1 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	`build job:'job1', parameters:[string(name:'name1', value:va1)]`
Run job in different agents	`node($agent_label) {...}`
Ask for user input	`stage('stage2'){ input "OK to go?" }`
Actively fail current pipeline job	`error("Build failed because of this and that..")`
Check whether property exists	`if (env.keep_failed_env)`
Jenkins Pipeline enable timestamps	`options{timestamps()}`
Set envs within a jenkins pipeline	`withEnv(["key1=$var1"])`
Install plugin via groovy	`Hudson.instance.updateCenter.getPlugin(plugin).deploy().get()`
Keep previous job run via groovy	`buildDiscarder(logRotator(daysToKeepStr: '20', numToKeepStr: '60'))`
Keep going when previous stage has failed	keep-going-with-errors.groovy
Send slack notification in pipeline	slack-notification.groovy
Pass parameter across jenkins jobs	jenkinsfile-pass-parameter.groovy
Set timeout & retry	jenkinsfile-timeout-retry.groovy
Use finally to do cleanup	jenkinsfile-finally.groovy
Run jenkins jobs in a sequential way	jenkinsfile-sequentially.groovy
Run jenkins jobs in parallel	jenkinsfile-parallelly.groovy
Reference	Link: Syntax Reference, Link: Jenkins User Documentation
Reference	Link: Groovy Language Documentation
Reference	Link: Example, Link: Example

1.2 Config Jenkins Via Groovy

Name	Comment
Set timezone for jenkins	timezone.groovy
Configure default view	jenkins-views.groovy
Configure Jenkins url	jenkins-url.groovy
Create a Jenkins user	create-jenkins-user.groovy
Groovy manages files/folders	files-folder.groovy
Configure max executors in Jenkins	master-executors.groovy
Configure slack plugin	config-slack.groovy
Configure pipeline shared libraries	config-pipeline-library.groovy
Get jenkins version from CLI	`java -jar /usr/share/jenkins/jenkins.war --version`
Reference	GitHub: cloudbees/jenkins-scripts, GitHub: jenkinsci/pipeline-examples

1.3 Jenkins Kubernetes Via Groovy

Name	Comment
Config jenkins kubernetes plugin	jenkins-kubernetes-cloud.groovy
Validate Kubernetes jenkins setup	validate-kubernetes-cloud.groovy
Kubernetes run with envs configured	runWithEnvVariables.groovy
Reference	GitHub: kubernetes-plugin pipeline examples

1.4 Jenkins View Via Groovy

Name	Comment
Add a list of jobs by regexp to a view	`myView.setIncludeRegex(".*Integration.*")`, addjobstoview-byregexp.groovy
Create jenkins views and add jobs to it	jenkins-views.groovy
Add a view of build monitor view plugin	build-monitor-views.xml
Change view description in groovy	`myView.doSubmitDescription`

1.5 Jenkins Job Via Groovy

Name	Comment
List all my jenkins jobs	`println Jenkins.instance.projects.collect { it.name }`
List all jenkins jobs	list-all-jobs.groovy
Create and trigger a job	create-jenkins-job.groovy
Manage jenkins jobs	manage-jenkins-jobs.groovy
Cancel queued jenkins jobs by regexp	kill-queued-jenkins.groovy
Support HTML for job and parameter descriptions	Link: OWASP Markup Formatter Plugin

1.6 Jenkins Different Parameters

Name	Comment
string	`string(name: 'key1', defaultValue: 'Default value', description: 'some parameter')`
text	`text(name: 'key1', defaultValue: 'Default value', description: 'some parameter')`
boolean	`booleanParam(name: 'key1', defaultValue: true, description: 'some parameter')`
choice	`choice(name: 'key1', choices: 'One\nTwo\nThree\n', description: 'some parameter')`
password	`password(name: 'key1', defaultValue: 'SECRET', description: 'Enter a password')`
file	`file(name: 'key1', description: 'Choose a file to upload')`

1.7 Jenkins Security Via Groovy

Name	Comment
logged-in users can do anything	logged-in-users.groovy
Enable ldap in Jenkins	enable-ldap.groovy
Create a jenkins secret text	create-secret-text.groovy
Configure authorization in Jenkins	matrix-authorization-strategy.groovy
Jenkins skip wizzard when initialization	-Djenkins.install.runSetupWizard=false
Slave To Master Access Control	00-slave-to-master-access.groovy
CSRF Protection	00-csrf.groovy
Disable CLI over Remoting	00-disable-cli-remoting.groovy
Disable jnlp	`jenkins.setSlaveAgentPort(-1)`
Access Control for Builds	jenkins.security.QueueItemAuthenticatorConfiguration.xml

1.8 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`

1.9 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
Log Parser Plugin	Parse the console output and highlight error/warning/info lines.
Build-timeout Plugin	Abort if job takes too long
Naginator Plugin	Retry failed a job
ThinBackup Plugin	Backup jenkins
JobConfigHistory Plugin	Backup job configuration
"Anything Goes" formatter	use JavaScript inside your project description
AnsiColor Plugin	Add support for ANSI escape sequences, including color, to Console Output
Build User Vars Plugin	Describe the user who started the build

1.10 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

1.11 Jenkins networking Via Groovy

Name	Comment
Get hostname	`println InetAddress.localHost.canonicalHostName`
Get IP address	`println InetAddress.localHost.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}")`

1.12 Jenkins with Kubernetes/Docker

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

1.13 Groovy Common Errors/Exceptions

Name	Comment
Illegal class name	JVM doesn't like class names with a hyphen

1.14 Groovy Basic

Name	Comment
Get environment variables	get-env.groovy, `println env.WORKSPACE`
Groovy execute command	execute-command.groovy
Get data type of a variable	`myObject.getClass()`
Print stdout	print.groovy echo 'Action is done', println "Hello World"
Use boolean parameter	if (istrue == "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

1.15 Groovy String/Regexp

Name	Comment
Check string startsWith	`assert s.startsWith("\t")`
Trim whitespaces	s=s.trim()
Concat string	`first = 'Joe'; last = 'Smith'; println("Name:  $first $last")`
Convert list to string	`l.join(";")`
Create string with multi-lines	multi-line-string.groovy
Convert string to list	split-string.groovy
Convert string to json	string-to-json.groovy
Remove tags	`input.replaceAll("\\<.*?>","")`
Regex match	regexp-match.groovy
Regex case insensitive	(item.name == ~/(?i).*NSX.*/)
Reference	Regular Expressions in Groovy

1.16 Groovy Array

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

1.17 Groovy File

Name	Comment
Read file into a string	`String fileContents = new File('/tmp/test.txt).text`
Read file content as a variable	`def env = System.getenv()`, `def content = readFile("/tmp/test.txt")`
Write file in pipeline	writeFile file: "output/my.txt", text: "This is a test"
Read a property file	def conf = readProperties file: "${env.WORKSPACE}@script/my.properties"
Read and write json files	json-file.groovy
Obtain a relative path	relative-path.groovy

1.18 Groovy Shell Command

Name	Comment
Run shell and get output	def out = sh script: command, returnStdout: true
Run shell and get exit code	def status = sh script: command, returnStatus: true

1.19 Groovy Dictionary

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

1.20 Groovy json

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

1.21 Groovy Date

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

1.22 Jenkins Agent

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

1.23 Jenkins Maintenance

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

1.24 More Resources

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

`https://github.com/fabric8io/jenkins-docker`