Multi-Layer Authorization Framework for a Representative Hadoop Ecosystem Deployment

Maanak Gupta, Farhan Patwa, James Benson, and Ravi Sandhu

Institute for Cyber Security and Department of Computer Science
University of Texas at San Antonio

22nd ACM Symposium on Access Control Models and Technologies (SACMAT)
June 21-23, 2017, Indianapolis, IN, USA

Agenda

- Introduction and Motivation
- Multi-layer Access Control
- Hadoop Ecosystem Authorization Architecture
- Access Control Mechanisms and Policy Configuration Points
- Conclusion

Big Data and Big Challenges

- IDC 2025 :
  - global “datasphere” – 163 zettabytes
  - 10x than 2016
- Opportunities: 21st century gold for data miners
- Big Data require “Big Systems”

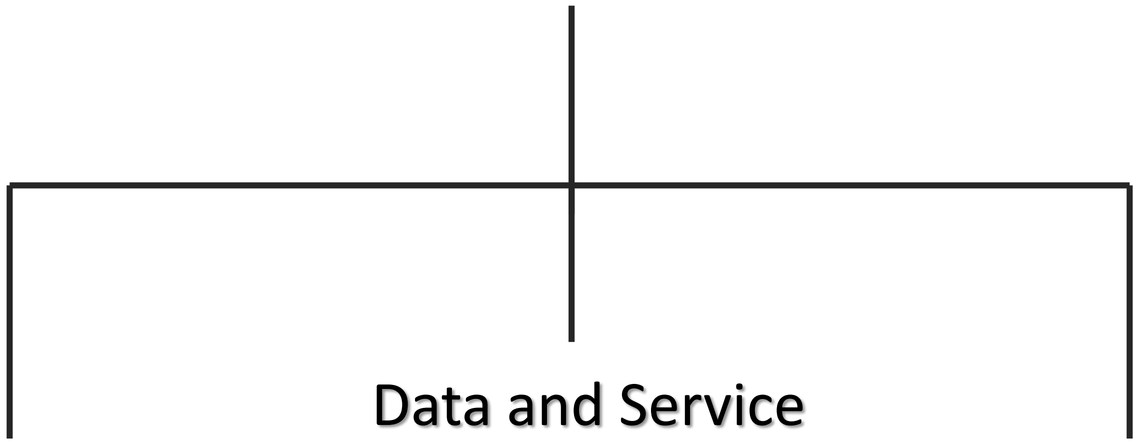
Security:

- Secure Storage
- Privacy Concerns (e.g. HIPPA)
- Fine granular access requirements

Hadoop Ecosystem

- Hadoop: resilient, cost efficient distributed storage (HDFS) and processing framework (MapReduce or YARN)
- Ecosystem = Hadoop core +
  Open-Source Projects
- Hadoop Data Lake
- Security Concerns

Multi-Layer Access Control



Services

HDFS NameNode,
YARN ResourceManager
Apache Hive

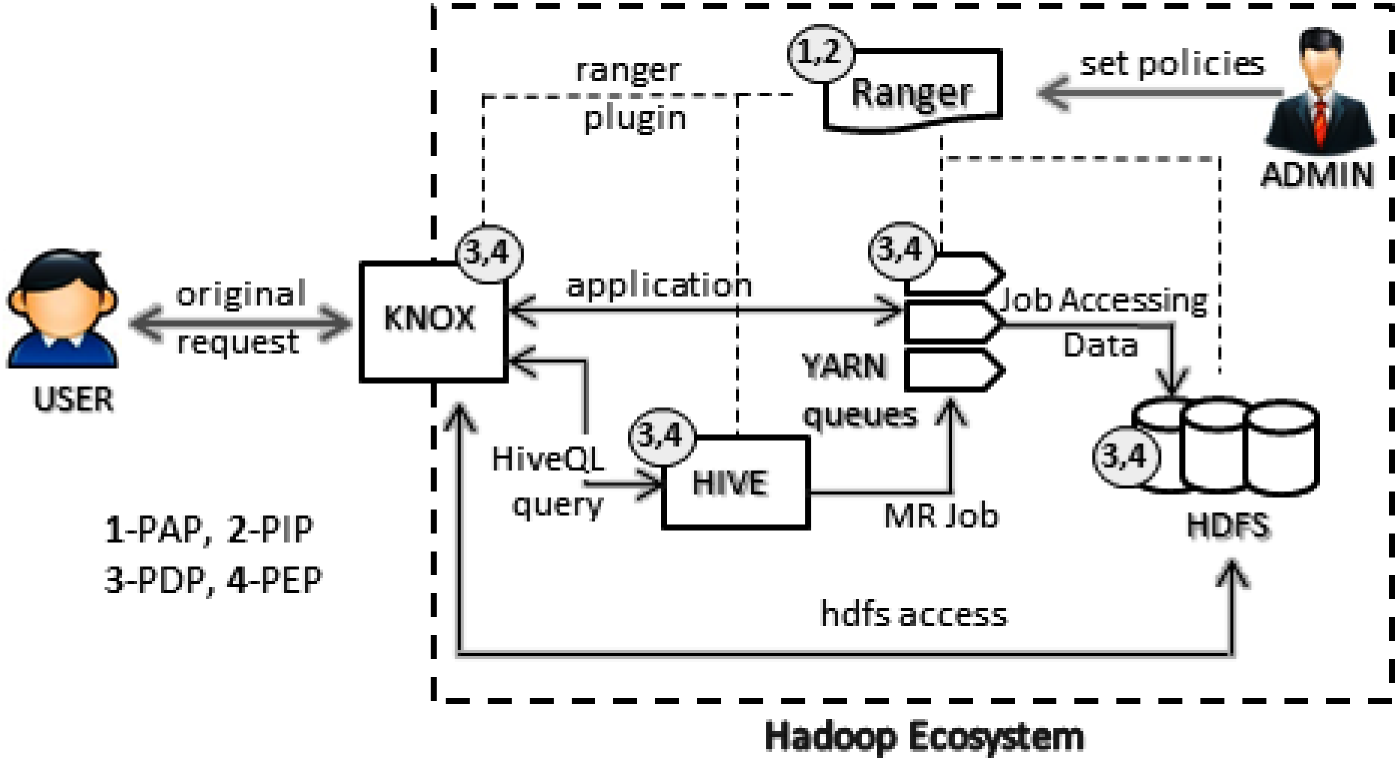
Data and Service Objects

HDFS Files,
Hive Tables
Kafka Topics

Cluster Resources and Applications

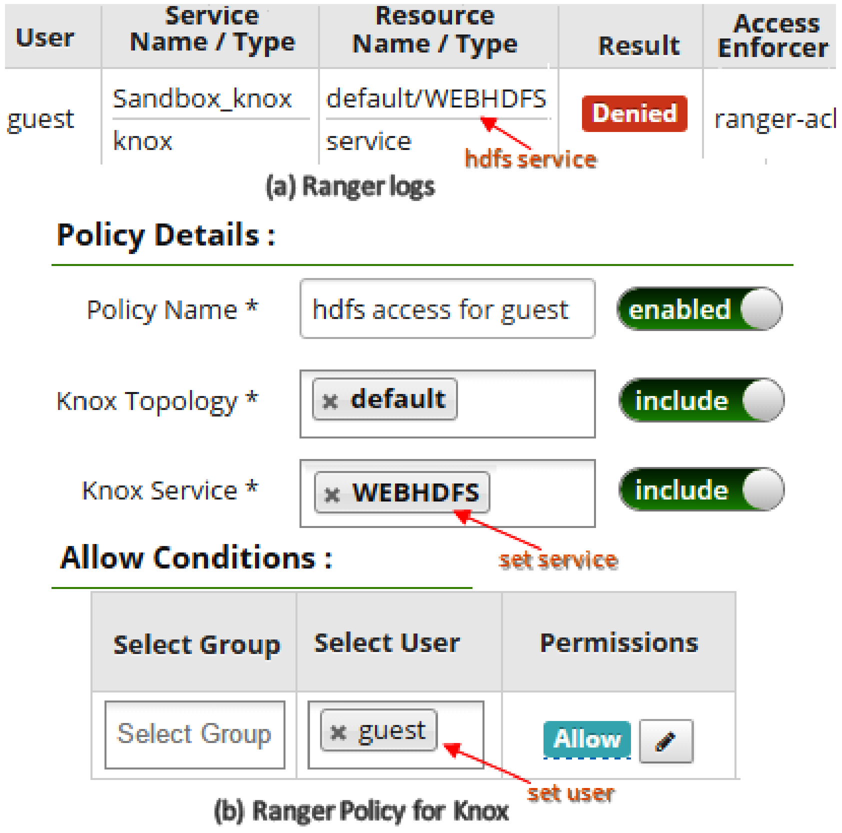
YARN Queues,
Cluster Nodes

Hadoop Ecosystem Authorization Architecture



1-PAP, 2-PIP
3-PDP, 4-PEP

Hadoop and Data Services Access



User	Service Name / Type	Resource Name / Type	Result	Access Enforcer
guest	Sandbox_knox knox	default/WEBHDFS service	Denied	ranger-acl

(a) Ranger logs

Policy Details :

Policy Name * hdfs access for guest — enabled

Knox Topology * × default — include

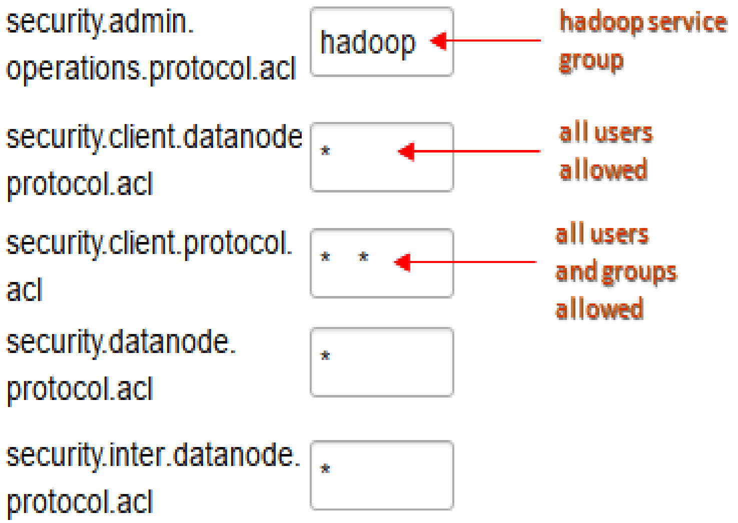
Knox Service * × WEBHDFS — include

Allow Conditions :

Select Group	Select User	Permissions
Select Group	× guest	Allow

(b) Ranger Policy for Knox

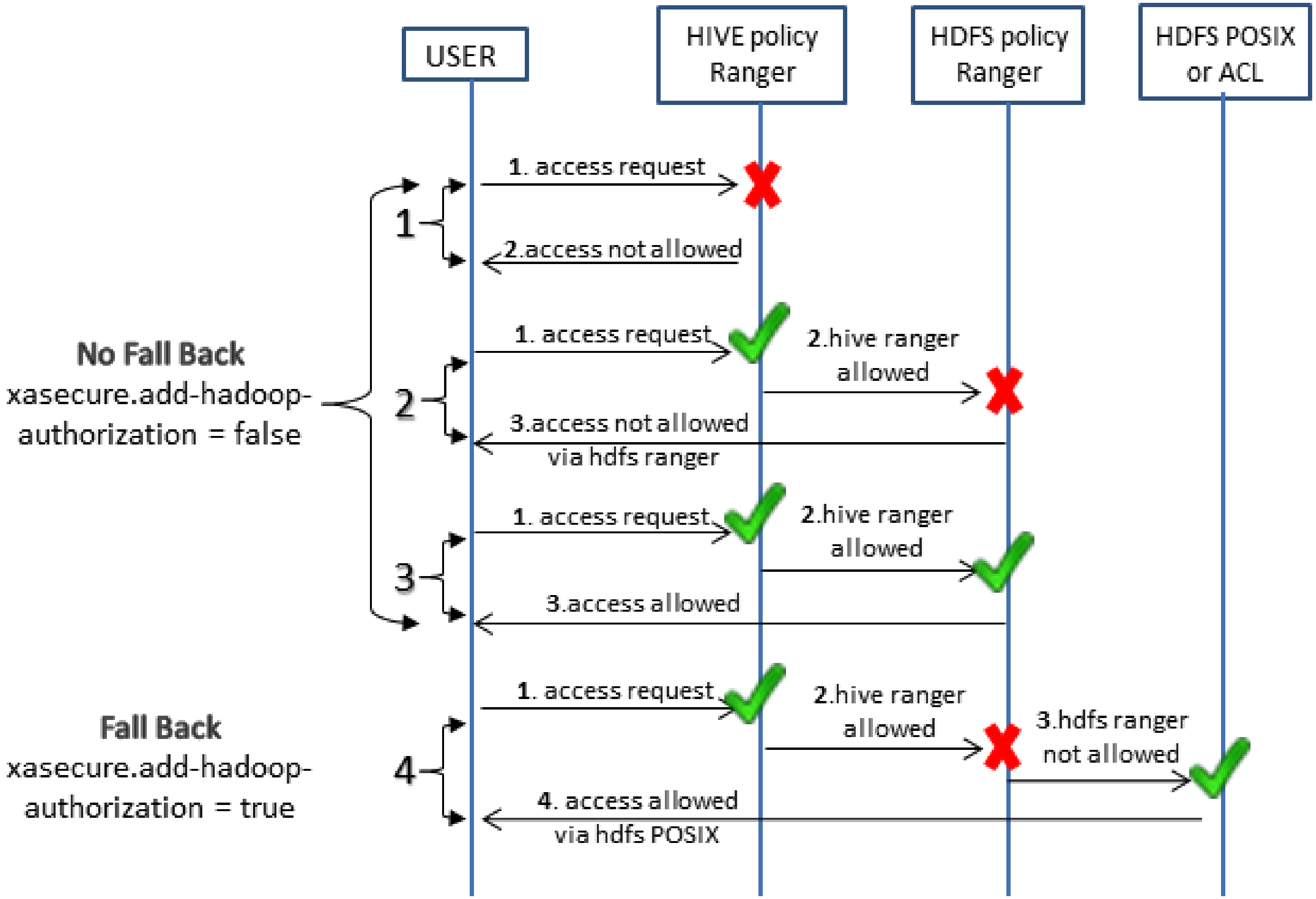
WebHDFS Access via Apache Knox



security.admin.operations.protocol.acl	hadoop
security.client.datanode protocol.acl	*
security.client.protocol.acl	* *
security.datanode.protocol.acl	*
security.inter.datanode.protocol.acl	*

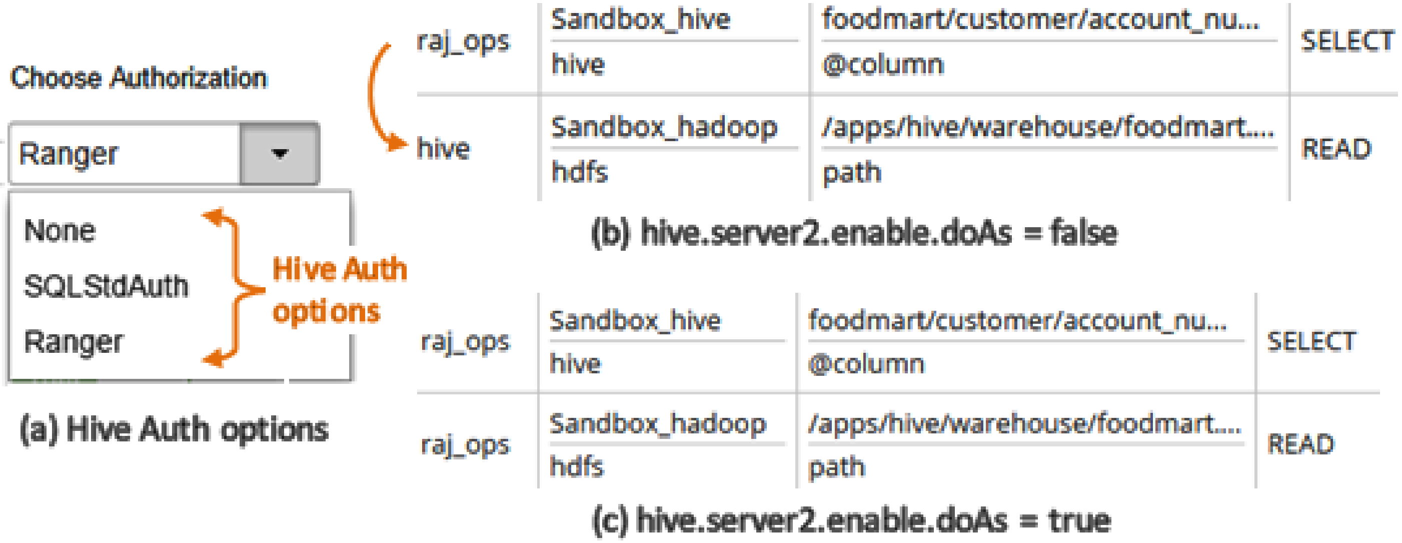
Hadoop Daemons Access Configuration

Data Objects Access



Hive and HDFS Access Configurations

Data Objects Access



Choose Authorization: Ranger

(a) Hive Auth options

raj_ops	Sandbox_hive hive	foodmart/customer/account_nu... @column	SELECT
hive	Sandbox_hadoop hdfs	/apps/hive/warehouse/foodmart.... path	READ

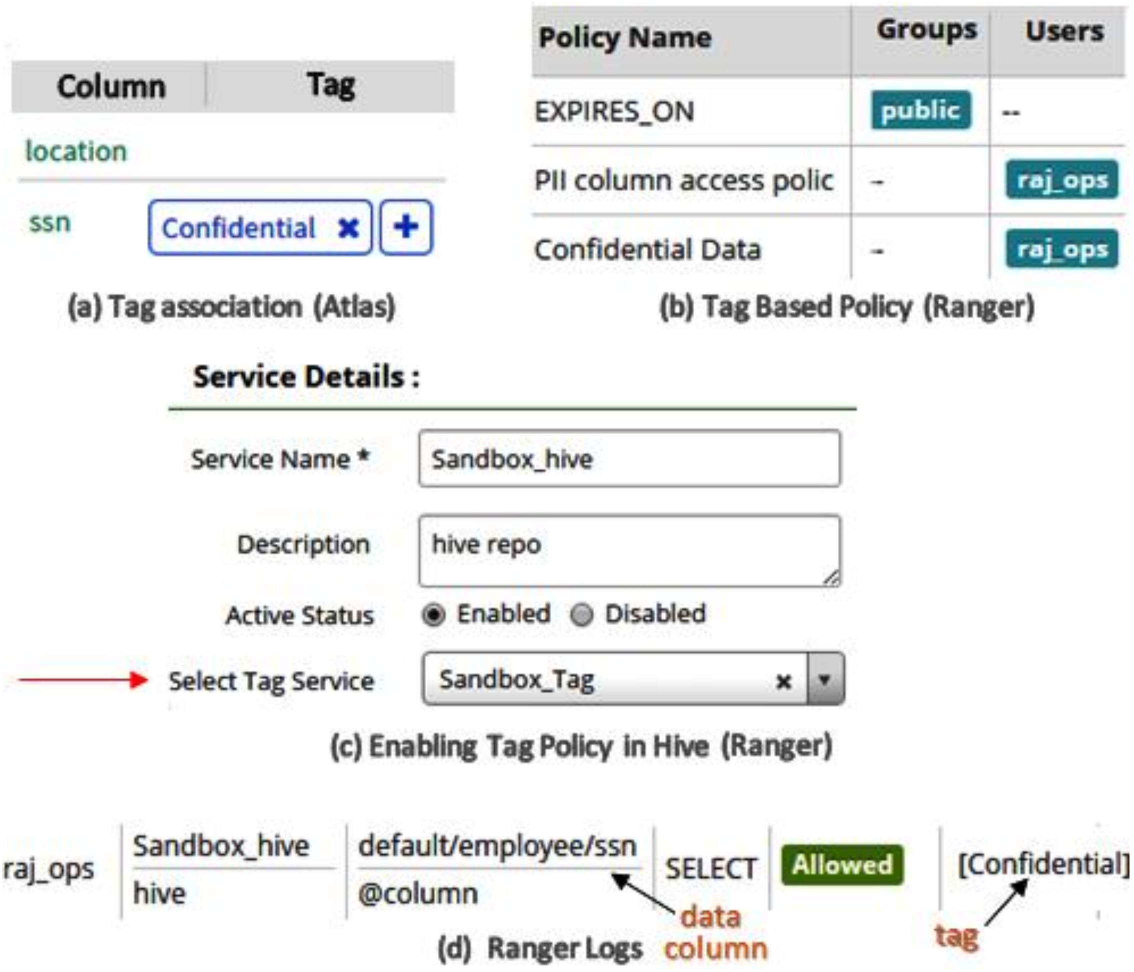
(b) hive.server2.enable.doAs = false

raj_ops	Sandbox_hive hive	foodmart/customer/account_nu... @column	SELECT
raj_ops	Sandbox_hadoop hdfs	/apps/hive/warehouse/foodmart.... path	READ

(c) hive.server2.enable.doAs = true

Authorization Options and End User Impersonation

Data Objects Access



Column	Tag
location	
ssn	Confidential ✖ +

(a) Tag association (Atlas)

Policy Name	Groups	Users
EXPIRES_ON	public	--
PII column access polic	-	raj_ops
Confidential Data	-	raj_ops

(b) Tag Based Policy (Ranger)

Service Details :

Service Name *	Sandbox_hive
Description	hive repo
Active Status	● Enabled ○ Disabled
Select Tag Service	Sandbox_Tag × ▾

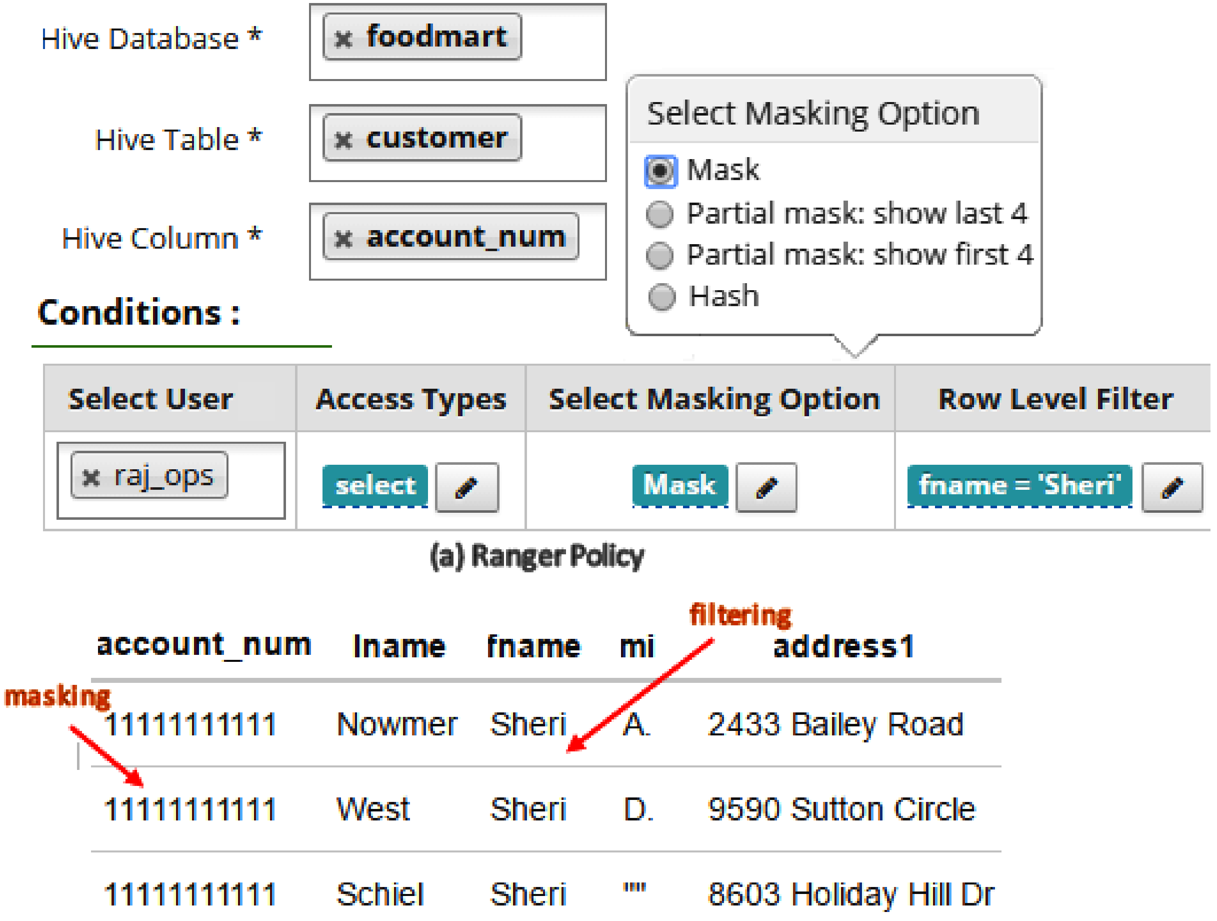
(c) Enabling Tag Policy in Hive (Ranger)

```
raj_ops | Sandbox_hive | default/employee/ssn | SELECT | Allowed | [Confidential]
        | hive         | @column              |        |         |
```

(d) Ranger Logs

Tag Based Policy Configuration

Data Objects Access



Hive Database * foodmart

Hive Table * customer

Hive Column * account_num

Select Masking Option

- ● Mask
- ○ Partial mask: show last 4
- ○ Partial mask: show first 4
- ○ Hash

Conditions :

Select User	Access Types	Select Masking Option	Row Level Filter
raj_ops	select	Mask	fname = 'Sheri'

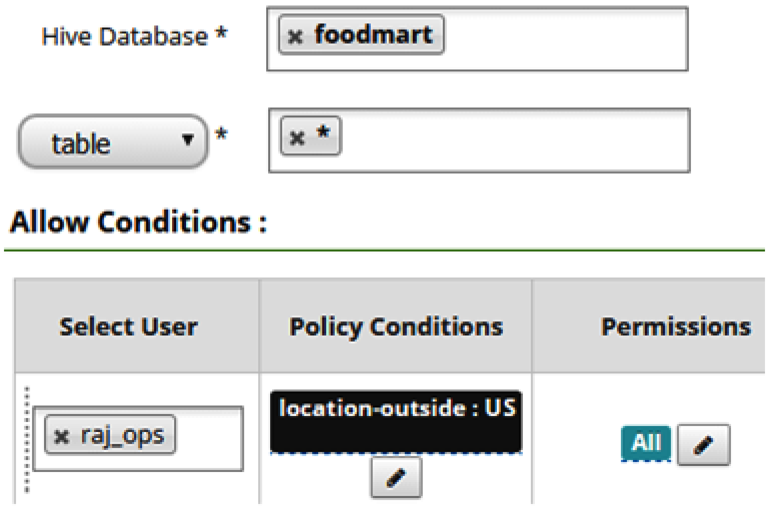
(a) Ranger Policy

account_num	lname	fname	mi	address1
11111111111	Nowmer	Sheri	A.	2433 Bailey Road
11111111111	West	Sheri	D.	9590 Sutton Circle
11111111111	Schiel	Sheri	""	8603 Holiday Hill Dr

(b) Hive View Table

Data Masking and Column Filtering

Context Enricher and Policy Conditions



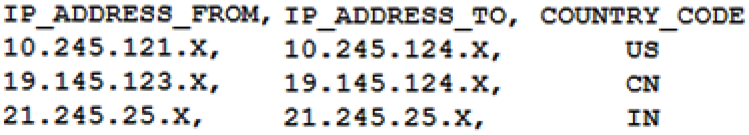
Hive Database * × foodmart

table ▾ * × *

Allow Conditions :

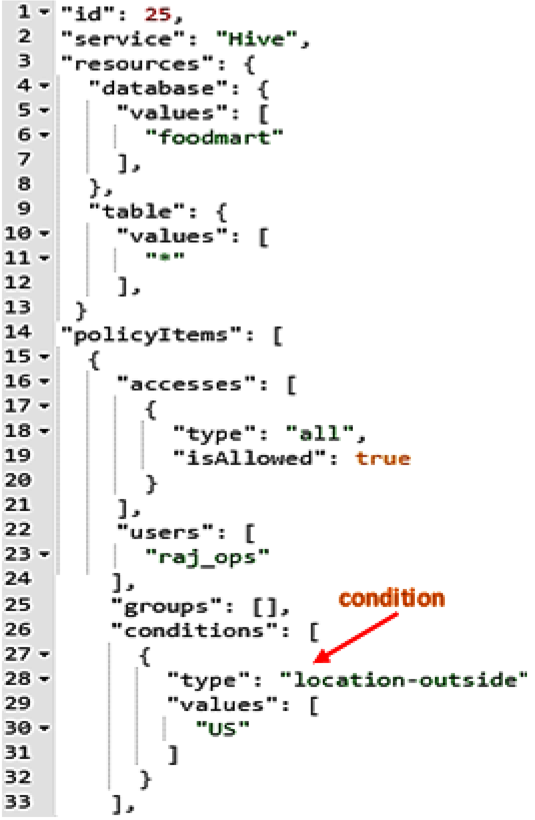
Select User	Policy Conditions	Permissions
× raj_ops	location-outside : US	All

(a) Ranger Policy



```
IP_ADDRESS_FROM, IP_ADDRESS_TO, COUNTRY_CODE
10.245.121.X,    10.245.124.X,       US
19.145.123.X,    19.145.124.X,       CN
21.245.25.X,     21.245.25.X,        IN
```

(b) Text File

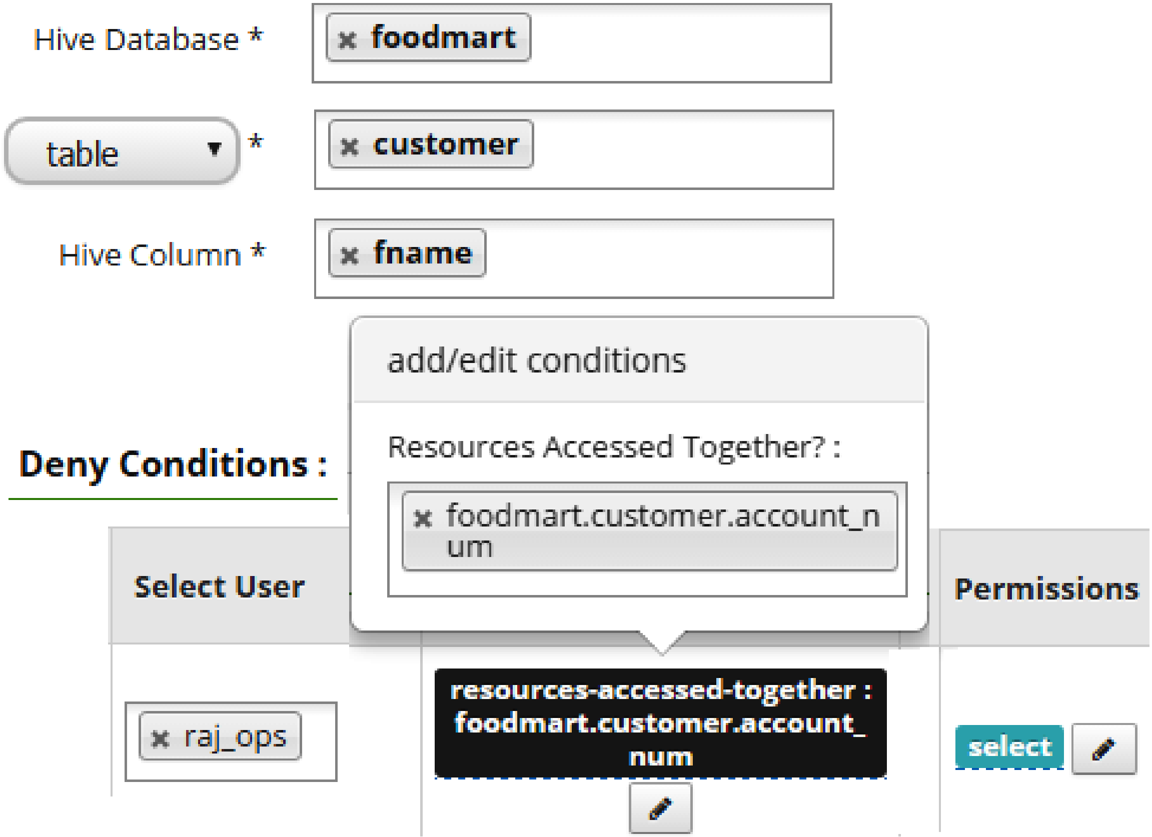


```
"id": 25,
"service": "Hive",
"resources": {
  "database": {
    "values": [
      "foodmart"
    ],
  },
  "table": {
    "values": [
      "*"
    ],
  }
"policyItems": [
  {
    "accesses": [
      {
        "type": "all",
        "isAllowed": true
      }
    ],
    "users": [
      "raj_ops"
    ],
    "groups": [],
    "conditions": [
      {
        "type": "location-outside"
        "values": [
          "US"
        ]
      }
    ],
```

(c) JSON Policy

Geo Location Based Policies

Context Enricher and Policy Conditions



Hive Database * × foodmart

table ▾ * × customer

Hive Column * × fname

add/edit conditions

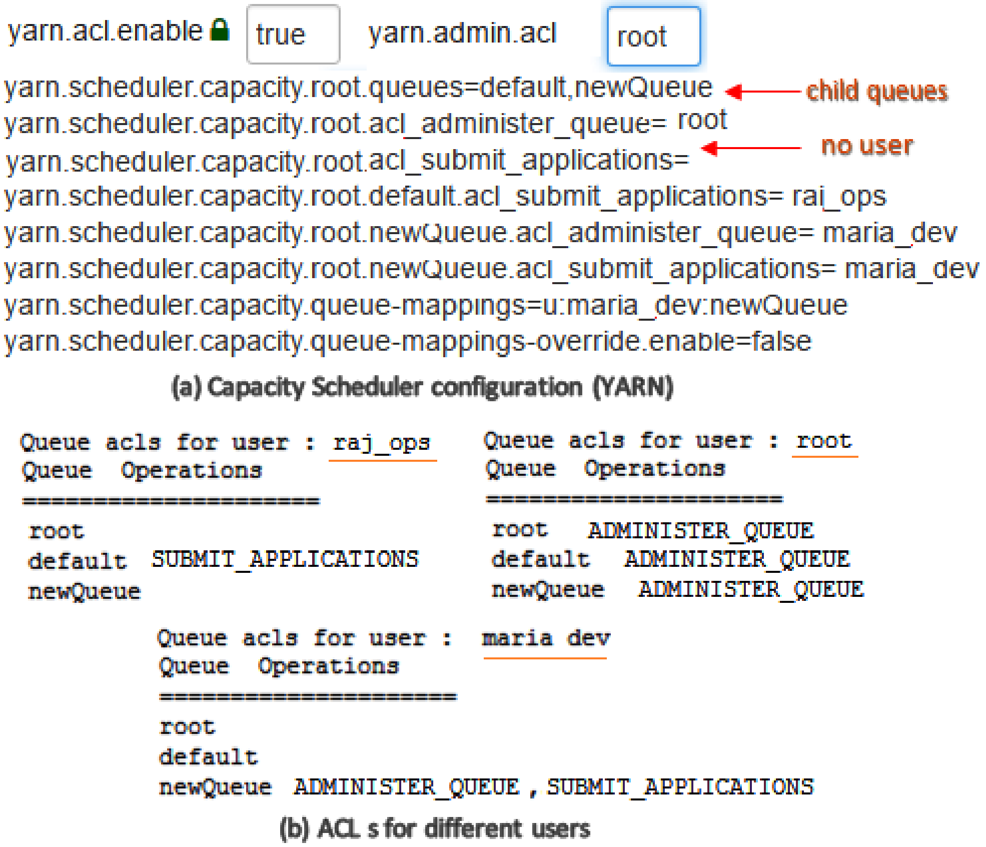
Resources Accessed Together? : × foodmart.customer.account_num

Deny Conditions :

Select User		Permissions
× raj_ops	resources-accessed-together : foodmart.customer.account_num	select

Data Combination Prohibition

Cluster Resource and Application Access



yarn.acl.enable 🔒 true   yarn.admin.acl root

```
yarn.scheduler.capacity.root.queues=default,newQueue
yarn.scheduler.capacity.root.acl_administer_queue= root
yarn.scheduler.capacity.root.acl_submit_applications=
yarn.scheduler.capacity.root.default.acl_submit_applications= raj_ops
yarn.scheduler.capacity.root.newQueue.acl_administer_queue= maria_dev
yarn.scheduler.capacity.root.newQueue.acl_submit_applications= maria_dev
yarn.scheduler.capacity.queue-mappings=u:maria_dev:newQueue
yarn.scheduler.capacity.queue-mappings-override.enable=false
```

child queues

no user

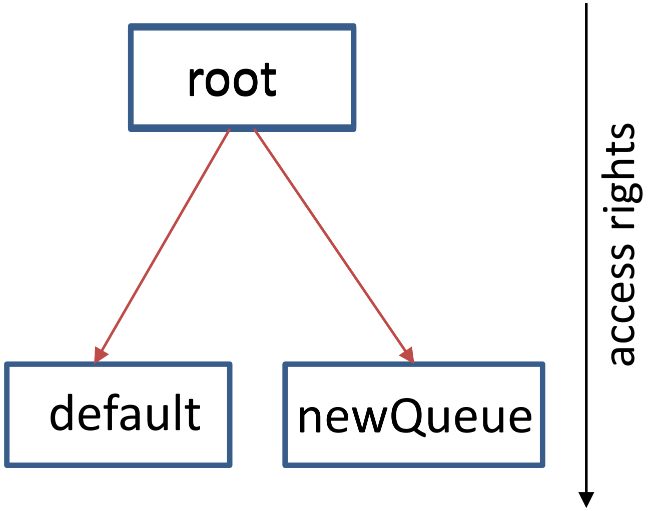
(a) Capacity Scheduler configuration (YARN)

```
Queue acls for user : raj_ops
Queue   Operations
======================
 root
 default  SUBMIT_APPLICATIONS
 newQueue
```

```
Queue acls for user : root
Queue   Operations
======================
 root    ADMINISTER_QUEUE
 default   ADMINISTER_QUEUE
 newQueue   ADMINISTER_QUEUE
```

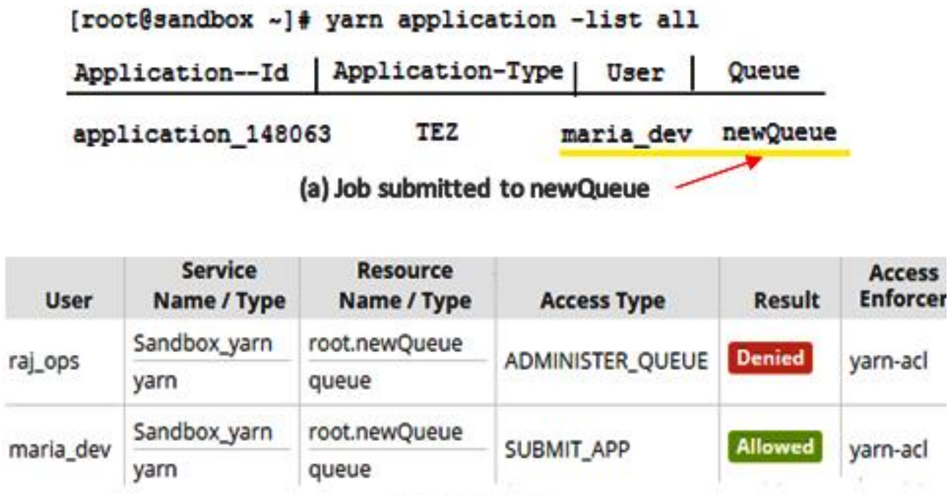
```
Queue acls for user : maria dev
Queue   Operations
======================
root
default
newQueue  ADMINISTER_QUEUE , SUBMIT_APPLICATIONS
```

(b) ACL s for different users



YARN Queue Access Control Configuration

Cluster Resource and Application Access



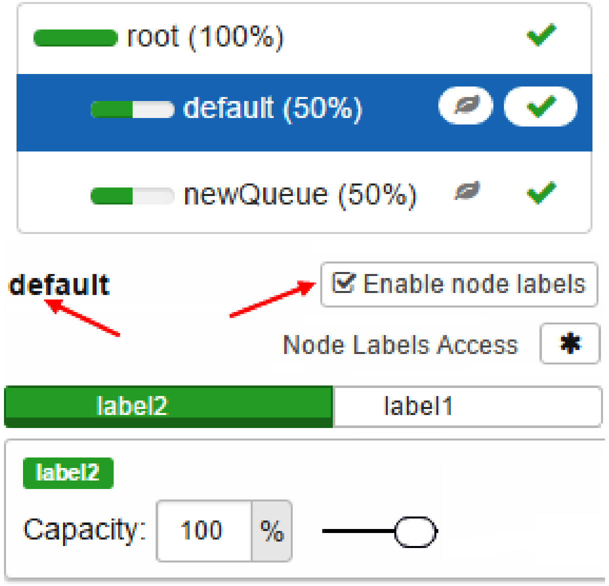
```
[root@sandbox ~]# yarn application -list all
 Application--Id | Application-Type | User | Queue
 application_148063      TEZ         maria_dev  newQueue
```

(a) Job submitted to newQueue

User	Service Name / Type	Resource Name / Type	Access Type	Result	Access Enforcer
raj_ops	Sandbox_yarn yarn	root.newQueue queue	ADMINISTER_QUEUE	Denied	yarn-acl
maria_dev	Sandbox_yarn yarn	root.newQueue queue	SUBMIT_APP	Allowed	yarn-acl

(b) Ranger logs

Queue and Job Level Access Control



root (100%) ✓

default (50%) ✓

newQueue (50%) ✓

default

☑ Enable node labels

Node Labels Access *

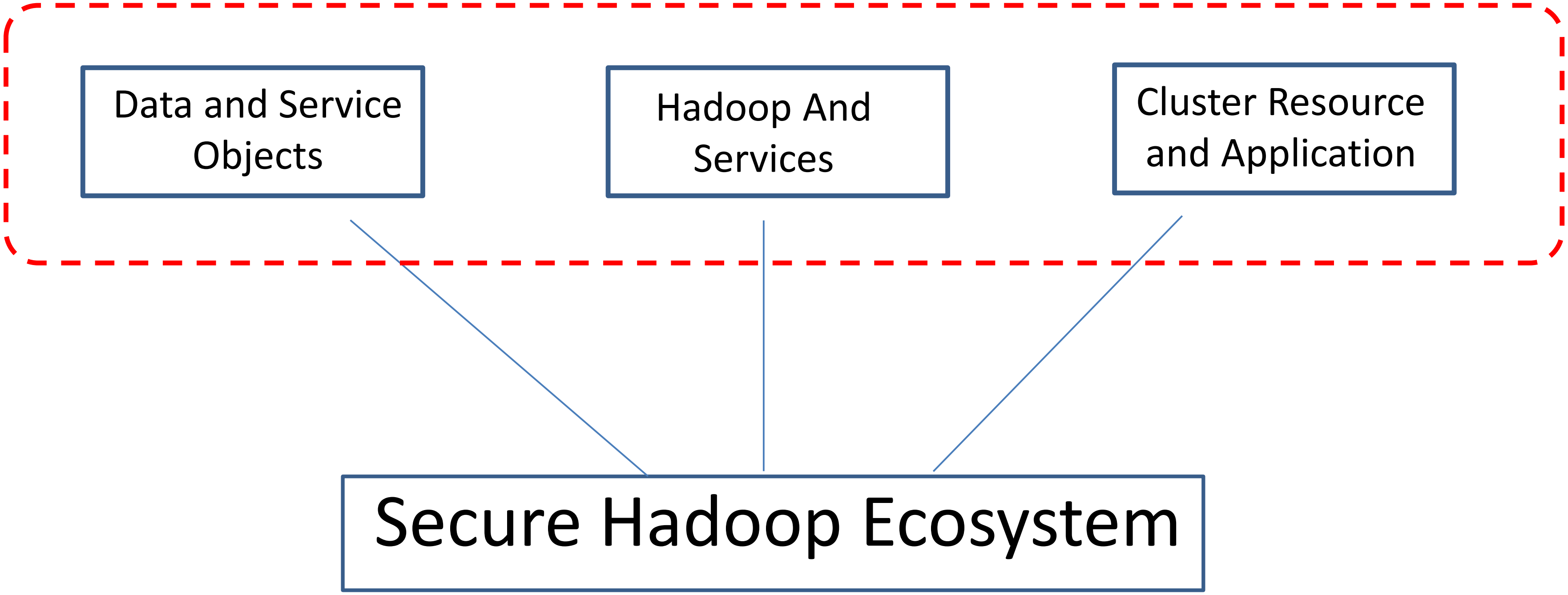
label2 | label1

label2

Capacity: 100 %

Cluster Nodes Configuration

Conclusion



‘Defense in Depth’