1 Introduction

Apache Ranger provides centralized security for Enterprise Hadoop ecosystem, including fine-grained access control and centralized auditing. Current version of Apache Ranger, 0.5, supports authorization policies that can allow access to resources when the specified conditions are met – conditions like user/groups, access-type and custom conditions. However, the model does not support policies that can explicitly deny access to resources. Also, the model does not support policies that can allow (or deny) access to a wider group (like employees, everyone) but exclude specific users/groups who might be part of the wider group.

Apache Ranger policy model has been enhanced, for Apache JIRA RANGER-606, with capability to explicitly deny access and specify exceptions (excludes) to both allow and deny conditions. This document covers various details of these enhancements, along with few examples.

2 Policy Model

Apache Ranger policy consists of two major parts:

1. Specification of resources for which the policy is applicable for – resources like HDFS files/directories, Hive database/tables/columns, HBase tables/columns-families, columns, etc.
2. Specification of conditions, like users/groups, access-types and custom-conditions, for which the access should be allowed

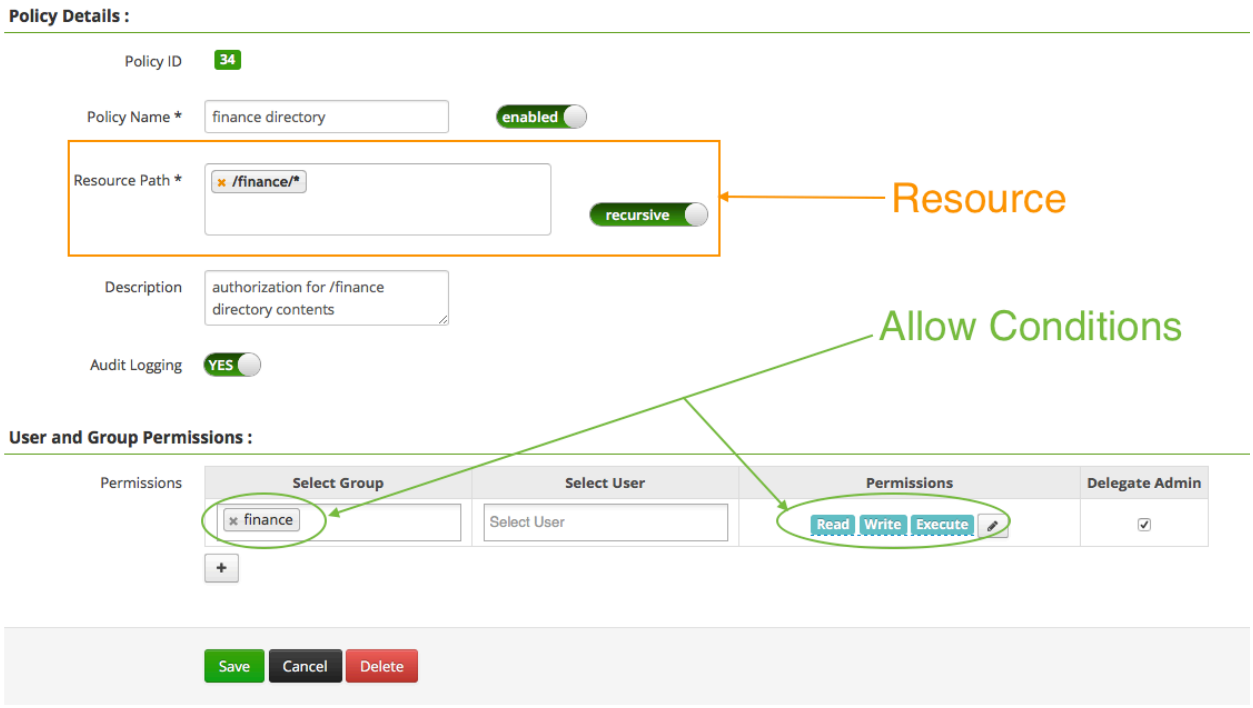
While the first part above remains unchanged with these enhancements, the policy model has been updated to support 4 categories of conditions (second part above), as listed below:

1. Allow (already exists in the current and earlier releases)
2. Deny
3. Allow Exceptions
4. Deny Exceptions

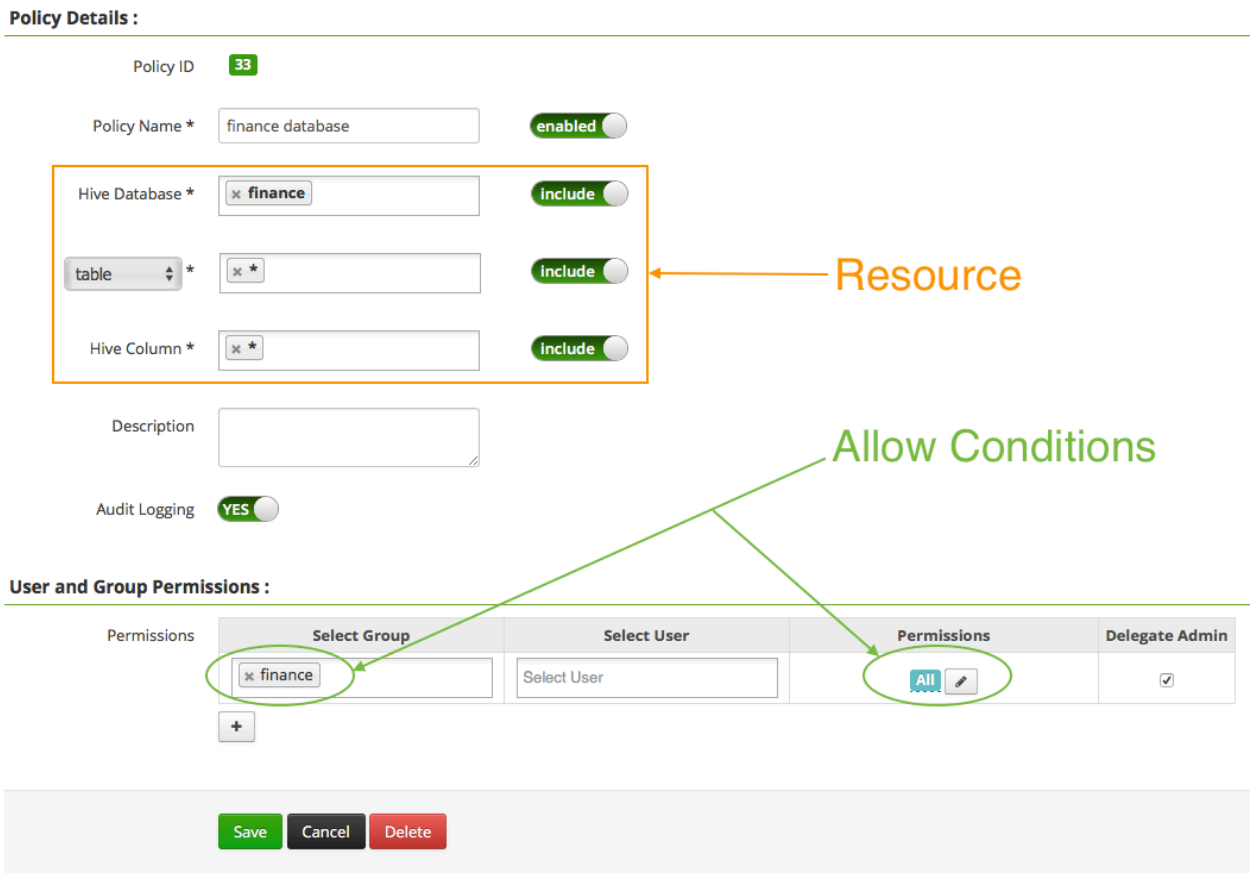
2.1 Apache Ranger 0.5 Policies

Apache Ranger policy model in releases 0.5 and earlier, supports policies that can explicitly allow access to resources. Following screenshots will help understand the details of couple of policies in Apache Ranger 0.5 version:

HDFS policy that allows all `finance` group users to access contents of `/finance` folder:



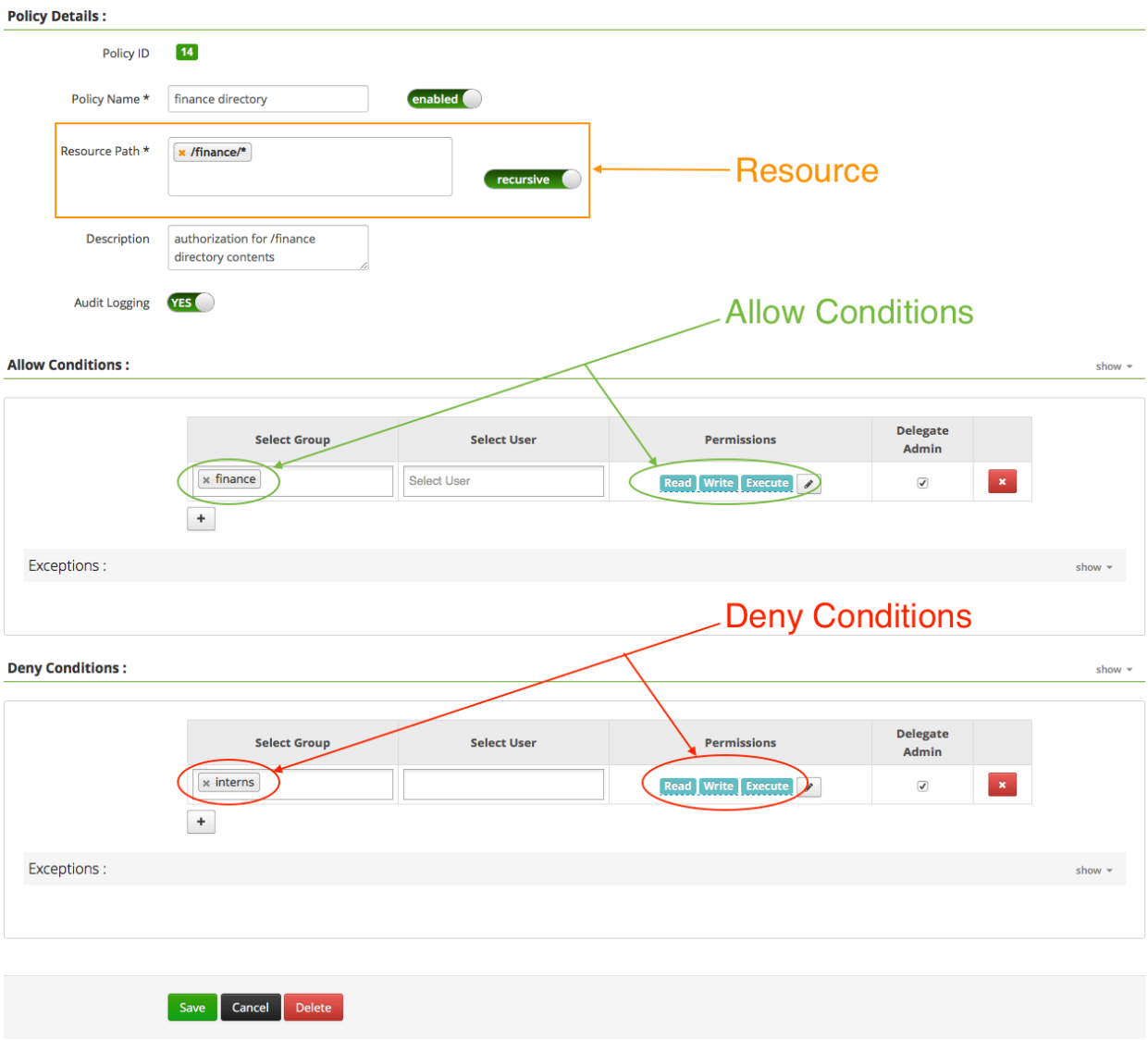
Hive policy that allows all `finance` group users to access contents of `finance` database:



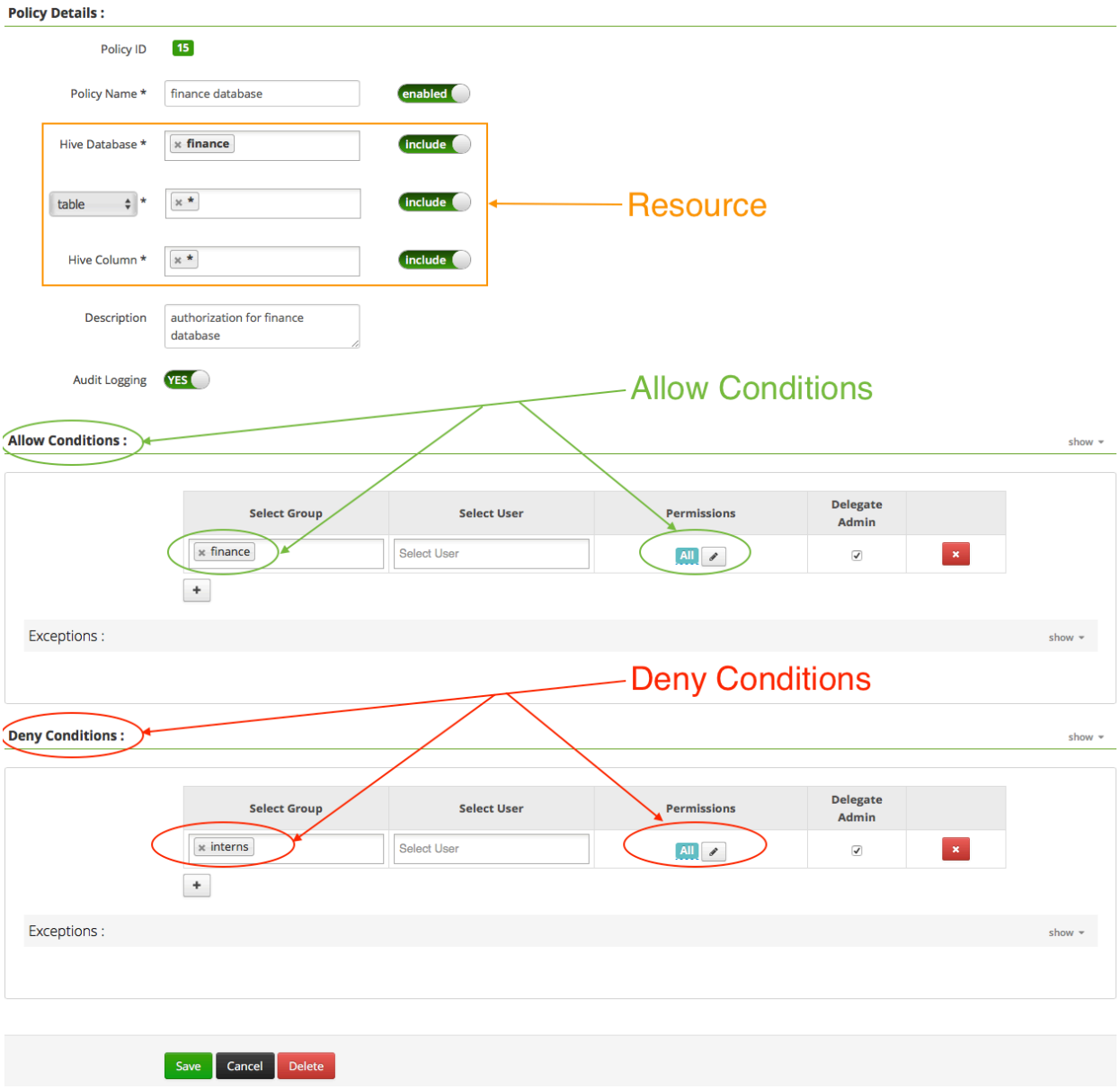
2.2 Enhanced Policy model

The policy model enhancements in RANGER-606 add the capability to explicitly deny access on the given conditions and also to specify exceptions to allow-conditions and deny-conditions. Let's use the same policies used in the previous section, but with an added condition to explicitly deny access to users in `interns` group.

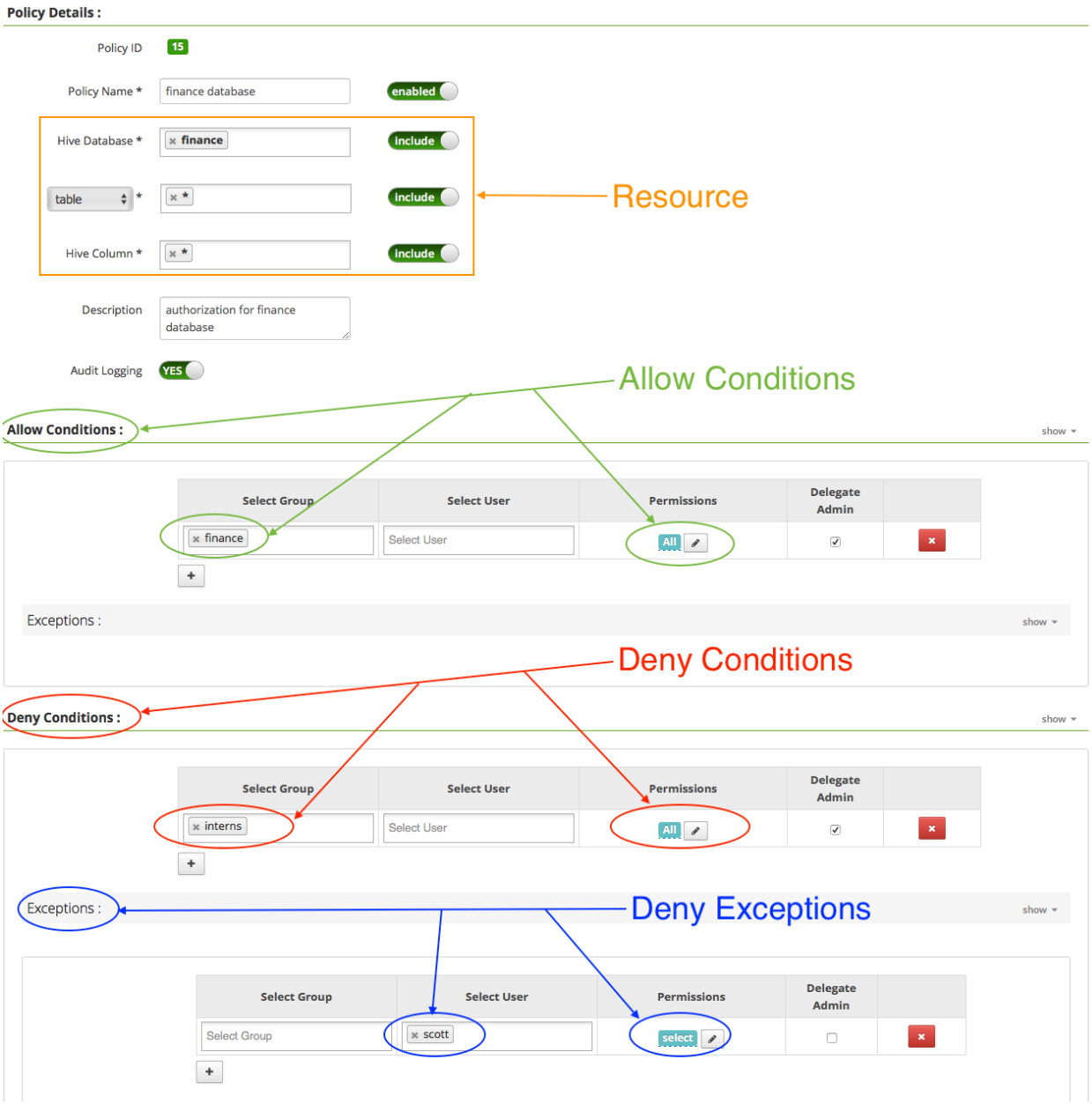
HDFS policy that allows all `finance` group users to access contents of `/finance` folder, but denies access to users in `interns` group. Users in `interns` group will be denied the access even if they are part of `finance` group.



Hive policy that allows all `finance` group users to access contents of `finance` database, but denies access to users in `interns` group. Users in `interns` group will be denied the access even if they are part of `finance` group.



Let's say one of the users in `interns` group, `scott`, is working on an assignment that requires `select` access to `finance` database . To support this, the authorization policy for the database should be updated by adding a deny-exception, as shown below:



3 Policy Evaluation

When the policy model supported only allow conditions, the order of policy evaluation did not impact the final result. Apache Ranger policy engine evaluated all policies for the resource until one of the policies allowed the access. When there is no policy to allow the access, the authorization request will typically be denied.

The introduction of deny conditions in the policy model requires the policies to be evaluated in a specific order to ensure that the final result is predictable. The policies for the accessed resource are evaluated by Apache Ranger policy engine in the following order:

- evaluate all deny-conditions
- when the request matches a deny-condition in a policy:
  - if the policy has no deny-exception or if the request does not match any deny-exception in the policy, the access will be **denied**
  - Else, continue evaluation of next deny-condition
- when the access is not denied by any deny-condition, evaluate all allow-conditions
- when the request matches an allow-condition in a policy:
  - If the policy has no allow-exception or if the request does not match any allow-exception in the policy, the access will be **allowed**
  - Else, continue evaluation of next allow-condition
- if no allow-condition matches the request, the access result will be **undetermined**. In this case, most components will deny the access. However, components like HDFS and YARN fallback to their native ACL to determine the access

4 Source code, build, install and upgrade

4.1 Build Apache Ranger

Apache Ranger policy model enhancement to support deny-conditions and exceptions is available in Apache branch named `tag-policy` (https://github.com/apache/incubator-ranger/tree/tag-policy). Instructions to build Apache Ranger from this branch from a Unix/Linux shell are given below. Please remember to set JAVA_HOME environment variable to appropriate value before executing these commands:

```
$ git clone git://git.apache.org/incubator-ranger.git
$ cd incubator-ranger
$ git checkout tag-policy
$ git pull
$ mvn clean compile package install assembly:assembly
```

Once the build completes, archive files containing the binaries should be available under target directory, as shown below:

```
$ ls -l target/*.tar.gz
target/ranger-0.5.0-admin.tar.gz
target/ranger-0.5.0-admin.tar.gz
target/ranger-0.5.0-hdfs-plugin.tar.gz
target/ranger-0.5.0-hive-plugin.tar.gz
...
```

4.2 Install Apache Ranger

Please follow the instructions available at Apache Ranger wiki page, to install Apache Ranger components (Admin, Usersync and plugins). Please make sure to build Apache Ranger using sources from `tag-policy` branch, as detailed in the previous section.

4.3 Upgrade existing Apache Ranger 0.5 deployment

To upgrade an existing Apache Ranger 0.5 version deployment, please do the following:

- stop ranger-admin and ranger-usersync applications
- backup existing database schemas used by Apache Ranger (for policy and audit stores)
- follow the instructions in the previous section ("Install Apache Ranger") to install Apache Ranger from `tag-policy` branch. While installing Apache Ranger components, make sure to specify existing database schema details in `install.properties`. This will upgrade the database schemas with necessary changes.

5 Backward compatibility

Once Apache Ranger admin is upgraded, it will be possible to create policies with deny-conditions and exceptions. However, please note that the plugins (HDFS/Hive/HBase/…) will need to be upgraded for the exceptions and deny-conditions to be effective. The plugins from previous versions will continue to work with upgraded Apache Ranger admin, but they will only process allow-conditions; all the exceptions and deny-conditions will be ignored by earlier version plugins.