Jenkins

Security

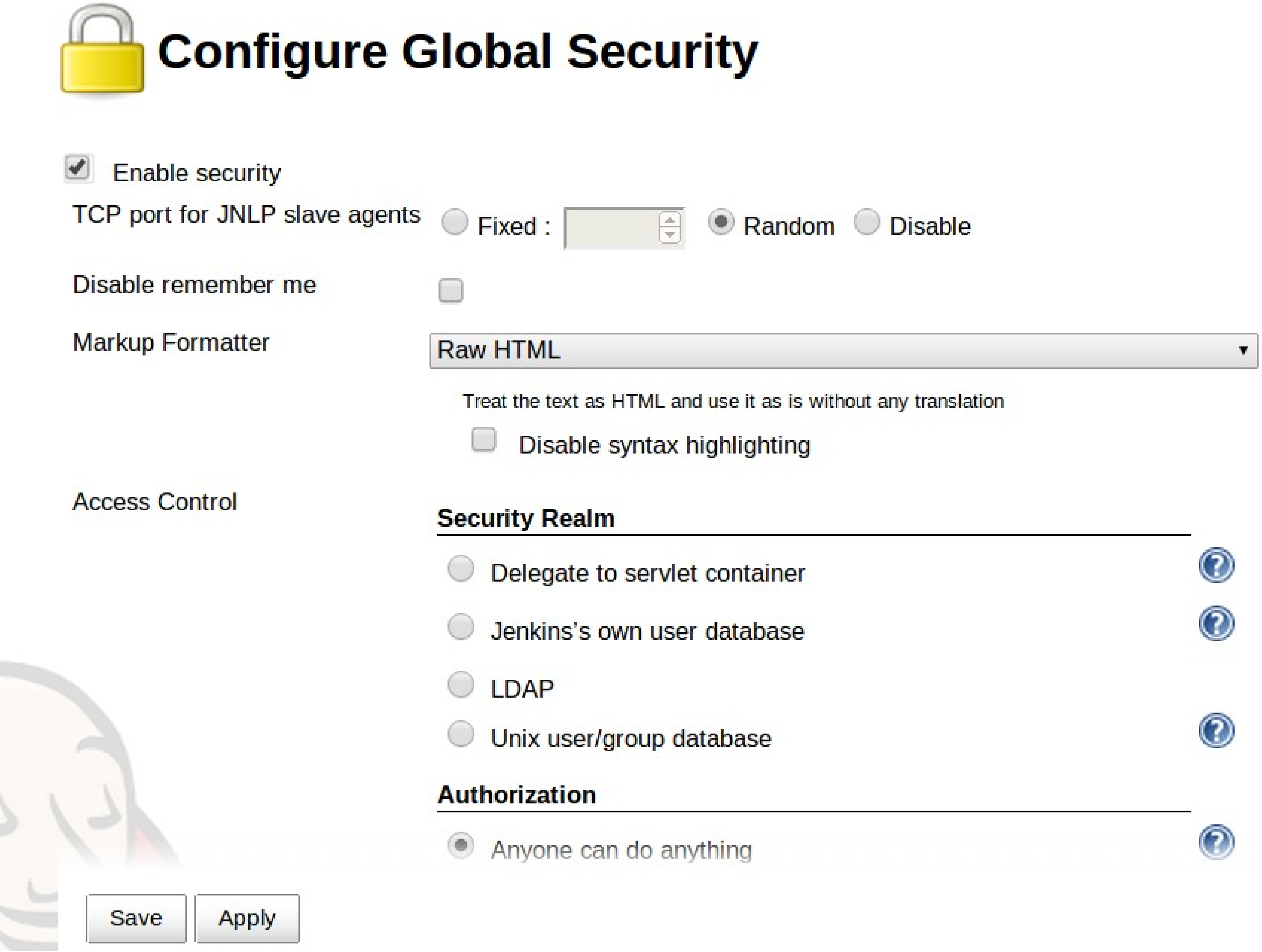
Section Contents

- Activating Security
- JNLP Port Management
- Markup Formatting
- Security Realms
- Setting up Simple Security

Activating Security

- Go to 'Manage Jenkins' and click on 'Enable Security'

Activating Security



Configure Global Security

☑ Enable security

TCP port for JNLP slave agents ○ Fixed : ● Random ○ Disable

Disable remember me □

Markup Formatter: Raw HTML

Treat the text as HTML and use it as is without any translation

□ Disable syntax highlighting

Access Control

Security Realm

○ Delegate to servlet container

○ Jenkins's own user database

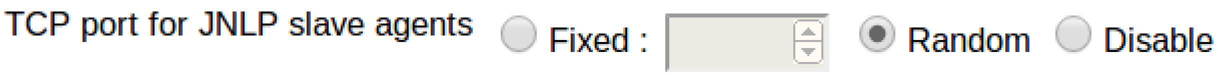
○ LDAP

○ Unix user/group database

Authorization

● Anyone can do anything

JNLP Port Management

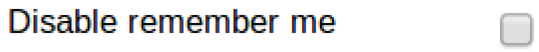


TCP port for JNLP slave agents ○ Fixed : ● Random ○ Disable

JNLP Port Management

- Gives the administrator the option to manage which port JNLP will be connected to
- Default is 'Random' for security purposes
- Although to effective work with a proxy server a fixed port may be required

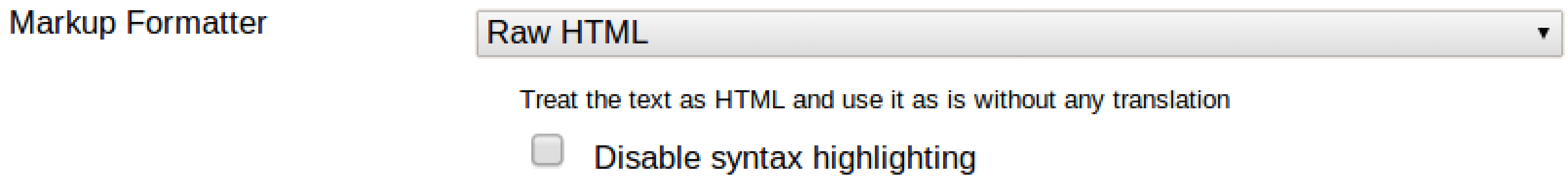
Disable Remember Me



Disable remember me ☐

- Disables Jenkins ability to remember usernames when logging in.

Markup Formatting



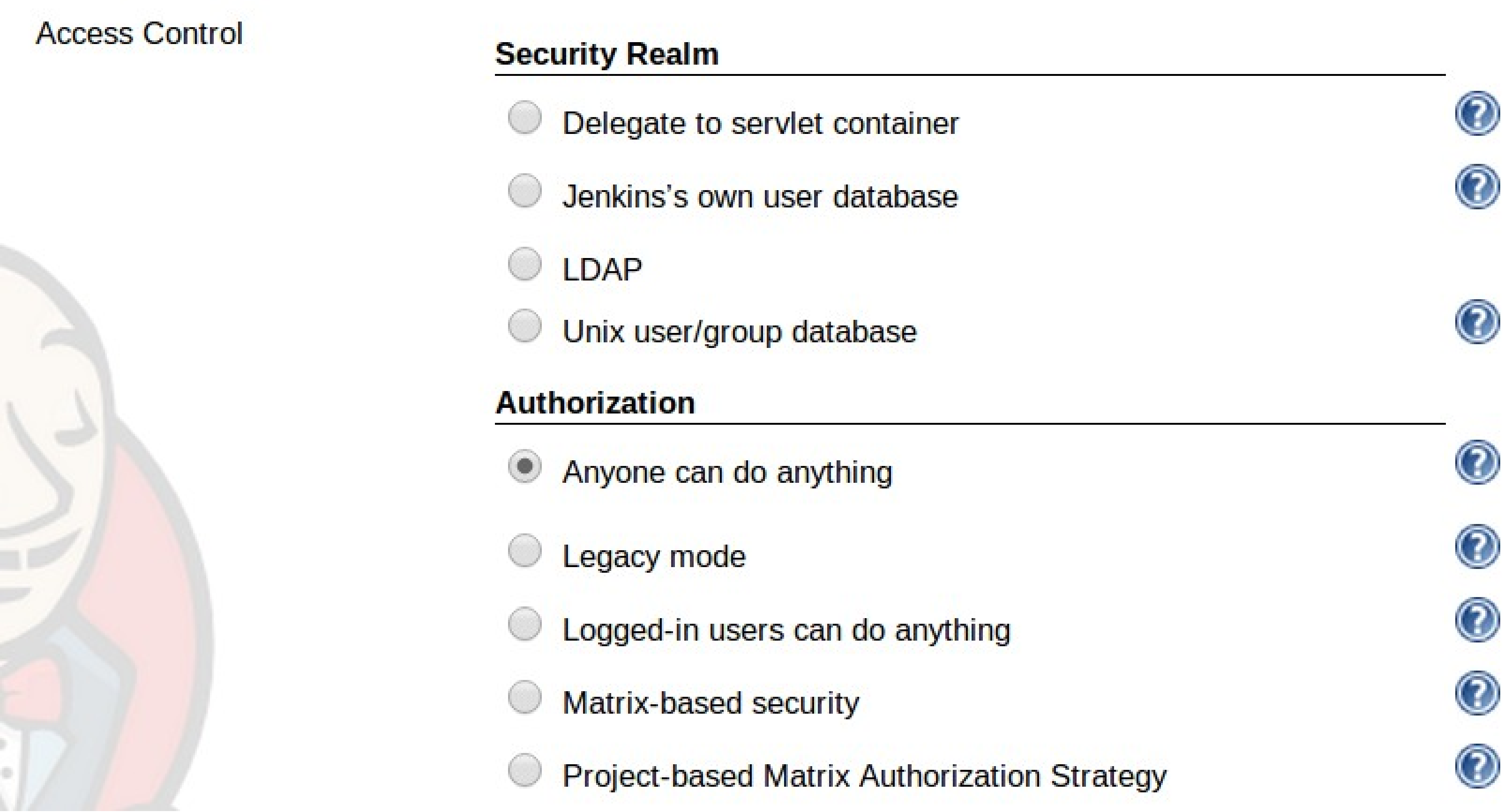
Markup Formatter | Raw HTML ▾

Treat the text as HTML and use it as is without any translation

□ Disable syntax highlighting

- Specify which markup formatting to use, if multiple renderers are available.
- Can disable formatting if it becomes annoying

Access Control



Access Control

Security Realm

- ○ Delegate to servlet container
- ○ Jenkins's own user database
- ○ LDAP
- ○ Unix user/group database

Authorization

- ● Anyone can do anything
- ○ Legacy mode
- ○ Logged-in users can do anything
- ○ Matrix-based security
- ○ Project-based Matrix Authorization Strategy

Security Realm

- Delegate to Servlet Container – allow username/password be defined by a java container
- Jenkins Own User Database – simple to use user database
- LDAP – LDAP Server Integration
- Unix User/Group Database – PAM integration

Authorization

- Anyone can do anything – signed in or not, the sky's the limit
- Legacy Mode – if "admin" then "total control" else "read only"
- Logged in Users can do anything – Anonymous users can still read though
- Matrix Based Security – Big Table and System Wide Detailed Security
- Project Based Authorization Strategy – Big Table and Per Project Detailed Security

Prevent Cross Site Request Forgeries

- Exploit that enables an unauthorized third party to take actions on a web site as you.
- Jenkins will check for a "crumb" or "ticket" to ensure that it any action is a "one time shot"

Setting up Simple Security

People vs. Manage Users

- Select "Jenkins Own User Database"
- The "People" section are the list of people that have committed to jobs or have been set up with security
- "Manage Users" in the "Manage Jenkins" section
  - are users that can log in to Jenkins
  - are a subset of the "People" section

Lab: Create a Login User

- Locate your username that was provided from subversion
- Create an account and password for the user name
- Verify that you can log in as that user

LDAP

LDAP

- Jenkins can
  - authenticate users using the LDAP repository
  - check group membership
  - retrieve the email address of authenticated users

Configuring LDAP

- Select "LDAP" from Security Realm
- Fill in the appropriate details about your LDAP server
- If you are using a non-standard port, you will need to provide this as well
- If you are using LDAPS, you will need to specify in the URL
- If LDAP has no anonymous binding Manager DN credentials will also need to be provided

Using LDAP with Security Matrix

- To use LDAP groups within a Security Matrix apply the word `ROLE_` to the LDAP group
- e.g. `ROLE_JENKINSADMIN`

Other Security Services

- Microsoft Active Directory
- Atlassian Crowd

Matrix Security

Matrix Security

- Detailed Security Management
- Different Users have different rights
- First you must create an Administrator before anyone else!
- The administrator doesn't have to be associated with a user who committed code
- Select "Matrix Based Security"

Matrix Security Setup

- Anonymous User Created Automatically
- Add the Administrator User to the Matrix and select every role
- Save and Log in

Lab: Setup Administrator Account

- Create an Administrator Account (administrator)
- Setup Matrix Security
- Add Administrator to the Matrix
- Select every role for the Administrator
- Log in to verify that you can login as an administrator

Matrix Permissions

Overall

- Basic System Wide Permissions
  - Administer - Lets a user make system-wide configuration changes.
  - Read – Read only access to all pages
    - For "anonymous" – able to view the jobs, not able to create or start the jobs
    - For all "authenticated" – create special user called "authenticated" and grant this permission

Slave (Nodes)

- Node Management and Creation Permissions
  - Create
    - Create and Configure Nodes
  - Delete
    - Delete Nodes

Job

- Create
  - Create New Jobs
- Delete
  - Delete Existing Jobs
- Configure
  - Configure Existing Jobs
- Read
  - View Existing Jobs
- Build
  - Start a Build Job
- Workspace
  - View and Download Workspace Contents
- Release
  - Run a Maven Release Plugin

Run

- Rights for the job records that have already run
- Delete - a build for the build history
- Update - update the description and other properties

View

- Rights managing Views
- Create – Create a new View
- Delete – Delete an existing View
- Configure – Configure an existing View

SCM

- Rights managing Version Control Systems
- Tag – Create a tag in the source code repository

Others

- Any other options based on plugin

I logged myself out

Solution 1: Turn Security Off

- Shutdown Jenkins
- Located the `config.xml` file in the `.jenkins` directory
- Locate the `<useSecurity>` element and change it to `false`
- Start up Jenkins Again
- Quickly reset up Security

Solution 2: Edit the full security profile

- Shutdown Jenkins
- Located the `config.xml` file in the `.jenkins` directory
- Update the Project Matrix with your name (see right)
- Start up Jenkins Again

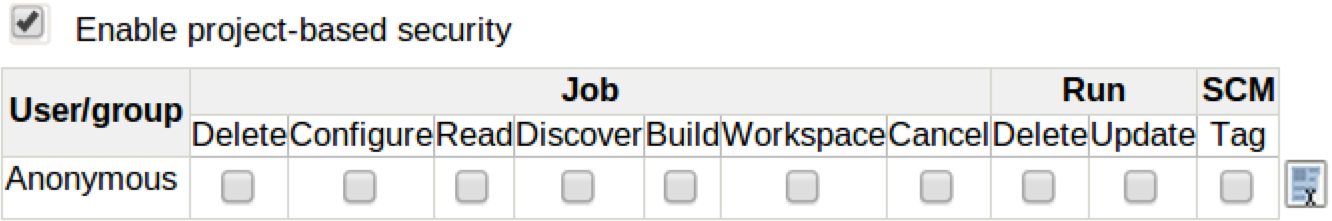
```
<authorizationStrategy class=”hudson.security.ProjectMatrixAuthorizationStrategy”>
  <permission>hudson.model.Computer.Configure:USERNAME</permission>
  <permission>hudson.model.Computer.Connect:USERNAME</permission>
  <permission>hudson.model.Computer.Create:USERNAME</permission>
  <permission>hudson.model.Computer.Delete:USERNAME</permission>
  <permission>hudson.model.Computer.Disconnect:USERNAME</permission>
  <permission>hudson.model.Hudson.Administer:USERNAME</permission>
  <permission>hudson.model.Hudson.Read:USERNAME</permission>
  <permission>hudson.model.Hudson.RunScripts:USERNAME</permission>
  <permission>hudson.model.Item.Build:USERNAME</permission>
  <permission>hudson.model.Item.Configure:USERNAME</permission>
  <permission>hudson.model.Item.Create:USERNAME</permission>
  <permission>hudson.model.Item.Delete:USERNAME</permission>
  <permission>hudson.model.Item.Read:USERNAME</permission>
  <permission>hudson.model.Item.Workspace:USERNAME</permission>
  <permission>hudson.model.Run.Delete:USERNAME</permission>
  <permission>hudson.model.Run.Update:USERNAME</permission>
  <permission>hudson.model.View.Configure:USERNAME</permission>
  <permission>hudson.model.View.Create:USERNAME</permission>
  <permission>hudson.model.View.Delete:USERNAME</permission>
  <permission>hudson.scm.SCM.Tag:USERNAME</permission>
</authorizationStrategy>
```

Project Matrix Security

- Same process as regular Matrix Security but per project
- Select “Project-based Matrix Authorization Strategy"
- Enter the default permissions as you would a regular matrix security profile
- Project Security can now be established in each project by selecting "Enable Project Based Security"

Project Matrix Security

- System wide Matrix overrides Project Security
- Typical setup is to give team members full access to your project, others read only



☑ Enable project-based security

User/group	Job							Run		SCM
	Delete	Configure	Read	Discover	Build	Workspace	Cancel	Delete	Update	Tag
Anonymous	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Lab: Give yourself access

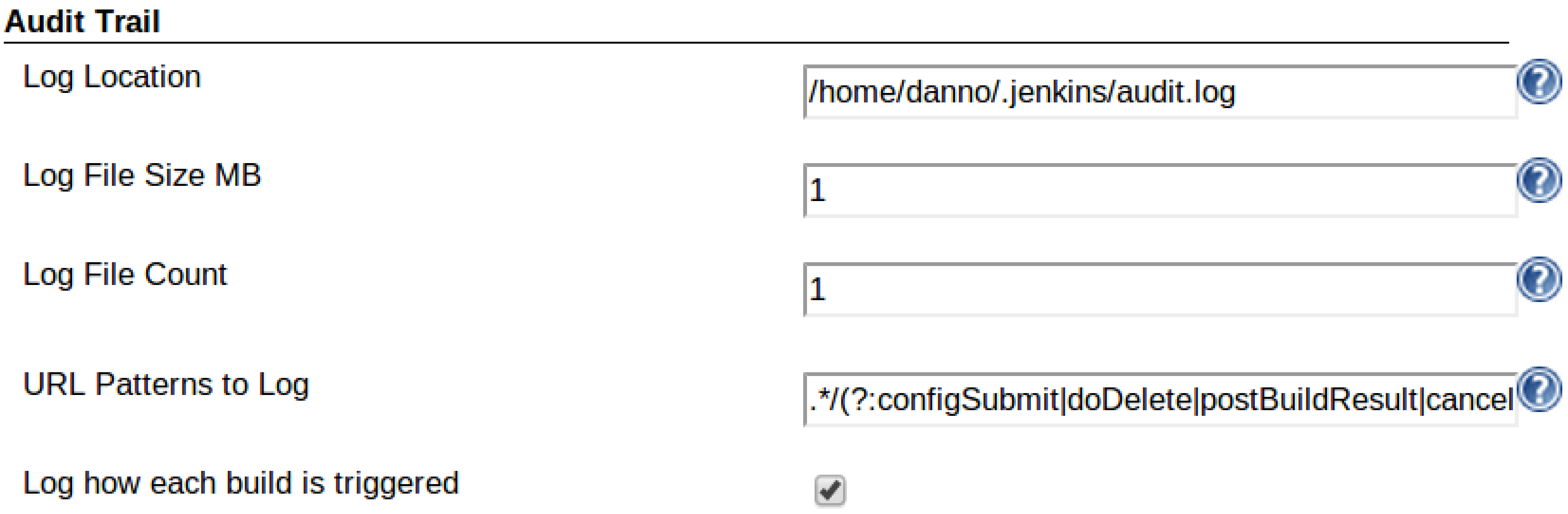
- Give yourself access to the simpleproject project
- Create a user of someone else in the class
- Ask them to log in and see if they can access certain parts of the project

Auditing Users

- To Audit Users you can use two Plugins:
  - Audit Trail
  - JobConfigHistory
- Both plugins can be installed from the Plugin Management

Auditing

Audit Trail Configuration



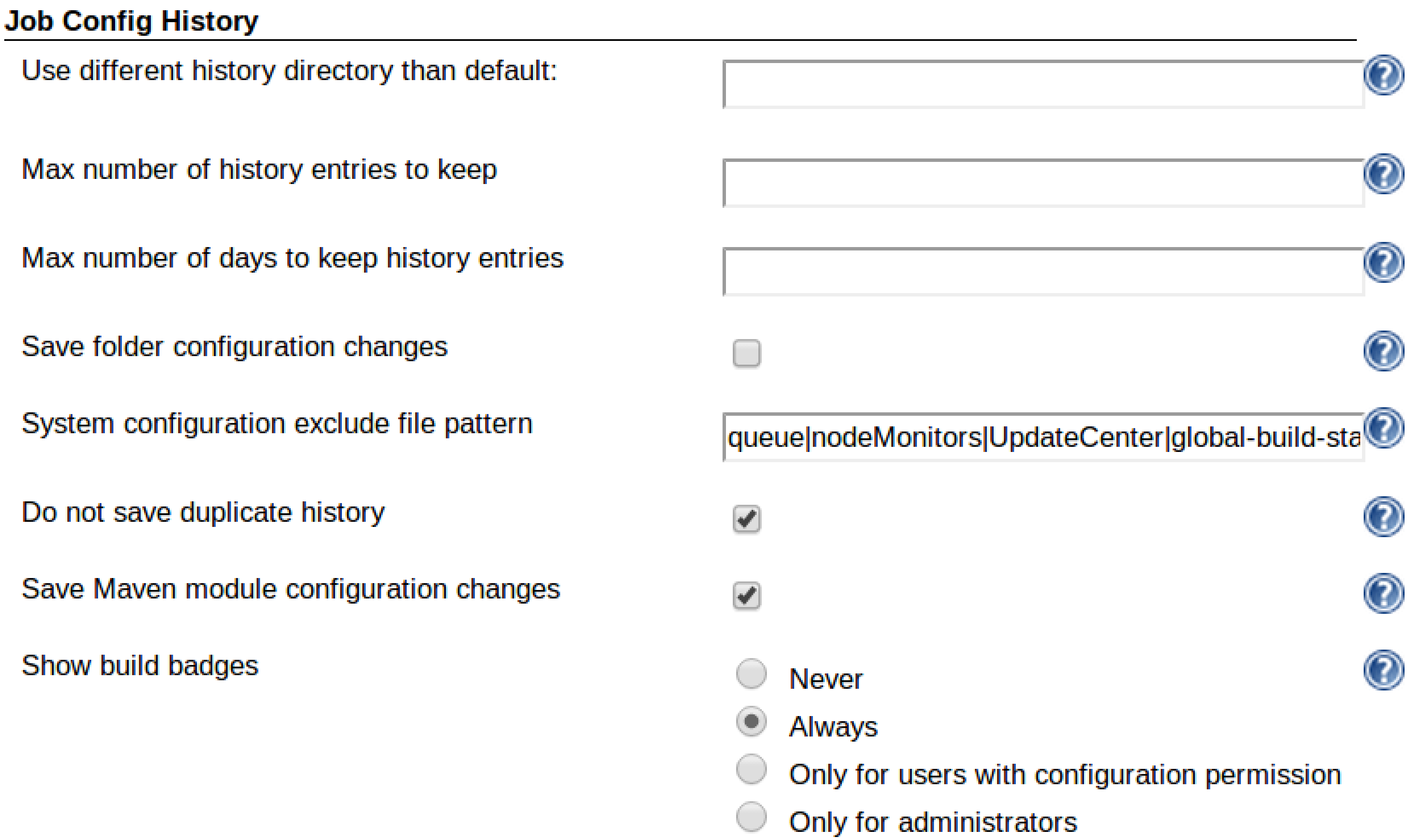
Audit Trail

Log Location	/home/danno/.jenkins/audit.log
Log File Size MB	1
Log File Count	1
URL Patterns to Log	.*/(?:configSubmit|doDelete|postBuildResult|cancel
Log how each build is triggered	☑

Audit Trail Configuration

- Audit trails produces log of Task and User
- Specify an absolute Log Location (default is `$JENKINS_HOME\audit.log`)
- Specify any size and number of files for the audit log
- Ensure that you have write access at said location
- Restart the Server
- Read the logs at the specified location

JobConfigHistory Configuration



Job Config History

Use different history directory than default:	
Max number of history entries to keep	
Max number of days to keep history entries	
Save folder configuration changes	☐
System configuration exclude file pattern	queue|nodeMonitors|UpdateCenter|global-build-sta
Do not save duplicate history	☑
Save Maven module configuration changes	☑
Show build badges	○ Never ● Always ○ Only for users with configuration permission ○ Only for administrators

JobConfigHistory Configuration

- Like Audit Trail but provides a fuller history
- Specify a root history folder either absolute or relative
- "Save System Configuration Changes" saves who changed configurations
- "Do not save duplicate history" will not save duplicates any time someone makes a change.

Lab: Auditing

- Setup both Audit Trail and JobConfigHistory Plugins

Thank you