Deploying Apache Ranger in the Big Data ecosystem at CERN

Emil Kleszcz @ CERN

09 Oct 2023
Presentation content

1. About CERN
2. Big Data ecosystem @ CERN
3. Overview of Apache Ranger project
4. Apache Ranger @ CERN
5. Summary
About CERN
Our goal is to understand the most fundamental particles and laws of the universe.

CERN is the world’s biggest laboratory for particle physics.
COLLABORATION
CERN was founded in 1954 with 12 European Member States

23 Member States
Austria – Belgium – Bulgaria – Czech Republic
Denmark – Finland – France – Germany – Greece
Hungary – Israel – Italy – Netherlands – Norway
Poland – Portugal – Romania – Serbia – Slovakia
Spain – Sweden – Switzerland – United Kingdom

3 Associate Member States in the pre-stage to membership
Cyprus – Estonia – Slovenia

7 Associate Member States
Croatia – India – Latvia – Lithuania – Pakistan
Türkiye – Ukraine

6 Observers
Japan – Russia (suspended) – USA
European Union – JINR (suspended) – UNESCO

Brazil signed Associate Membership Agreement on 3 March 2022. Ratification process in Brazil is underway.

Around 50 Cooperation Agreements with non-Member States and Territories

CERN’s annual budget is 1200 MCHF (equivalent to a medium-sized European university)

As of 31 December 2021
Employees: 2676 staff, 783 fellows
Associates: 11175 users, 1556 others
What is the universe made of?

We study the elementary building blocks of matter and the forces that control their behavior.
How did the universe begin?

We reproduce the conditions a fraction of a second after the Big Bang, to gain insight into the structure and evolution of the universe.
Large Hadron Collider (LHC)

- 27 km in circumference
- About 100 m underground
- Superconducting magnets steer the particles around the ring
- Particles are accelerated close to the speed of light
~20 projects other than LHC with > 1200 physicists
The LHC produces more than 1 billion particle collisions per second

The energy of the particles in collision is converted into new particles.
CERN DC and WLCG (Worldwide LHC Computing Grid)

CERN Science is Data Intensive Science

- Run 1 (2009-2013) we stored 65 PB
- Run 2 (2015-2018) we stored 209 PB
- Run 3 (2022-2026) we expect to store 600 PB

Solution is to combine the power of all our computing centres

- CERN Data Centre in Meyrin
  - 13k servers, 450k CPU cores, 320 PB of storage
  - New data centre being built in Prevessin site
Big Data ecosystem @ CERN
Holistic view on the Big Data components

- **Kafka**: Data streaming
- **Zookeeper**: Coordination of distributed systems
- **Spark**: Large scale data processing
- **Hive**: SQL
- **HBase**: NoSql columnar store
- **Sqoop**: Data exchange with RDBMS
- **Presto**: Low latency SQL
- **HDFS**: Hadoop Distributed File System
- **YARN**: Cluster resource manager
Hadoop ecosystem @ CERN

Stream Source → Events → Staging area → Real time stream processing

Stream Source → Events → Staging area → Flush immediately

Stream Source → Events → Staging area → Flush periodically

Big Files
- HDFS

Batch processing

Recent data

Indexed data

Fast data access

OLAP

OLTP

Real time stream processing

09 Oct 2023
Emil Kleszcz | Apache Ranger @ CERN
Key users and workload

- Users:
  - IT Security
  - IT Monitoring
  - Experiments
  - Accelerators Monitoring

- HBase and HDFS clusters
Key statistics

- **5 prod clusters** (+ 3QA & multiple DEV)
- Hardware: Intel-based servers, continuous capacity expansion, hyper-threading enabled, HDDs and SSDs, **180+ bare metal, 60+ VMs**
- **Users**: 600+ (general-purpose), 1000+ (accelerators) and growing
- Storage capacity: **40PB+**
- Service started in 2013
Current security state

- Integration with CERN’s LDAP (e-groups)
- Kerberos and SSL
- FW rules defined
- Potential improvements:
  - Policies and ACLs defined manually
  - Lack of central management
  - No user-friendly auditing (raw logs)
  - No CERN SSO integration for Web UIs

Apache Ranger to improve security and ease management?

# changing owner
hdfs dfs -chown <owner> <hdfs_path>

# granting new permissions to a user
hdfs dfs -setfacl -m user:<grantee>:<permission> <hdfs_path>

# granting new permissions to a group or an e-group
hdfs dfs -setfacl -m group:<group_name>:<permission> <hdfs_path>
Overview of Apache Ranger project
About the tool

- Apache Ranger™ is a framework to enable, monitor and manage comprehensive data security across the Hadoop platform and beyond.
- It supports multiple projects such as: Apache Hadoop, Apache HBase, Apache Kafka, YARN, Apache Hive and some more…
- The plugin agent pulls the policy-changes using REST API.
- Replaced Apache Sentry.

https://github.com/apache/ranger

https://ranger.apache.org/
What Apache Ranger offers

- Fine-Grained Access Control
- Centralized Policy Management
- Dynamic Policy Enforcement
- Centralized auditing
- Delegate administration of policies to group owners
- Integration with LDAP
- Resource and tag-based Policies
- Extensible Lightweight Plugin Architecture
- No single point of failure (if configured)
WebUI and REST APIs

Apache Ranger is a framework to enable, monitor and manage comprehensive data security across the Hadoop platform. Apache Ranger currently provides a centralized security administration, fine-grain access control and detailed auditing for user access within Apache Hadoop, Apache Hive, Apache HBase and other Apache components.

Asset REST

- GET /assets/accessAudit
- GET /assets/assets
- POST /assets/assets
- GET /assets/assets/count
- POST /assets/assets/featConfig
- DELETE /assets/assets/{id}
- GET /assets/assets/{id}
- PUT /assets/assets/{id}
- GET /assets/credential
- GET /assets/credential
- GET /assets/credential/count
- DELETE /assets/credential/{id}
- GET /assets/credential/{id}
- PUT /assets/credential

Allow Conditions:

Select Role | Select Group | Select User
---|---|---
Select Roles | Select Groups | Select Users

Delegate Admin

Read | Write | Execute | Select/Unselect All

Add Permissions |
Web interface experience

![Ranger Web Interface Screenshot]

<table>
<thead>
<tr>
<th>Policy ID</th>
<th>Policy Name</th>
<th>Policy Labels</th>
<th>Status</th>
<th>Audit Logging</th>
<th>Roles</th>
<th>Groups</th>
<th>Users</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>all - table, column-family, column</td>
<td></td>
<td>Enabled</td>
<td>Enabled</td>
<td></td>
<td>ranger</td>
<td>ethnoc</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>UC1: deny all and allow klescz on (be...</td>
<td></td>
<td>Enabled</td>
<td>Enabled</td>
<td></td>
<td>kleszcz</td>
<td>ethnoc</td>
<td></td>
</tr>
</tbody>
</table>
Building the tool regular way

- Included only plugins for our components in use
- Tested against the master branch and latest release 2.4.0
- Build in CentOS 7 with Java 11

```
mvn clean compile package install -DskipTests=true -Dspotbugs.skip=true
-Dchkstyle.skip=true -Dassembly.plugin.version=3.1.0 -Dhadoop.version=3.3.0
-Dhbase.version=2.3.4 -Dzookeeper.version=3.6.1 -Dhive.version=3.0.0
-Dmysql-connector-java.version=8.0.28
-pl '!plugin-ozone, !plugin-solr, !plugin-nifi, !plugin-nifi-registry, !plugin-kudu,
!plugin-kms, !ranger-ozone-plugin-shim, !storm-agent, !ranger-storm-plugin-shim,
!ranger-solr-plugin-shim, !ranger-atlas-plugin-shim, !plugin-atlas, !plugin-kylin,
!ranger-kylin-plugin-shim'
```
Our setup
Usersync plugin

- No performance overhead
- Integrates well with thousands of users and groups
- Responsive/robust
- Supports internal and external sync source
- Great for building Roles
- Integrated with the policies’ definitions
Usersync plugin
Apache Ranger

Plugin deployment
Demo

---

**Ranger UI Overview**

The image shows the user interface of Apache Ranger, a security management tool for Apache Hadoop. The interface is focused on the list of policies for HDFS. Each policy is listed with details such as Policy Name, Policy Labels, Status, Audit Logging, Roles, Groups, and Users. The interface provides options for editing, viewing service details, and implementing policies.

- **Policy List**:
  - Policy ID: 1, Policy Name: all - path, Status: Enabled, Role: Ranger
  - Policy ID: 2, Policy Name: hdfs_audit_path, Status: Enabled, Role: hdfs
  - Policy ID: 3, Policy Name: hdfsAudit, Status: Enabled, Role: hdfs
  - Policy ID: 4, Policy Name: test against cloudera, Status: Enabled, Role: hdfs
  - Policy ID: 5, Policy Name: test against cloudera, Status: Enabled, Role: hdfs
  - Policy ID: 6, Policy Name: test against cloudera, Status: Enabled, Role: hdfs
  - Policy ID: 7, Policy Name: test against cloudera, Status: Enabled, Role: hdfs
  - Policy ID: 8, Policy Name: test against cloudera, Status: Enabled, Role: hdfs
  - Policy ID: 9, Policy Name: test against cloudera, Status: Enabled, Role: hdfs

---

**Notes**

- By default, all policies are enabled if access cannot be determined by Ranger policies, authorization will fall back to HDFS ACLs. If this behavior needs to be changed, modify HDFS plugin config - cassandra:addAuditor: authorization.

---

**Additional Information**

- **Emil Kleszcz | Apache Ranger @ CERN**
- **09 Oct 2023**
Setting resource policies

![Diagram of resource policies with labels: Resource, Allow Conditions, Deny Conditions]
Resource-based policy access flow

1. Resource access Request
2. Find policies for the accessed resource
3. Has more policies with Deny Conditions?
   - Yes
   - Request matches a deny condition in the policy?
     - Yes
     - Request matches a deny exclude?
       - Yes
       - DENY
       - No
       - ALLOW
     - No
   - No
4. Has more policies with Allow Conditions?
   - Yes
   - Request matches an allow condition in the policy?
     - Yes
     - Request matches an allow exclude?
       - Yes
       - DENY / PASS-THROUGH
       - No
     - No
     - ALLOW
   - No

---

Emil Kleszcz | Apache Ranger @ CERN
More details

- Support for autocompletion
- User can be set with one of the following roles: KEYADMIN, ADMIN, USER
- ROLE is assigned for a specific user by the Administrator
- **Delegated-Admin** permission allows other resource administrators to manage permissions for their managed-resources
- **USER** role gives ability to manage only resources for which the user has been granted with delegated-admin privilege
- Ranger provides **tag-based policies** too
Example with 2 policies

Policies:

1. Block access for a {USER} to /user/*
2. Override to allow {USER} access to /user/{USER}

> kinit cllucas

```
[root@ithdpdev-mmartinm01 ranger-2.4.0-hdfs-plugin]# hdfs dfs -ls /user/cllucas
Found 1 items
drwxr-xr-x   - cllucas supergroup   0 2023-09-11 17:08 /user/cllucas/.sparkStaging
```

```
[root@ithdpdev-mmartinm01 ranger-2.4.0-hdfs-plugin]# hdfs dfs -ls /user/ekleszcz
ls: Permission denied: user=cllucas, access=EXECUTE, inode="/user/ekleszcz"
```
Our experience

- Works with HA clusters (setting namespace)
- No performance overhead
- Extra properties to be set for clusters with Kerb auth in web UI
- Lack of documentation is very painful
- `dfs.namenode.rpc-address.<cluster_namespace>/<hostname>` must be updated to be done when aliases change (hosts upgraded)
Apache Ranger

Plugin deployment
Yarn resources at CERN

### Cluster Metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apps Submitted</td>
<td>194399</td>
</tr>
<tr>
<td>Apps Pending</td>
<td>31</td>
</tr>
<tr>
<td>Apps Running</td>
<td>35</td>
</tr>
<tr>
<td>Apps Completed</td>
<td>194333</td>
</tr>
<tr>
<td>Containers Running</td>
<td>400</td>
</tr>
<tr>
<td>Memory Used</td>
<td>5.38 TB</td>
</tr>
<tr>
<td>Memory Total</td>
<td>30.07 TB</td>
</tr>
<tr>
<td>MR</td>
<td>7 GB</td>
</tr>
</tbody>
</table>

### Cluster Nodes Metrics

<table>
<thead>
<tr>
<th>Node Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Nodes</td>
<td>58</td>
</tr>
<tr>
<td>Decommissioning Nodes</td>
<td>0</td>
</tr>
<tr>
<td>Decommissioned Nodes</td>
<td>0</td>
</tr>
<tr>
<td>Lost Nodes</td>
<td>0</td>
</tr>
<tr>
<td>Unhealthy Nodes</td>
<td>0</td>
</tr>
</tbody>
</table>

### Scheduler Metrics

<table>
<thead>
<tr>
<th>Scheduler Type</th>
<th>Scheduling Resource Type</th>
<th>Minimum Allocation</th>
<th>Maximum Allocation</th>
</tr>
</thead>
</table>

### Application Queues

**Legend:**
- Steady Fair Share
- Instantaneous Fair Share
- Used
- Used (over fair share)
- Max Capacity

**Trees:**
- `root`
- `+ root.certlogcompress`
- `+ root.hconfig`
- `+ root.monitops`
- `+ root.certaner`
- `+ root.perezde`
- `+ root.emonitors`
- `+ root.tapeops`
- `+ root.atlevind`
- `+ root.backup`
- `+ root.hmonitor`

---

**Yarn Fair scheduler**
Setting policies for a fair scheduler

queue name is: root.{group}.{user}

- **submit-app**: submit apps to the queue
- **admin-queue**: administer a queue (be able to kill an app)
Plugin deployment
HBase ACLs management

Around 10k tables in multiple namespaces
Setting policies

{namespace}:{table}

Permissions:

- Read
- Write
- Create
- Admin
- Execute

Disabling an HBase policy doesn’t reset the previous state.
## Auditing

### Ranger Dashboard

**Access**
- Login Sessions
- Plugins
- Plugin Status
- User Sync

**Admin**
- Resource Policies
- Tag Policies
- Audit
- Security Zone
- Settings
- Reports

### Auditing Log

<table>
<thead>
<tr>
<th>Operation</th>
<th>Audit Type</th>
<th>User</th>
<th>Date (heure d'été d'Europe centrale)</th>
<th>Actions</th>
<th>Session ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>User updated cluscas</td>
<td>Ranger User</td>
<td>admin</td>
<td>09/12/2023 10:06:55 AM</td>
<td>Update</td>
<td>674</td>
</tr>
<tr>
<td>User updated cluscas</td>
<td>Ranger User</td>
<td>admin</td>
<td>09/12/2023 09:23:53 AM</td>
<td>Update</td>
<td>664</td>
</tr>
<tr>
<td>User updated cluscas</td>
<td>Ranger User</td>
<td>admin</td>
<td>09/12/2023 09:19:49 AM</td>
<td>Update</td>
<td>698</td>
</tr>
<tr>
<td>User profile updated cluscas</td>
<td>User Profile</td>
<td>admin</td>
<td>09/12/2023 09:19:49 AM</td>
<td>Update</td>
<td>668</td>
</tr>
<tr>
<td>Policy updated UC1: deny user cluscas on root.lit-hadoop-a...</td>
<td>Ranger Policy</td>
<td>admin</td>
<td>09/11/2023 04:55:59 PM</td>
<td>Update</td>
<td>626</td>
</tr>
<tr>
<td>Policy updated UC1: deny user cluscas on root.lit-hadoop-a...</td>
<td>Ranger Policy</td>
<td>admin</td>
<td>09/11/2023 04:55:53 PM</td>
<td>Update</td>
<td>626</td>
</tr>
<tr>
<td>Policy updated UC2: deny summer-students in root.lit-had...</td>
<td>Ranger Policy</td>
<td>admin</td>
<td>09/11/2023 04:55:33 PM</td>
<td>Update</td>
<td>626</td>
</tr>
<tr>
<td>Policy updated UC2: deny summer-students in root.lit-had...</td>
<td>Ranger Policy</td>
<td>admin</td>
<td>09/11/2023 04:55:12 PM</td>
<td>Update</td>
<td>626</td>
</tr>
<tr>
<td>Policy updated UC2: deny summer-students in root.lit-had...</td>
<td>Ranger Policy</td>
<td>admin</td>
<td>09/11/2023 03:42:42 PM</td>
<td>Update</td>
<td>626</td>
</tr>
<tr>
<td>Policy updated all - table, column-family, column</td>
<td>Ranger Policy</td>
<td>admin</td>
<td>09/11/2023 02:51:44 PM</td>
<td>Update</td>
<td>623</td>
</tr>
<tr>
<td>Policy updated UC1: deny cluscas and allow kleszcz on d...</td>
<td>Ranger Policy</td>
<td>admin</td>
<td>09/11/2023 02:51:32 PM</td>
<td>Update</td>
<td>623</td>
</tr>
<tr>
<td>Policy updated all - table, column-family, column</td>
<td>Ranger Policy</td>
<td>admin</td>
<td>09/11/2023 02:51:23 PM</td>
<td>Update</td>
<td>623</td>
</tr>
<tr>
<td>Policy updated UC1: deny cluscas and allow kleszcz on d...</td>
<td>Ranger Policy</td>
<td>admin</td>
<td>09/11/2023 02:51:13 PM</td>
<td>Update</td>
<td>623</td>
</tr>
<tr>
<td>Policy updated all - table, column-family, column</td>
<td>Ranger Policy</td>
<td>admin</td>
<td>09/11/2023 02:51:01 PM</td>
<td>Update</td>
<td>623</td>
</tr>
</tbody>
</table>
Auditing: logs kept in HDFS

Stored in /ranger/audit/{plugin}/{date:YYYYMMDD}/{filename}

Example: hdfs dfs -cat

```
/hranger/audit/hdfs/20230912/hdfs ranger audit ithdpdev-mmartinm02.cern.ch.log
```

Result of the request
1 is allowed, 0 is denied

Policy ID used to process the request
-1 means that no Ranger policy cover the request
Our experience and concerns

- Documentation
  - Legacy or missing or linked to Ambari or Cloudera but not standalone
  - Little activity for reported issues in GitHub/mailing list
- Compatibility
  - Certain component versions must be respected - no docs
  - Last official release 2.4.0 from 2017
- Integrations
  - Auditing not supported for OpenSearch, only ES 7.10.2
  - Some missing dependencies in yarn plugin (jars missing, eg. apache-commons-lang3)
- Resource intensive - 1G+ of heap needed with certain plugins enabled
- Pull mechanism only - no enforcement of the current state
- Yarn autocompletion didn’t work with fair-scheduler (?)
Upstream contributions

- Clear way to contribute
- A contribution submitted to upstream but not followed up
- A few issues reported to the mailing list
  - (dev@ranger.apache.org)
Future plans

- Puppetize and deploy to PROD
- SSO integration (Apache KNOX)
- Auditing with OpenSearch backend
  - We moved away from ElasticSearch/OpenDistro
- Outsource mgmt. to project managers via Roles
- Explore HA deployment
- Fix autocompletion for Apache YARN
- Dynamic management of the configs
Summary
At CERN, we collect 100s of PBs of data from various physics experiments.

We have a lot of users who access the data from our Big Data services such as HDFS.

We need to protect this data and optimize the access rights management.

Also, we lack a centralized auditing system.

Apache Ranger is a proven-to-work mature project that integrates well with our main components and provides the solution for our needs.

With Ranger, we faced a few issues that still need to be tackled, such as:
  - integration with OpenSearch for auditing
  - autocompletion for the YARN plugin

The tool will be deployed into the prod clusters.
Thank you!

Contact: emil.kleszcz@cern.ch