

在 Amazon EMR 上使用 Apache Ranger 来实现认证和审计

Contents

Apache Ranger	2
架构	2
演练	3
预先要求	4
设置 AD 服务器	4
设置 Ranger 服务器	7
创建 EMR 集群	9
使用 CloudFormation 模板来创建集群	9
使用 AWS CLI 来创建集群	11
测试集群	13
访问 Web 界面	13
使用 HDFS	13
使用 Hive 查询	13
更新安全策略	14
列屏蔽和行过滤	15
行级过滤	15
列屏蔽	17
审计	18
结论	19
附录	19

本文来源于 <https://aws.amazon.com/cn/blogs/big-data/implementing-authorization-and-auditing-using-apache-ranger-on-amazon-emr/>，并针对中国区（北京区和宁夏区）做了相应修改。

基于角色的访问控制（RBAC）对于多租户的 Hadoop 集群来说是一项重要的安全需求，但是在长期和短暂运行的集群上很难设置和维护。

想象一个组织在使用 Active Directory 的用户和组来完成基于角色的访问控制。他们希望在一个中央安全策略服务器上管理并且强制在 AWS 上所有的 Hadoop 集群都遵循。这个策略服务器同时也必须存放访问和审计信息来保证合规性的需求。

在本文中，我将提供使用 Apache Ranger 来 Amazon EMR 启用认证和审计的步骤。

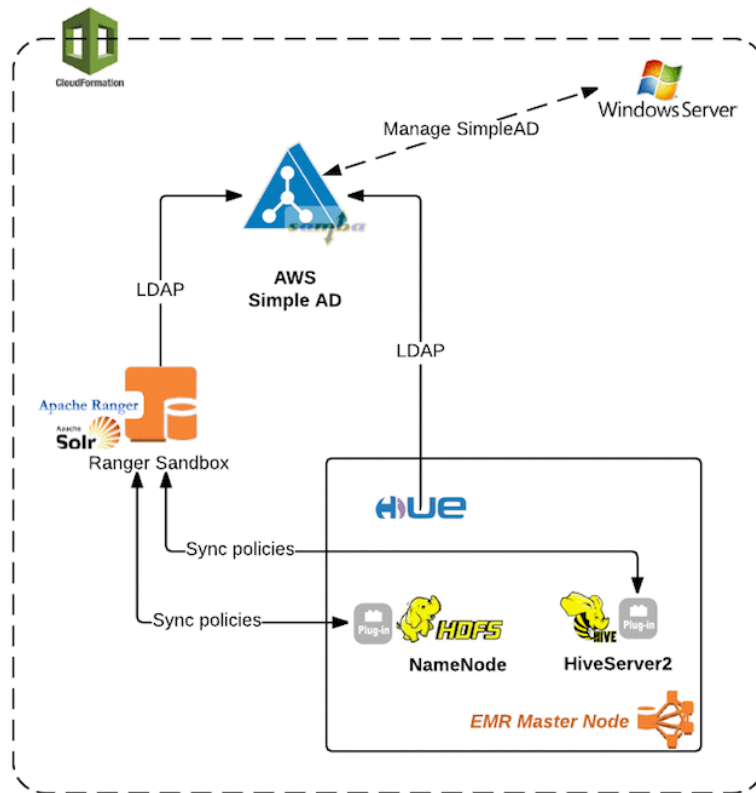
Apache Ranger

Apache Ranger 是一个开源框架，它用来启用，监控和管理 Hadoop 平台上复杂的数据安全。它的功能包括中央化的安全管理，跨越许多 Hadoop 组件（Hadoop，Hive，HBase，Storm，Knox，Solr，Kafka 和 YARN）的精细化认证和中央审计。它使用代理去同步策略和用户，和在 Hadoop 组件同一进程中运行的插件，例如 NameNode。

架构

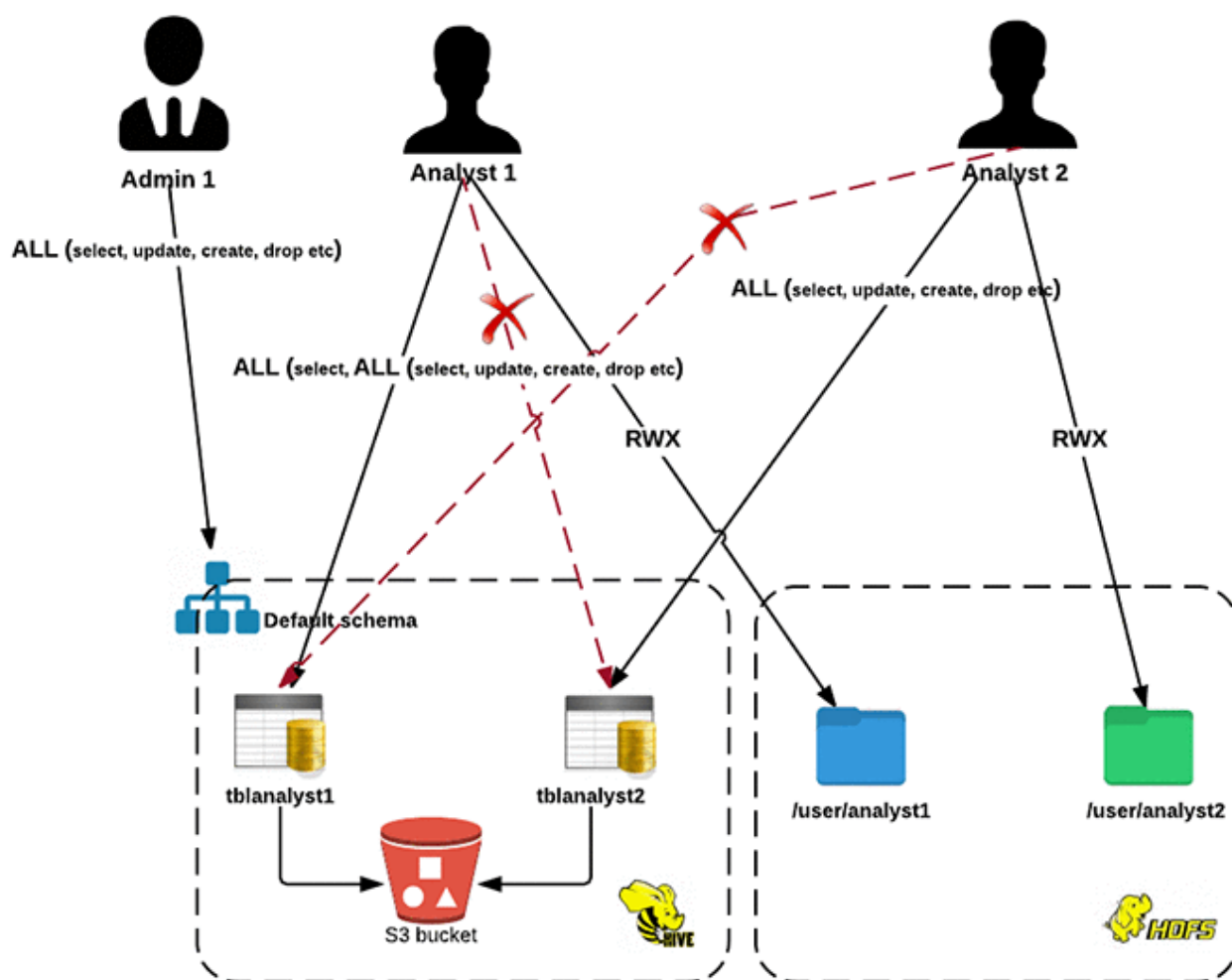
如下图所示，多个 EMR 集群可以通过单独的安全策略服务器来同步策略。这个想法和多个 EMR 集群使用共享的 Hive metastore 类似。（在中国区的实现中，AWS

Simple AD 被替换成单独的一台 Windows EC2，并启用了 Active Directory）



演练

在这个演练中，如下图所示预先创建了三个用户—analyst1, analyst2 和 admin1—作为初始用户。我会在 Ranger 的管理界面上演示如何修改访问权限。这些修改会被传递到 EMR 集群，最后通过 Hue 来验证。



为了管理用户，组和密码，原文使用了 AWS 托管服务 Simple AD。由于中国区尚无此服务，我改用了一台 Windows 2016 EC2，并且手动启用了 Active Directory 功能。然后设置了安全策略服务器（Ranger）和配置 EMR 集群。最后测试了安全策略并且更新它们。

预先要求

以下步骤假定你有一个包含至少两个子网的 VPC，如果是私有子网，则已经配置了 NAT。并且 VPC 中 enableDnsSupport 和 enableDnsHostnames 都是设置为 Yes。

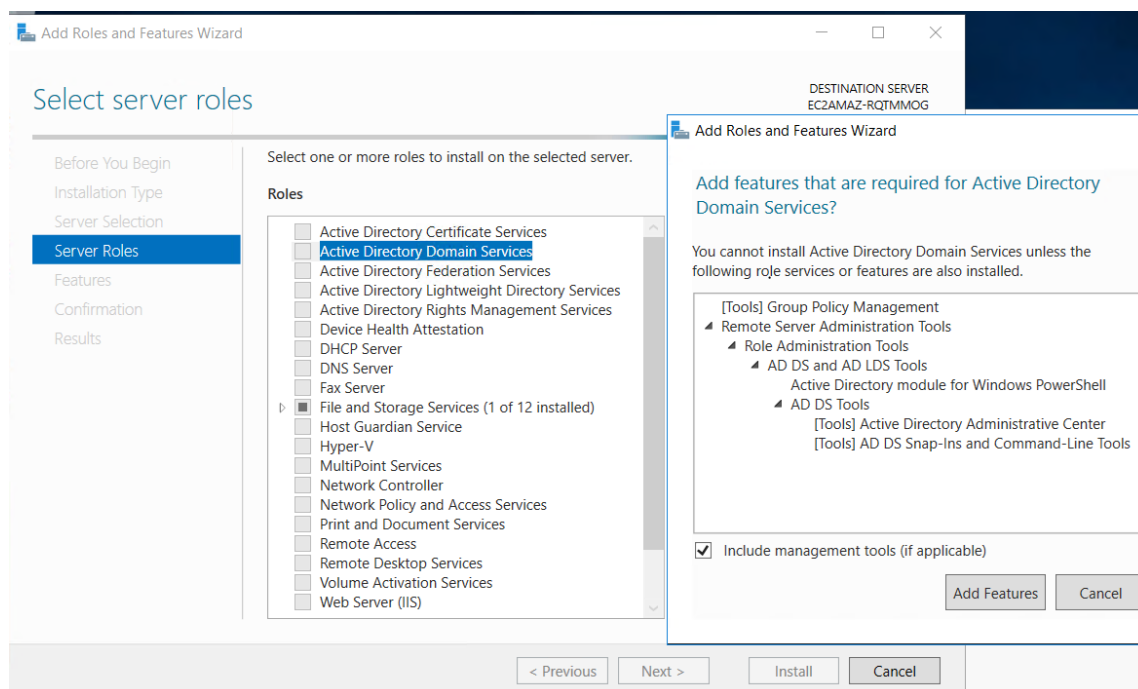
设置 AD 服务器

由于中国区暂时没有 SimpleAD 服务，本文就利用了 EC2 上 Windows 搭建了一个 AD，具体信息如下：

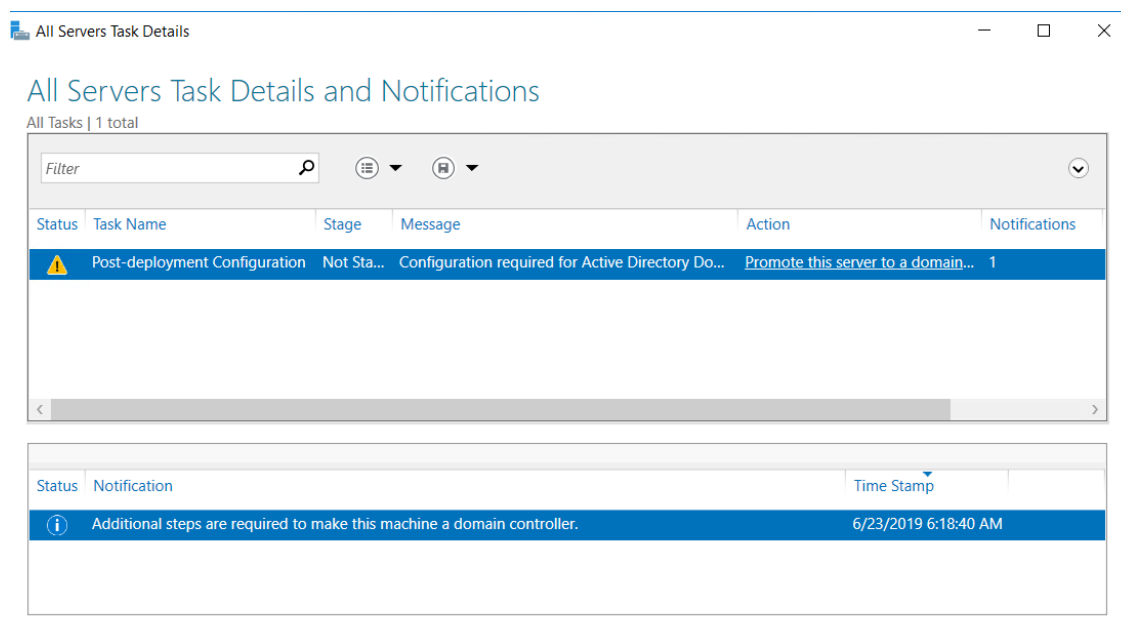
域名	corp.emr.local
管理员密码	Password@123

为节约篇幅，这里仅提供了几个重要步骤：

1. 在上文 VPC 的某一子网内创建一台 Windows 2016 服务器。
2. 安装 Active Directory Domain Service 角色，如下图：



3. 安装完之后将服务器提升为域控制器。



4. 创建 corp.emr.local 的域

Active Directory Domain Services Configuration Wizard

TARGET SERVER
EC2AMAZ-RQTMMOG

Review Options

Deployment Configuration

Domain Controller Options

DNS Options

Additional Options

Paths

Review Options

Prerequisites Check

Installation

Results

Review your selections:

Configure this server as the first Active Directory domain controller in a new forest.

The new domain name is "corp.emr.local". This is also the name of the new forest.

The NetBIOS name of the domain: CORP

Forest Functional Level: Windows Server 2016

Domain Functional Level: Windows Server 2016

Additional Options:

Global catalog: Yes

DNS Server: Yes

Create DNS Delegation: No

These settings can be exported to a Windows PowerShell script to automate additional installations

View script

More about installation options

< Previous

Next >

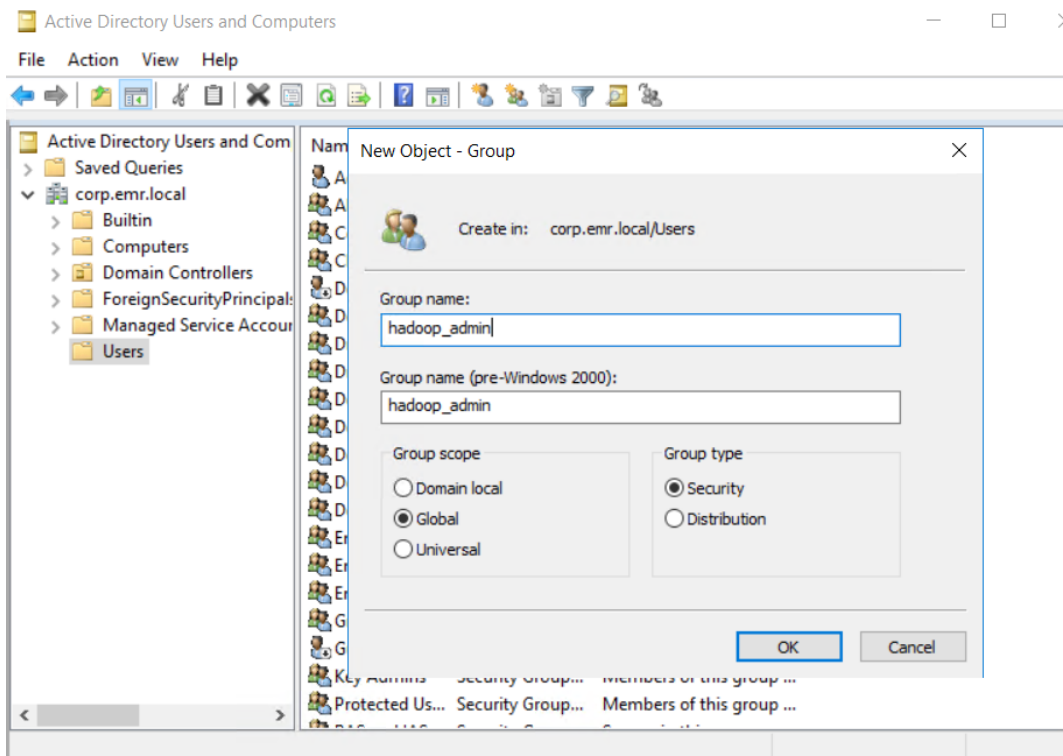
Install

Cancel

5. 创建测试用户组：

组名	hadoop_admin, hadoop_analyst
范围	Global
类型	Security

如下图：



6. 创建四个用户，并放入相应的组

用户名	binduser	analyst1	analyst2	admin1
First name	Bind	Hadoop	Hadoop	Hadoop
Last name	User	Analyst1	Analyst2	Admin1
Full name	Bind User	Hadoop Analyst1	Hadoop Analyst2	Hadoop Admin1
Logon name	binduser	analyst1	analyst2	admin1
Password	Bond@U123	Had00p@User1	Had00p@User2	Had00p@User3
Group		hadoop_analyst	hadoop_analyst	hadoop_admin

设置 Ranger 服务器

上面设置好 AD 服务器和用户后，我们现在就可以开始设置安全策略服务器（Ranger）。我们将在 Amazon Linux 实例上安装和运行 Ranger。我们可以通过 CloudFormation 来快速搭建它。

CloudFormation 的模板在 [https://aws-bigdata.s3.cn-north-](https://aws-bigdata.s3.cn-north-1.amazonaws.com.cn/artifacts/cloudformation/ranger-server.template.json)

1.amazonaws.com.cn/artifacts/cloudformation/ranger-server.template.json。你可以在 AWS 控制台的 CloudFormation 界面运行它，需要参数如下：

InstanceType	实例类型
KeyName	EC2 Key Pair 名字
Subnet	子网

VPC	与上文相同 VPC
myDirectoryBaseDN	dc=corp,dc=emr,dc=local
myDirectoryBindUser	binduser@corp.emr.local
myDirectoryBindPassword	Bond@U123
myDirectoryIPAddress	Windows AD 服务器地址
rangerVersion	Ranger 版本，可选 0.6，0.7 和 1.0
s3artifactsRepoHttp	所需脚本和安装文件路径，请使用默认路径 https://aws-bigdata.s3.cn-north-1.amazonaws.com.cn/artifacts

具体设置如下图：

Stack name

ranger-server

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

InstanceType
Instance type of the Ranger Server

m4.xlarge

KeyName
Name of an existing EC2 KeyPair to enable SSH to the instances

■

Subnet
Subnet ID for creating the EMR cluster

subnet-010ecbe114603ff99 (10.101.0.0/24) (public0-ssm-demo)

VPC
VPC ID for creating the EMR cluster

vpc-020946dea1950dd55 (10.101.0.0/16) (vpc-ssm-demo)

myDirectoryBaseDN
Base DN SimpleAD server

dc=corp,dc=emr,dc=local

myDirectoryBindPassword
BindPassword AD server

Bond@U123

myDirectoryBindUser
BindUser AD server

binduser@corp.emr.local

myDirectoryIPAddress
IP Address of the AD server

10.101.0.74

rangerVersion
RangerVersion

0.7

s3artifactsRepoHttp
Git Repo URL for this blog.

<https://aws-bigdata.s3.cn-north-1.amazonaws.com.cn/artifacts>

当 CloudFormation 顺利完成时，可以查看 output 中输出的 Range 服务器地址。这时可以登陆 Ranger 管理界面：<http://<Ranger 服务器地址>:6080/login.jsp>。缺省的用户名密码是：admin/admin。

创建 EMR 集群

最后该创建 EMR 集群并且配置所需要的插件了。你可以使用 AWS CLI 或者 CloudFormation 模板来创建 EMR 集群。要注意：不是所有的安全配置都被 CloudFormation 支持。

使用 CloudFormation 模板来创建集群

<https://bigdata-cn-northwest-1/artifacts>

<https://bigdata-cn-northwest-1/artifacts>

<https://aws-bigdata.s3.cn-north-1.amazonaws.com.cn/artifacts>

你可以使用 <https://aws-bigdata.s3.cn-north-1.amazonaws.com.cn/artifacts/cloudformation/emr-template.template.json> 模板创建一个 EMR 集群，以下是所需参数：

CoreInstanceCount	Core 节点数目
CoreInstanceType	Core 节点类型
EMRClusterName	EMR 集群名字
EMRLogDir	EMR 日志存放路径
KeyName	EC2 Key pair 名字
LDAPServerIP	Windows AD 服务器地址
MasterInstanceType	Master 节点类型
RangeHostname	Ranger 服务器地址
Subnet	子网
VPC	与上文相同 VPC
emrReleaseLabel	EMR 版本
rangerVersion	Ranger 版本，可选 0.6，0.7 和 1.0
s3artifactsRepoHttp	所需脚本和安装文件路径，请使用默认路径 s3://bigdata-cn-northwest-1/artifacts

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

CoreInstanceCount

Number of core instances

CoreInstanceType

Instance Type of the core node

EMRClusterName

Cluster name for the EMR

EMRLogDir

Log Dir for the EMR cluster

KeyName

Name of an existing EC2 KeyPair to enable SSH to the instances

LDAPServerIP

IP address of the LDAP server

MasterInstanceType

Instance Type of the master node

RangerHostname

Internal IP address of the Ranger Server

Subnet

Subnet ID for creating the EMR cluster

VPC

VPC ID for creating the EMR cluster

emrReleaseLabel

Release label for the EMR cluster

myDirectoryBindPassword

BindPassword AD server

myDirectoryBindUser

BindUser AD server

rangerVersion

Version of the Ranger Server.

s3artifactsRepo

Git Repo URL for this blog.

当 CloudFormation 顺利完成时，可以查看 output 中输出的 EMR Master 服务器地址。

使用 AWS CLI 来创建集群

```
aws emr create-cluster --applications Name=Hive Name=Spark Name=Hue --tags 'Name=EMR-Security' \

--release-label emr-5.17.0 \

--ec2-attributes 'SubnetId=<subnet-xxxxx>, InstanceProfile=EMR_EC2_DefaultRole, KeyName=<Key name>' \

--service-role EMR_DefaultRole \

--instance-count 4 \

--instance-type m4.xlarge \

--log-uri 's3 location for logging' \

--bootstrap-actions '[{"Path": "s3://bigdata-cn-northwest-1/artifacts/scripts/download-scripts.sh", "Args": ["s3://bigdata-cn-northwest-1/artifacts"], "Name": "Download scripts"}]' \

--steps '[{"Args": ["/mnt/tmp/aws-blog-emr-ranger/scripts/emr-steps/install-hive-hdfs-ranger-policies.sh", "<ranger host ip>", "s3://bigdata-cn-northwest-1/artifacts/inputdata"], "Type": "CUSTOM_JAR", "MainClass": "", "ActionOnFailure": "CONTINUE", "Jar": "s3://cn-northwest-1.elasticmapreduce/libs/script-runner/script-runner.jar", "Properties": "", "Name": "InstallRangerPolicies"}, {"Args": ["spark-submit", "--deploy-mode", "cluster", "--class", "org.apache.spark.examples.SparkPi", "/usr/lib/spark/examples/jars/spark-examples.jar", "10"], "Type": "CUSTOM_JAR", "MainClass": "", "ActionOnFailure": "CONTINUE", "Jar": "command-runner.jar", "Properties": "", "Name": "SparkStep"}, {"Args": ["/mnt/tmp/aws-blog-emr-ranger/scripts/emr-steps/install-hive-hdfs-ranger-plugin.sh", "<ranger host ip>", "0.6", "s3://bigdata-cn-northwest-1/artifacts"], "Type": "CUSTOM_JAR", "MainClass": "", "ActionOnFailure": "CONTINUE", "Jar": "s3://cn-northwest-1.elasticmapreduce/libs/script-runner/script-runner.jar", "Properties": "", "Name": "InstallRangerPlugin"}, {"Args": ["/mnt/tmp/aws-blog-emr-ranger/scripts/emr-steps/loadDataIntoHDFS.sh", "us-east-1"], "Type": "CUSTOM_JAR", "MainClass": "", "ActionOnFailure": "CONTINUE", "Jar": "s3://cn-northwest-1.elasticmapreduce/libs/script-runner/script-runner.jar", "Properties": "", "Name": "LoadHDFSData"}, {"Args": ["/mnt/tmp/aws-blog-emr-ranger/scripts/emr-steps/createHiveTables.sh", "us-east-1"], "Type": "CUSTOM_JAR", "MainClass": "", "ActionOnFailure": "CONTINUE", "Jar": "s3://cn-northwest-1.elasticmapreduce/libs/script-runner/script-runner.jar", "Properties": "", "Name": "CreateHiveTables"}]' \
```

```
--configurations '[{"Classification":"hue-
ini","Properties":{},"Configurations":[{"Classification":"desktop","Properties":{},"C
onfigurations":[{"Classification":"auth","Properties":{"backend":"desktop.auth.backen
d.LdapBackend"},"Configurations":[]}, {"Classification":"ldap","Properties":{"bind_dn"
:"binduser","trace_level":"0","search_bind_authentication":"false","debug":"true","ba
se_dn":"dc=corp,dc=emr,dc=local","bind_password":"Bond@U123","ignore_username_case":
"true","create_users_on_login":"true","ldap_username_pattern":"uid=<username>,cn=users
,dc=corp,dc=emr,dc=local","force_username_lowercase":"true","ldap_url":"ldap://<ip
address of simple ad
server>","nt_domain":"corp.emr.local"},"Configurations":[{"Classification":"groups","
Properties":{"group_filter":"objectclass=*","group_name_attr":"cn"},"Configurations":
[]}, {"Classification":"users","Properties":{"user_name_attr":"sAMAccountName","user_f
ilter":"objectclass=*"},"Configurations":[]}]}}]]]' \

--service-role EMR_DefaultRole --name 'SecurityPOCCluster' --region cn-northwest-1
```

HUE 所用到的 LDAP 相关配置是使用 `-configurations` 的选项传递的，具体可以参考 [Configure Hue for LDAP Users](http://docs.aws.amazon.com/ElasticMapReduce/latest/ReleaseGuide/hue-ldap.html) (<http://docs.aws.amazon.com/ElasticMapReduce/latest/ReleaseGuide/hue-ldap.html>) 和 EMR `create-cluster` CLI 参考 (<https://docs.aws.amazon.com/cli/latest/reference/emr/create-cluster.html>)。

EMR 的 steps 执行了如下操作：

- 安装和配置 Ranger HDFS 和 Hive 插件
- 使用 Ranger REST API 来更新资源库和认证策略。（注意：该 step 只需要在最初执行一次，以后新建的 EMR 集群无需包含此 step）
- 创建 Hive 表（tblAnalyst1 和 tblAnalyst2），并拷贝样本数据
- 创建 HDFS 目录（/user/analyst1 和 /user/analyst2），并拷贝样本数据
- 使用 spark 提交动作来运行一个 SparkPi 任务以验证集群安装设置。

为验证所有 step 操作已执行成功，可以在 EMR 集群中查看 Steps 部分，如下图：

Steps						
Filter:		All steps	Filter steps ...	5 steps (all loaded)		
	ID	Name	Status	Start time (UTC+8)	Elapsed time	Log files
▶	s-3RY22J9IKAWIY	InstallRanger Policies	Completed	2019-06-23 17:48 (UTC+8)	6 seconds	View logs
▶	s-3LC6A6OF8WBGK	InstallRanger Plugin	Completed	2019-06-23 17:47 (UTC+8)	56 seconds	View logs
▶	s-35S6LH0QSWLUX	LoadHDFS Data	Completed	2019-06-23 17:46 (UTC+8)	24 seconds	View logs
▶	s-2WVTADB0EJN2C	SparkStep	Completed	2019-06-23 17:46 (UTC+8)	36 seconds	View logs
▶	s-3H81XHTEITDOL	CreateHiveTables	Completed	2019-06-23 17:45 (UTC+8)	38 seconds	View logs

注意：集群的创建可能需要 10 到 15 分钟。

测试集群

祝贺您！您已经成功的配置了 EMR 集群，并能够使用 Ranger 来管理认证策略。具体是怎么样工作呢？你可以测试一下 HDFS 和 Hive 查询。

访问 Web 界面

我们需要访问 Range 管理界面和 Hue 来完成测试，你可以使用以下链接：

1. Ranger 管理界面：<http://<Ranger服务器地址>:6080/login.jsp>
2. Hue 的网页界面：<http://<EMR Master IP>:8888>

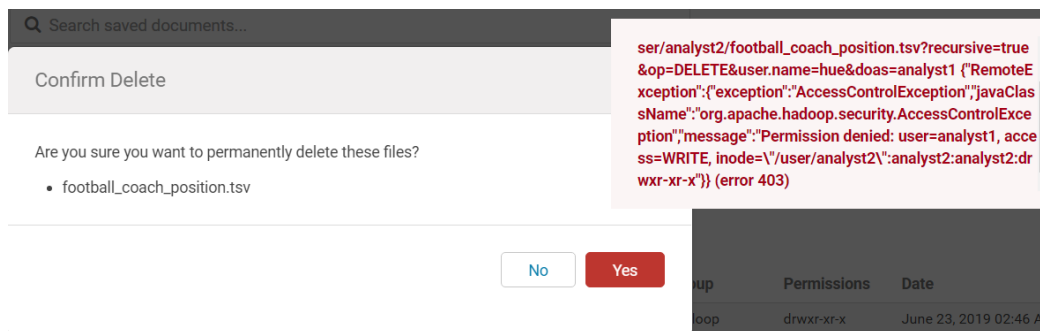
如果你的 EMR 是运行在私有子网，请参考 blog (<https://aws.amazon.com/blogs/big-data/securely-access-web-interfaces-on-amazon-emr-launched-in-a-private-subnet/>) 来访问网页界面。同样的步骤可以适用于访问 Ranger 给管理界面。

使用 HDFS

使用 analyst1 用户登陆 Hue，尝试删除一个 analyst2 所拥有的文件。若想了解如何访问 Hue，请参考 Launch the Hue Web Interface

(<https://docs.aws.amazon.com/ElasticMapReduce/latest/ReleaseGuide/accessing-hue.html>)。

1. 使用 analyst1 登陆 Hue（密码: Had00p@User1）
2. 打开/user/analyst2 HDFS 目录，删除文件 football_coach_position.tsv。
3. 你应该看到 “Permission denied” 的错误，这是符合预期的。



使用 Hive 查询

使用 HUE SQL Editor 来执行以下查询。

这些查询使用外表，Hive 利用 EMRFS 来访问存储在 S3 上的数据。因为 HiveServer2（Hue 提交查询的目标）在访问 S3 的任何数据之前就通过与 Ranger 检查来确定是授权还是拒绝，所以你可以创建精细化基于 SQL 语句的权限赋予用户，尽管该集群被指定了一个单独的 EC2 角色（该角色被用作于所有访问 S3 的请求）。若了解详情，请参看 Additional Features of Hive on Amazon EMR (<http://docs.aws.amazon.com/ElasticMapReduce/latest/ReleaseGuide/emr-hive-additional-features.html>)。

```
SELECT * FROM default.tblanalyst1
```

这语句会按照预期返回结果。现在，运行以下语句：

```
SELECT * FROM default.tblanalyst2
```

你会看到以下错误：

```
Error while compiling statement: FAILED: HiveAccessControlException Permission denied: user [analyst1] does not have [SELECT] privilege on [default/tblanalyst2/*]
```

这是合理的。用户 analyst1 没有表 tblanalyst2 的 SELECT 权限。

反之，用户 analyst2 如果去访问表 tblanalyst1 的时候，也会遇到类似错误。用户 admin1 可以运行任何查询，因为它拥有管理员权限。

更新安全策略

你已经验证了策略被正确执行了。现在让我们来更新它们。

1. 登陆到 Ranger 管理界面
 - a. <http://<Ranger>服务器地址>:6080/login.jsp>
 - b. 缺省的用户名密码是：admin/admin。
2. 选择 hivedev 来查看所有 Raner 中 Hive 策略

Service Manager

hivedev Policies

Access











Masking

Row Level Filter

List of Policies : hivedev



Search for your policy...

Add New Policy

Policy ID	Policy Name	Status	Audit Logging	Groups	Users	Action
2	all - database, table, column	Enabled	Enabled	--	polycmgr_hive	 
3	all - database, udf	Enabled	Enabled	--	polycmgr_hive	 
6	Analyst1Policy	Enabled	Enabled	--	analyst1	 
7	Analyst2Policy	Enabled	Enabled	--	analyst2	 
8	Admin1Policy	Enabled	Enabled	--	admin1	 

3. 选择策略 Analyst2Policy
4. 编辑该策略，增加 analyst1 用户在表 tblanalyst2 上的 SELECT 权限

Allow Conditions :

Select Group	Select User	Permissions	Delegate Admin	
<div>xx</div> <div>xx</div> <div>xx</div> <div>Select Group</div>	<div>xx</div> <div>xx</div> <div>xx</div> <div>analyst2</div>	<div>All</div> <div></div>	<div><input type="checkbox"/></div>	<div></div>
<div>xx</div> <div>xx</div> <div>xx</div> <div>Select Group</div>	<div>xx</div> <div>xx</div> <div>xx</div> <div>analyst1</div>	<div>select</div> <div></div>	<div><input type="checkbox"/></div>	<div></div>
<div>+</div>				

5. 保存修改

这个策略修改会被 EMR 上的 Hive 插件拉过去。等待至少 60 秒后，该策略就会生效了。

回到刚才所做的 Hue 测试，看一下修改是否已经生效。

1. 以用户 analyst1 登陆 Hue
2. 在 Hive SQL Editor 中，运行之前失败的查询：

`SELECT * FROM default.tblanalyst2`

现在查询应该运行成功了。

	tblanalyst2.request_begin_time	tblanalyst2.ad_id	tblanalyst2
1	2009-04-12 13:59:53	AAv8arUW6Uw8HsXfssxearjVRbIOU9	00c9KeVGr
2	2009-04-12 13:43:19	S15U6hxbUmrFNdJvQLi9KtQDRlwko	00jUJRp8o:

列屏蔽和行过滤

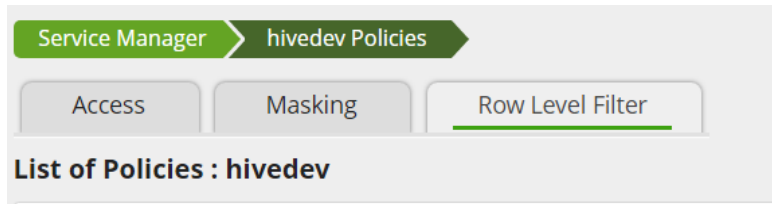
Apache Ranger 提供了列屏蔽和行过滤的功能。

假设我们想允许用户 analyst1 只能看到表 tblanalyst1 中的一些行并且屏蔽其中一个列的值。我们可以按照以下步骤来设置：

1. 登陆到 Ranger 管理界面
 - a. <http://<Ranger>服务器地址>:6080/login.jsp>
 - b. 缺省的用户名密码是：admin/admin。
2. 在 Service Manager 界面选择 hivedev

行级过滤

- 在页面上部选择 “Row Level Filter”，选择 “Add New Policy”



- 以下列参数来创建一个新策略
 - Policy Name: analyst1filer
 - Hive Database: default
 - Hive Table: tblanalyst1

Policy Details :

Policy Type **Row Level Filter**

Policy Name * **enabled**

Hive Database *

Hive Table *

Audit Logging **YES**

Description

- 在 Row Filer Conditions 下 :
 - Select User: analyst1
 - Access Type: select
 - Row Level Filter: page='yelp.com'

Row Filter Conditions :

Select Group	Select User	Access Types	Row Level Filter
<input type="text" value="Select Group"/>	<input type="text" value="x analyst1"/>	<input type="button" value="select"/> <input type="button" value="edit"/>	<input type="text" value="page='yelp.com'"/> <input type="button" value="edit"/>
<input type="button" value="x"/>			

- 选择 "Add" 来启用插件

列屏蔽

- 在相同的 “hivedev” 策略下，选择标签页 “Masking”
- 选择 “Add New Policy”
- 以下列参数来创建一个新策略
 - Policy Name: analyst1mask
 - Hive Database: default
 - Hive Table: tblanalyst1
 - Hive Column: request_begin_time

Policy Details :

Policy Type	Masking
Policy Name *	analyst1mask enabled
Hive Database *	× default
Hive Table *	× tblanalyst1
Hive Column *	× request_begin_time
Audit Logging	YES
Description	Mask column for user <u>analyst1</u>

- 在 Mask Conditions 下：
 - Select User: analyst1
 - Access Type: select
 - Select Masking Option: Partial mask: show first 4

Mask Conditions :

Select Group	Select User	Access Types	Select Masking Option	
<div>⋮ Select Group</div>	<div>× analyst1</div>	<div>select </div>	<div>Partial mask: show first 4 </div>	<div>×</div>
<div>+</div>				

- 选择 “Add” 来启用插件。

这个策略修改会被 EMR 上的 Hive 插件拉过去。等待至少 60 秒后，该策略就会生效了。

回到刚才所做的 Hue 测试，看一下修改是否已经生效。

1. 以用户 analyst1 登陆 Hue
2. 在 Hive SQL Editor 中，运行之前的查询：

```
SELECT * FROM default.tblanalyst1
```

现在查询应该运行成功并且只显示 page 为 yelp.com 的行。列 request_beging_time 应该只显示前 4 个字符。

Query History 🔍 📅 Saved Queries 🔍 Query Builder Results (100+) 🔍 📄				
	tblanalyst1.request_begin_time	tblanalyst1.ad_id	tblanalyst1.impression_id	tblanalyst1.page
1	2009-xx-xx xx:xx:xx	JL77gwkPJxno6llCtF98wMEumeGOEX	0nXQCJqoTjQkql6K6gxnTE6j4X2hmJ	yelp.com
2	2009-xx-xx xx:xx:xx	pkkup4TlqVR75ph1s6QHSrac7o1chn	0plADQJQcwbICwPJVEE2cV2Su5nfbr	yelp.com
3	2009-xx-xx xx:xx:xx	h0sBxhbUSkttDhe4Q6itkhFFHLsJfb	14N93keNXWUu6Xu62QDiGogs9ekDjm	yelp.com
4	2009-xx-xx xx:xx:xx	Ck1ukPPvXkPX6TeAMBLsgRRijFSmrh	1A6JRQsrTLDCEPMFmjdeJH6Kpmj8lh	yelp.com
5	2009-xx-xx xx:xx:xx	mSqbxhHBBbT7wAMXwUjF7c7uuFKuan	1f2Lwx9C9Jpw5PDEoCtpxM6C6TJgRs	yelp.com

审计

现在你能发现谁曾经试图访问过 Hive 表，并且是被拒绝或者允许的吗？

1. 登陆到 Ranger 管理界面
 - a. <http://<Ranger>服务器地址>:6080/login.jsp>
 - b. 缺省的用户名密码是：admin/admin。
2. 选择 Audit，并按照 analyst1 过滤
 - o 用户 analyst1 在访问表 tblanalyst2 时被拒绝过：

Access	Admin	Login Sessions	Plugins	Plugin Status
🔍 START DATE: 06/23/2019 USER: analyst1 RESULT: Denied ⓘ				

Policy ID	Event Time ▼	User	Service	Resource	Access Type	Result	Access Enforcer
			Name / Type	Name / Type			
...	06/23/2019 11:13:01 PM	analyst1	hivedev hive	default/tblanalyst2/ad_id @column	SELECT	Denied	ranger-acl
...	06/23/2019 10:32:31 PM	analyst1	hadoopdev hdfs	/user/analyst2/football_coach_position.tsv path	WRITE	Denied	hadoop-acl

当策略更改后，访问被允许了，并且也被记录了。

7	06/23/2019 11:29:15 PM	analyst1	hivedev hive	default/tblanalyst2/ad_id,clicked,day,hour,im... @column	SELECT	Allowed	ranger-acl
---	------------------------	----------	-----------------	-------------------------------------------------------------	--------	---------	------------

这些同样的审计信息也保存在 SOLR，以便执行更加复杂和完整的搜索。这 SOLR 是被安装在与 Ranger 相同的服务器上。

- 打开 Solr 界面：
http://<Ranger服务器地址>:8983/solr/#ranger_audits/query
- 执行一个文档搜索

结论

在本文中，我们一步步完成了在 EMR 上使用 Apache Ranger 来启用认证和审计功能。我们也通过 CloudFormation 模板达到了自动化。

如果你有任何问题或建议，欢迎赐教。

附录

本文中所用到的软件和脚本，你可以在北京区或者宁夏区的 S3 上取得：

`aws s3 sync s3://bigdata-cn-north-1/artifacts/ .`

`aws s3 sync s3://bigdata-cn-northwest-1/artifacts/ .`