
CERN

CERN DB On Demand

A DBaaS platform

ignacio.coterillo@cern.ch

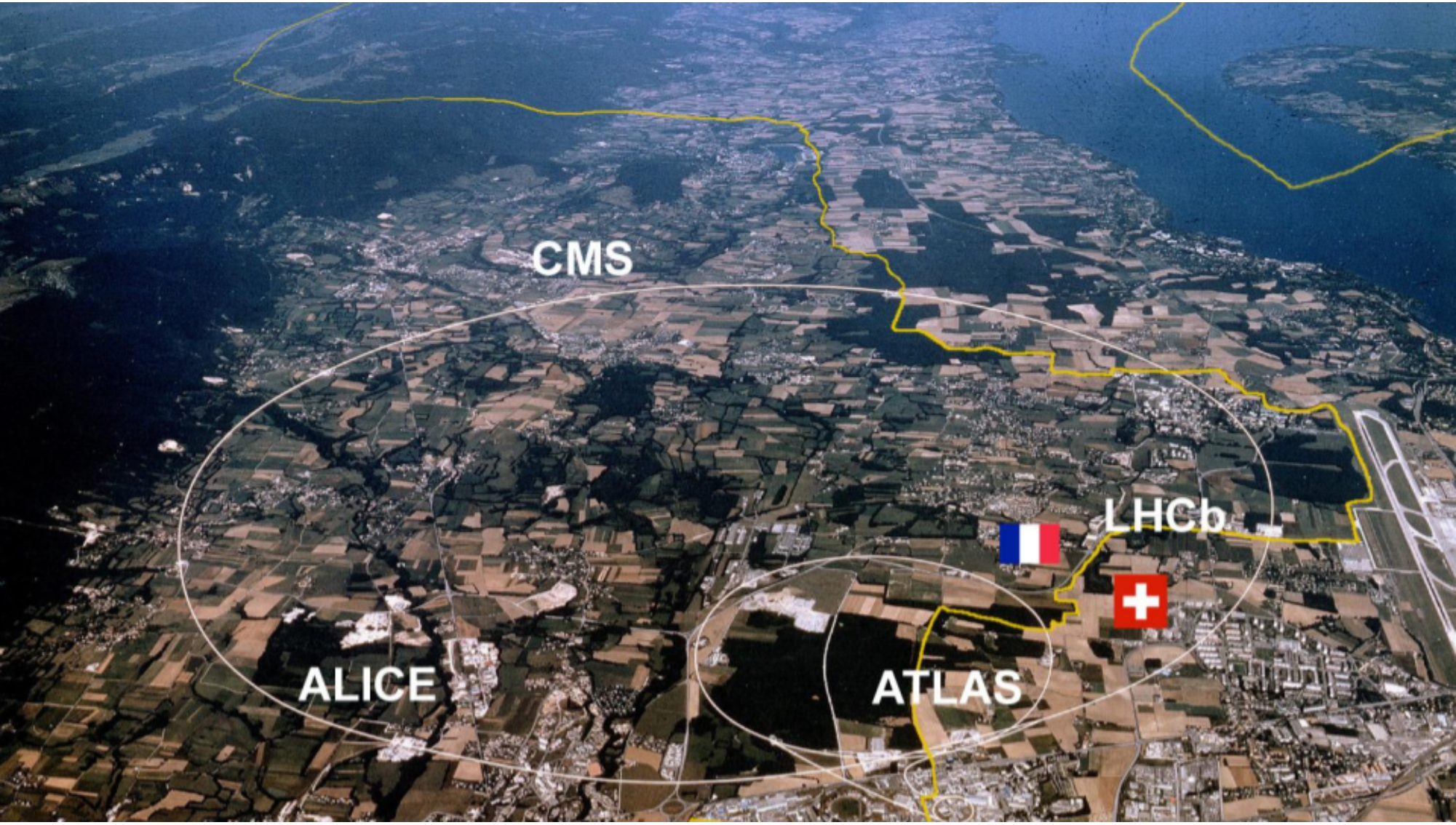
7 Nov 2017

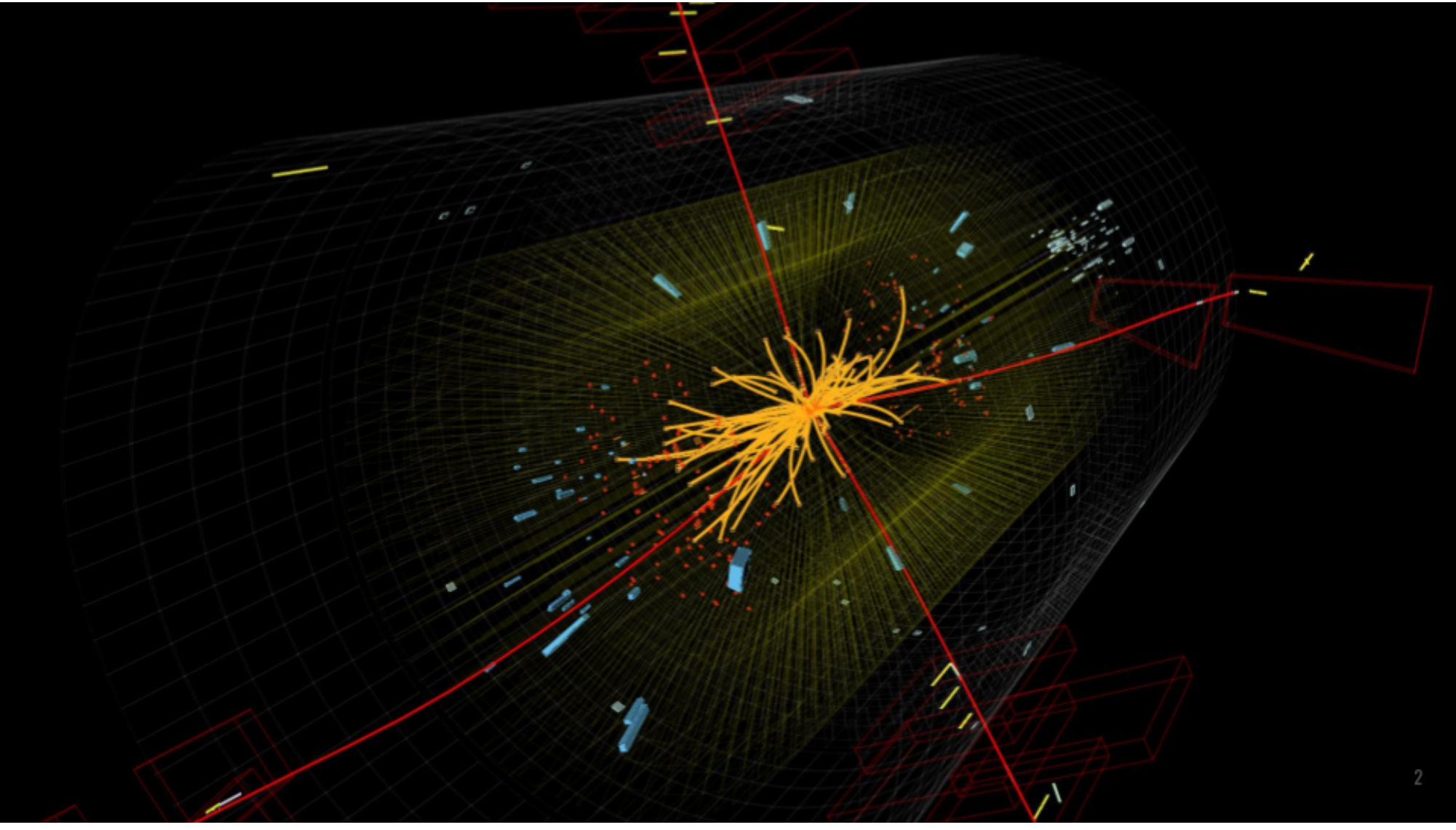
Table of Contents

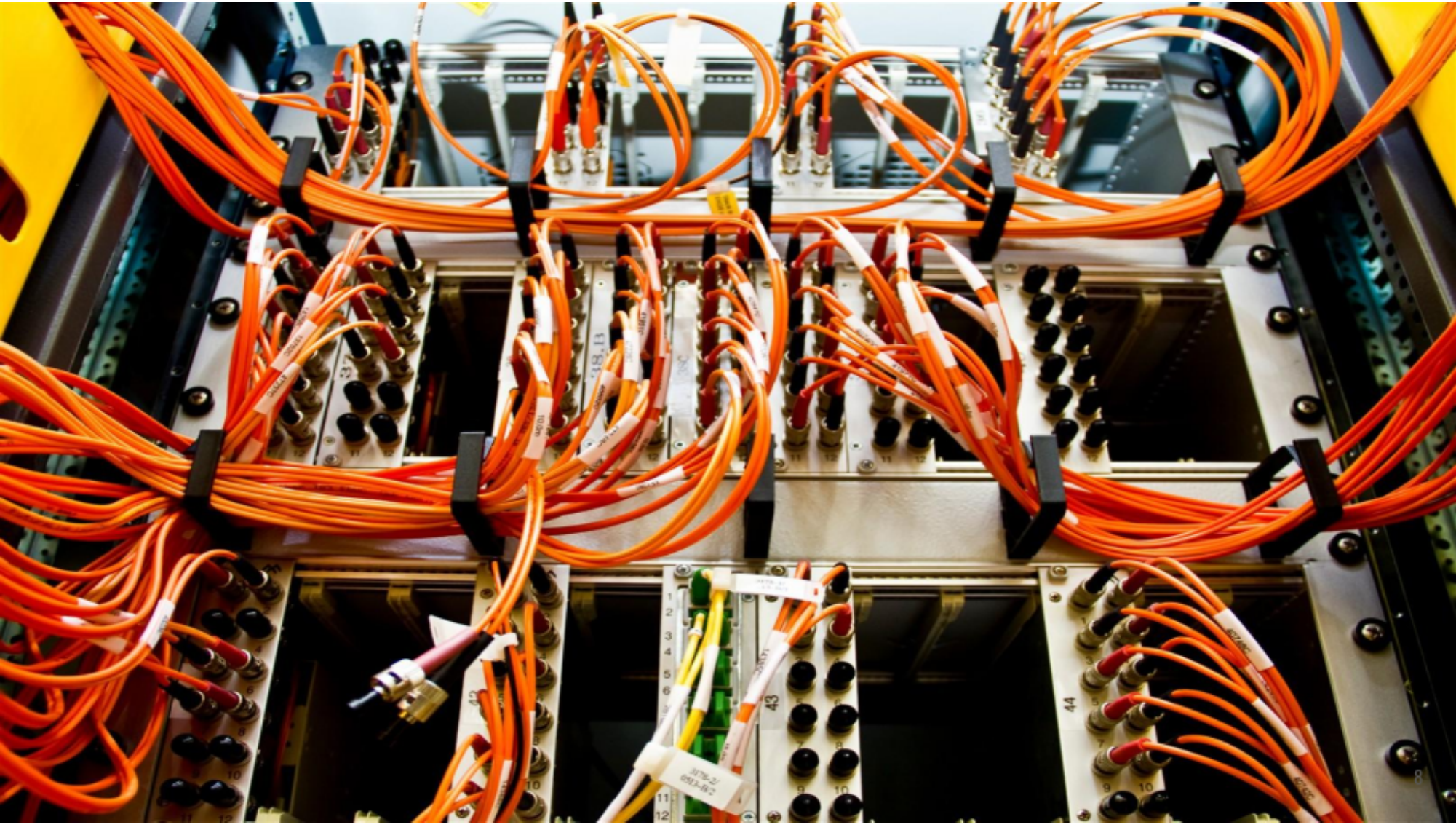
Table of Contents


On the French-Swiss border
22 member states
Collaborators from all around the world
What do we do?


Large Hadron Collider (LHC)


CMS
LHCb
ALICE
ATLAS






StorageTek

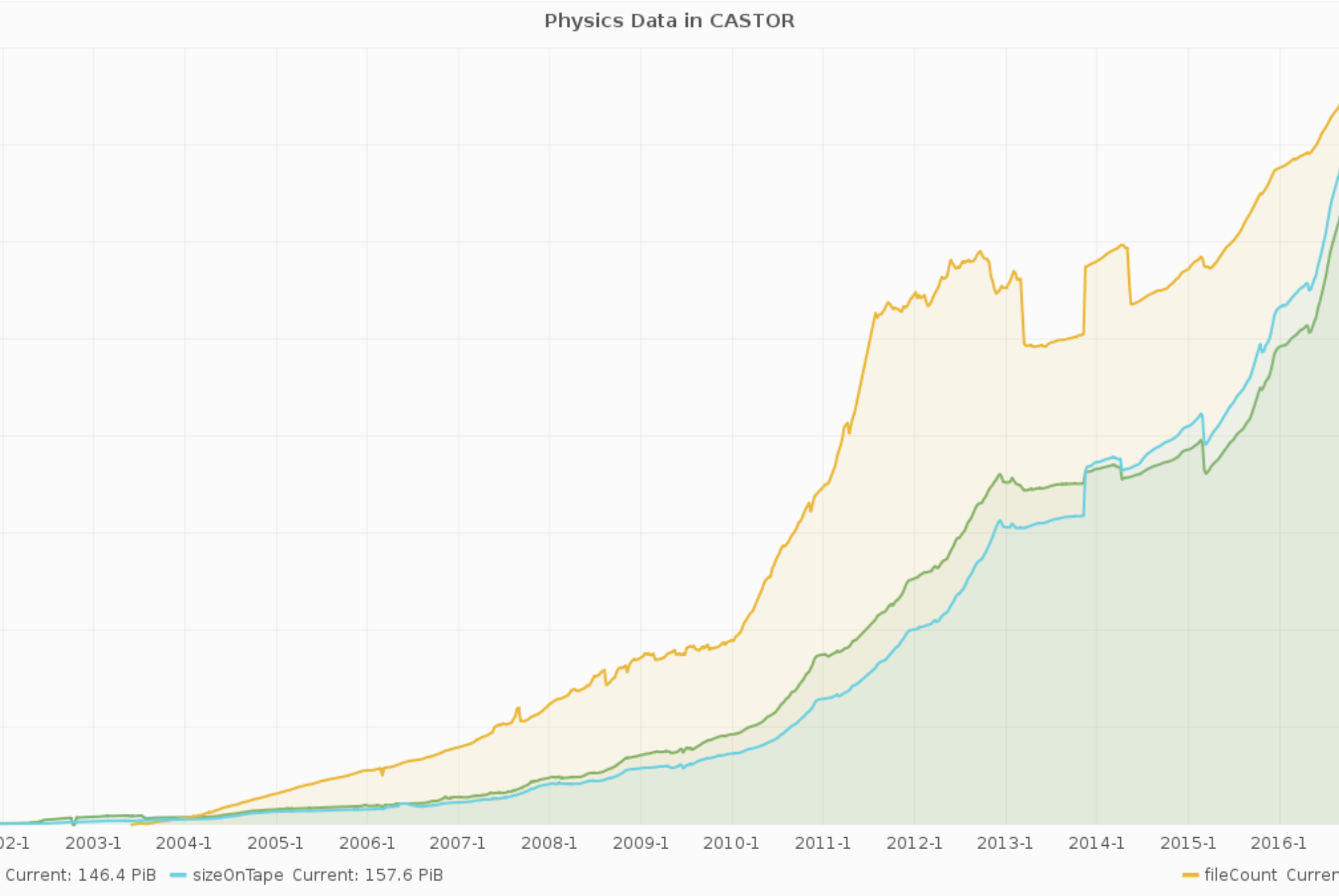

Physics Data in CASTOR
02-1
2003-1
2004-1
2005-1
2006-1
2007-1
2008-1
2009-1
2010-1
2011-1
2012-1
2013-1
2014-1
2015-1
2016-1
Current: 146.4 PiB
sizeOnTape Current: 157.6 PiB
fileCount Curren

Table of Contents

Databases Group

- Part of the IT Department
- Around $\sim$ 30 people
- Manages database services for the whole CERN community
  - Experiments (CMS, Atlas, LHCb, Alice, ...)
  - Accelerator databases
  - Access control, Human Resources, Purchases, etc
  - IT tools (Puppet, OpenStack, ...)
- Manage also application servers

Oracle Databases

Oracle at CERN: Key facts

- $\sim$ 70 databases (most of them RAC) (plus $\sim$ 30 replicas)
- Running versions 11g (11.2.x) and 12c (12.1.x)
- Over 700 TBi of data files for *production* databases

Database on Demand

Key facts

- 9 Oracle special cases (being decommisioned)
- 71 InfluxDB (time Series)
- 123 PostgreSQL
- 386 MySQL instances

Other "databases"

Hadoop

- 4 clusters (3 prod and 1 QA)

ElasticSearch

- $\sim$ 25 Clusters[1]

MS SQL Server

- $\sim$ 10s of databases for MS Applications[2]

[1] Managed by the IT Monitoring team
[2] Managed by the application owners

Table of Contents

Service Timeline

- **2011 Q?** IT-DB hosts MySQL databases for the Drupal service
- **2011 Q4**: Started operations as a service intended for managing single instance **MySQL** databases
- **2012 Q2**: Added **Oracle** instances
- **2013 Q2**: Added **HA Cluster** instances
- **2013 Q4**: Added **PostgreSQL** instances
- ... Infrastructure migration (puppetization)
- **2015**: Introduced **replication** support, New platform architecture
- **2016**: New architecture, SSL, InfluxDB

Overview

Key facts

User is DBA

Database owners have *almost full* DBA privileges[3]

Multi DBMS support

MySQL, PostgreSQL, Oracle[4], InfluxDB

Common Backup and Recovery operations

- B&R based on storage level snapshots.
- Point in Time Recovery depending on each DB system capabilities

[3] Not allowed to modify host system critical parameters
[4] Phasing out

Overview

Resource allocation

- High consolidation. Several databases hosted on the same servers/VM's.
- Each application/service gets its own separated database process/cgroup
- Each database uses two NFS volumes (data, binary logs)
- Standard initial configuration.

Overview

Functionality

- Start/Stop
- File management
  - Configuration files
  - Log files
- Backup: manual and automatic backups based on storage snapshots)
- Recovery: To an specific snapshot and PITR
- Upgrade: Enabled only when a certain version is made available
- Database and host monitoring

Table of Contents

Simplified architecture

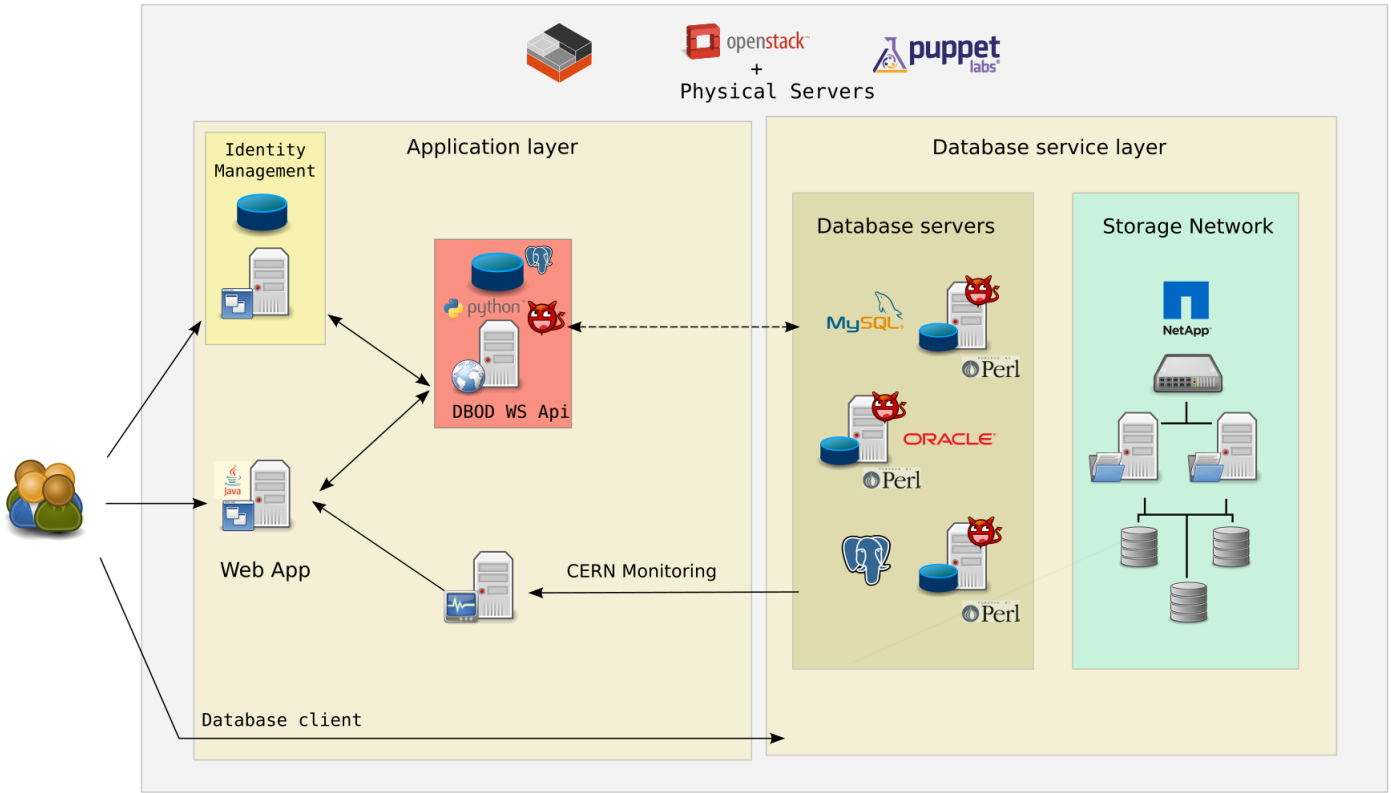
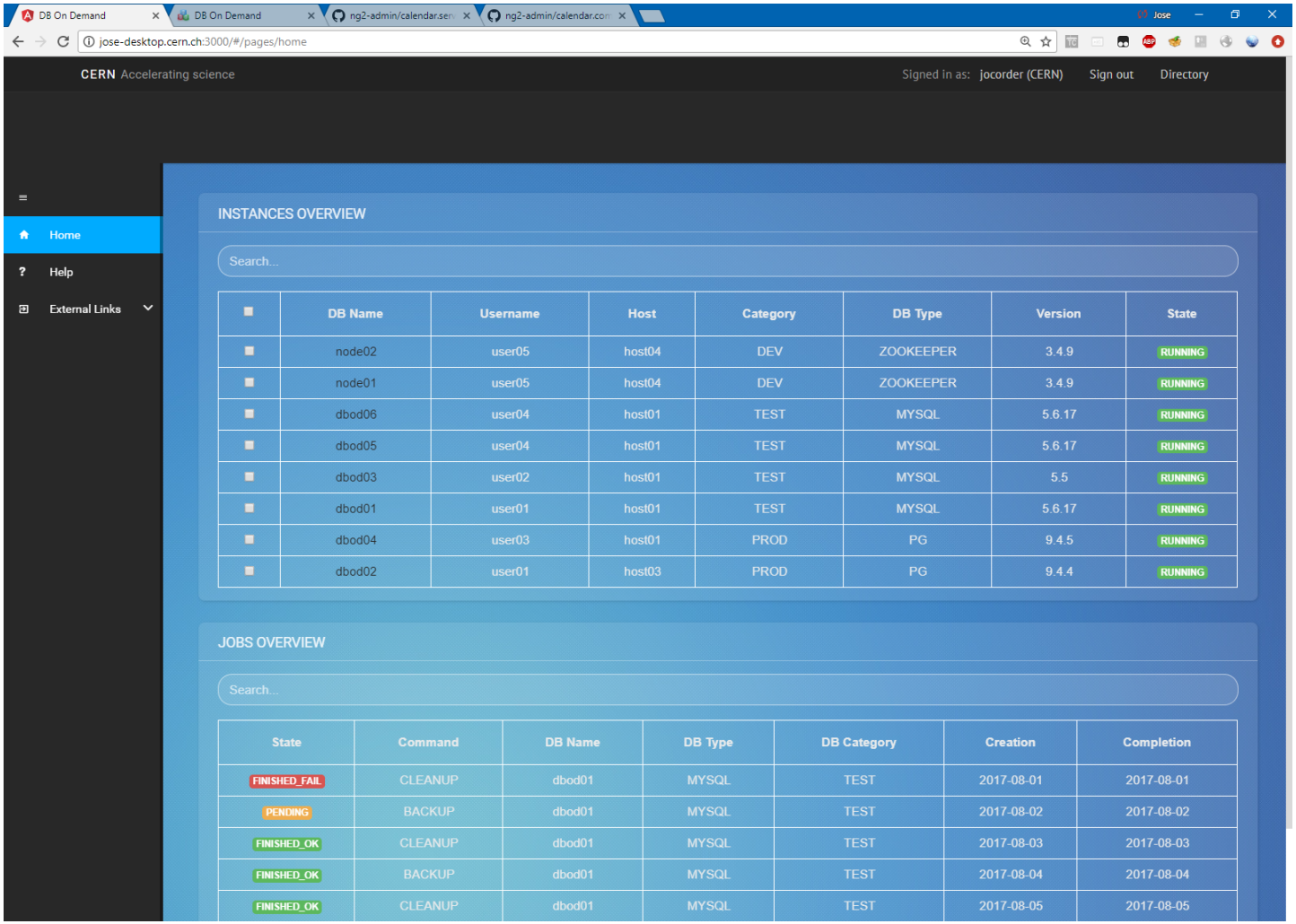
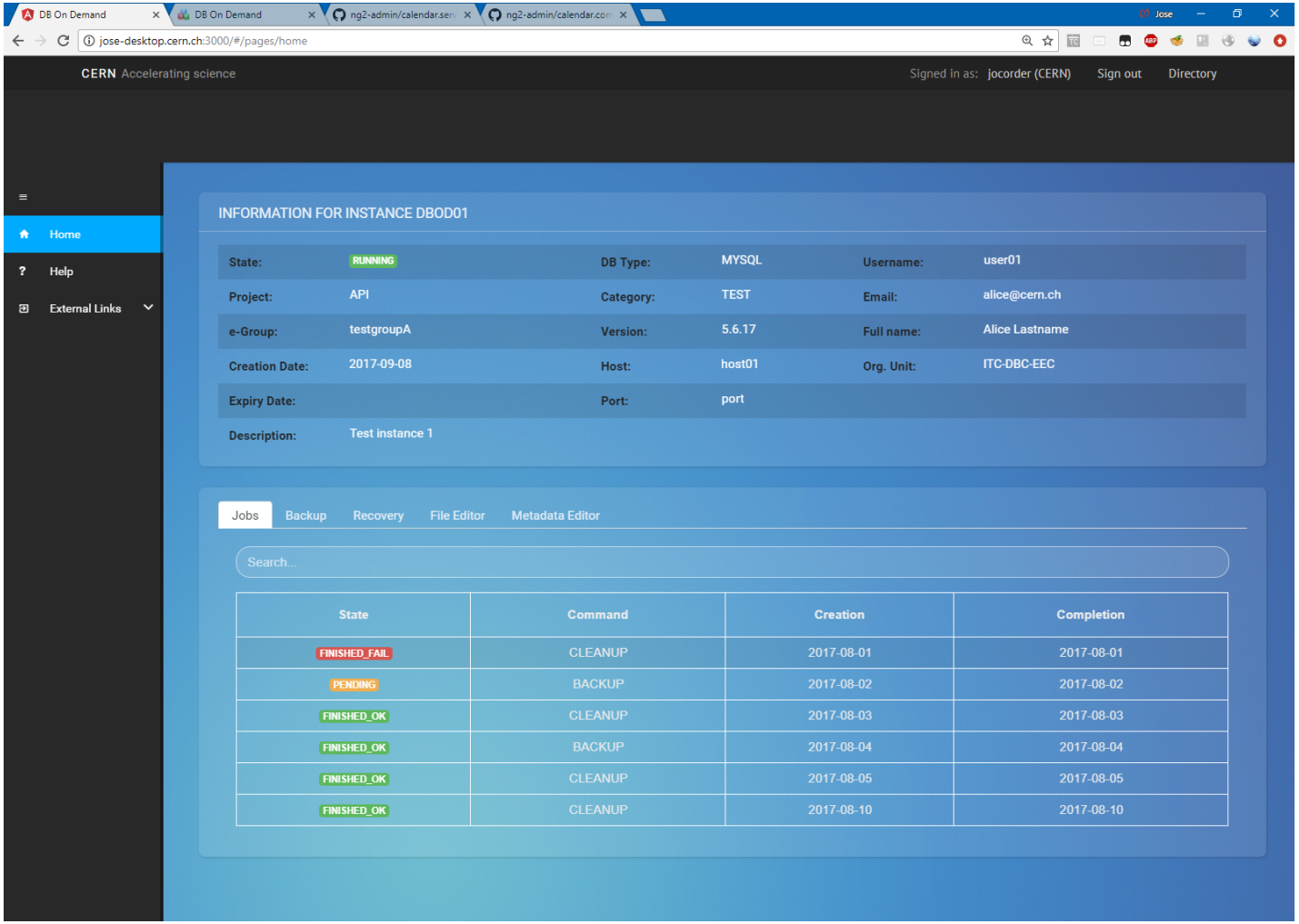


Table of Contents

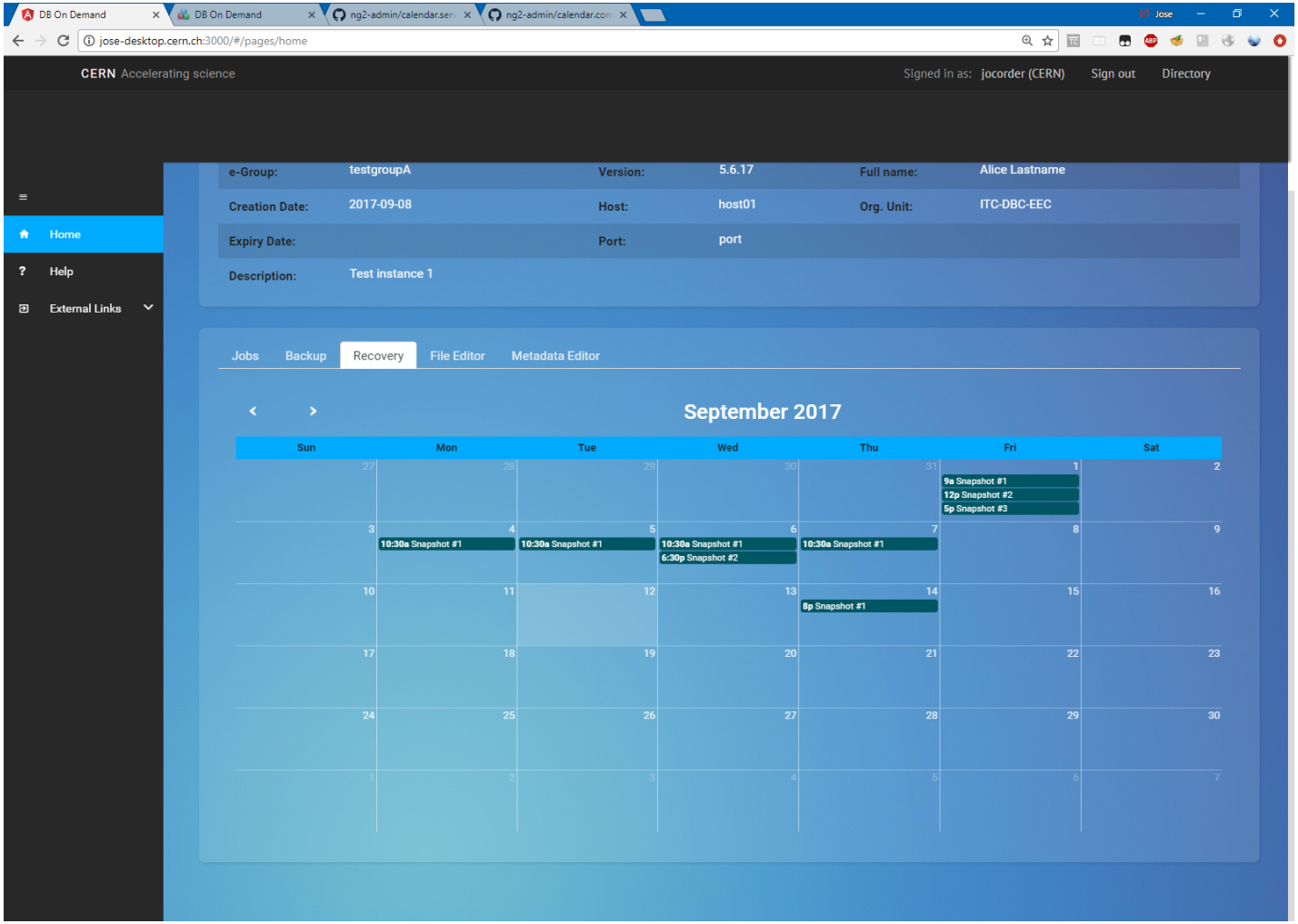
Web Interface



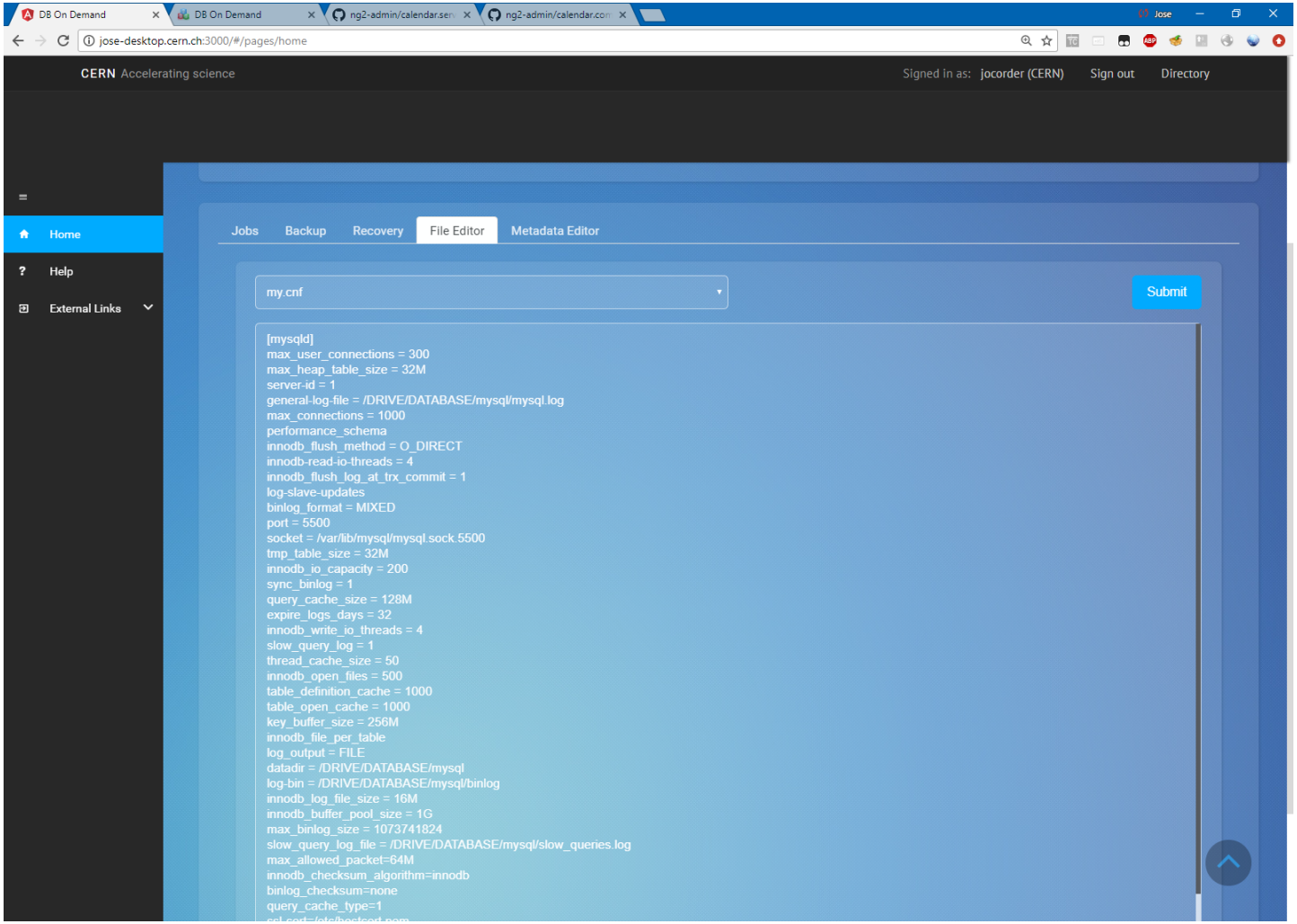
Web Interface



Web Interface



Web Interface



Web Interface

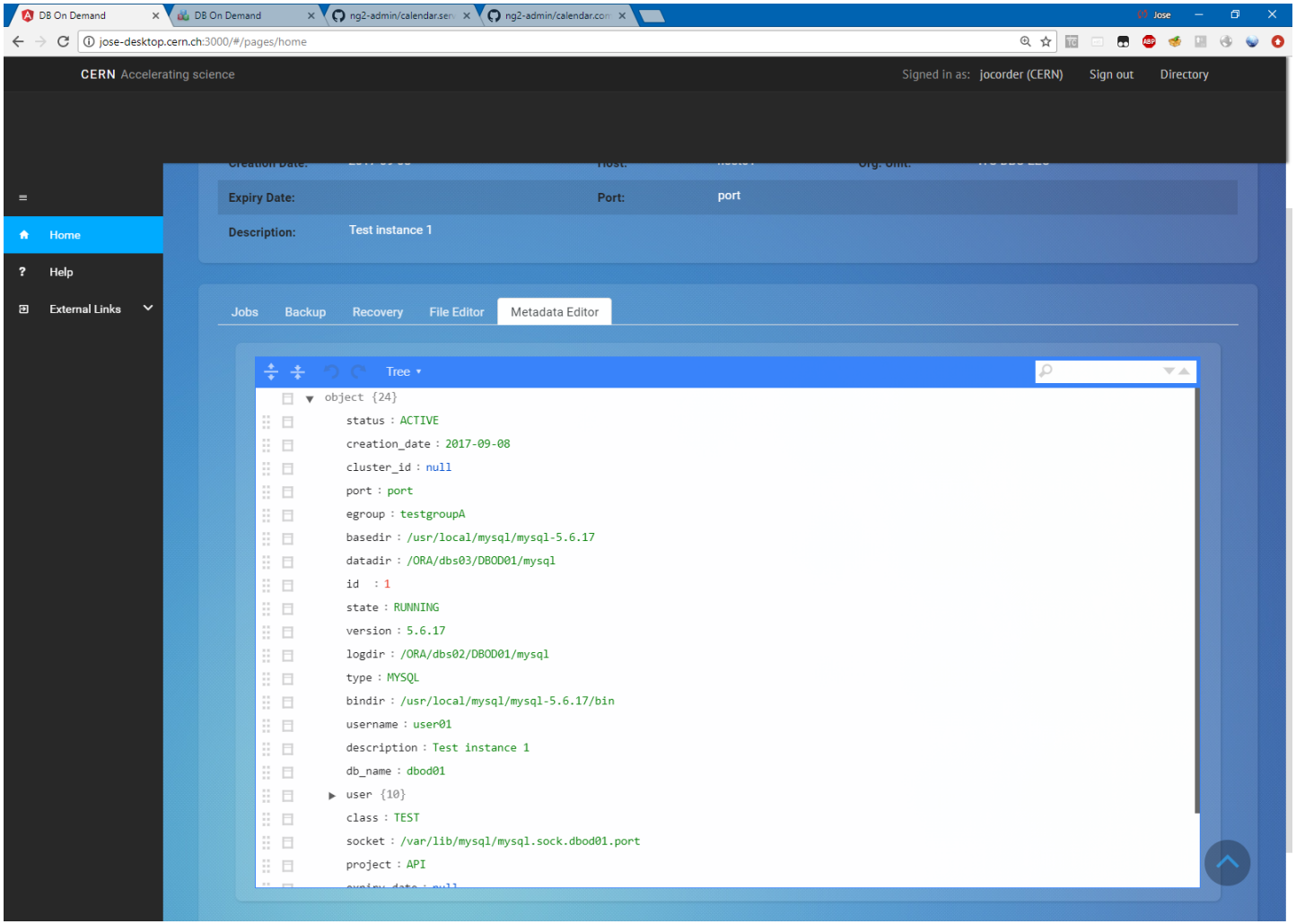


Table of Contents

What is apiato?

apiato

A REST API sitting in the middle of the DB On Demand and Nile platforms[5] working as both a single entry point for service data, and as a gateway to support services such as Rundeck, FIM, *OpenStack*, or the *Storage API*)

HTTP spoken

External APIs integration and the system database all accessed through HTTP(s)

[5] A similar platform ofering Streaming solutions on demand (Apache Kafka)

API server

Facts

- Based on Tornado[6] a Python web framework and asynchronous networking library
- Hosted on `GitHub`[7]. Open sourced under the **GPLv3** License
- Interaction with external API's, authentication layer, ...
- 92% of code covered by tests
- Available for installation via **pip (Python Package Index)**

[6] `https://tornadoweb.org`
[7] `https://github.com/cerndb/dbod-api`

What is apiato?

Elements

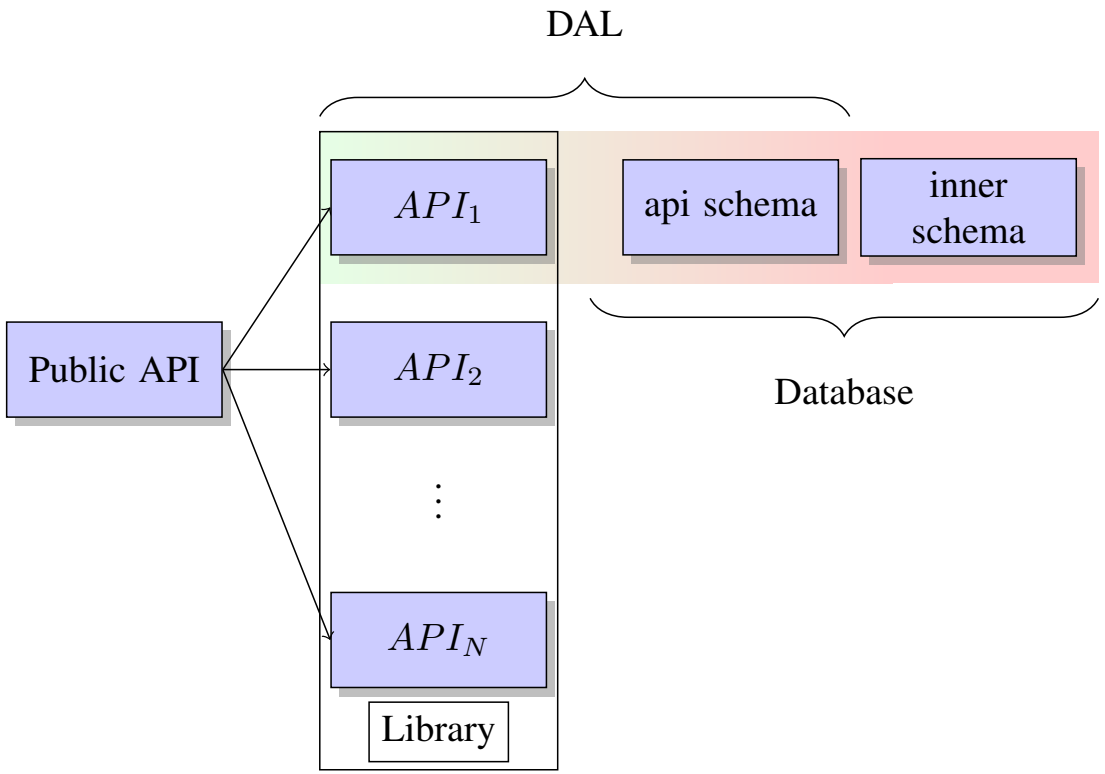
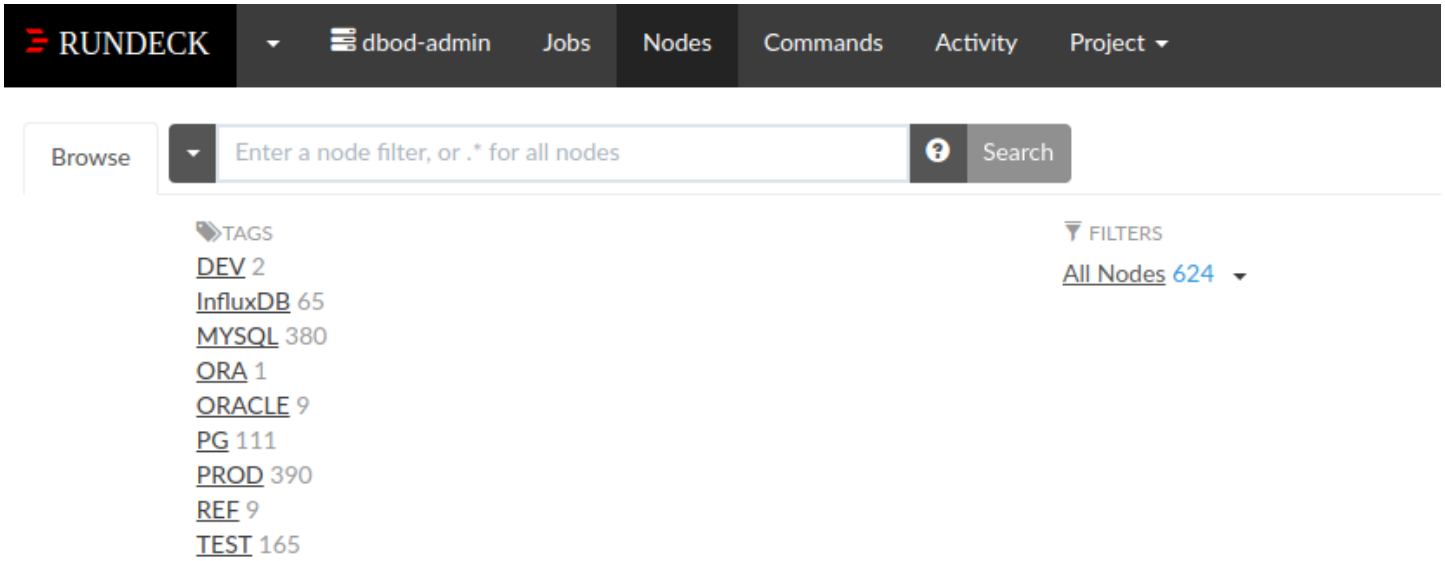
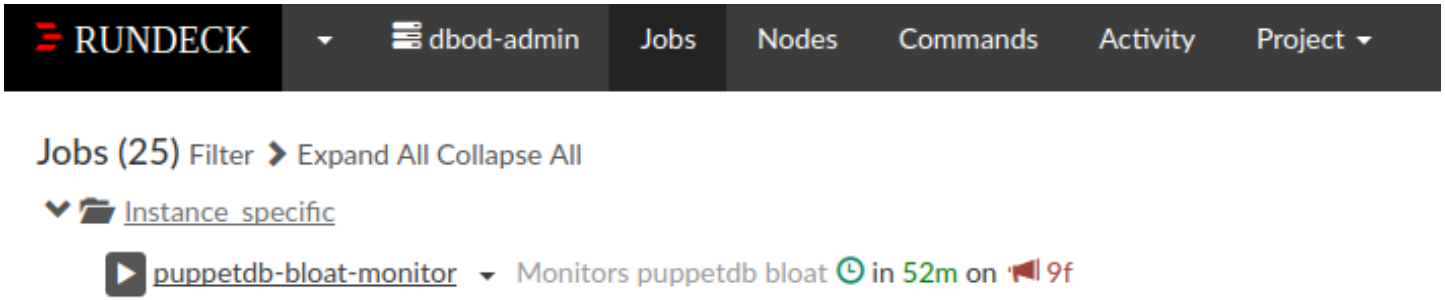


Table of Contents

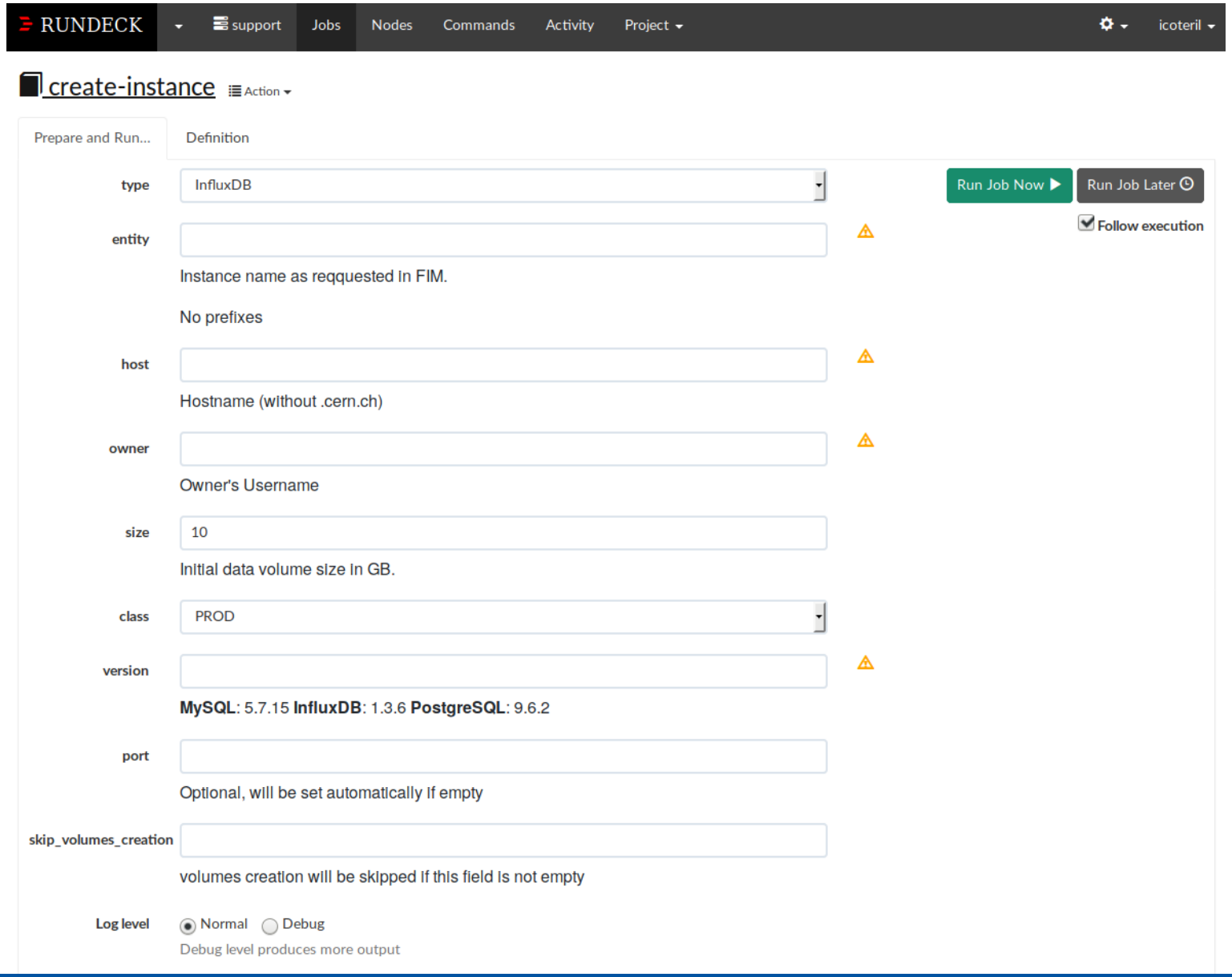
Rundeck, an Automation/Orchestration tool



Rundeck


RUNDECK
dbod-admin
Jobs
Nodes
Commands
Activity
Project
Jobs (25) Filter Expand All Collapse All
Instance_specific
puppetdb-bloat-monitor
Monitors puppetdb bloat
in 52m on
9f

Rundeck


RUNDECK
support
Jobs
Nodes
Commands
Activity
Project
icoteril
create-instance
Action
Prepare and Run...
Definition
type
InfluxDB
Run Job Now
Run Job Later
Follow execution
entity
Instance name as reqquested in FIM.
No prefixes
host
Hostname (without .cern.ch)
owner
Owner's Username
size
10
Initial data volume size in GB.
class
PROD
version
MySQL: 5.7.15 InfluxDB: 1.3.6 PostgreSQL: 9.6.2
port
Optional, will be set automatically if empty
skip_volumes_creation
volumes creation will be skipped if this field is not empty
Log level
Normal
Debug
Debug level produces more output

Rundeck

Apiato

- Resource file generation
- Gateway to Rundeck API for Job execution and scheduling

e.g: Rundeck resources.xml

```
<project>
<node name="ac3a_db" description="" hostname="dbod-ac3a-db" username="dbod" typ
subcategory="PG" port="6605" tags="PG,PROD"/>
<node name="acc_test" description="" hostname="dbod-acc-test.cern.ch" username=
type="TEST" subcategory="MYSQL" port="5506" tags="MYSQL,TEST"/>
```

Table of Contents

Puppet Module

Custom fact

```
Facter.add(:dbod_instances) do
setcode do
begin
# API url and base endpint to host metadata
base_url = 'https://dbod-api.cern.ch:5443/api/v1/host'
uri = URI([base_url,
Socket.gethostname.split('.')[0],
'metadata', ].join('/'))
conn = Net::HTTP.new(uri.host, uri.port)
conn.use_ssl = true
conn.ca_file = '/etc/ssl/certs/CERN-bundle.pem'
conn.read_timeout = 5
conn.open_timeout = 5
conn.verify_mode = OpenSSL::SSL::VERIFY_PEER
response = conn.start { |http|
request = Net::HTTP::Get.new(uri.path)
http.request request
}
case response
when Net::HTTPNotFound
# No instances defined for host
{}
else
format( (JSON.parse response.body)['response'] )
end
rescue SocketError, Timeout::Error, Errno::ECONNREFUSED
# Return cached result
if File.exist? '/etc/dbod/cache/entities.json'
format(
JSON.parse File.read('/etc/dbod/cache/entities.json')
)
end
end
end
end
```

Puppet Module

In hostgroup manifest

```
# Operate on dbod_instances fact
if (is_hash($::dbod_instances)) {
$defaults = {
require => [
Class['::dbod::certificates'],
Class['::dbod::users'],
]
}
create_resources(dbod::instance, $::dbod_instances,
}
```

Monitoring

Metrics

- Database metrics are collected using **Telegraf** and sent to an **InfluxDB**[8] instance
- Visualization dashboards are implemented with **Grafana**[9], integrating database, system and storage system metrics

Server Logs and Slow Query logs

- Both server logs and slow query logs are sent to an **ElasticSearch**[10] Cluster
- Visualization is being integrated in the platform Web Interface

[8] https://www.influxdata.com/

[9] https://grafana.com/

[10] https://www.elastic.co/


InfluxData | Docum
x/talks/present: disp
Grafana - MySQL Mo
Secure | https://dbod-mon.web.cern.ch/dashboard/db/mysql-monitoring?refresh=10s&orgId=1&var-server=itrac5146.cern.ch&var-instance=magnum
Database Process Metrics
Process CPU Time
User
System
IO Wait
Process Threads
Threads
Process MEM
RSS
VMS
SWAP
Connections
Aborted_clients
Aborted_connects
Connections
Activity Type
Select
Update
Insert
Delete
Commit
Rollback
Transactions per second
Commit
Rollback
Innodb Buffer Pool Usage
Used
Dirty
Size
InnoDB Buffer Pool Usage (Pages)
Total
Free
Misc
Dirty
Data
Temporary Table Activity
Tables
Files
Disk Tables
Thread Cache
Running
Cached
Connected
Created
Query Cache Usage
Cache Size
Free Memory
Not cached
Inserts
Hits
NFS: Data Volume
NFS: WAL Volume
WAL Log Volume Throughput
WAL Volume IOPs
max
avg
2017-09-12 14:19:00
Average:
olume Latencies

kibana

130,724 hits

New Save Open Share Last 15 minutes

Search...

- Discover
- Visualize
- Dashboard
- Timelion
- Indices
- Own Home
- Management

itdb_dbod-inflog-*

Selected Fields

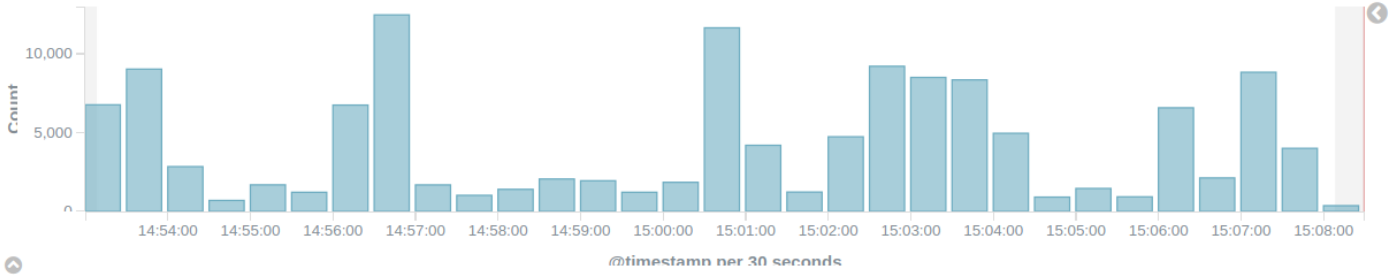
? _source

Available Fields

Popular

- @timestamp
- t httpd_timestamp
- msg_timestamp
- t tags
- t @version
- t _id
- t _index
- # _score
- t _type
- t beat.hostname
- t beat.name
- ? beat.version
- t host
- t httpd_ip
- t httpd_req_method
- t influx_endpoint
- t influx_msg
- t influx_process
- t influx_response
- t influx_user
- t input_type
- t log_source
- t logstash_source
- t message
- # offset
- t source
- t status_code
- t type
- ? user_agent
- ? website

September 12th 2017, 14:53:08.356 - September 12th 2017, 15:08:08.356 — Auto



Time	_source
September 12th 2017, 15:08:06.000	**offset:** 37,768,755 **logstash_source:** dbod-logstash-2.cern.ch **input_type:** log **source:** /ORA/dbs03/HLTMON/influxd.log **message:** [I] 2017-09-12T13:08:06Z SELECT uptime / 3600 FROM plancton_hltdev.autogen.container WHERE started = 'True' AND time > now() - 12h AND uptime > 120 service=query **type:** inflog **tags:** beats_input_codec_plain_applied, matched **@timestamp:** September 12th 2017, 15:08:06.000 **log_source:** DBOD DB LOG MESSAGE **influx_process:** I **@version:** 1
September 12th 2017, 15:08:06.000	**website:** http://alihltcloud2.cern.ch:3000/dashboard/db/plancton-monitor?from=now-12h&to=now **status_code:** 200 **offset:** 37,769,284 **logstash_source:** dbod-logstash-2.cern.ch **input_type:** log **httpd_timestamp:** 12/Sep/2017:15:08:06 **source:** /ORA/dbs03/HLTMON/influxd.log **message:** [httpd] 128.141.224.178,137.138.116.75 - plancton_ro [12/Sep/2017:15:08:06 +0200] "GET /query?db=plancton hltdev&epoch=ms&q=SELECT+%22uptime%22%2F3600+FROM+%22container%22+WHE
September 12th 2017, 15:08:06.000	**offset:** 37,769,646 **logstash_source:** dbod-logstash-2.cern.ch **input_type:** log **source:** /ORA/dbs03/HLTMON/influxd.log **message:** [I] 2017-09-12T13:08:06Z SELECT mean(uptime) FROM plancton_hltdev.autogen.container WHERE started = 'True' AND time > now() - 12h AND uptime > 120 GROUP BY time(30s) fill(previous) service=query **type:** inflog **tags:** beats_input_codec_plain_applied, matched **@timestamp:** September 12th 2017, 15:08:06.000 **log_source:** DBOD DB LOG MESSAGE
September 12th 2017, 15:08:06.000	**offset:** 37,769,812 **logstash_source:** dbod-logstash-2.cern.ch **input_type:** log **source:** /ORA/dbs03/HLTMON/influxd.log **message:** [I] 2017-09-12T13:08:06Z SELECT sum(last) FROM plancton_hltdev.autogen.sum_containers WHERE time > 1505178306s AND time < 1505221506s GROUP BY time(3m) service=query **type:** inflog **tags:** beats_input_codec_plain_applied, matched **@timestamp:** September 12th 2017, 15:08:06.000 **log_source:** DBOD DB LOG MESSAGE **influx_process:** I
September 12th 2017, 15:08:06.000	**offset:** 37,769,968 **logstash_source:** dbod-logstash-2.cern.ch **input_type:** log **source:** /ORA/dbs03/HLTMON/influxd.log **message:** [I] 2017-09-12T13:08:06Z SELECT mean(cpu_eff) FROM plancton_hltdev.autogen."measurement" WHERE time > now() - 12h GROUP BY time(5m), hostname service=query **type:** inflog **tags:** beats_input_codec_plain_applied, matched **@timestamp:** September 12th 2017, 15:08:06.000 **log_source:** DBOD DB LOG MESSAGE **influx_process:** I **@version:** 1
September 12th 2017, 15:08:06.000	**website:** http://alihltcloud2.cern.ch:3000/dashboard/db/plancton-monitor?from=now-12h&to=now

Table of Contents

The dbod-core framework

What is it?

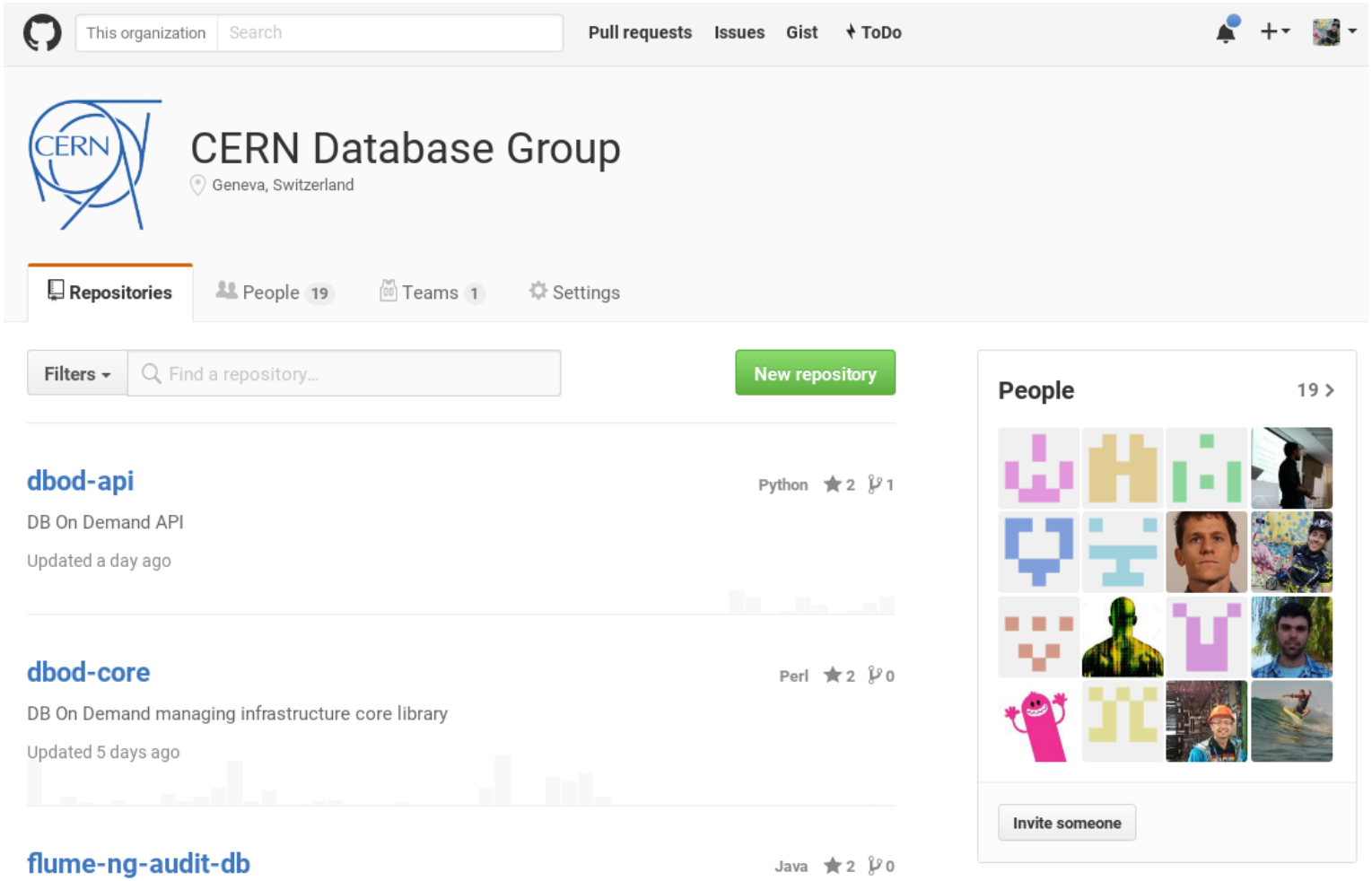
The actual programs executed (typically) on the database hosts performing top level functionality tasks (backups, recoveries, upgrades, etc.)

Why Perl

- Re-using pre-existing code
- NetApp SDK Perl distribution support superior to alternatives[11]

[11] At the time. Soon it will no longer be required

https://github.com/cerndb



Questions?


https://home.cern
https://visit.cern

www.cern.ch

Database SERVICES

At the heart of CERN, LHC and Experiment Operations

CERN IT Department

http://cern.ch/it-dep/db/

IT/DB GROUP

Experiment Offline Databases

Streams

Experiment Online Databases

Data

Streams

Tier-1 Centres

dedicated 10 Gbit/s links

NDGF Nordic Countries
GridKa Karlsruhe, Germany
BNL Brookhaven, NY - USA
RAL Oxfordshire, UK
ASGC Taipei
SARA-NIKHEF Amsterdam, Netherlands
PIC Barcelona, Spain
FNAL Batavia, IL - USA
CCIN2P3 Lyon, France
INFN - CNAF Bologna, Italy
TRIUMF Vancouver, Canada

LHC Experiments

LHCb

CMS

ATLAS

ALICE

RAW Data

CASTOR

CERN Advanced STORage manager

Middleware

Administrative/IT/ Engineering Databases

Accelerators ACC

LHC Operations

EDMS

AIS

PARTNERS:

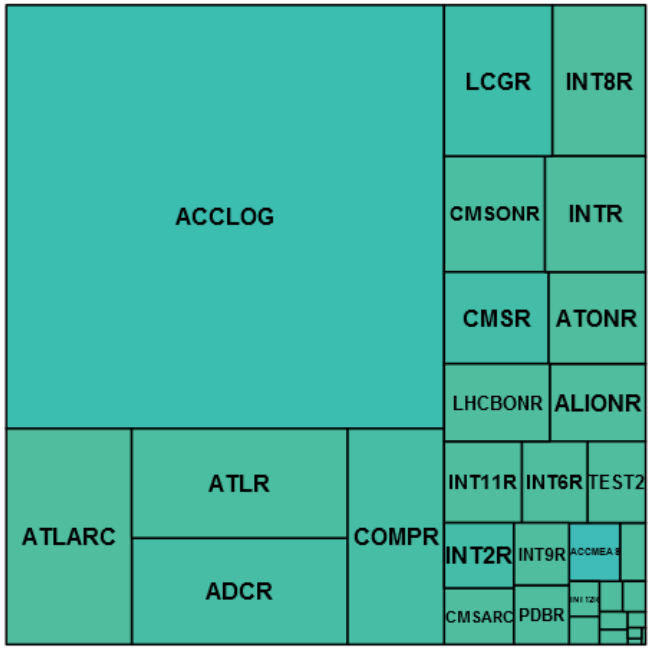
CERN openlab

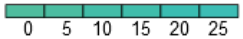
ORACLE

CERN

Current status on June 2013

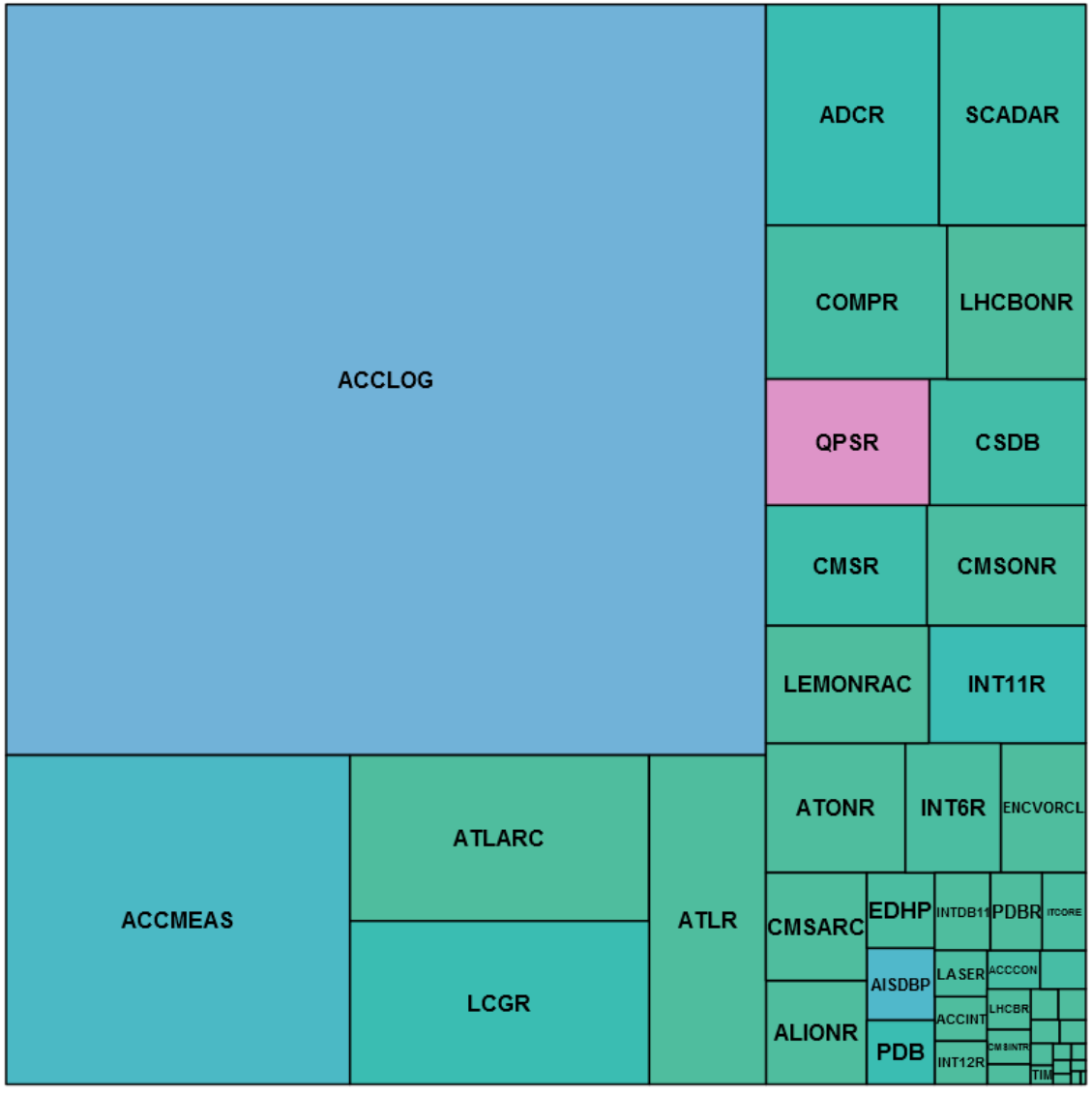
CERN databases October 2012
area is size (total=299TB)

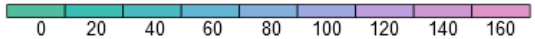




TB redo per month, sum=120 TB

CERN databases May 2016
area is size (total=865TB), color is redo activity





TB redo per month, sum=503 TB

What is apiato?

Basic Concepts

Instance

An individual element, generally linked to a process which run on a certain computing resource (host, vm, container), can be a member of a cluster, and can own resources.

Cluster

A set of instances related in a certain way

Resource

A *computing* element used by one or more instances (Storage volumes, *containers*, *virtual machines*, ...)

API server

Sample URIs

- server:port/api/v1/instance/*:name*
- server:port/api/v1/instance/*:name*/metadata
- server:port/api/v1/host/*:name*
- server:port/api/v1/host/*:name*/metadata
- server:port/api/v1/cluster/*:name*
- server:port/api/v1/cluster/*:name*/metadata
- server:port/api/v1/rundeck/resources.xml
- server:port/api/v1/rundeck/job/*:jobid*/*:instance*

API server

Nile cluster response


```
{
"response": [
{
"username": "certsoc",
"master_name": "zookeeper-sec-test",
"name": "kafka-sec-test",
"master_hosts": [
"nile-zk-sec-test-02.cern.ch",
"nile-zk-sec-test-01.cern.ch",
"nile-zk-sec-test-03.cern.ch"
],
"version": "0.10.2.0",
"class": "DEV",
"project": "SECURITY TEAM Pilot",
"instances": [
{
"username": "certsoc",
"cluster_id": 5,
"version": "kafka-0.10.1.1-scala-2.11",
"class": "DEV",
"port": null,
"db_name": "kafka-sec-test-node-02",
"hosts": [
"nile-kafka-sec-test-02.cern.ch"
],
{
"volumes": [
"mounting_path": "/nbapps/sas/dev_kafka007"
}
],
"attributes": {
"broker_id": "2"
},
"type": "KAFKA",
"id": 19
},
],
"lb_alias": null,
"hosts": [
"nile-kafka-sec-test-02.cern.ch",
"nile-kafka-sec-test-04.cern.ch",
"nile-kafka-sec-test-03.cern.ch",
"nile-kafka-sec-test-01.cern.ch",
"nile-kafka-sec-test-05.cern.ch"
],
"e_group": "it-db-nile-developers",
"attributes": {
"super.users": "User:kafka;User:aromerom;User:ce
"log.retention.hours": "120",
"replication_factor": "3",
"partitions_number": "3",
"zk_chroot": "/kafka",
"port_ssl": "9093"
```

API server

DBOD instance metadata response (Oracle instance)


```
{
"active": true,
"attributes": {
"home": "/ORA/dbs01/oracle/product/rdbms1",
"listener_log": "/ORA/dbs01/oracle/product/rdbms1/network/admin",
"listner_config": "/ORA/dbs01/oracle/product/rdbms1/network/admin",
"listner_trace": "/ORA/dbs01/oracle/product/rdbms1/network/trace",
"log": "/ORA/dbs00/CSTEST1/bdump",
"notifications": "true",
"orainst": "/ORA/dbs01/oracle/product/rdbms1/oraInst.loc",
"port": "10121",
"rman_command": "/ORA/dbs01/syscontrol/projects/rman/cmd/rman_templates",
"tnsnames": "/ORA/dbs01/oracle/product/rdbms1/network/admin",
"tsm_server": "TSM615_ORA",
"version": "oracle-rdbms-1-11.2.0.4.0.cern5-1.x86_64"
},
"basedir": "/ORA/dbs01/oracle/product/rdbms",
"bindir": "/ORA/dbs01/oracle/product/rdbms",
"class": "TEST",
"datadir": "/ORA/dbs03/CSTEST1",
"db_name": "cstest1",
"db_type": "ORACLE",
"hosts": [
"itrac5175"
],
"id": 160,
"logdir": "/ORA/dbs02/CSTEST1",
"port": "10121",
```

Data abstraction layer

External *(api)* database schema

- One[12] view per external API endpoint
- One update procedure per view

```
postgres@dbod-dbod01:dbod> \d
+----------+--------------------+--------+---------+
| Schema   | Name               | Type   | Owner   |
|----------+--------------------+--------+---------|
| api      | attribute          | view   | dbod    |
| api      | fim_data           | view   | dbod    |
| api      | functional_aliases | view   | dbod    |
| api      | host               | view   | dbod    |
| api      | host_aliases       | view   | dbod    |
| api      | instance           | view   | dbod    |
| api      | metadata           | view   | dbod    |
| api      | rundeck_instances  | view   | dbod    |
| api      | volume             | view   | dbod    |
+----------+--------------------+--------+---------+
```

[12] At this moment

Data abstraction layer

postgrest

What

- Standalone web server which turns a PostgreSQL database directly into a RESTful API[13]
- Single binary
- Running locally on the same node as the API servers

Simple use

```
postgrest postgres://dbod@dbod-dbod01.cern.ch:6601/dbod -a c
```

[13] https://postgrest.com/en/v0.4/intro.html

Data abstraction layer

curl localhost:3000 | jq


```
[
{
"schema": "api",
"name": "attribute",
"insertable": true
},
{
"schema": "api",
"name": "fim_data",
"insertable": true
},
{
"schema": "api",
"name": "functional_aliases",
"insertable": true
},
{
"schema": "api",
"name": "host",
"insertable": true
},

{
"schema": "api",
"name": "host_aliases",
"insertable": false
},
{
"schema": "api",
"name": "instance",
"insertable": true
},
{
"schema": "api",
"name": "metadata",
"insertable": false
},
{
"schema": "api",
"name": "rundeck_instances",
"insertable": false
},
{
"schema": "api",
"name": "volume",
"insertable": true
}
]
```