

CheatSheet: Jenkins & Groovy

LANGUAGES

- PDF Link: [cheatsheet-jenkins-groovy-A4.pdf](#), Category: languages
- Blog URL: <https://cheatsheet.dennyzhang.com/cheatsheet-jenkins-groovy-A4>
- Related posts: [Jenkins CheatSheet](#), [#denny-cheatsheets](#)

File me Issues or star this repo.

1.1 Jenkins Pipeline

Name	Comment
Specify parameter to run jobs	<code>build job:'job1', parameters:[string(name:'name1', value:val)]</code>
Run job in different agents	<code>node(\$agent_label) {...}</code>
Ask for user input	<code>stage('stage2'){ input "OK to go?" }</code>
Actively fail current pipeline job	<code>error("Build failed because of this and that..")</code>
Check whether property exists	<code>if (env.keep_failed_env)</code>
Jenkins Pipeline enable timestamps	<code>options{timestamps()}</code>
Keep going when previous stage has failed	<code>keep-going-with-errors.groovy</code>
Send slack notification in pipeline	<code>slack-notification.groovy</code>
Pass parameter across jenkins jobs	<code>jenkinsfile-pass-parameter.groovy</code>
Set timeout & retry	<code>jenkinsfile-timeout-retry.groovy</code>
Use finally to do cleanup	<code>jenkinsfile-finally.groovy</code>
Run jenkins jobs in a sequential way	<code>jenkinsfile-sequentially.groovy</code>
Run jenkins jobs in parallel	<code>jenkinsfile-parallelly.groovy</code>
Reference	Link: Syntax Reference , Link: Jenkins User Documentation
Reference	Link: Example , Link: Example

1.2 Config Jenkins Via Groovy

Name	Comment
Set timezone for jenkins	<code>timezone.groovy</code>
Configure default view	<code>jenkins-views.groovy</code>
Configure Jenkins url	<code>jenkins-url.groovy</code>
Create a Jenkins user	<code>create-jenkins-user.groovy</code>
Groovy manages files/folders	<code>files-folder.groovy</code>
Configure max executors in Jenkins	<code>master-executors.groovy</code>
Config jenkins kubernetes plugin	<code>jenkins-kubernetes-cloud.groovy</code>
Configure slack plugin	<code>config-slack.groovy</code>
Reference	GitHub: cloudbees/jenkins-scripts , GitHub: jenkinsci/pipeline-examples

1.3 Jenkins Job Via Groovy

Name	Comment
List all my jenkins jobs	<code>println Jenkins.instance.projects.collect { it.name }</code>
List all jenkins jobs	<code>list-all-jobs.groovy</code>
Add a list of jobs by regexp to a view	<code>addjobstoview-byregexp.groovy</code>
Create jenkins views and add jobs to it	<code>jenkins-views.groovy</code>
Create and trigger a job	<code>create-jenkins-job.groovy</code>
Manage jenkins jobs	<code>manage-jenkins-jobs.groovy</code>
Cancel queued jenkins jobs by regexp	<code>kill-queued-jenkins.groovy</code>

1.4 Jenkins Security Via Groovy

Name	Comment
logged-in users can do anything	<code>logged-in-users.groovy</code>
Enable ldap in Jenkins	<code>enable-ldap.groovy</code>
Create a jenkins secret text	<code>create-secret-text.groovy</code>
Configure authorization in Jenkins	<code>matrix-authorization-strategy.groovy</code>
Jenkins skip wizzard when initialization	<code>-Djenkins.install.runSetupWizard=false</code>

1.5 Load Jenkins settings via folder copy

Name	Comment
Add default jobs	Copy jobs/ /usr/share/jenkins/ref/jobs/
Copy custom built plugins	COPY plugins/*.hpi /usr/share/jenkins/ref/plugins/
Use jenkins cli	COPY config/jenkins.properties /usr/share/jenkins/ref/
Add jenkins groovy scripts	COPY config/*.groovy /usr/share/jenkins/ref/init.groovy.d/
Configure Jenkins with some defaults	COPY config/*.xml /usr/share/jenkins/ref/
Install jenkins plugins	/usr/local/bin/install-plugins.sh < /usr/share/jenkins/ref/plugins.txt

1.6 Jenkins Plugins

Plugin	Summary
Kubernetes Plugin	Jenkins plugin to run dynamic agents in a Kubernetes/Docker environment
Credentials Plugin	Load the ssh key
SiteMonitor Plugin	Monitor URLs
Timestamper Plugin	Add timestamp to job output
Dashboard View Plugin	Create dashboard
Build-timeout Plugin	Abort if job takes too long
Naginator Plugin	Retry failed a job
ThinBackup Plugin	Backup jenkins
JobConfigHistory Plugin	Backup job configuration
Build User Vars Plugin	Describe the user who started the build

1.7 Jenkins Git Via Groovy

Name	Comment
Git checkout code	git-checkout.groovy
Get all git commits since last success	git-commits-before-fail.groovy
List git tags and branches	git-list-tags-and-branches.groovy

1.8 Jenkins networking Via Groovy

Name	Comment
Get hostname	println InetAddress.getLocalHost().canonicalHostName
Get IP address	println InetAddress.getLocalHost().hostAddress
Get hostname by ip	get-ip-by-hostname.groovy
validate user input: ip address	assert ip_address.matches("\\d{1,3}\\. \\d{1,3}\\. \\d{1,3}\\. \\d{1,3}")

1.9 Jenkins with Kubernetes/Docker

Name	Comment
Kubernetes Plugin	Jenkins plugin to run dynamic agents in a Kubernetes/Docker environment
Config jenkins kubernetes plugin	jenkins-kubernetes-cloud.groovy
Cleanup for Docker stale containers/images/volumes	docker-cleanup.groovy

1.10 Groovy Basic

Name	Comment
Get environment variables	get-env.groovy
Groovy execute command	execute-command.groovy
Print stdout	print.groovy echo 'Action is done', println "Hello World"
Use boolean parameter	if (istru == "false") {...}
Basic integer caculation	def a = 3, b = 7; println "\$a + \$b = \${a + b}"
Run groovy online	SaaS: Groovy Web console
Run groovy script from Jenkins	Link: Jenkins Script Console
Reference	Link: Apache Groovy

1.11 Groovy String/Regex

Name	Comment
Check string startsWith	<code>assert s.startsWith("\t")</code>
Trim whitespaces	<code>s=s.trim()</code>
Convert list to string	<code>l.join(",")</code>
Create string with multi-lines	<code>multi-line-string.groovy</code>
Convert string to list	<code>split-string.groovy</code>
Convert string to json	<code>string-to-json.groovy</code>
Remove tags	<code>input.replaceAll("\\<.*?>", "")</code>
Regex match	<code>regex-match.groovy</code>

1.12 Groovy Array

Name	Comment
Iterate a list	<code>for(item in [1,2,3,4]){ println item }</code>
Iterate a list	<code>(1..3).each { println "Number \${it}" }</code>
Add item to list	<code>def alist = [10, 9, 8]; alist << 7</code>
List size	<code>def alist = [10, 9, 8]; alist.size()</code>
Split string with delimiter	<code>'1128-2'.tokenize('-')</code>

1.13 Groovy File

Name	Comment
Read file content as a variable	<code>def env = System.getenv(), def content = readFile("/tmp/test.txt")</code>
Read and write json files	<code>json-file.groovy</code>
Obtain a relative path	<code>relative-path.groovy</code>

1.14 Groovy Dictionary

Name	Comment
Create a map	<code>def m = ['fruit':'Apple', 'veggie':'Carrot']</code>
Add an item to map	<code>m.put('denny', 'hello')</code>
Check if key exists	<code>m.containsKey('key1')</code>
Loop a map	<code>loop-map.groovy</code>

1.15 Groovy json

Name	Comment
Convert string to json	<code>string-to-json.groovy</code>
Convert dictionary to json	<code>dict-to-json.groovy</code>
Read and write json files	<code>json-file.groovy</code>

1.16 Groovy Date

Name	Comment
Date to string	<code>new Date().format("yyyy-MM-dd'T'HH:mm:ss'Z'", TimeZone.getTimeZone("UTC"))</code>
String to date	<code>Date.parse("yyyy-MM-dd'T'HH:mm:ss'Z'", "2001-01-01T00:00:00Z")</code>
String to date	<code>Date.parse("yyyy-MM-dd'T'HH:mm:ss'Z'", "2001-01-01T00:00:00+0000")</code>

1.17 Jenkins Agent

Name	Comment
Check jenkins slave jar version	<code>check-slave-jar-version.groovy</code>
Find dead executors and remove them	<code>find-dead-executors.groovy</code>
Set env for each agent	<code>set-agent-env.groovy</code>

1.18 Jenkins Maintenance

Name	Comment
Delete jenkins job by regexp	<code>delete-job-by-regexp.groovy</code>
Deploy Jenkins via docker	https://hub.docker.com/r/jenkins/jenkins/
Clean up old builds	Link: CloudBees Best Strategy for Disk Space Management

1.19 More Resources

<http://groovy-lang.org/documentation.html#gettingstarted>

<https://github.com/fabric8io/jenkins-docker>

License: Code is licensed under MIT License.