored as long as the API remains the same
- Each component released as a separate repository -> encourage the community to write their own implementations.

Engagement

Projects used as:

- learning resources
  - Workshops, conferences
- Technology showcases
- Basis for customised solutions

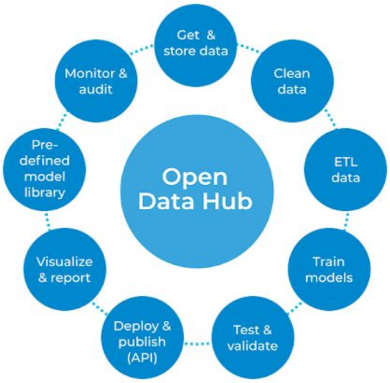


OpenDataHub

OpenDataHub

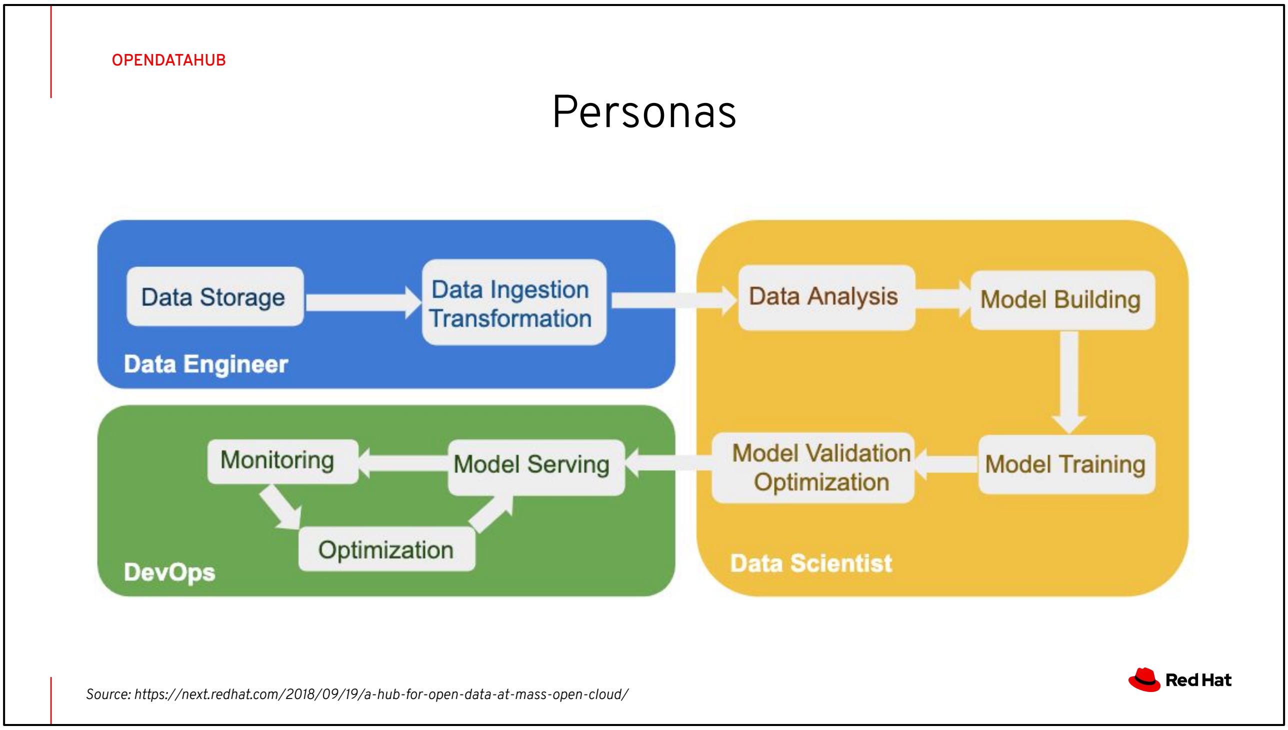
A **reference architecture** for an AI and Machine Learning as a **service platform** for OpenShift built using **open source tools**

End-to-End



End-to-end Security & Compliance

Source: https://next.redhat.com/2018/09/19/a-hub-for-open-data-at-mass-open-cloud/



OPENDATAHUB

Personas

Data Storage

Data Ingestion Transformation

Data Engineer

Data Analysis

Model Building

Model Training

Model Validation Optimization

Data Scientist

Model Serving

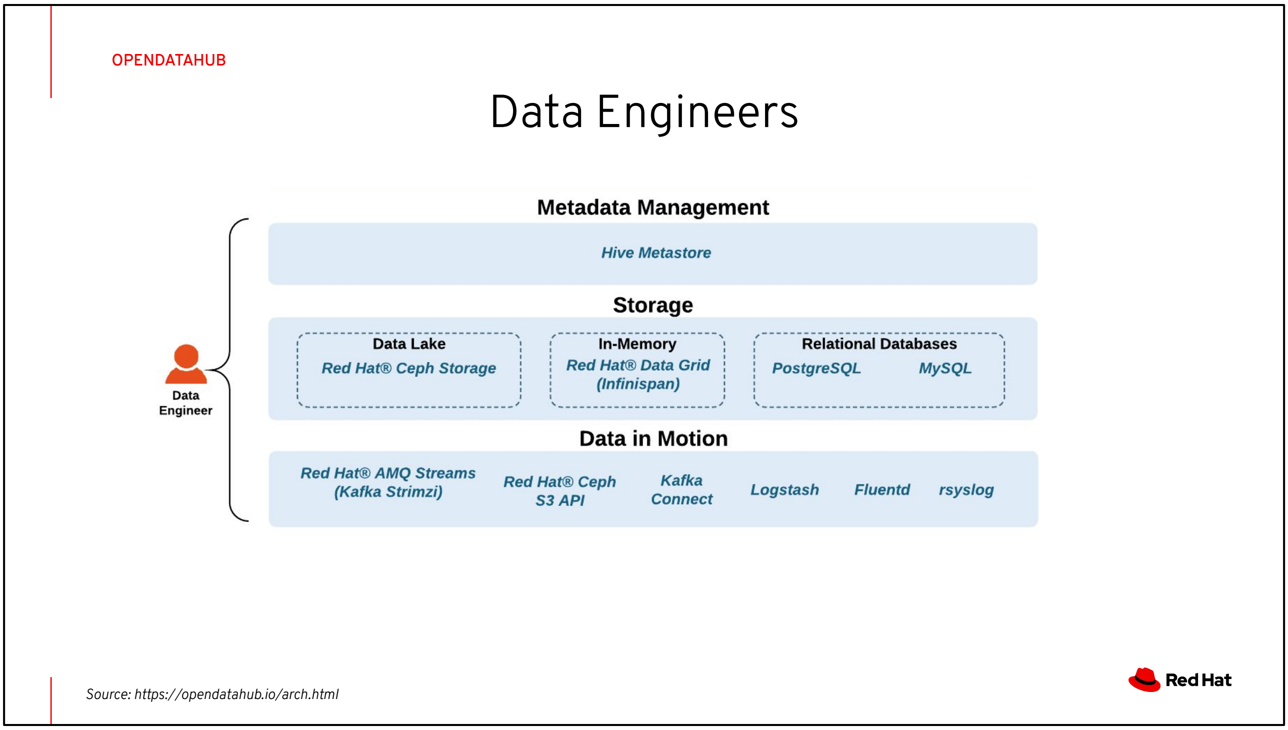
Monitoring

Optimization

DevOps

Source: https://next.redhat.com/2018/09/19/a-hub-for-open-data-at-mass-open-cloud/

- typical AI workflow step
- aimed at multiple personas
- their fit in in a end-to-end AI workflow



OPENDATAHUB

Data Engineers

Data Engineer

Metadata Management

Hive Metastore

Storage

Data Lake
Red Hat® Ceph Storage

In-Memory
Red Hat® Data Grid (Infinispan)

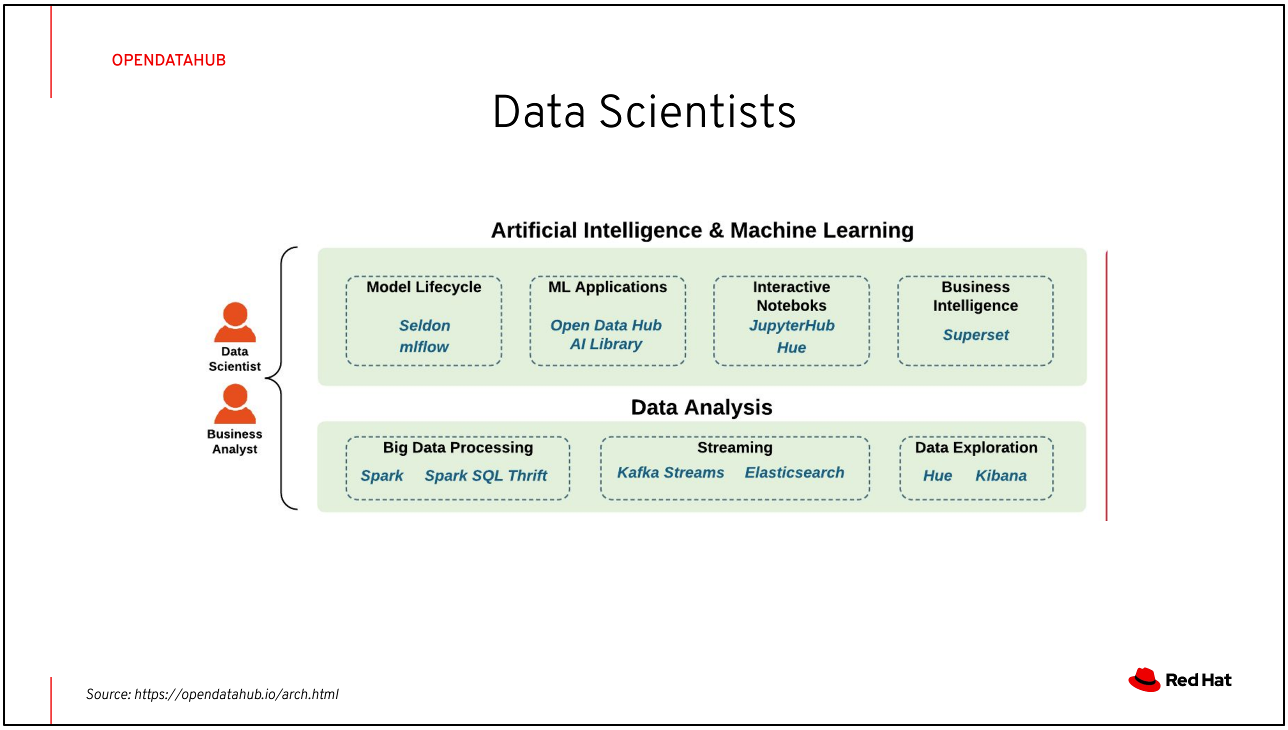
Relational Databases
PostgreSQL *MySQL*

Data in Motion

Red Hat® AMQ Streams (Kafka Strimzi) *Red Hat® Ceph S3 API* *Kafka Connect* *Logstash* *Fluentd* *rsyslog*

Source: https://opendatahub.io/arch.html

- **Data in Motion**
  - data resides in multiple locations
  - support data stored in legacy systems
  - Hybrid Cloud => sharing data between different cloud systems
  - Tools:
    - Red Hat AMQ Streams
    - Kafka
    - Logstash
    - native data transfer capabilities
- **Storage**
  - Data Lake/Databases/In-Memory
  - distributed files
  - {block, object} storage
  - relational databases + document-oriented databases
  - RHDG -> Ceph
  - High performance in-memory -> Infinispan (fast data access needed)
- **Metadata Management**
  - Hive Metastore-> SQL interface to access the metadata information



OPENDATAHUB

Data Scientists

Artificial Intelligence & Machine Learning

Data Scientist

Business Analyst

Model Lifecycle
Seldon
mlflow

ML Applications
Open Data Hub
AI Library

Interactive Noteboks
JupyterHub
Hue

Business Intelligence
Superset

Data Analysis

Big Data Processing
Spark *Spark SQL Thrift*

Streaming
Kafka Streams *Elasticsearch*

Data Exploration
Hue *Kibana*

Source: *https://opendatahub.io/arch.html*

Red Hat

- **Data Analysis**
  - Apache Spark (operator) -> OCP distributed cluster
  - Support for ephemeral Spark clusters
  - Data Exploration:
    - Hue -> SQL interface to query the data and basic visualization
    - Kibana
    - Elasticsearch

- **Artificial Intelligence and Machine Learning**
  - Model Lifecycle tools
  - Seldon: model hosting + metric collection
  - MLflow: parameter tracking for models

Mass Open Cloud (MOC)

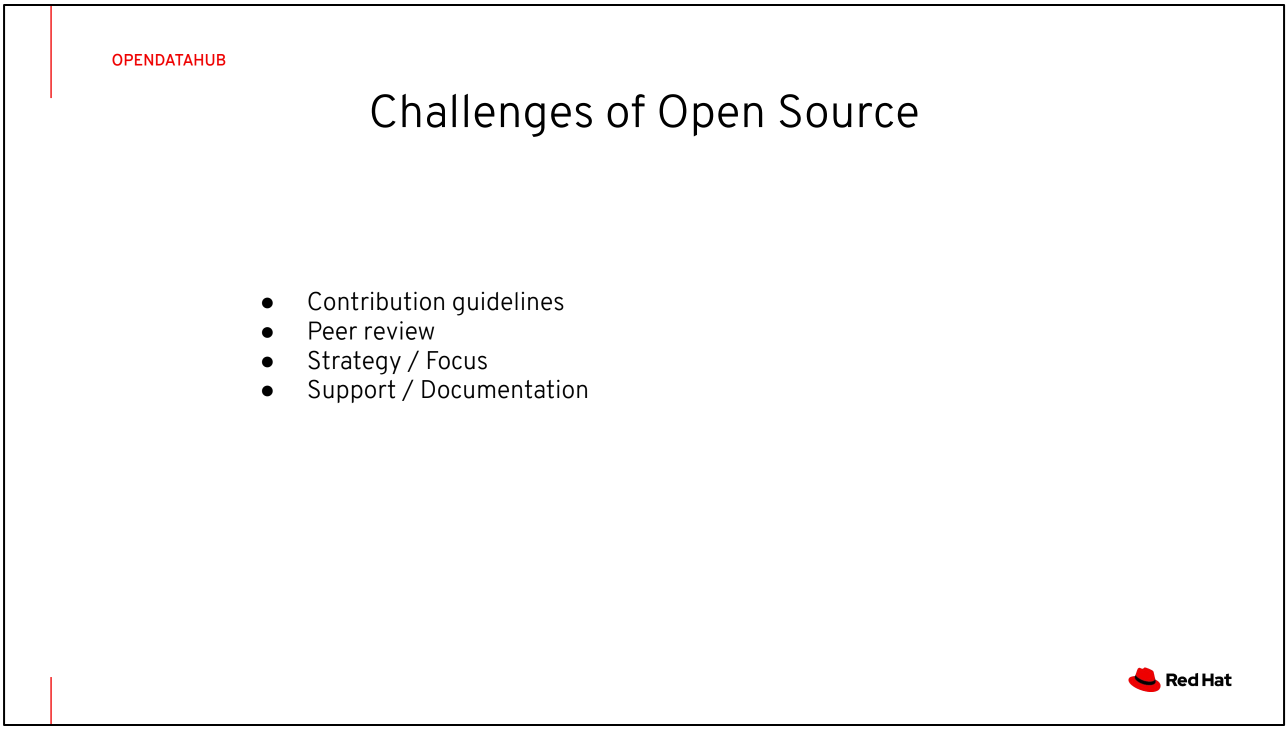
1. To create an inexpensive and efficient at-scale production cloud utility suitable for sharing and analyzing massive data sets and supporting a broad set of applications.
2. To create and deploy the OCX model, enabling a healthy marketplace for industry to participate at all levels in the cloud and profit from doing so.
3. To create a testbed for research in and prototyping of cloud technology, empowering a broad community of researchers, open source developers and companies to develop new cloud computing technologies.

OPENDATAHUB

Mass Open Cloud (MOC)

Project’s core partners:

- Academic (Boston University, Harvard University, Northeastern University, MIT)
- Government (Massachusetts Technology Collaborative, United States Air Force)
- Non-profit (MGHPCC)
- Industry (Cisco, Intel, NetApp, Red Hat, Two Sigma)



OPENDATAHUB

Challenges of Open Source

- Contribution guidelines
- Peer review
- Strategy / Focus
- Support / Documentation

Red Hat

- Building a successful community?

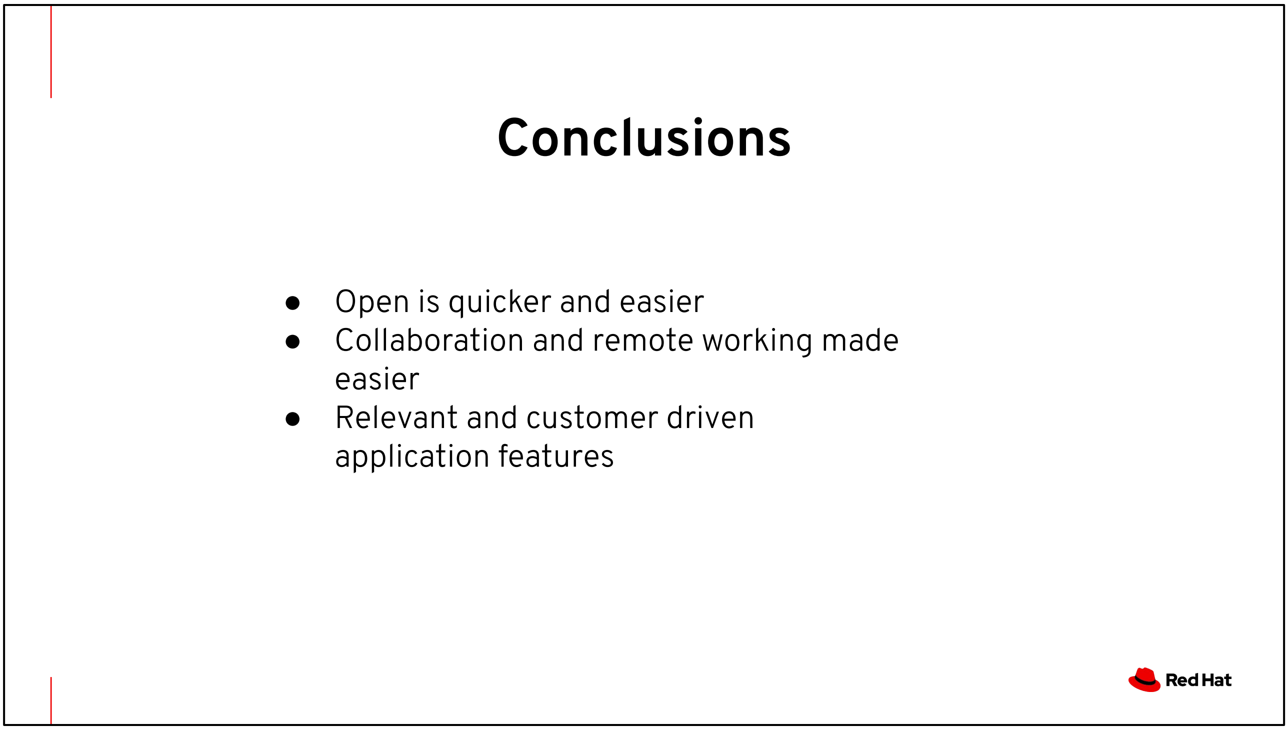
Conclusions

Lessons learnt

- Open needs to be planned
- Communities need to be nourished to succeed

BUT

- You can have a hobby project
- Experiment and find your ideal spot



Conclusions

- Open is quicker and easier
- Collaboration and remote working made easier
- Relevant and customer driven application features

How you can get involved

https://radanalytics.io/
https://opendatahub.io/

Contact us:

rsimmond@redhat.com
rui@redhat.com