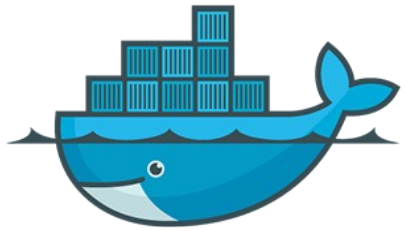
Scalable Processing for Storage Events

Leveraging dCache events for handling scientific data

Michael Schuh, P.Fuhrmann, T.Hartmann, P.Millar, T.Mkrtchyan, M.Sahakyan
Jan 27 2020

Software stack event driven computing

Cloud Computing
Container Orchestration
Software Defined Networking
Infrastructure as code



kubernetes

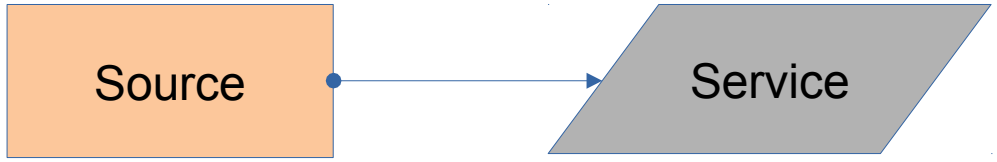
Peta-scale distributed storage
Storage events
Kafka Stream API

Function as a service
Distributed computing
Science Notebooks

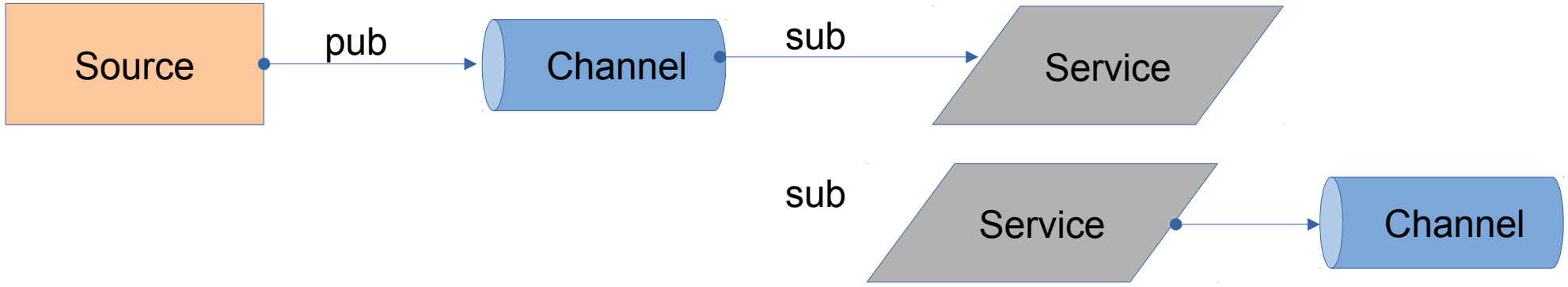
Scalable Event-Driven Service Architecture

Producers	Sensors, detectors, storage systems, user
Channels	Message-oriented middleware, Kafka Topics, direct peer to peer
Consumers	Cloud Functions (processing or filter+transform+forward)

Direct messaging



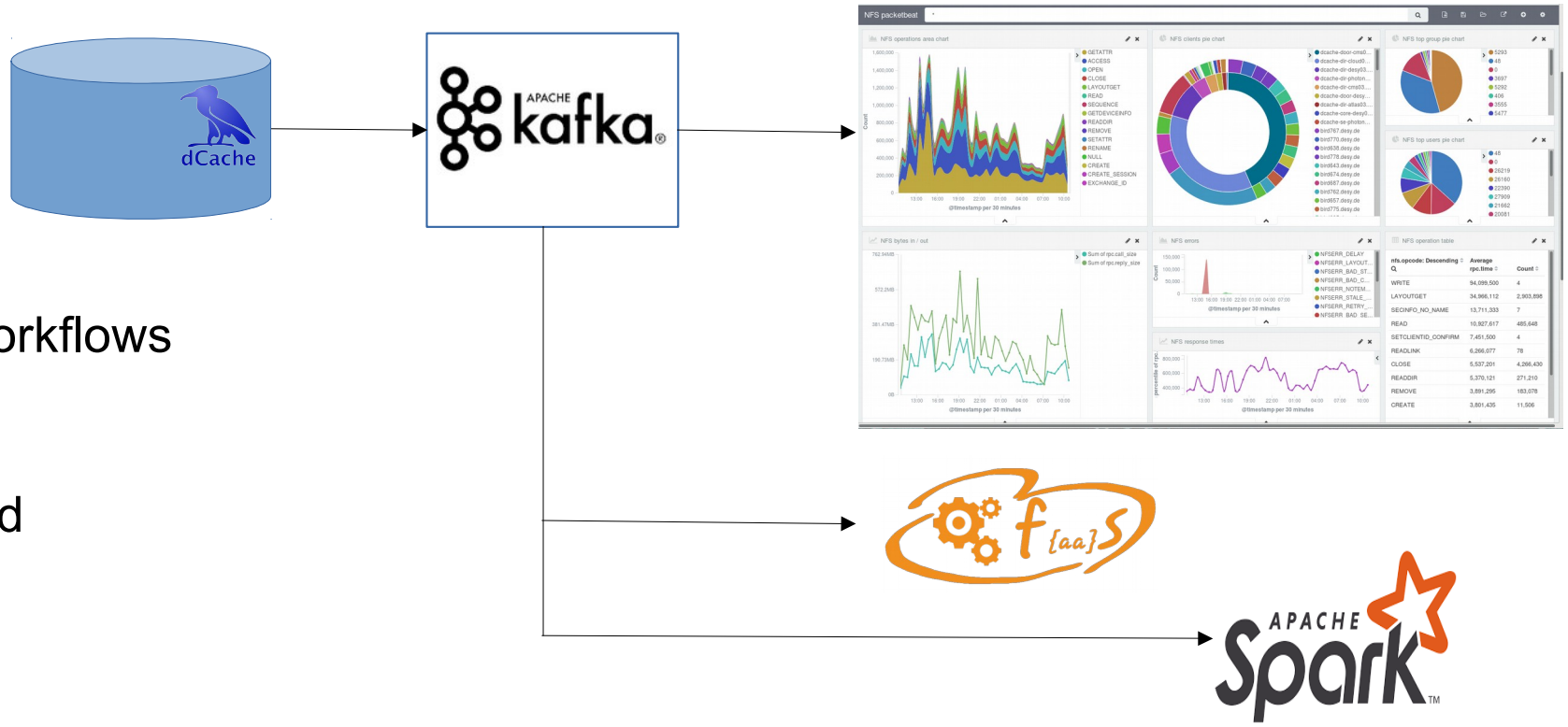
Publish/Subscribe messaging, complex processing



Based on: https://knative.dev/docs/eventing/

Primary Example: Storage Events

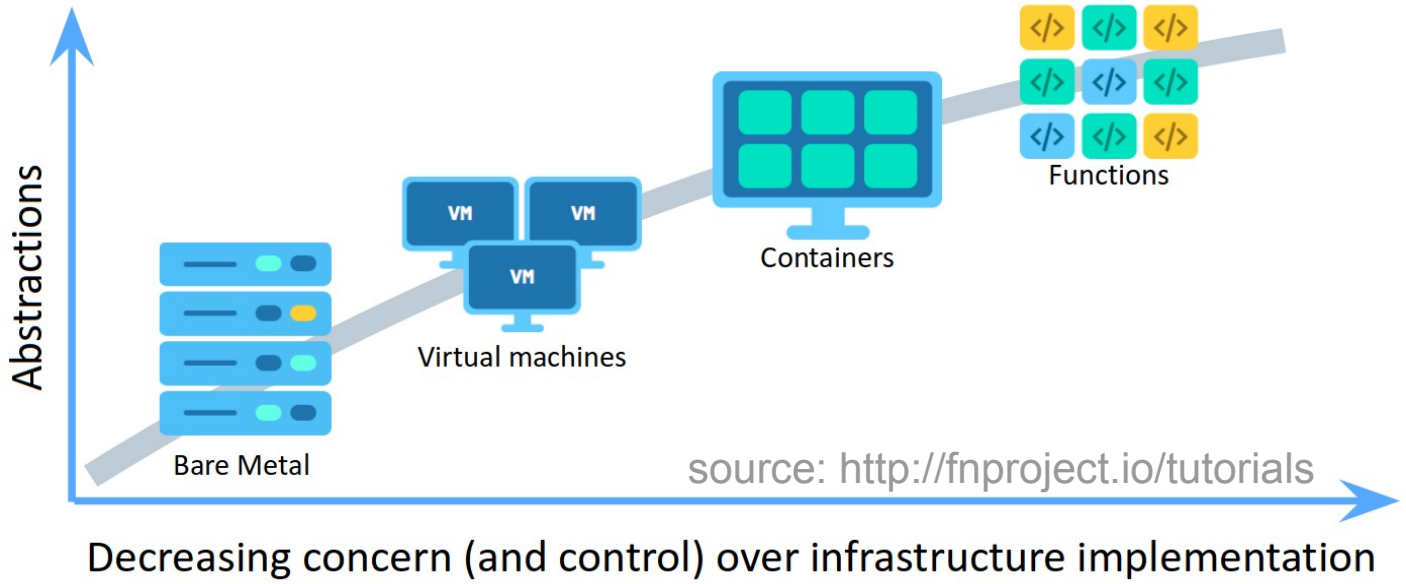
- Leverage the storage provider log
  - Analyse and tune dCache, a petascale mass storage system
  - Real-time analysis, large scale time series (Apache Spark)
  - Export to Elastic Search & Kibana
  - Trigger functions (uncompress, malware scan, ...)



- FaaS for data-driven workflows
  - Push data
  - Subscribe to events
  - Run codes automated

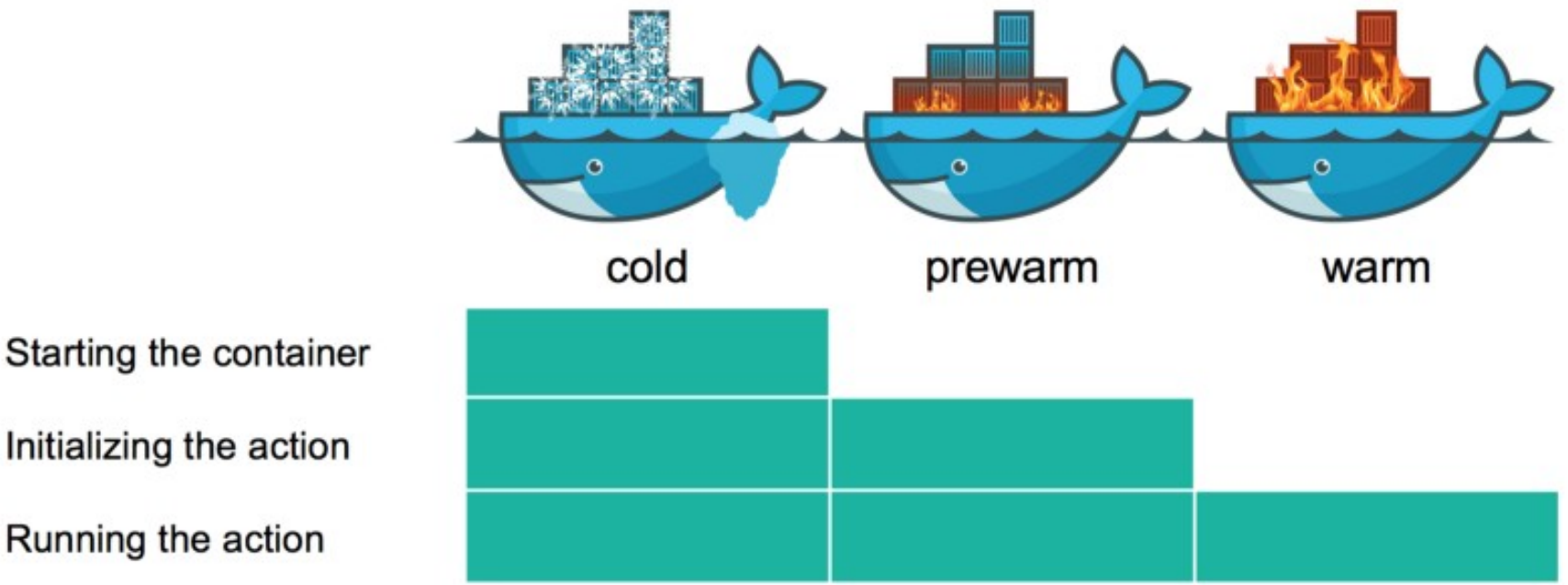
Function-as-a-Service

- Event-driven code execution
- Auto-scaling
- All programming languages



- Basic: Invoke function by FaaS API or client, receive output & logs
  Create function from file
  Create function from container
- Advanced: Expose URL for function, leverage HTTP features
- Expert: Build complex RESTful APIs from functions

Scaling on Demand, Avoidance of Idling



- scale-to-zero, accept cold start penalty O(0.1s)
- scale-to-one, reserve memory, keep low latency functions pre-warm
- re-use stateless containers

https://medium.com/openwhisk/squeezing-the-milliseconds-how-to-make-serverless-platforms-blazing-fast-aea0e9951bd0

Auto-scaling, event-triggered micro services

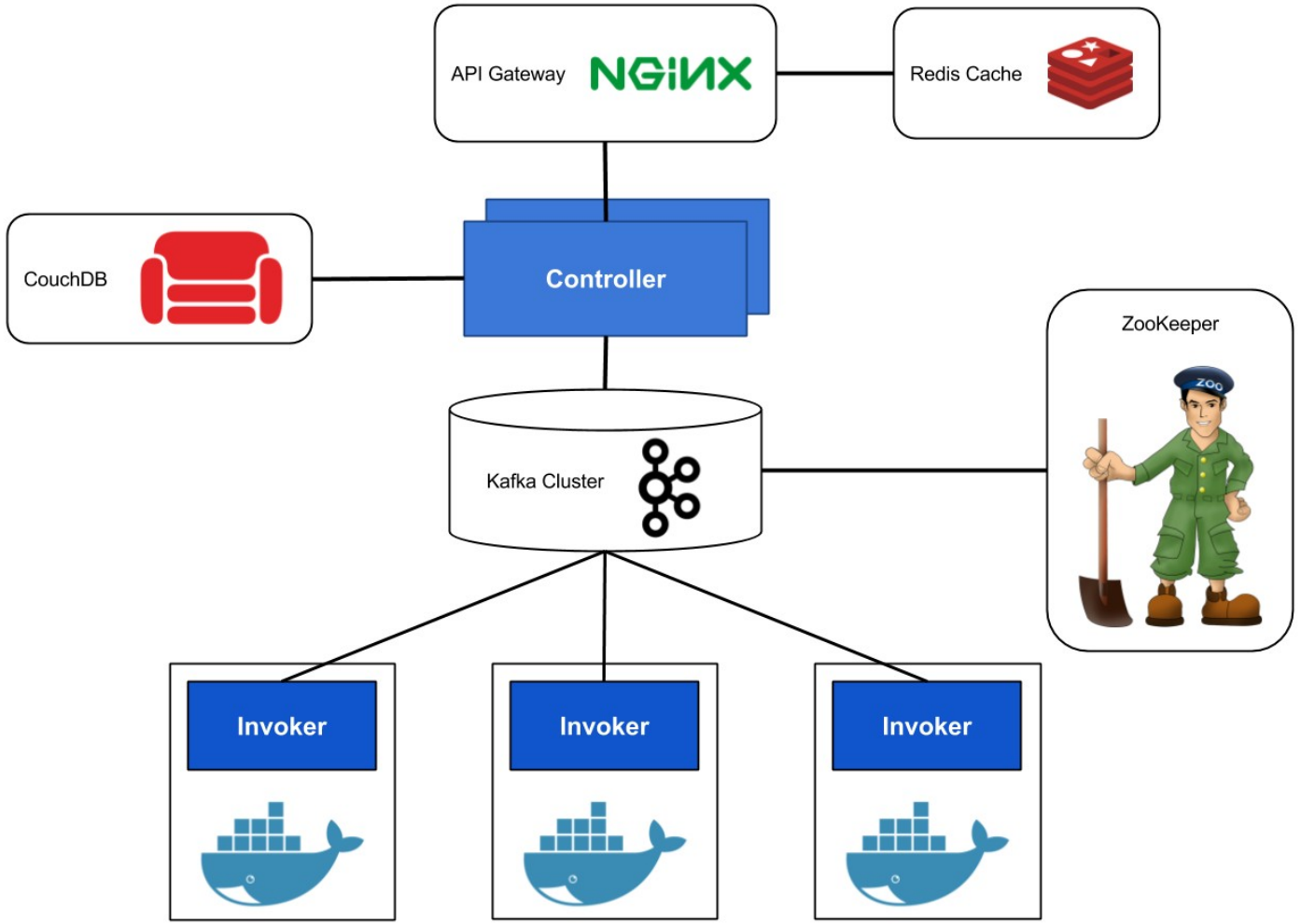
Dockerfile	Required	Hidden	Hidden	Hidden	Hidden	Hidden
Image Repo	Required	Required	Required	None	None	None
Local Docker	Required	Required	Required	None	None	None
Base Image	Required	Required	Required	Required	None	None

Control of Docker environment required
Orchestrate source-to-container builds
Managing traffic for service mesh-grid

Control of Docker environment optional
Add function codes to language runtimes
System manages scaling resources

Source: https://blogs.cisco.com/cloud/examining-the-faas-on-k8s-market & fonk-apps.io

Openwhisk architecture



Source: Figure 6.2 in Oreilly's Docker-for-serverless applications

Redis HA on k8s

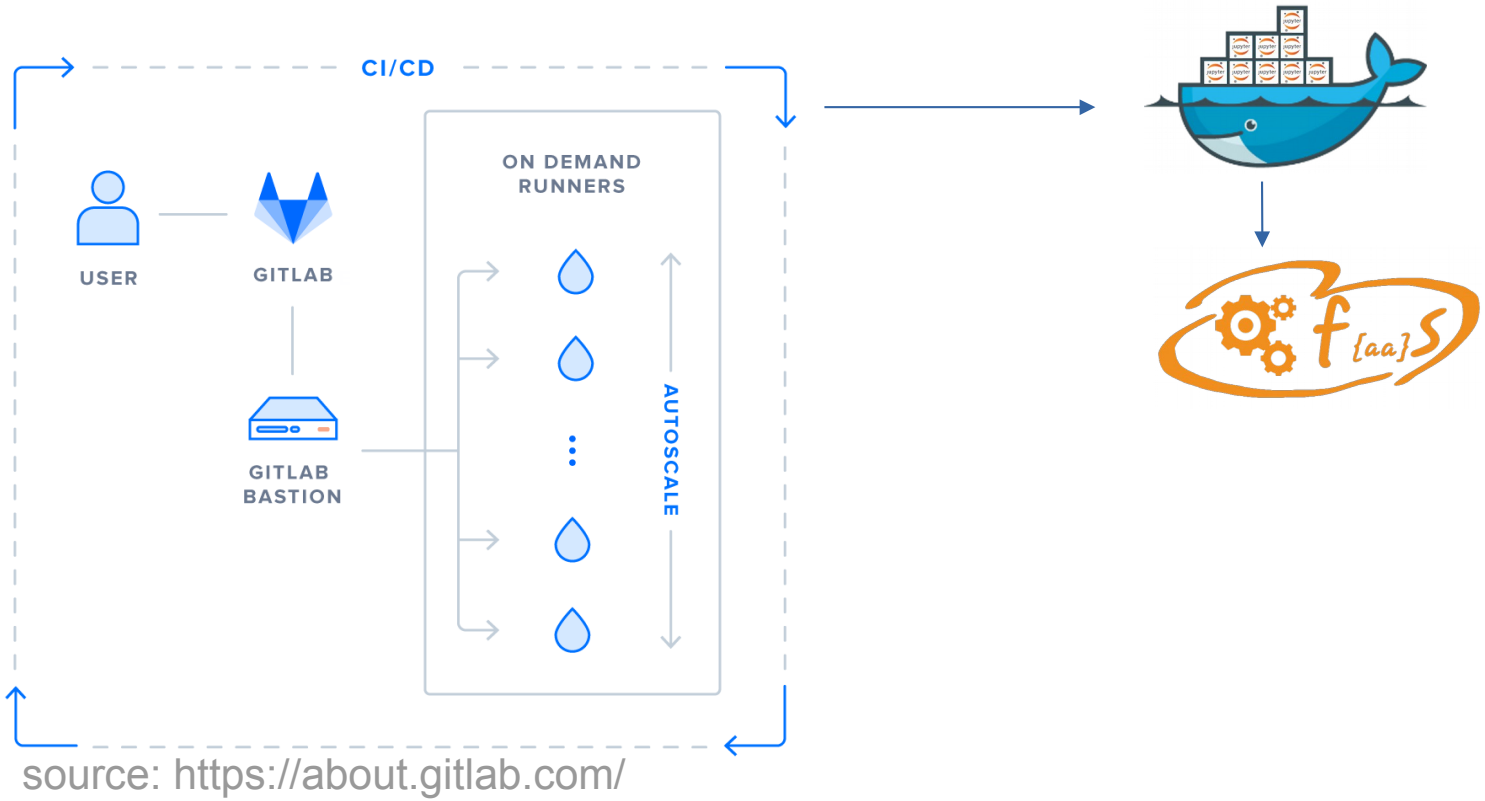
CouchDB HA on k8s

Internal messaging
Kafka/Zookeeper

Invokers start/stop
Docker Containers

Continuous Integration, Delivery & Deployment (CI/CD)

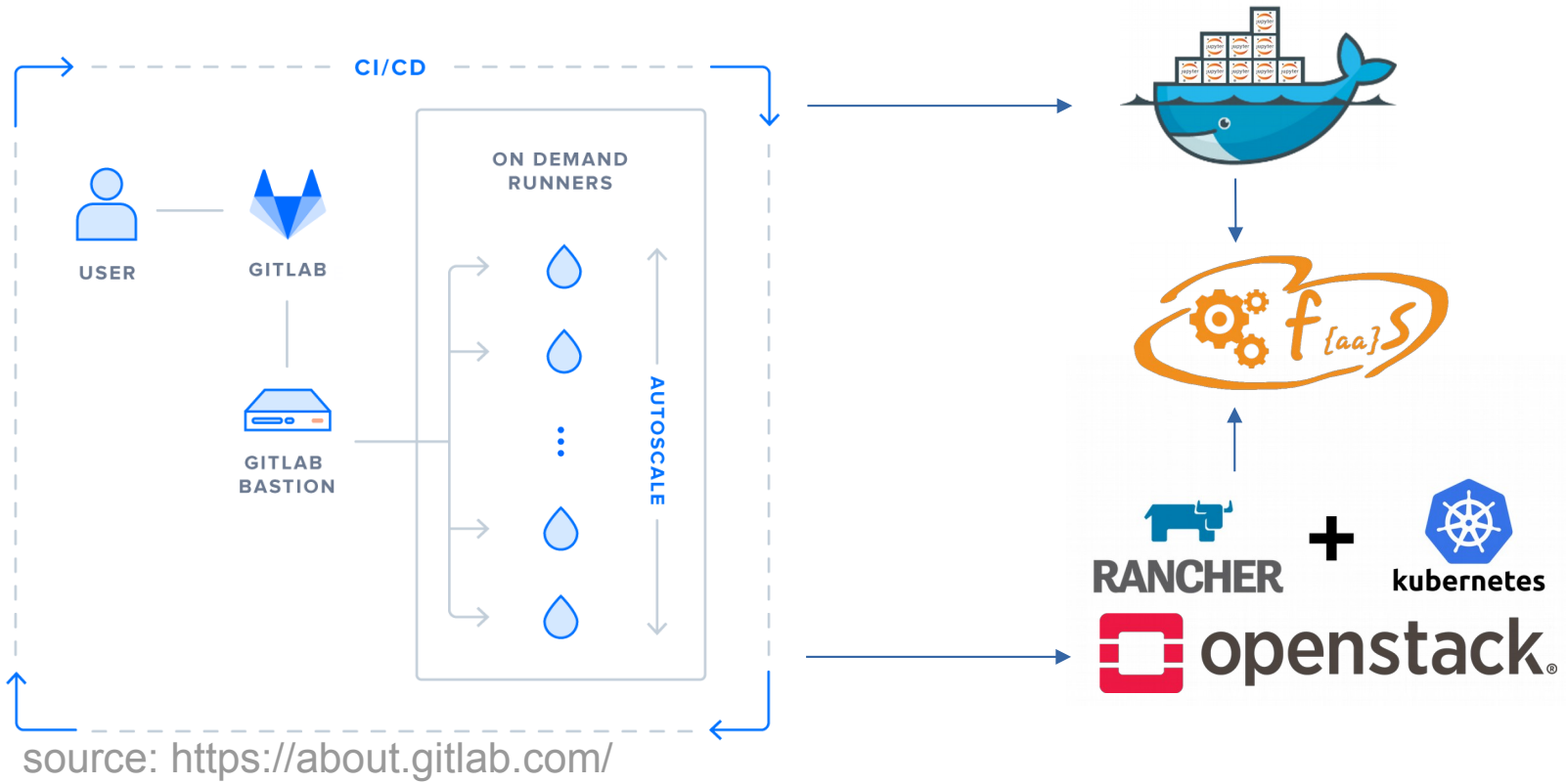
- GitLab as frontend
- Push code, go live
  - Functions
  - Dockerfiles
- Version control
- Auto-scaling CI/CD
- Container registry
- Secret management
  - per user, group, project
  - per CI/CD job



source: https://about.gitlab.com/

Same CI/CD as for platform services

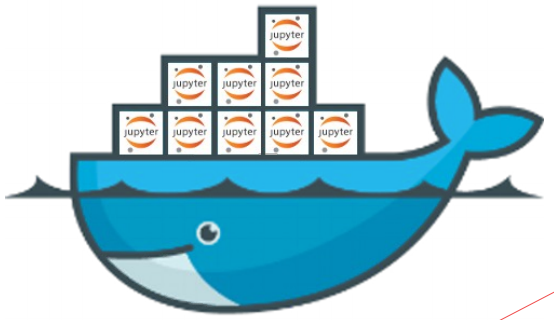
- GitLab as frontend
- Push code, go live
  - Functions
  - Dockerfiles
- Version control
- Auto-scaling CI/CD
- Container registry
- Secret management
  - per user, group, project
  - per CI/CD job



source: https://about.gitlab.com/

Note:
Same system provides the platform as-a-service

Docker Machine Openstack Driver



Docker Machine is now in maintenance mode

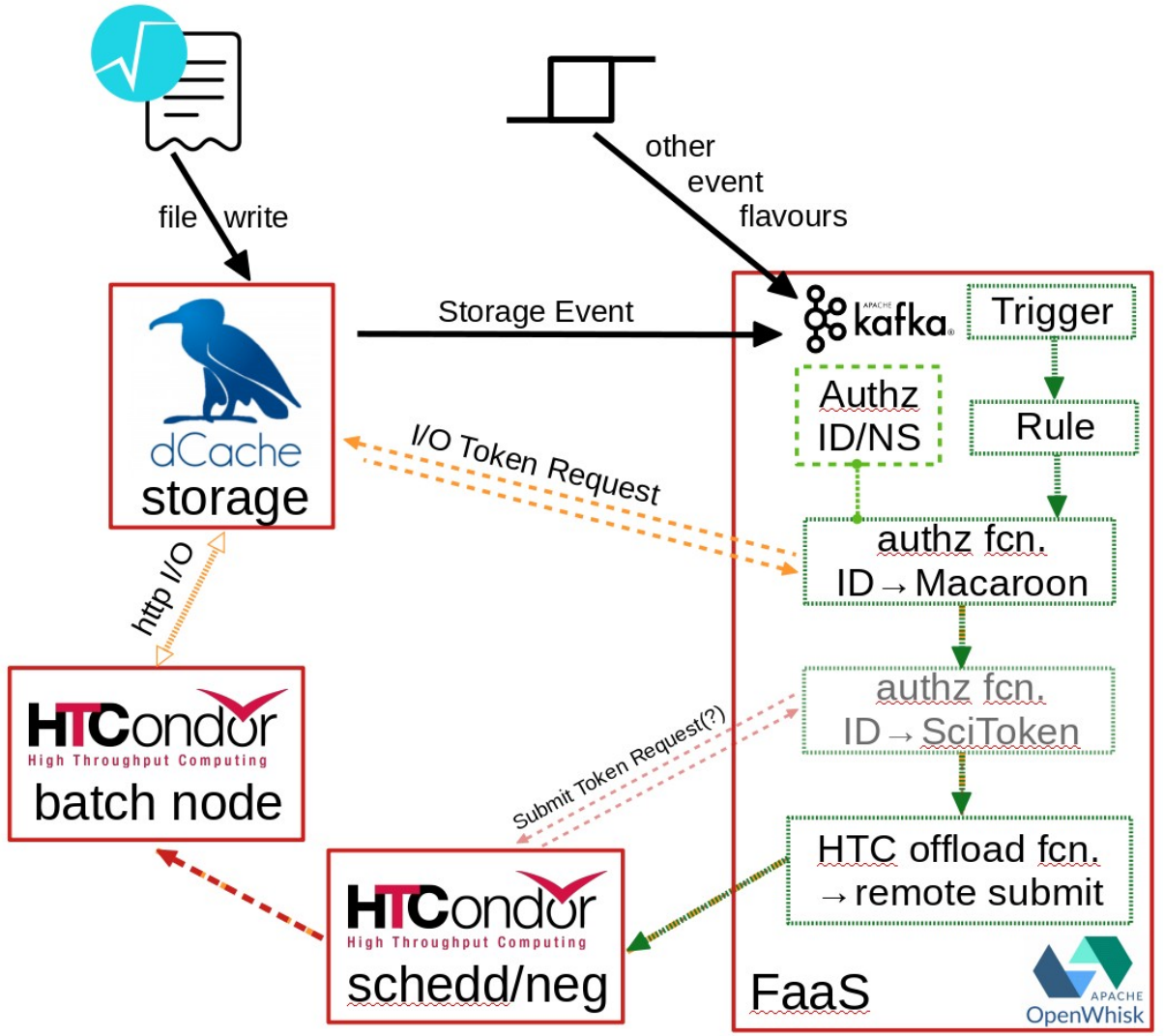
- Openstack driver used in:
  - Gitlab docker+machine runners
  - Rancher
- github.com/docker/machine/issues/4537

- Amazon Web Services
- Microsoft Azure
- DigitalOcean
- Exoscale
- Generic
- Google Compute Engine
- Linode (unofficial plugin, not supported by Docker)
- Microsoft Hyper-V
- OpenStack
- Rackspace
- IBM Softlayer
- Oracle VirtualBox
- VMware vCloud Air
- VMware Fusion
- VMware vSphere
- VMware Workstation (unofficial plugin, not supported by Docker)
- Grid 5000 (unofficial plugin, not supported by Docker)
- Scaleway (unofficial plugin, not supported by Docker)
- Hetzner Cloud (unofficial plugin, not supported by Docker)

Source: https://docs.docker.com/machine/drivers/

Heavy Lifting: Event-driven access to HTC cluster

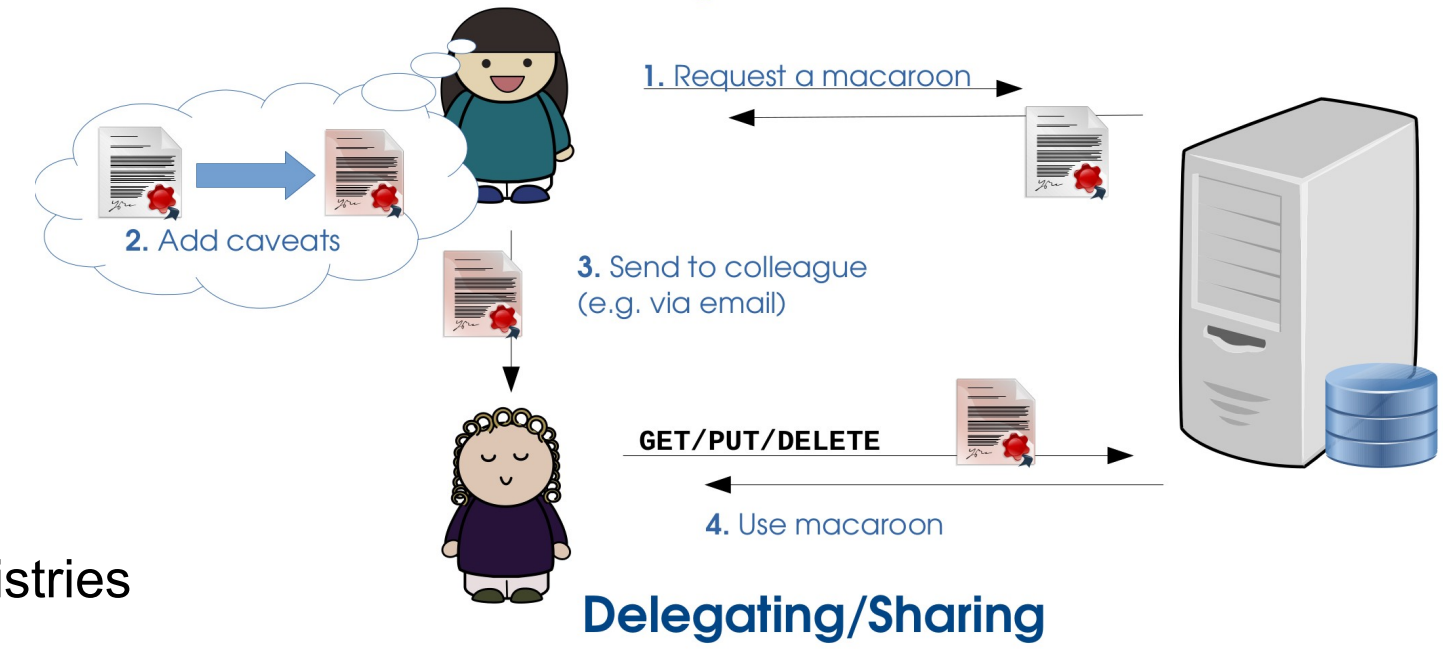
- Automated HTC services on data upload
  - Transfer between sites
  - Restaging from tape
  - Ingest from detector, instrument
- HTC available in FaaS
  - Offloading of compute/IO-intensive jobs
  - FaaS as “glue code”
- FaaS available in HTC
  - Consolidate common tasks as services
  - Function based accounting
  - Usage statistics for users and codes



Source: T.Hartmann (DESY)

Token-based Authentication and Authorization

- Secure multi-tenancy
- Access delegation
- Private/public events, data, functions
- Tokens for
  - dCache to read/write data (Macaroons, SciTokens, OIDC)
  - HTCondor access (SciTokens)
  - FaaS namespace access (OIDC, custom…)
  - GitLab repositories, container registries (OIDC, custom)
  - ...
- In discussion (WLCGAuthorizationWG): WLCG-JWT



As-a-Service Different from as-Infrastructure

- Shift from user based brokering to matching the current demand by a service
- Shift to supporting federated identity and new authorization standards
- Auto-scaling FaaS and batch systems
- Scaling out to transient resources (multi-cloud, EOSC, ...)
- Scale containers, also down to zero (or one to reduce latency)
- Re-use accessible, interoperable microservices rather than re-produce
- Consistent version control and CI/CD for user container builds and runs

Thank you

Contact

DESY. Deutsches Elektronen-Synchrotron

www.desy.de

Michael Schuh
Scientific Computing
michael.schuh@desy.de
+49 040 8998 2316