

# Role Strategy Plugin

Adds a new role-based strategy to manage users' permissions.

## Plugin Information

View Role-based Authorization Strategy on the [plugin site](#) for more information.

Older versions of this plugin may not be safe to use. Please review the following warnings before using an older version:

- [CSRF vulnerability in security configuration](#)
- [Dangerous permissions can be configured independently of Administer permission](#)

## About this plugin

This plugin adds a new role-based strategy to ease and fasten users management. This strategy allows:

- Creating **global roles**, such as admin, job creator, anonymous, etc., allowing to set Overall, Slave, Job, Run, View and SCM permissions on a global basis.
- Creating **project roles**, allowing to set only Job and Run permissions on a project basis.
- Creating **slave roles**, allowing to set node-related permissions.
- Assigning these roles to users.

## Table of contents

- [About this plugin](#)
- [User guide](#)
  - [Getting started](#)
  - [Global Roles vs. Project Roles](#)
  - [Built-in Roles](#)
- [Macros support \(since 2.1.0\)](#)
  - [Available macros](#)
  - [Macro usage](#)
- [External add-ons](#)
- [Version history](#)
  - [Version 2.10 \(Feb 11, 2019\)](#)
  - [Version 2.9.0 \(Aug 27, 2018\)](#)
  - [Version 2.8.2 \(Aug 07, 2018\)](#)
  - [Version 2.8.1 \(May 25, 2018\)](#)
  - [Version 2.8.0 \(May 16, 2018\)](#)
  - [Version 2.7.0 \(Feb 06, 2018\)](#)
  - [Version 2.6.1 \(Oct 04, 2017\)](#)
  - [Version 2.6.0 \(Aug 28, 2017\)](#)
  - [Version 2.5.1 \(July 10, 2017\)](#)
  - [Version 2.5.0 \(Jun 02, 2017\)](#)
  - [Version 2.4.0 \(Apr 10, 2017\)](#)
  - [Version 2.3.2 \(06/13/2016\)](#)
  - [Version 2.3.0 \(06/07/2016\)](#)
  - [Version 2.2.0 \(06/29/2014\)](#)
  - [Version 2.1.0 \(07/20/2013\)](#)
  - [Version 1.1.3 \(07/10/2013\)](#)
  - [Version 1.1.2 \(10/14/2011\)](#)
  - [Version 1.1.1 \(09/19/2011\)](#)
  - [Version 1.1 \(06/08/2011\)](#)
  - [Version 1.0 \(09/20/2010\)](#)

## User guide

### Getting started

Using the plugin is fairly simple:

1. Activate the **Role-Based Strategy** by using the standard **Manage Jenkins > Configure System** screen:

Enable security

TCP port for JNLP slave agents

Fixed :   Random  Disable

Access Control

### Security Realm

---

Delegate to servlet container

LDAP

Hudson's own user database

Unix user/group database

### Authorization

---

Matrix-based security

Logged-in users can do anything

Legacy mode

Project-based Matrix Authorization Strategy

Anyone can do anything

Role-Based Strategy

2. Define and assign roles by using the **Manages Roles** item which appears in the **Manage Jenkins** screen:

## Manage Hudson



### [Configure System](#)

Configure global settings and paths.



### [Reload Configuration from Disk](#)

Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.



### [Manage Plugins](#)

Add, remove, disable or enable plugins that can extend the functionality of Hudson.



### [System Information](#)

Displays various environmental information to assist trouble-shooting.



### [System Log](#)

System log captures output from `java.util.logging` output related to Hudson.



### [Load Statistics](#)

Check your resource utilization and see if you need more computers for your builds.



### [Hudson CLI](#)

Access/manage Hudson from your shell, or from your script.



### [Script Console](#)

Executes arbitrary script for administration/trouble-shooting/diagnostics.



### [Manage Nodes](#)

Add, remove, control and monitor the various nodes that Hudson runs jobs on.



### [Manage Roles](#)

Handle permissions by creating roles and assigning them to users/groups

You then get following options:



# Manage and Assign Roles



[Manage Roles](#)  
Manage Roles



[Assign Roles](#)  
Assign Roles



[Role Strategy Macros](#)  
Provides info about macro usage and available macros

1. **Manage Roles** is the place where to set up roles:



## Manage Roles

### Global roles

Role	Overall		Slave		Job						Run			View			SCM	
	Administer	Read	Configure	Delete	Create	Delete	Configure	Read	Build	Workspace	Delete	Update	Artifacts	Create	Delete	Configure	Tag	
admin	<input checked="" type="checkbox"/>																	
anonymous	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
job-creator	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Role to add

Add

### Project roles

Role	Pattern	Job					Run		
		Delete	Configure	Read	Build	Workspace	Delete	Update	Artifacts
		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
		<input checked="" type="checkbox"/>							
		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Sample	Sample.*	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Role to add

Pattern

Add

Save

There's nothing much to say here, this is self-explanatory. The only tricky field is the **Pattern** one. This field consists in a regular expression aimed at matching the full name (including the folder name, if you're using Cloudbees Folders Plugin) of the jobs which the role will apply to. For example, if you set the field to "Roger-.\*", then the role will match all jobs which name starts with "Roger-". Note that the pattern is case-sensitive. To perform a case-insensitive match, use (?i) notation: upper, "Roger-.\*" vs. lower, "roger-.\*" vs. case-insensitive, "(?i)roger-.\*". If you have a nested folder structure where you want to provide the particular access to the second folder (or deeper), consider having a two-level security structure as well (Say you want to provide exclusive write/ modify type access to foo/bar and not everything else under "foo": First, assign that user/ group to read/ discover permissions with pattern "^foo.\*", then assign that same user/ group to the more particular permissions with pattern "^foo/bar.\*" - Similar to what you'd do in a Unix/ Linux environment.

2. **Assign Roles** is the place where to assign the defined roles to users:



## Assign Roles

### Global roles

User/group	admin	anonymous	job-creator	
██████████	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="✖"/>
██████████	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="✖"/>
██████████	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="✖"/>
██████████	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="✖"/>
██████████	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="✖"/>
Anonymous	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="✖"/>

User/group to add

Add

### Project roles

User/group	████	████	Sample	████	
██████████	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="✖"/>
Anonymous	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="✖"/>

User/group to add

Add

Save

## Global Roles vs. Project Roles

It should be noted that the Global Roles **override** anything you specify in the Project Roles. That is, when you give a role the right to Job-Read in the Global Roles, then this role is allowed to read **all** Jobs, no matter what you specify in the Project Roles.

It may therefore be advisable to leave most (all) options unchecked in Job, Run and SCM in the Global Roles section for normal users.

## Built-in Roles

There are two built-in roles:

\*Anonymous\* — Users who have not logged in

**authenticated** — Logged in users

## Macros support (since 2.1.0)

Macros allow to extend analysis of user's access rights (see @RoleMacroExtension). If user's sid meets criteria in Roles and Assignments, then analysis will be propagated to extension, which makes decisions according to instance and parameters.

## Available macros

You can get list of available macros and theirs descriptions at the `JENKINS_URL/role-strategy/list-macros` page. At the current state, plugin has minimal set of available macros, but they can be added by extensions from plugins.

Known macros:

- Built-in: @BuildableJob
- Ownership plugin: Ownership-based roles: @Owner and @CoOwner (will be released in ownership-0.2)

## Macro usage

Macros can be used in the following fields:

- RoleMacros - name of the role
- UserMacros - Not supported yet
- Macro format: @macroName[id][(parameter1, parameter2, ...)]>
- macroName - name of the macro (see available macros in the table below)
- id - identifier of the macro. Technical parameter, which allows to use same macros for multiple patterns
- parameter - additional parameters. At the current state, they don't support variables or TokenMacros

Macro string examples:

- @BuildableJob - Primitive macro invocation. Such invocation can be used only once in each roles category.
- @BuildableJob:1 - Macro with id
- @ParameterizedMacro(param1) - Invokes macro with one parameter
- @ParameterizedMacro:2(param1,param2) - Invokes macro with two parameters. Id prevents naming conflicts

## External add-ons

The management interface becomes difficult to use with a large number of users and/or roles. Several Greasemonkey userscripts exist to make the UI easier to use ([Jira issue](#)):

### Jenkins Role Strategy UI enhancer

This userscript adds a tooltip to the checkboxes indicating the row (e.g. user name) and column (e.g. permission).

**Rôles de projets**

Utilisateur/groupe	A-build-only	A-configure	A-read-only	B-build-only	B-read-only
<input checked="" type="checkbox"/> Anonyme	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> Ned Stark	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> Jaime Lannister	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> Robb Stark	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> Jon Snow	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

User: Ned Stark  
Role: A-configure

### Jenkins Role Strategy Role Management Enhancer and Jenkins Role Strategy Role Assignment Enhancer

These userscripts rotate the text in the column title cells on the Role Strategy configuration pages by 90 degrees so they use less horizontal space. Additionally, the first (header) column is repeated at the end of the table.

**Project roles**

User/group	A-build-only	A-configure	A-read-only	B-build-only	B-read-only	User/group
<input checked="" type="checkbox"/> Jaime Lannister	<input type="checkbox"/>	Jaime Lannister				
<input checked="" type="checkbox"/> Jon Snow	<input type="checkbox"/>	Jon Snow				
<input checked="" type="checkbox"/> Ned Stark	<input type="checkbox"/>	Ned Stark				
<input checked="" type="checkbox"/> Robb Stark	<input type="checkbox"/>	Robb Stark				
<input checked="" type="checkbox"/> Anonymous	<input type="checkbox"/>	Anonymous				

## Version history

**Version 2.10 (Feb 11, 2019)**

- Jenkins 2.60.3 is now the minimal requirement of the plugin
- [JENKINS-44472](#) - "Manage roles" table now supports preview of jobs matching the regular expression
- [PR #45](#) - REST API: *getRole* now also returns SID assignments
- [JENKINS-55804](#), [JENKINS-55803](#) - Improve performance of the plugin on instances with many roles
- [JENKINS-49102](#) - "Manage roles" page now displays patterns in quotes to properly visualize whitespace patterns
- [JENKINS-45942](#) - REST API: Throw error when a non-existent permission is added in the *addRole* call
- [JENKINS-54900](#) - REST API: Prevent concurrency issues when permissions are checked in parallel with REST API calls

Internal changes:

- [JENKINS-55916](#) - RoleWalker iterator now can be aborted by the handler
- [JENKINS-55933](#) - First version of the Role Strategy performance testing instance

## Version 2.9.0 (Aug 27, 2018)

- [PR #42](#) - Add REST API for retrieving particular roles
  - **Example:** `curl -XGET 'http://localhost:8080/jenkins/role-strategy/strategy/getRole?type=globalRoles&roleName=admin'`

## Version 2.8.2 (Aug 07, 2018)

- [PR #41](#) - *getAllRoles* REST API call was not closing output writer properly, and responses were missing in some cases
- [PR #40](#) - Prevent `NullPointerException` when *getAllRoles* REST API is called for non-existent role type

## Version 2.8.1 (May 25, 2018)

- [PR #39](#) - Prevent `NullPointerException` when adding role to an empty type from REST API

## Version 2.8.0 (May 16, 2018)

- [PR #37](#) - *getAllRoles* method now allows retrieving Job and Agent roles
  - **Example:** `curl -X GET localhost:8080/role-strategy/strategy/getAllRoles?type=globalRoles ("projectRoles" or "slaveRoles")`

## Version 2.7.0 (Feb 06, 2018)

- [PR #36](#) - Improve API to provide integration with [Configuration As Code Plugin](#)
  - [Configuration Example](#)

## Version 2.6.1 (Oct 04, 2017)

- [JENKINS-47265](#) - The plugin does not require extra dangerous permission enabler flags to be set with [Matrix Authorization Strategy Plugin 1.5+](#)
- [PR #33](#) - Improve diagnostics of invalid cases when Roles get created with null permissions

## Version 2.6.0 (Aug 28, 2017)

- [PR #30](#) - Add REST API endpoints to get and unassign roles
  - Examples:
    - **Unassign role:** `curl -X POST localhost:8080/role-strategy/strategy/unassignRole --data "type=globalRoles&roleName=AMD&sid=username"`
    - **List roles:** `curl -X GET localhost:8080/role-strategy/strategy/getAllRoles`
- Update Jenkins core minimal requirement to 1.625.3

## Version 2.5.1 (July 10, 2017)

- [Fix security issue](#)

## Version 2.5.0 (Jun 02, 2017)

- [JENKINS-37178](#) - Add REST API, which allows managing roles and assignments
  - Examples:
    - **Add Role:** `curl -X POST localhost:8080/role-strategy/strategy/addRole --data "type=globalRoles&roleName=ADM&permissionIds=hudson.model.Item.Discover,hudson.model.Item.ExtendedRead&overwrite=true"`

- **Remove Role(s):** `curl -X POST localhost:8080/role-strategy/strategy/removeRoles --data "type=globalRoles&roleNames=ADMIN,DEV"`
- **Assign Role:** `curl -X POST localhost:8080/role-strategy/strategy/assignRole --data "type=globalRoles&roleName=ADMIN&sid=username"`
- **Delete SID from all roles:** `curl -X POST localhost:8080/role-strategy/strategy/deleteSid --data "type=globalRoles&sid=username"`
- Parameters:
  - **Type:** *globalRoles, projectRoles, slaveRoles*
- [JENKINS-18377](#) - Improve speed of fetching roles by permission
- [JENKINS-43058](#) - Stop mentioning "slaves" in the plugin UI and Javadoc

## Version 2.4.0 (Apr 10, 2017)

This change is a part of the Security release in Jenkins.

- [SECURITY-410](#) - Prohibit dangerous permissions by default
  - Permissions like "Jenkins.RUN\_SCRIPTS" cannot be granted to non-admin users by default
  - After the upgrade to 2.4.0, such dangerous permission configurations will be disabled and reported in the Administrative Monitor
  - "[org.jenkinsci.plugins.rolestrategy.permissions.DangerousPermissionHandlingMode.enableDangerousPermissions](#)" system property can be used to allow these dangerous permissions (**not recommended**)
  - See the referenced issue for more info
- Fixed escaping of descriptions in the Role Strategy Macros list ([JENKINS-38230](#))

### Warning!

After the update the dangerous permissions will be disabled, hence some Jenkins instances may require reconfiguration if they rely on dangerous configurations (e.g. RUN\_SCRIPTS without ADMINISTER)

## Version 2.3.2 (06/13/2016)

- Performance: Disable user authorities resolution in permission checks by default ([JENKINS-35515](#))
  - It has been done due to the reported performance degradation in 2.3.0
  - The 2.3.0 behavior can be restored by the `org.jenkinsci.plugins.rolestrategy.Settings.treatUserAuthoritiesAsRoles` system property
  - If you enable it, the performance can be also tweaked by `org.jenkinsci.plugins.rolestrategy.Settings.userDetailsCacheMaxSize` and `org.jenkinsci.plugins.rolestrategy.Settings.userDetailsCacheExpirationTimeSec`
- Authorities resolution: Catch Runtime Exceptions from underlying Security Realms. Prevents Jenkins DoS in such case ([JENKINS-35652](#))
- Generalize the help message for role patterns ([JENKINS-35250](#))

2.3.1 is skipped due to the typo in the property name

## Version 2.3.0 (06/07/2016)

- Threat user authorities as roles (<https://github.com/jenkinsci/role-strategy-plugin/pull/13>)
- Escape all form entry fields by default (prevent unintentional HTML injection by admins)
- Migration to the new Jenkins plugin parent POM
- Fixes of minor issues discovered by FindBugs

### WARNING!

There are performance regressions reported to this version. Upgrade only after testing

## Version 2.2.0 (06/29/2014)

- Support of Create Job permissions since jenkins-1.566 ([JENKINS-19934](#))
  - The permission requires the specific item name validation strategy, which should be selected in Jenkins global configuration
- Fixed help links in manage-roles pages ([JENKINS-15030](#))
- Slave permissions: Allow assignment of permissions, which don't belong to "Slave" group ([JENKINS-18978](#))

## Version 2.1.0 (07/20/2013)

- Added support of individual permission assignments for slave nodes ([JENKINS-18748](#))
- Added support of Macro roles ([JENKINS-18700](#))

## Version 1.1.3 (07/10/2013)

- Prevented exceptions in case of missing roles ([JENKINS-18648](#))
- Prevented exceptions in case of deleted Permissions
- Support of folders plugin ([JENKINS-17482](#))
- Upgraded to Jenkins 1.424

## Version 1.1.2 (10/14/2011)

- Implemented [JENKINS-9325](#): Permissions contributed by plugins can now be managed at the project roles level
- Upgraded to Jenkins 1.409

## Version 1.1.1 (09/19/2011)

- Fixed [JENKINS-8058](#): "<" and ">" characters were not supported in regular expression patterns

## Version 1.1 (06/08/2011)

- SCM permissions (e.g. Tag) can now be handled at the project roles level
- Improved UI to handle large installations:
  - Deletion buttons are now also displayed on the left of each table
  - When having table with more than 20 entries, a footer is now added which repeats header
  - It is now possible to edit already defined patterns by double-clicking on them in the **Project roles** table
- Fixed some typos
- Fixed some image display issues

## Version 1.0 (09/20/2010)

- Initial release