This document is a list of answers to the things the lab asks you to figure out on your own. Each answer is on a separate page so you don’t see them by accident.

**Looking in the Nexus task log on the command line**

Task:

Now find the main nexus.log

Solution:

cd /nexus-data/log

tail –f nexus.log

**Looking in the Nexus task log on the command line**

Task:

In your browser, re-run your Groovy task. How many lines do you see added to the tail output? It’s not 8 lines. Can you figure out the difference between the nexus log and task log?

Solution:

6 lines

|  |  |
| --- | --- |
| **Nexus Log** | **Task Log** |
| Task name start | Task information |
| Task state change to start | Task id |
| Location of log file | Task type |
| 2 lines output | Task name |
| Task state change to waiting | Task start |
|  | 2 lines output |
|  | Task complete |

**Writing your own script in Nexus**

Task:

Now try to write your own Groovy script that prints out a countdown.

Solution:

(10..1).each{ log.info '--> ' + it }

log.info 'Blast off!'

**log vs println**

Task:

Check the output in both the browser and command line task log. Where do you see it?

Solution:

println only goes to the task log. This output does not appear in the nexus log in the browser.

**Writing your own script in Jenkins**

Task:

Fill in the blank to print out all the strings that end with “day”.

def list = ['monday', 'wednesday', 'chocolate', 'friday']

println \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Solution:

def list = ['monday', 'wednesday', 'chocolate', 'friday']

println list.grep { it.endsWith 'day' }

**Creating a user**

Task:

Add four more lines of code so Owen, Sam, Sophia and Daisy get accounts.

Solution:

import jenkins.model.Jenkins

def instance = Jenkins.getInstance()

def realm = Jenkins.getInstance().securityRealm

realm.createAccount('olivia', 'olivia')

realm.createAccount('owen', 'owen')

realm.createAccount('sam', 'sam')

realm.createAccount('sophia', 'sophia')

realm.createAccount('daisy', 'daisy')

instance.save()

**Setting up the authorization strategy**

Task:

Update the groovy code to set up the sea lions.

Solution:

auth.doAddRole('globalRoles', 'sea-lion-team', readOnly, 'true', '.\*')

auth.doAddRole('projectRoles', 'sea-lion-team', allJobAndRunPermissions, 'true', 'sea-lion.\*')

auth.doAssignRole('globalRoles', 'sea-lion-team', 'sophia')

auth.doAssignRole('globalRoles', 'sea-lion-team', 'sam')

auth.doAssignRole('projectRoles', 'sea-lion-team', 'sophia')

auth.doAssignRole('projectRoles', 'sea-lion-team', 'sam')

auth.doAssignRole('projectRoles', 'authenticated', 'sophia')

auth.doAssignRole('projectRoles', 'authenticated', 'sam')

**Editing a pipeline job**

Task:

The Sea Lions have decided they want to deploy to Nexus on every other build. For even numbered builds they want to just print a message to the console that they are skipping deploy. For odd numbered builds, they want the pipeline to behave normally.

Solution:

stage ('Publish') {

def buildNum = env['BUILD\_NUMBER'] as Integer

if (buildNum % 2 == 0) {

println 'skip publish'

} else {

withCredentials…

}

}

**Learn what objects are available**

Task:

Write your own Groovy task to output the four classes in use her

Solution:

log.info "Core: ${core.class}"

log.info "Repository: ${repository.class}"

log.info "Blob Store: ${blobStore.class}"

log.info "Security: ${security.class}"

Core: class org.sonatype.nexus.internal.provisioning.CoreApiImpl

Repository: class org.sonatype.nexus.script.plugin.internal.provisioning.RepositoryApiImpl

Blob Store: class org.sonatype.nexus.internal.provisioning.BlobStoreApiImpl

Security: class org.sonatype.nexus.security.internal.SecurityApiImpl