

SOP on PROVISIONING AUTOMATION (RJIL IDC)

Contents

[Scope 5](#_Toc419214311)

[Phase-1 (ILO Configuration & Firmware Upgrade) 6](#_Toc419214312)

[Firmware Upgrade 8](#_Toc419214313)

[Phase-2 (Raid Configuration) 9](#_Toc419214314)

[Phase-3 (Non-PXE method – LINUX OS Installation) 12](#_Toc419214315)

[Phase-3 (PXE method – LINUX/ESXI OS Installation) 13](#_Toc419214316)

[VM-Image Deployment 15](#_Toc419214317)

[VM-Creation-and-OS-Installation 16](#_Toc419214318)

Amendment Record / Control Sheet (ACS)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rev. No. | Date | Reason for Issue | Prepared by | Reviewed by | Approved by | Issued by | Details of Amendments / Revisions |
| 1.0 | 24/03/2015 | For Management Review & update | Hardik V Shah |  |  |  | New Document |
| 2.0 | 30/03/2015 | For Management Review & update | Hardik V Shah |  |  |  | Updated Document as screenshots required for each phase and action |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

# Scope

This document can be referred for below hardware models and operating systems installation.

**Hardware details:**

1. HP DL380P (Gen7, Gen8)
2. HP DL580 (Gen7)
3. HP DL980 (Gen7)

**OS details:**

1. RHEL-(5.5 to 6.5)
2. CentOS-(5.5 to 6.5)
3. Windows 2008, 2012R2
4. ESXI-(5.0, 5.1, 5.5)

# Phase-1 (ILO Configuration & Firmware Upgrade)

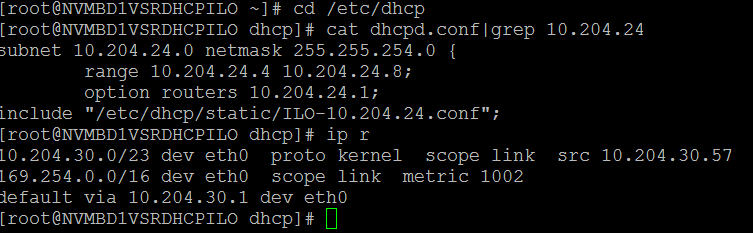
**Purpose:** To assigning static IP to iLO interface, setting ILO parameters including IRS settings and firmware upgrade.

**Scope:** It is applicable to configure iLO and firmware upgrade on HP servers.

**Prerequisite:**

1. Network/VLAN has to be enabled/patched properly for ILO IP range.
2. DHCP relay should be enabled for (ILO DHCP server – 10.204.30.57) by which server can reach to the ILO DHCP server. [ it can be checked through tcpdump -i eth0 |grep <mac-id of the server> ]
3. PXE should be enabled for ILO on the server. [ It has to be checked manually through connecting monitor physically as there is no any ILO IP assigned to the server at this time ]
4. Appropriate ILO range should be in place on ILO DHCP server at /etc/dhcp/dhcpd.conf

e.g. for 10.204.24.\* it should be declared as below:

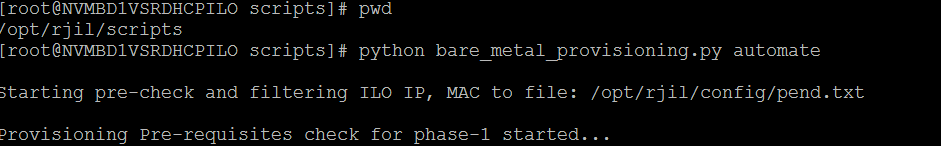


Actual provisioning starts with executing below script.

**Script name:** bare\_metal\_provisioning.py

[To execute run, “python bare\_metal\_provisioning.py project\_id” (project\_id should be what is given in IGF provided by customer)

e.g. python bare\_metal\_provisioning.py automate ]

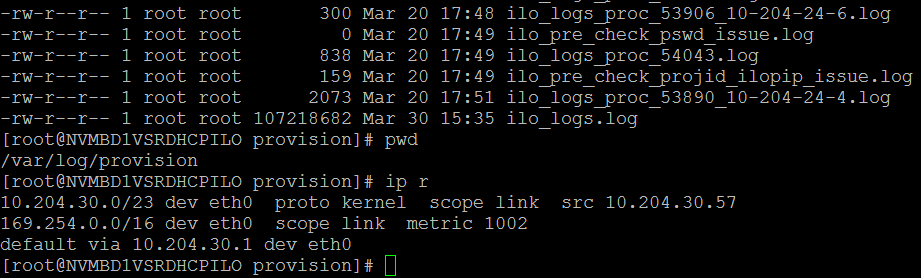


**Script path:** 10.204.30.57:/opt/rjil/scripts/ (Login to ILO DHCP server, 10.204.30.57 as root)

**Log path:** 10.204.30.57:/var/log/provision/ilo\_logs\_proc\_<pid>.log, /var/log/provision/ilo\_pre\_check\_pswd\_issue.log,

/var/log/provision/ilo\_pre\_check\_<projid>\_issue.log,

/var/log/provision/ilo\_logs\_proc\_<pid>\_<ilo\_ip>.log



**Note: This script is not running through cron, so need to run through ILO DHCP server whenever required.**

**Script Flow:**

Script is executing mainly three functions:

1. Pre-check for ILO:

It checks /var/log/messages on ILO DHCP server and capture iLO MAC and iLO IP and stores them to /opt/rjil/config/dhcp\_info.txt.

* It gets the serial number through curl and verify the project\_id, default password, status, if IP is already assigned etc... And send mail to provisioning/deployment team accordingly.
* On the successful pre-check, it creates the file which will be used for actual execution of ILO configuration and firmware upgrade.

1. New\_provision\_parallel:

* It starts ILO configuration in parallel for servers exist in /opt/rjil/config/dhcp\_info.txt
* Execution of servers in parallel depends on load/capacity of ILO DHCP server.
* It calls ilo\_initial\_config which does as below:

1. It checks for console power status and switch it ON if it is OFF.
2. Activate License key on iLO IP.
3. Assign static IP, netmask, gateway on iLO
4. Set server name as its serial number
5. Create idcadm user on iLO
6. Apply IRS setting
7. Update the status/ILORSA\_STATUS with **“IIC” (ILO IP Configured)** in t\_server\_provisioning and t\_equipment\_physical
8. Check the current firmware version and upgrade if required
9. Update the status **FUI/FUN/FUF (Firmware Upgrade Initiated, Firmware Upgrade Not-required and Firmware Upgrade Failed)** in t\_server\_provisioning and t\_equipment\_physical
10. create\_ilo\_mac\_entry\_to\_dhcpd:

* Based on T\_DHCP\_INFO on cmdb, it creates entry in /etc/dhcp/static/ILO-[first-3-octet].conf file on iLO DHCP server.
* It creates new file in /etc/dhcp/static/ if required and updates the same in /etc/dhcp/dhcpd.conf
* If there is any new entry or new file, it will reload the dhcpd daemon.

## Firmware Upgrade

**Purpose:** In case if you just want to upgrade firmware on the servers.

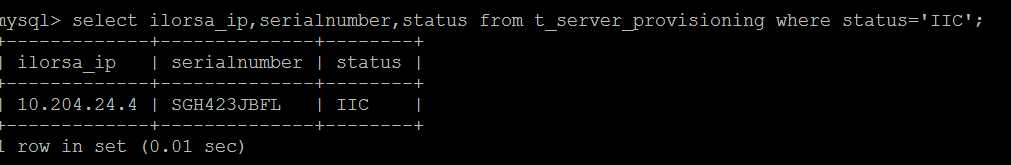
**Scope:** ILO IP of the server is required to follow this document.

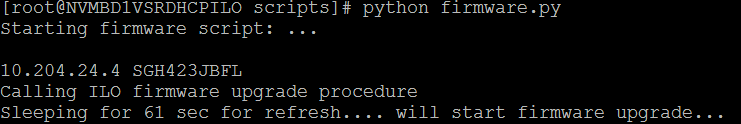
**Prerequisite:**

1. ILO IP should be reachable from ILO DHCP server
2. Make sure HPSPP.iso file available on ILO DHCP server at below path: /var/www/html/HPSPP.iso

**Script name:** firmware.py [To execute run, “python firmware.py”]

It checks and start firmware upgrade on all the servers having ‘IIC’ status in t\_server\_provisioning.





**Script path:** 10.204.30.57:/opt/rjil/scripts/

**Log path:** 10.204.30.57:/var/log/provision/ilo\_logs.log

**Note: This script is not running through cron, so need to run through ILO DHCP server whenever required.**

**Script Flow:**

1. It gets the list of ILO IP’s and serial no from t\_server\_provisioning table of cmdb which are with status “IIC”
2. It checks the current version of firmware and start firmware upgrade if required.
3. It updates the status in t\_server\_provisioning and t\_equipment\_physical with FUN/FUF/FUI depending on Firmware Upgrade Not-required, Firmware Upgrade Failed, Firmware Upgrade Initiated

[Firmware upgrade would take around 30-45 minutes, so wait for some time before proceeding to the next step]

**Note: HPSPP.iso is upgrading firmware to latest version “1.70” or “1.51” depending on hardware model. If newer version is available for the firmware, one has to download HPSPP.iso for latest version and replace that with the above mentioned iso file.**

# Phase-2 (Raid Configuration)

**Purpose:** It describes the steps to create a magic ISO, hpdiscovery, RAID configuration, update DHCP configuration for Data IP.

**Scope:** It captures the server hardware details and applies the RAID configuration. It does not do any OS installation.

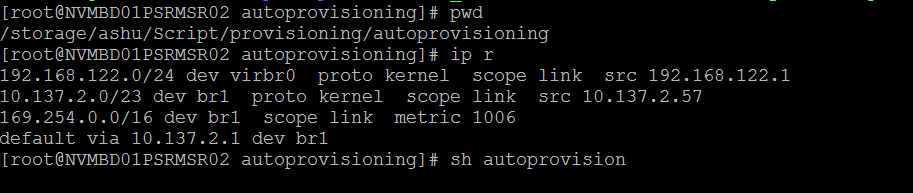
**Prerequisite:**

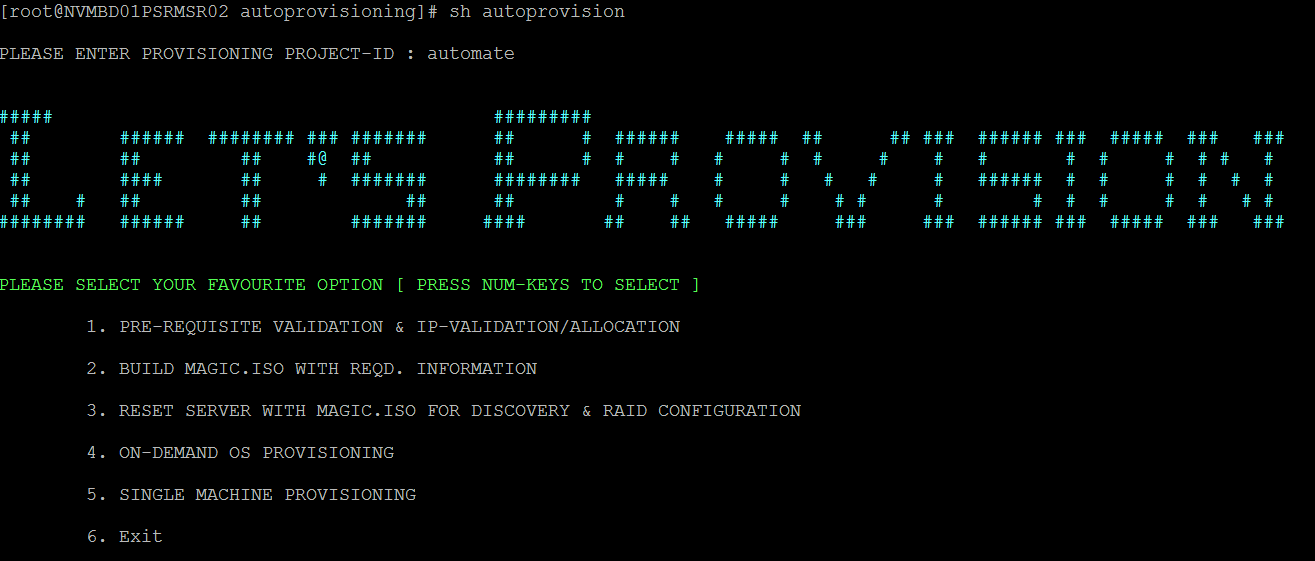
* Store Team will have updated server details in to the CMDB table (t\_strore)
* Deployment Team will have racked and staged and updated into CMDB table (t\_equipment\_physical).
* Server will have patched properly and updated into the CMDB table (t\_server\_provisoining).
* Network team will have passed VLAN\_ID to the server according to server hall and Environment and update CMDB table (t\_server\_provisioning).
* Provisioning Team will have filled required provision details into CMDB table (t\_server\_provisioning).
* Make sure that iLO has been configured and updated in CMDB.
* DHCP server will have configured as per subnet by using scripts.
* Cross check scripts availability which are essentials for provisioning.

**Involved directories and paths:**

* Install server script path
  + /storage/ashu/Script/provisioning/autoprovisioning/
* Configuration Script’s path and Virtual Media’s XML file Path
  + /storage/repo/scripts/provisioning/pre\_scripts
* SSTK Directory
  + /storage/repo/os/utilities/SSSTK/
* TOOLKIT Script’s path
  + /storage/repo/os/utilities/SSSTK/scripts/
* Magic ISO’s Path
  + /storage/repo/magiciso/
* CoreOS installation Components.
  + /storage/repo/os/CoreLinuxOS\_Beta
  + /storage/repo/scripts/provisioning/XML\_files
  + /storage/osinstall

**Script name:** **autoprovision** [To execute run, “sh **autoprovision**”]





Enter the project id and follow the option for phase-2.

**Script path:** 10.137.2.57:/storage/ashu/Script/provisioning/autoprovisioning/

**Note: This script is not running through cron, so need to run through 10.137.2.57 whenever required.**

**Script Flow:**

1. autoprovision is user defined menu script; we can choose option based upon the step you want to execute.

NOTE:

* 1. Please don’t skip or jump the step.
  2. For each and every step there is individual module designed, as per requirement you can execute it separately.

1. Pre-check script execution:
   1. You can execute this script by select option no. 1 from menu OR separately from the concern path.
   2. This scripts validate ILO related information into the t\_server\_provisioning like ilorsa\_ip, ilorsa\_gateway and ilorsa\_subnet\_mask {should not be null or Blank}; also validating ilorsa\_status should be IIC (ILO IP configured)
   3. Validate the status as FWU / FUN column in t\_server\_provisioning table.
   4. It will GO ahead for the next validation process only for the servers which are successfully passed **step b & c.**
   5. Pre-check steps defined into two major parts.
      1. Mandatories Validations
         1. NOT NULL OR BLANK (VLAN\_ID, SERVERHALL, FACILITY\_ID)
         2. BLANK (DATA\_IP, DATA\_SUBNET\_MASK AND DATA\_GATEWAY)
      2. Optional Validations
         1. PROFILE
         2. ROLE\_PROFILE
         3. HARDWARE\_PROFILE
         4. JOB\_ID
   6. After Successfully validation of step no 2.e.i {1, 2}, It assigns data\_ip based upon the vlan\_id by calling script **“ipman”.**
   7. **ipman** plays two major role in phase-2 of provisioning process.
      1. **IP Validation.**
         1. If data\_ip and other required information has already been populated into t\_server\_provisioning ; it will be validate information in t\_ipstatus server.
      2. **IP Allocation.**
         1. Based upon the step 2.e.i {1, 2} ; it will allocate data\_ip and required information into t\_server\_provisioning as well as t\_jpstatus
         2. It checks status FREE in t\_ipstatus table and ping FREE IP, before allocate data\_ip from t\_ipstatus depends upon vlan\_id and serverhall and facility\_id mentioned into t\_server\_provisioning and finally update t\_ipstatus colume and set serialnumber, hostname and ip\_status=’USED’
2. magiciso script to build magic.iso and stich required information into the same.
   1. You can execute this module by selecting option no.2 OR individually as needed.
   2. Takes care of discovery, get network port and switch information, RAID configuration, kickstart creation and 10gig PXE enablement.
   3. It gives liberty to create new or edit the existing **job\_id** depends upon the requirement.
      1. JOB\_ID is alphanumeric number, which will generate for one project and update the same in t\_server\_provisioning table.
      2. JOB\_ID.TXT is file which contains serialnumber,dataip, subnet and gateway ,vlan\_id information from t\_server\_provisioning table and will be built along with magic iso.
   4. Update JOB\_ID to t\_server\_provisioning; Creation of JOB\_ID.xml files under Virtual Media’s XML file Path.
   5. Building of Magic ISO (HP smart scripting toolkit) with latest JOB\_ID.txt files along with ***toolkit scripts*** under Magic iso’s path and store the job\_id.iso under /storage/repo/magiciso/ and update status=**’MIC’** in cmdb.
3. attachmagiciso script reset server using ILO API.
   1. You can execute this module by selecting option no.3 OR individually as needed
   2. Reset ILO and attach job\_id .iso specified in t\_server\_provisioning in job\_id column.
   3. Refer /storage/repo/magiciso for attachment of magiciso.

# Phase-3 (Non-PXE method – LINUX OS Installation)

**Purpose:** It describes the steps for actual **LINUX OS installation** on the server. In this phase we are creating minimal OS of 38mb in size and installing required OS.

**Script Flow:**

1. This PHASE divided into 2 parts **A. CoreOS IMAGE CREATION B. Required OS INSTALLATION**
2. **CoreOS image Creation.**
   1. Integrating kickstart file and other required information into CoreOSLinux.iso during creation process.
   2. Before creating CoreOS .iso ; it validates the required information like status=’RFA’ and data\_ip, data\_subnet\_mask , data\_gateway are not ‘NULL’OR Blank.
   3. If Step-3 passed successfully it creats CoreOSLinux.iso under /storage/osinstall directory with serialnumber.iso nameand update cmdb with status=**’OIC’**
   4. Also it generates serialnumber.xml file under /storage/repo/scripts/provisioning/XML\_files directory.
   5. It validates other required directories and script.
3. **Required OS INSTALLATION**
   1. Validate the required information like ilorsa\_ip , status=**’OIC’** for specific serialnumber.
   2. Attachment of serialnumber.xml file to iLO specified into table (t\_server\_provisioning ).
   3. Reset the ILO and attach Created serialnumber.iso to ILO for next process.

NOTE: Before running it always recommends to enable **spanning edge tree ON** for particular switch port.

Once you will enable that, booting of OS will be fast and it won’t be wait for any manual intervention by chance.

**TOOLKIT SCIRPTING (SetIP.sh)**

**Functionalities:**

1. Cross check serial number of HP Bare Metal from JOB\_ID.txt
2. Assigns DATA IP as per JOB.ID.txt file.
3. Execute hpdiscovery.py script and STORE component into the centralize server/ install server under /storage/hpdiscovery.
4. Based upon the hardware\_profile mentioned into the table, it executes RAID Configuration module and make status as RFA into CMDB.
5. After successfully RAID Configuration and status=’RFA’; create kickstart on the install server.
6. Execute CDPR on the serialnumber and update cmdb table nw\_server\_switch\_port.
7. Enable PXE on 10gig Ethernet card.

# Phase-3 (PXE method – LINUX/ESXI OS Installation)

**Purpose:** It describes the steps for actual **LINUX/ESXI OS installation** on the server through PXE method.

**Scope:** It can be followed for RHEL/CentOS/Windows OS installation via PXE BOOT.

**Prerequisite:** Server’s data mac should be reachable to 10.137.2.57 (dhcp server). Server status should be “RFA” in t\_server\_provisioning in cmdb. ROLE\_PROFILE in t\_server\_provisioning needs to be updated with whatever OS needs to be installed. [ reference: T\_OSPROFILES in cmdb ]. Raid configuration is done via 1st Option of this script [or] booting mac address is rightly updated in t\_server\_provisoning table.

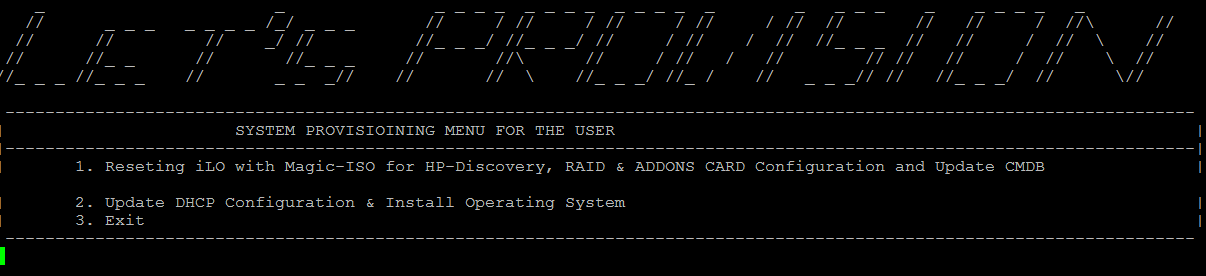
**Script path:** 10.137.2.57:/opt/rjil/scripts/

**Script name:** os\_installation.sh [To execute run, “sh os\_installation.sh”]

Step1: ssh [root@10.137.2.57](mailto:root@10.137.2.57)

Step2: cd /opt/rjil/scripts/

Step3: sh os\_installation.sh



Step4: Type 2 [To choose OS Installation option]

Step5: Press Enter [to Begin preparation and Installation of OS]

Note: This script is not running through cron, so need to run through DHCP server (10.137.2.57) whenever required.

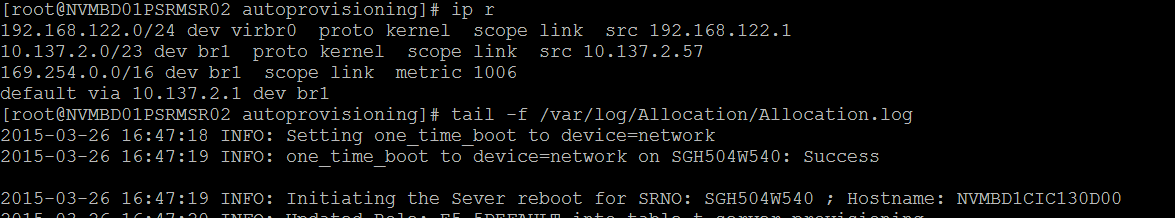
**Script Flow:**

* It gets the host information from the table and updates the same in host\_info file locally for further reference.
* It updates DHCP configuration for Data IP if it is not already done.
* It sets the parameters for PXE boot, create kickstart file and restart the server for network boot and set STATUS=’IIP’ [Installation In Progress]in t\_server\_provisioning.
* OS installation should be done in 20-30 mins with POST installation of enabling services like BA, Salt, HPOM, NTP, IDCADM a/c, generic account creation etc.
* After OS Install the script will update the t\_server\_provisioning STATUS=’RFR’ [Ready For Release] against its SerialNumber in t\_server\_provisioning table in the CMDB.

**Log Path:** Complete logs can be referred at 10.137.2.57:/var/log/Allocation/Allocation.log

**TIP:** You can open another session in 10.137.2.57 server and see running logs by executing

`tail -f /var/log/Allocation/Allocation.log`



# VM-Image Deployment

**Purpose:** It describes the steps to deploy a Virtual machine from a Golden-Template/Default-Image

**Scope:** It can be used to spin 19 different VM images as follows.

1. RHEL 5.7, 5.8 & 6.1,6.2,6.3,6.4,6.5 [7 Images]
2. CentOS 6.2,6.3,6.4,6.5 [4 Images]
3. W\_2K8\_ENT & R2, W\_2K8\_STD & R2, W\_2K12\_DC & R2, W\_2K12\_STD & R2 [8 Images]

**Prerequisite:** Esxi-Hypervisor should be reachable from DHCP Server. Should have Esxi-Hypervisor credentials. Manually update the VM information in t\_vm\_provisioning tables and get IP through manual process [via IP\_Assignemt portal].

**Script name:** VM\_AutoProvision.sh [To execute run, “sh VM\_AutoProvision.sh”]

**Script path:** 10.137.2.57:/opt/rjil/scripts/

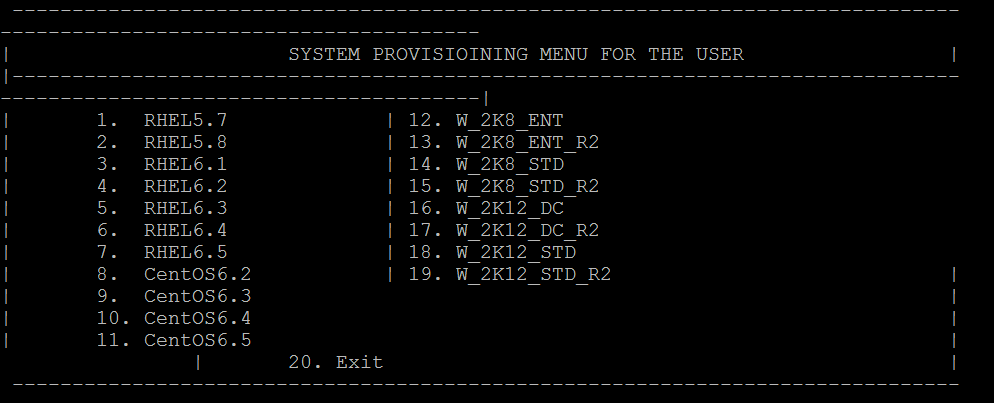
**Script Flow:**

Step1: ssh [root@10.137.2.57](mailto:root@10.137.2.57)

Step2: cd :/opt/rjil/scripts/

Step3: sh VM\_AutoProvision.sh

This script provides a CLI-based UI to choose the required image to be deployed. Please check below image.



Step5: Select an option from 1-19 for VMs; option 20 to exit.

1. Once the option is selected, It prompts for Esxi-Hypervisore’s credentials like “IP, USER-ID & PASSWORD”. User must provide the destination esxi-ip details here.
2. It further prompts for destination VM-Name & datastore where it needs to be placed/created.
3. Once all the required information is provided a Virtual Machine with default-OS will be created on the destination Esxi-Hypervisor in less than 3 minutes.

**Note:**

1. All the VM’s are created with defauldt 100Gig size.
2. As VMs (Linux & Windows) are cloned from a Golden-templates/Images, few configurations {Network, salt, hpomagent, Licences} needs to be done after VMs creation.
3. You have to update the CMDB tables manually.

# VM-Creation-and-OS-Installation

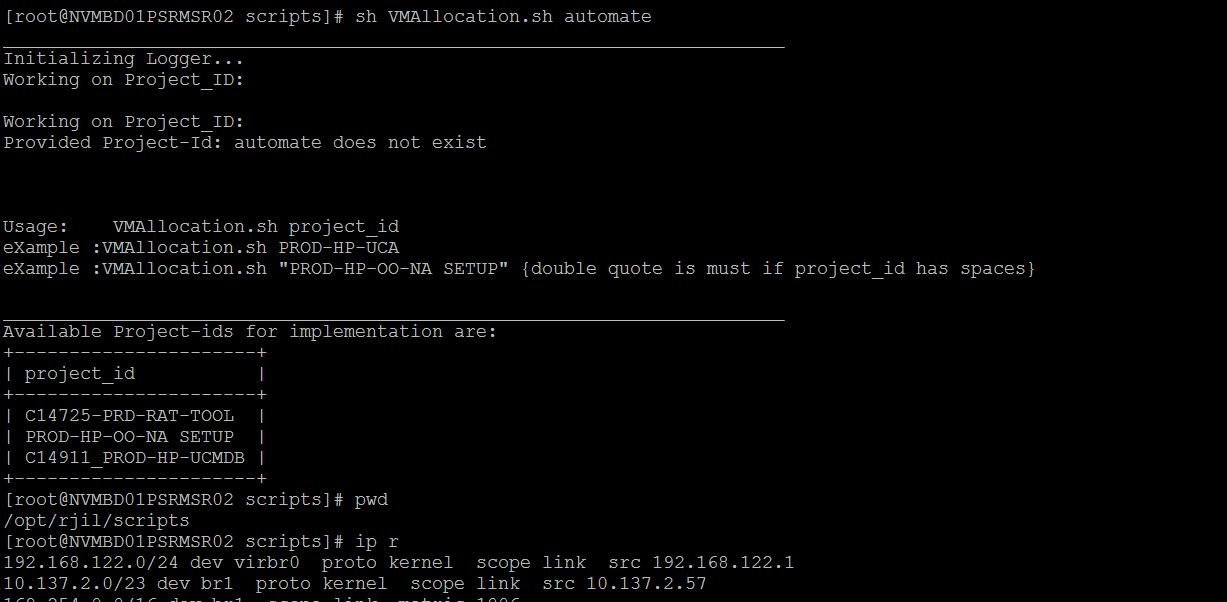
**Purpose:** It describes the steps to create a virtual machine on an esxi-hypervisor.

**Scope:** It can be followed for Windows/RHEL/CentOS Linux OS installation.

**Prerequisite:** Esxi-Hypervisor should be reachable from DHCP-server. All the required fields needs to be updated in t\_vm\_provisioning table in CMDB. VM\_ROLE\_PROFILE in t\_vm\_provisioning needs to be updated with whatever OS needs to be installed. [Reference: T\_OSPROFILES in cmdb ]

**Script path:** 10.137.2.57:/opt/rjil/scripts/

**Script name:** VMAllocation.sh [To execute run, “sh VMAllocation.sh <project-id>”]



**Script Flow:**

1. It gets the VM details from t\_vm\_provisioning where status=’RFD’ and reserves data-IP as per VLAN-ID & SERVERHALL values.
2. It gets the VM Hardware specs from t\_vm\_provisioning table and creates a VM in destination esxi as per BASE\_DATA\_IP value.
3. It gets the vm information[vm\_hostname,base-host info, ip-info, h/w spec,mac-address, vm\_role\_profile etc] from t\_vm\_provisioning table and updates the same in vm\_host\_info file locally for further reference.
4. It prompts for entering the datastore name where it needs to be created.
5. It waits for user to Press Enter-Button after deleting and re-creating the Ethernet via vCentre Application.
6. It updates DHCP configuration for Data IP if it is not already done.
7. It sets the parameters for PXE boot, create kickstart file for each VM as per its VM\_ROLE\_PROFILE value and restart the vm for network boot and set STATUS=’**IIP**’ [Installation In Progress] in t\_vm\_provisioning table in CMDB.
8. OS installation should be done in 20-30 mins with POST installation of enabling services like BA, Salt, HPOM, NTP, IDCADM a/c, generic account creation[as per kickstart file] etc.
9. After OS Install the script will update the t\_vm\_provisioning STATUS=’**RFR**’ [Ready For Release] against its VM\_HOSTNAME where OS is linux. For windows we have to manually change the status after verifying the OS Installation.