# **Jenkins Migration**

This is in-depth tutorial for migrating Jenkins jobs and plugins from one instance to another.

When we’re done, you should have a fully functional Jenkins environment like your existing Jenkins instance.

**Pre-Requirement:**

* A functional Jenkins Instance, we will call it as Source Jenkins server
* Python 2.X installed, to run the scripts

Following scripts are used for Jenkins Migration activity. In this activity, we are installing plugins and jobs from source Jenkins server to destination Jenkins server.

* resource.txt: This file needs to be updated before running jenkinsMigration.py to provide source and destination Jenkins url, username, password, token and local file system path to save backup
* jenkinsMigration.py: This script needs to be run to start Jenkins Migration. This is menu driven utility which will invoke the performtask.py to execute different functions related to Plugin and Job migration as per user selection.
* performTask.py: This file has functions to
  + install Plugins from source Jenkins server to destination Jenkins server using /pluginManager/api/
  + create backup of list of plugins installed on source Jenkins server in text file
  + install Jobs from source Jenkins server to destination Jenkins server using the source job config.xml file
  + create backup of config.xml files from source Jenkins server in local file system

**jenkinsMigration.py** is menu-driven utility, so it will prompt user to enter choice from the menu displayed on screen. Following are the menus provided and their action items:

1. **Plugin Migration**: This menu deals with plugin migration activity. It has two sub-menus
   1. **Create Plugin Backup**: This menu will call the function createPluginBackup() from performtask.py to get the list of plugins their versions installed on source Jenkins server using REST API call {SOURCE\_JENKINS\_SERVER}/pluginManager/api/json?depth=1 and write it to pluginBackup.txt which is stored in local file system.
   2. **Install Plugin**: This menu will call the function installPluginOnDestination() from performtask.py to get the list of plugins from source Jenkins server and install it on destination jerkins server using POST method {DESTINATION\_JENKINS\_SERVER}/pluginManager/installNecessaryPlugins . This POST call requires additional attributes
      1. final\_url={DESTINATION\_JENKINS\_SERVER}/pluginManager/installNecessaryPlugins
      2. data=<jenkins><install plugin="{PLUGIN\_NAME}@latest"/></jenkins> (where, latest will fetch the latest version of the plugin\_name)
      3. auth=(destination\_jenkins\_username, destination\_jenkins\_password)
      4. header={crumb\_field:crumb\_value,"Content-Type":"application/xml”} (where crumb\_field=Jenkins-Crumb and get crumb value using API call {DESTINATION\_JENKINS\_SERVER}/crumbIssuer/api/json
2. **Job Migration:** This menu deals with job migration activity. It has following sub-menus
   1. **Create Job Config Backup**: This menu will call the function createJobBackup() from performtask.py first to get the list of all jobs present using REST API call {SOURCE\_JENKINS\_SERVER}/view/All/api/json and then to get the config.xml file for each individual job using the jobURL {SOURCE\_JENKINS\_URL}/job/{JOB\_NAME}. We will then save this {JOB\_NAME}\_config.xml file in local file system.
   2. **Install Jobs**: This menu provides further sub menus to create jobs using either the existing backed up job config.xml files or create job using REST API calls to fetch job config.xml content. To create jobs on destination jenkins server, we first get the job config.xml file from source jenkins server. Then we use POST call with following attributes to install jobs on destination jenkins server.
      1. final\_url= {DESTINATION\_JENKINS\_SERVER}/createItem?name={JOB\_NAME}
      2. data= {JOB\_NAME}\_config.xml file content
      3. auth=(destination\_jenkins\_username, destination\_jenkins\_password)
      4. header={crumb\_field:crumb\_value,"Content-Type":"application/xml”} (where crumb\_field=Jenkins-Crumb and get crumb value using API call {DESTINATION\_JENKINS\_SERVER}/crumbIssuer/api/json