IBM

developerWorks®

Automate VIOS updates in NIM environment with ‘Puppet AIX and VIOS Automation’ module

Oct 26th ,2018

Updating IBM® AIX® VIOS system *in a large-scale infrastructure is now possible with Puppet®. Setting attributes in a Puppet manifest file enables automatic updates of a list of VIOS pairs from the NIM server in NIM push mode.*

Introduction

This article details how to use Puppet to automate VIOS updates on IBM® AIX® systems. The current documentation comes in addition to “*Automate AIX infrastructure updates in NIM environment with ‘Puppet AIX and VIOS Automation’ module*” which describes the ‘Puppet AIX® and VIOS’ module: hardware configuration, installation process, and how to use Puppet. Common explanations already provided in that document are not repeated here. This current article explains the additional service for automating VIOS updates with Puppet**.**

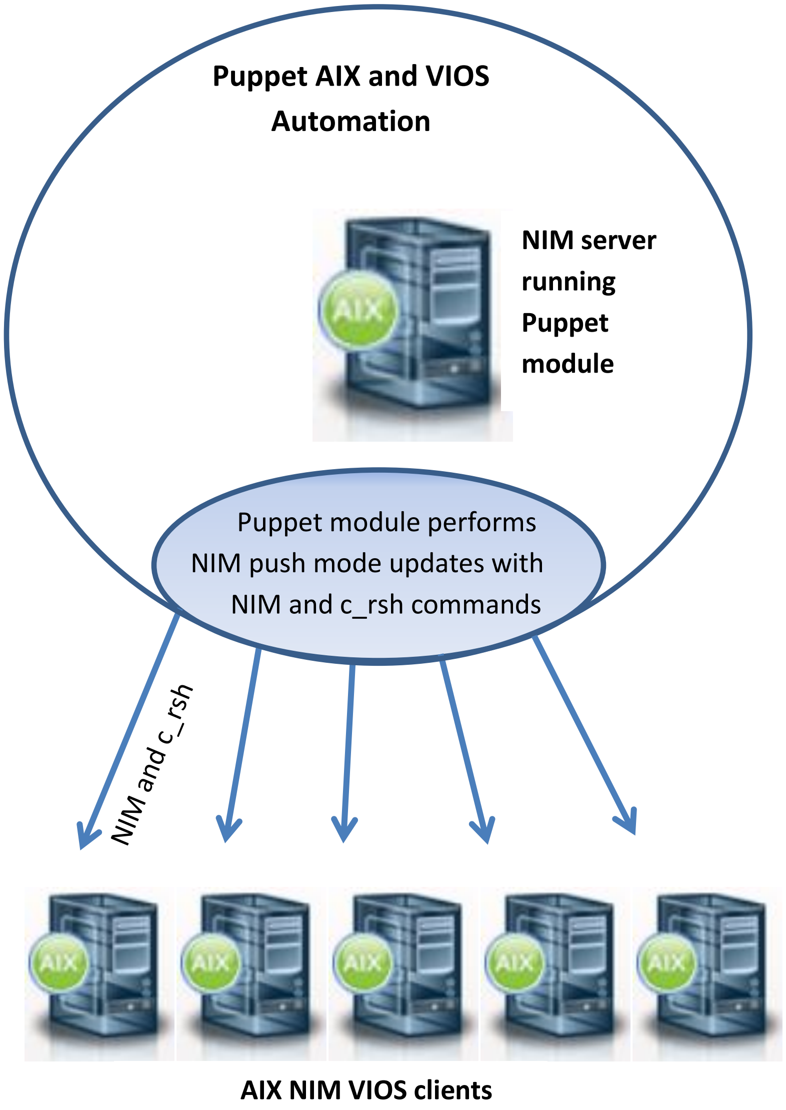
Main components involved in **Puppet AIX and VIOS automation** module are described first, and general explanations are provided. Functionalities offered by this module and description of the different steps to securely update list of VIOS pairs are then detailed. Then starts the internal description of the module and how to use it. If syntax of new ‘vios’ custom type is not repeated as already given in README.md, several examples are provided to show how to individually run these steps, or how to combine them to update dual VIOS without service interruption, this operation being called “VIOS rolling update”.

See table of contents at the end.

Main components

This service is delivered as a Puppet module called “**Puppet AIX and VIOS Automation module**”.

This service for managing VIOS system updates with Puppet® requires a NIM (Network Installation Management) environment and of course Puppet.


Puppet AIX and VIOS Automation
NIM server running Puppet module
AIX
Puppet module performs NIM push mode updates with NIM and c_rsh commands
NIM and c_rsh
AIX
AIX
AIX
AIX
AIX
AIX NIM VIOS clients

The diagram above describes the main components involved in **Puppet AIX and VIOS Automation module**.

At the bottom are the AIX NIM VIOS clients: they are the targets of the updates performed by Puppet AIX and VIOS Automation module, they need to be part of the NIM configuration. They can be automatically updated either on an individual basis or by pair: usually VIOS are configured as a pair of virtual servers, offering redundant configurations on LPAR disks access. Puppet does not need to be installed on these clients.

At the top is the NIM server, on which Puppet Agent needs to be installed. Puppet AIX and VIOS Automation module runs on this instance of Puppet. Connections between NIM server and NIM clients is exclusively performed through NIM's nimsh protocol, (c_rsh is used, but no ssh required): updates are performed from NIM server to NIM client by using NIM push mode.

Software component

This service for managing VIOS system updates with Puppet® is an Open Source development available on **AIXOSS GitHub repository**. It is added to the existing AIX VIOS systems update offers built on Ansible® and Chef® platforms.

General explanations

Puppet orchestrates (in a potentially large-scale infrastructure) the AIX and VIOS automation module but relies on the underlayers of functionalities provided by the NIM server. VIOS update is performed through NIM '**updateios'** command in NIM push mode.

- VIOS update can be run on a single VIOS, or a pair of VIOS, or several pairs of VIOS.
- Checks are made before performing the VIOS update: first a 'health check' to verify each VIOS pair are 'dual' (consistent as far as their configuration are concerned), then 'SSP check' to consider the fact that VIOS may belong to a SSP cluster.
- VIOS update can be run in preview mode only.
- VIOS are updated to the level provided by a NIM lpp_source resource which needs to be existing and ready to use into NIM configuration before launching Puppet AIX Automation module. Each VIOs can be associated with its own NIM lpp_source resource, and then can be individually updated to its own level.
- Auto-commit of applied LPPs on each VIOS can be performed beforehand, so that VIOS update can apply afterwards.
- Before VIOS update is performed, a copy of the rootvg is made to ensure a safe path to the previous state in case of errors. Several ways of doing this safe copy are proposed, and the fact that rootvg may be mirrored is considered.
- Logs and journals of what has run are taken, for trackability and better understanding.

VIOS update functionalities offered by 'Puppet AIX and VIOS automation' module

With Puppet AIX and VIOS automation module, updating a VIOS as being part of a VIOS pair enables:

- to introduce a **health check** step to verify as a preamble that both VIOS share the same view of mappings on all types of storage and virtual adapters: vSCSI, vFC, SEA, VNIC, etc. (this is done through actions='health' in manifest file).
- to consider **potential SSP cluster** hosted by both VIOS: as NIM VIOS update can only be run while SSP cluster node is stopped, each SSP cluster node is stopped before VIOS update itself is done and restarted afterwards (this is done through actions='check' in manifest file).

With Puppet AIX and VIOS automation module, you can associate a NIM **lpp_source resource on an individual VIOS basis**. For example, you can update one VIOS to 2.2.6.23 version and another VIOS to 2.2.6.31 version, (this is done through 'vios_lpp_source' attribute in manifest file)

With Puppet AIX and VIOS automation module, the updates can be performed in **preview mode** only, or can be committed, depending on an optional parameter. By default, installation is committed, but it can be a safe path to first 'preview' updates before doing them for real, (preview is done through options='preview' in manifest file).

Moreover, to be able to perform update of a VIOS, previous installation needs to be committed. The AIX and VIOS Automation module can perform 'commit' of existing installation as a preliminary step, before performing VIOS update step, this functionality is called '**autocommit'**, (autocommit is done through actions='autocommit' in manifest file).

With Puppet AIX and VIOS automation module, a **safe copy of rootvg** is kept insuring a rollback path. Then alternate disk copy issues, potential mirroring of rootvg of VIOS, SSP clusters the VIOS may belong to are considered, (safe copy of rootvg is controlled through actions='save' and through 'altinst_rootvg_force' and 'vios_altinst_rootvg' attributes in manifest file).

- If any SSP cluster exists, SSP cluster nodes needs to be stopped during updates. If they are stopped, they are restarted afterwards. If both SSP nodes are started, or both SSP node are stopped at the beginning of Puppet AIX and VIOS Automation module, the overall processing of updates for both VIOS of the pair can be performed. But if SSP nodes are not in the same state at the beginning, the overall process of updates for both VIOS of the pair cannot be performed.
- To be able to perform alternate disk copy of the rootvg (this is done through actions='save'), and if this rootvg is mirrored, un-mirroring (and symmetric operation) proves to be necessary.

These operations won't be done by default. They will be done only if user explicitly authorizes them by adding a 'unmirror' into 'actions' parameter of manifest file (so this is done through actions='autocommit'). Several possibilities exist as far as alternate disk copy is concerned. If one alternate disk copy already exists at the very beginning, this one cannot be overridden by default. To be able to create a new alternate disk copy, user must explicitly require it through the 'altinst_rootvg_force' parameter. By setting this 'altinst_rootvg_force' parameter to 'yes', a new alternate disk copy will be taken, potentially overriding the existing one, and user should use this value cautiously. By setting this 'altinst_rootvg_force' parameter to 'reuse', no new alternate disk copy will be taken if one already exists, as this very one is then considered as valuable: reusing an existing alternate disk copy spares time.

Moreover, to go beyond NIM updateios return code (which does not always provide right status of execution!), some **heuristics** have been introduced to parse output and fix, as far as it is possible, the status returned to the user. In some cases, a "bad" return code from NIM updateios command can be transformed into a "normal" return code by Puppet module logic. In any case it is always safer to go through log file and verify the status of NIM updateios command and advise. Specific log exists for NIM outputs into '**./output/vios/**'.

VIOS update steps of 'Puppet AIX and VIOS automation' module

When Puppet Automation is launched to perform VIOS update on a pair of VIOS or on a pair of VIOS, pseudo **algorithm of runtime steps** are:

a) Gather all necessary information using **facters**:
    a. Read manifests/init.pp containing user declarations expressed in Puppet DSL language, and containing instructions of what needs to be done.
    b. Run Puppet facters to retrieve configuration information: VIOS, HMC, SSP, global variables, … Only facters which are needed by the contents of manifest file are run.

b) Perform **checks**: health checks and then SSP checks:
    a. Perform a health check on VIOS pairs, to verify they exactly share same vision of configuration, to verify that both VIOS of the pair share the same view of mappings on all types of storage and virtual adapters: vSCSI, vFC, SEA, VNIC, etc.
    b. Check the existence of a SSP cluster on VIO pairs, and the state of their nodes, as it would be necessary to temporarily stop SSP cluster on each node to perform update.

c) **Loop** for each pair of VIOS, and then for each VIOS of the pair:

a. Ensure an **alternate disk copy** for the rootvg of the VIOS for backup purpose exists or is taken (in case of failure during update, a safe copy allows to come back on previous version). If taking a new alternate disk copy proves to be necessary, check the potential mirroring of the rootvg, as it is necessary to temporarily un-mirror this VG to take this new alternate disk copy of rootvg.
   i. Un-mirror rootvg
   ii. Find best disk to host alternate disk copy
   iii. Take an alternate disk copy of rootvg
   iv. Mirror rootvg back

b. Potentially **stop** the SSP cluster on this node

c. Perform the **update** using NIM updateios.
   i. Perform auto-commit of previous installation if necessary
   ii. Perform NIM updateios

d. Potentially **restart** the SSP cluster on this node

The unique entry point for the end user to declare its resource and the desired state of these resources is located in the manifest file: ‘**./aixautomation/manifests/init.pp**’.

Structure of ‘Puppet AIX and VIOS automation’ module

The structure of the Puppet AIX and VIOS automation module source tree is already explained into “*Automate AIX infrastructure updates in NIM environment with ‘Puppet AIX and VIOS Automation’ module*” and is not repeated here. Only what has been added to specifically support VIOS update is explained below. You can find the source tree under **/etc/puppetlabs/code/environments/production/modules/aixautomation** after installation.

Puppet AIX and VIOS automation module, as far as VIOS update is concerned, implements specific “facters” (a facter is a Puppet module, gathering facts) for VIOS and HMC, and these facters run at the beginning:

- ‘vios’: to gather data on NIM VIOS objects and to verify they can be used.
- ‘hmc’: to gather data on NIM HMC objects and to verify they can be used.

These two facters are added to the existing ones: ‘applied_manifest’, ‘standalones’, ‘props’, ‘servicepacks’ already explained into “*Automate AIX infrastructure updates in NIM environment with ‘Puppet AIX and VIOS Automation’ module*”.

Puppet AIX and VIOS automation module, as far as VIOS update is concerned, implements a specific custom type and its service provider:

- ‘vios’ custom type implemented by the ‘viosmngt’ service provider.

This custom type and its provider are added to the three existing ones, already explained into “*Automate AIX infrastructure updates in NIM environment with ‘Puppet AIX and VIOS Automation’ module*”.

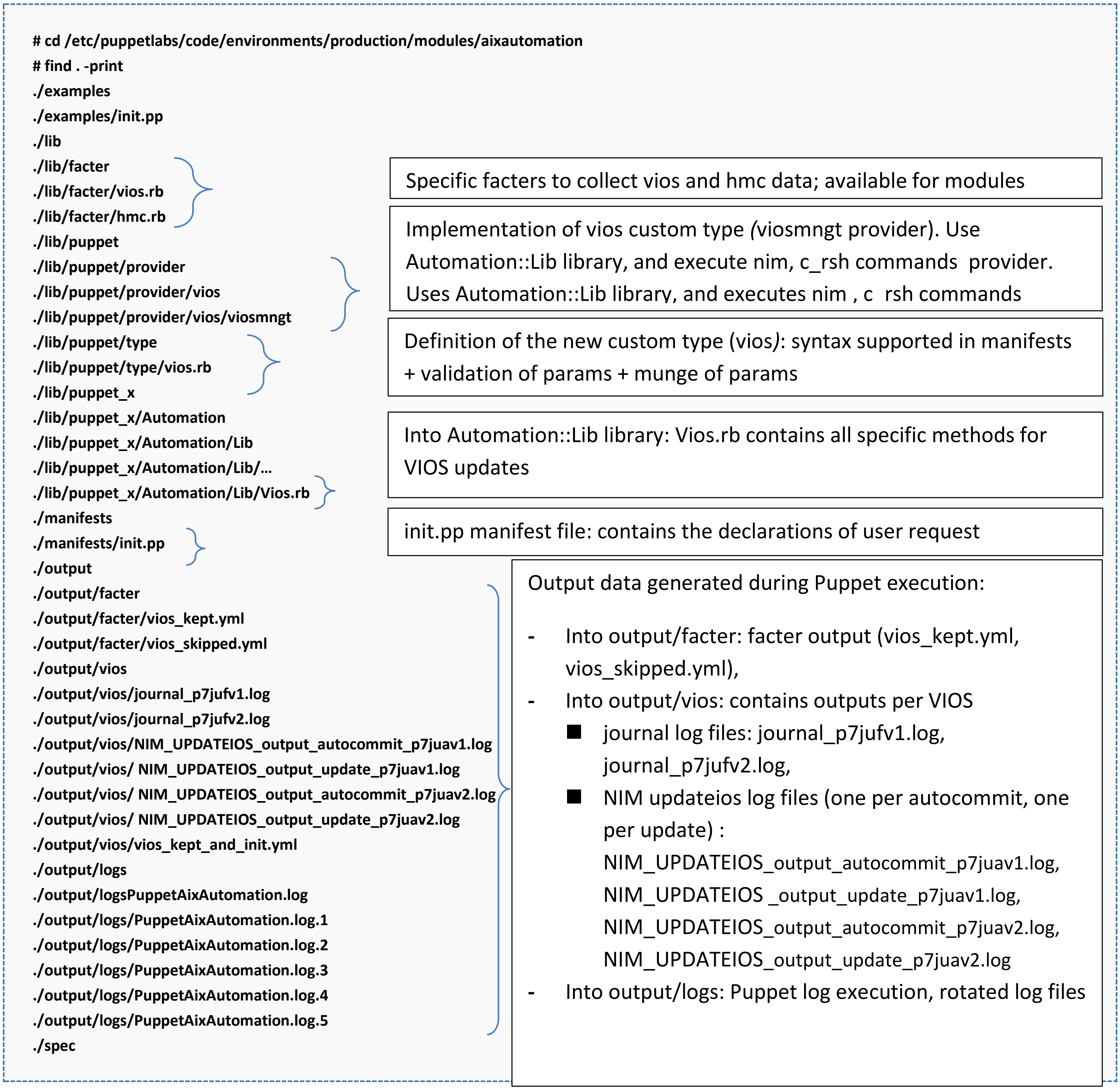
Puppet AIX and VIOS automation module provides a library (already explained into “*Automate AIX infrastructure updates in NIM environment with ‘Puppet AIX and VIOS Automation’ module*”), this library is enriched with one specific file for VIOS update:

- Automation::Lib::Vios.rb contains all methods for VIOS update.

Puppet AIX and VIOS automation module provides samples of manifest files and documentation

- ./examples/init.pp
  - User should refer to these examples to start writing his own manifest file.
- ./manifests/init.pp
  - User should write his own declarations into this file
- README.md
  - Complete explanations provided

Below is summarized the source files added for VIOS update:



```
# cd /etc/puppetlabs/code/environments/production/modules/aixautomation
# find . -print
./examples
./examples/init.pp
./lib
./lib/facter
./lib/facter/vios.rb
./lib/facter/hmc.rb
./lib/puppet
./lib/puppet/provider
./lib/puppet/provider/vios
./lib/puppet/provider/vios/viosmngt
./lib/puppet/type
./lib/puppet/type/vios.rb
./lib/puppet_x
./lib/puppet_x/Automation
./lib/puppet_x/Automation/Lib
./lib/puppet_x/Automation/Lib/…
./lib/puppet_x/Automation/Lib/Vios.rb
./manifests
./manifests/init.pp
./output
./output/facter
./output/facter/vios_kept.yml
./output/facter/vios_skipped.yml
./output/vios
./output/vios/journal_p7jufv1.log
./output/vios/journal_p7jufv2.log
./output/vios/NIM_UPDATEIOS_output_autocommit_p7juav1.log
./output/vios/ NIM_UPDATEIOS_output_update_p7juav1.log
./output/vios/ NIM_UPDATEIOS_output_autocommit_p7juav2.log
./output/vios/ NIM_UPDATEIOS_output_update_p7juav2.log
./output/vios/vios_kept_and_init.yml
./output/logs
./output/logsPuppetAixAutomation.log
./output/logs/PuppetAixAutomation.log.1
./output/logs/PuppetAixAutomation.log.2
./output/logs/PuppetAixAutomation.log.3
./output/logs/PuppetAixAutomation.log.4
./output/logs/PuppetAixAutomation.log.5
./spec
```

Specific facters to collect vios and hmc data; available for modules

Implementation of vios custom type (viosmngt provider). Use Automation::Lib library, and execute nim, c_rsh commands provider. Uses Automation::Lib library, and executes nim , c rsh commands

Definition of the new custom type (vios*)*: syntax supported in manifests + validation of params + munge of params

Into Automation::Lib library: Vios.rb contains all specific methods for VIOS updates

init.pp manifest file: contains the declarations of user request

Output data generated during Puppet execution:

- Into output/facter: facter output (vios_kept.yml, vios_skipped.yml),
- Into output/vios: contains outputs per VIOS
  - ■ journal log files: journal_p7jufv1.log, journal_p7jufv2.log,
  - ■ NIM updateios log files (one per autocommit, one per update) : NIM_UPDATEIOS_output_autocommit_p7juav1.log, NIM_UPDATEIOS _output_update_p7juav1.log, NIM_UPDATEIOS_output_autocommit_p7juav2.log, NIM_UPDATEIOS_output_update_p7juav2.log
- Into output/logs: Puppet log execution, rotated log files

‘Puppet AIX and VIOS automation’ module DSL language: ‘vios’ custom type

Syntax of manifest file is explained in details into ‘**./aixautomation/README.md**’**.** These explanations are not repeated here. In order to write customized ‘**./aixautomation/manifests/init.pp**’, the ‘**./aixautomation/README.md**’ file provides a full reference for the four custom-types : ‘**download**’, ‘**patchmngt**’, ‘**fix**’, and ‘**vios**’ and their attributes. Only ‘**vios**’ custom type is necessary for VIOS updates. Semantics of attributes are provided, as well as details about values: mandatory or optional values, possible values, default values.

In addition to this README.md file, the user can refer to commented samples provided into ‘**./aixautomation/examples/init.pp**’.

‘Puppet AIX and VIOS automation’ module configuration

NIM configuration, Puppet configuration, Puppet AIX and VIOS automation module (installation, configuration) are not explained in this document. User should refer to “*Automate AIX infrastructure updates in NIM environment with ‘Puppet AIX and VIOS Automation’ module*” which is a pre-requisite reading.

Building a NIM lpp_source resource to perform VIOS update

Building the NIM lpp_source resource is necessary to update VIOS through NIM. This preliminary step needs to be manually done before launching Puppet AIX and VIOS automation module.

Below is described one method to build such NIM resource, this method is provided as a sample, other ways of doing may exist.

- From FixCentral website https://www-945.ibm.com/support/fixcentral/, fill in form to indicate:

  ```
  Product Selector: PowerVM I/O Server
  Installed version: 2.2.6.23(this value is provided as an example)
  ```

- You’ll be invited to download your fix, which can be downloaded using web interface. You may as well choose to download it using wget, for example through something like:
  - `wget --no-check-certificate \`
    `https://delivery04.dhe.ibm.com/hsb/iso/H86201476.iso`

- When iso file is downloaded on your NIM server (often a large file of 4 or 5 GB) , you have to build your NIM lpp_source resource using something like:

```
nim -o define -t lpp_source -a
source=/export/vios/download/H19819269.iso -a server=master -a
location=/export/vios/NIM/22631_iso -a packages=all
vios_22623_22631
Preparing to copy install images (this will take several
minutes)...
  /export/vios/NIM/22631_iso/bos.terminfo.dec.data.6.1.1.0.U
  ...
  /export/vios/NIM/22631_iso/wio.fcp.6.1.8.0.I
  Now checking for missing install images...
  All required install images have been found. This lpp_source is
  now ready.
```

Running ‘Puppet AIX and VIOS automation’ module

The user declarations are located in the manifest file ‘**./aixautomation/manifests/init.pp**’.

Puppet AIX and VIOS automation is launched with the following command:

```
#/opt/puppetlabs/bin/puppet apply --debug --
modulepath=/etc/puppetlabs/code/environments/production/modules/ -e
"include aixautomation"
```

Output files – Logs

Puppet AIX and VIOS automation module outputs files into:

./aixautomation/output/
./aixautomation/output/facter: output of facter
./aixautomation/output/vios: output of updates done on VIOS, journal per VIOS. And NIM output per VIOS.
./aixautomation/output/logs: Puppet AIX and VIOS Automation logs

The log file from Puppet AIX and VIOS automation process is in: **./output/logs/PuppetAixAutomation.log**
The log file is automatically rotated and 12 files of 1MB are kept.

Use cases for updating VIOS systems with ‘Puppet AIX and VIOS automation’ module

This section describes some examples and nominal use cases for using ‘vios’ custom type in the manifest file “**./manifests/init.pp**”.

Hereunder examples were taken mostly from examples provided in **./aixautomation/examples/init.pp** For each example are provided some log and trace files extracts, so that user can better understand Puppet Aix Automation output in these various cases.

First example of manifest/init.pp to only perform health check on a single VIOS or on a VIOS pair:

```
class aixautomation {
 vios { 'vios_health_check_only_1_2':
   actions    => 'health',
   vios_pairs => '(p7jufv1,p7jufv2)',
  }
}
```

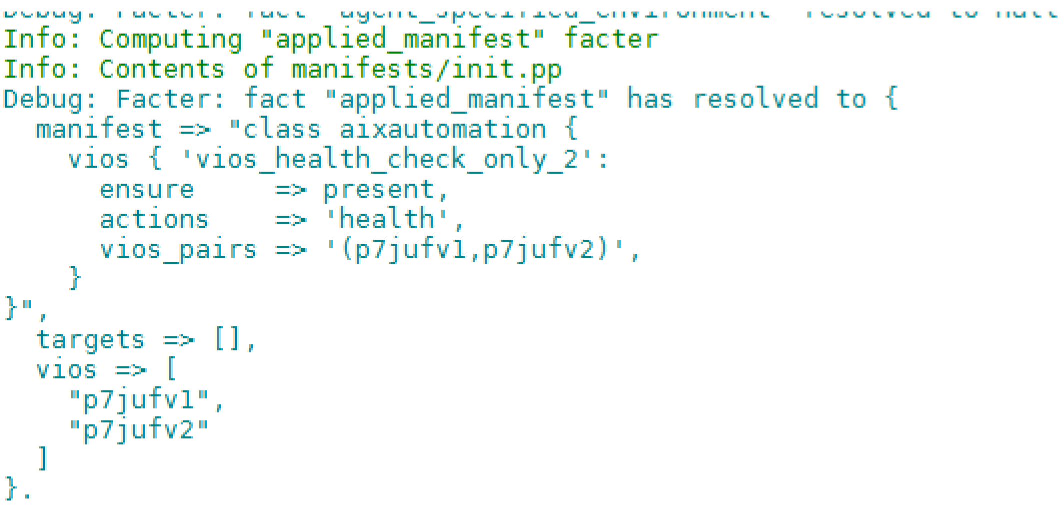
Class automation to encapsulate the declaration

Attributes of custom type ‘vios’: this declaration called ‘vios_health_check_only_1_2’ only launches the health check on the pair of VIOS

The ‘actions’ parameter is a multi-valuated attribute which takes only one value in this case: ‘health’. As matter of fact, actions are supposed to be run one after the other, and a failing ‘health’ check on a VIOS pair prevents other steps from being run. We’ll see that on other examples.
The ‘vios_pairs’ attribute contains in this case only one pair of VIOS: two VIOS into brackets, but this attribute is designed to receive pairs of VIOS, each pair containing two VIOS into brackets. Note that a pair could as well contain only one value (even if it does not make sense for health checking).

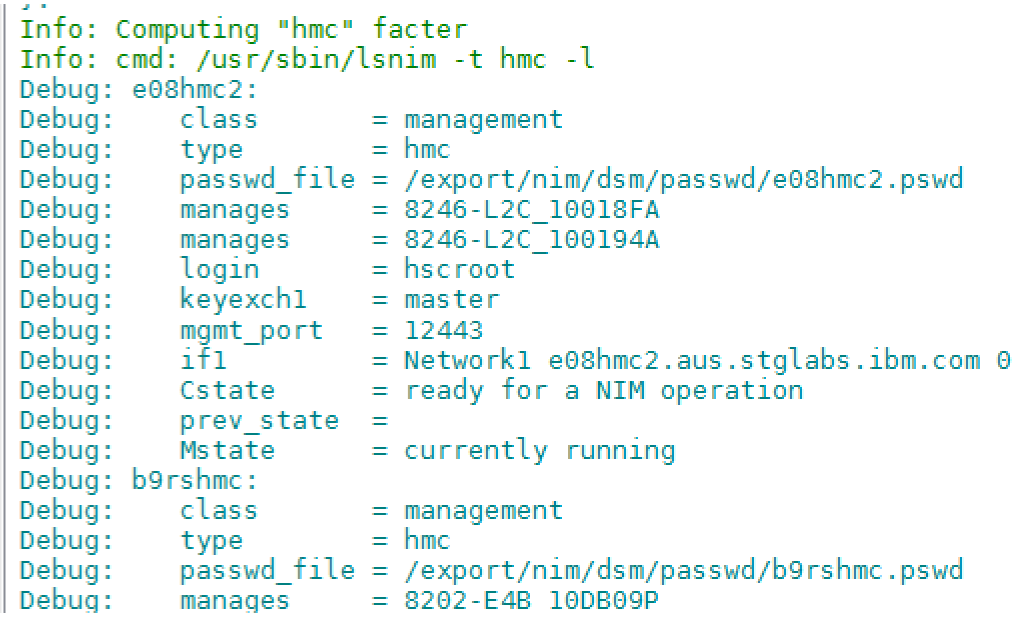
At the beginning, the facters are run, please note that it takes some time to gather all these data. Below are shown once the facters specific to this module, but other facters are run as well from Puppet framework.

Facteur ‘**applied_manifest**’ produces following output showing the contents of the manifests/init.pp which is applied:



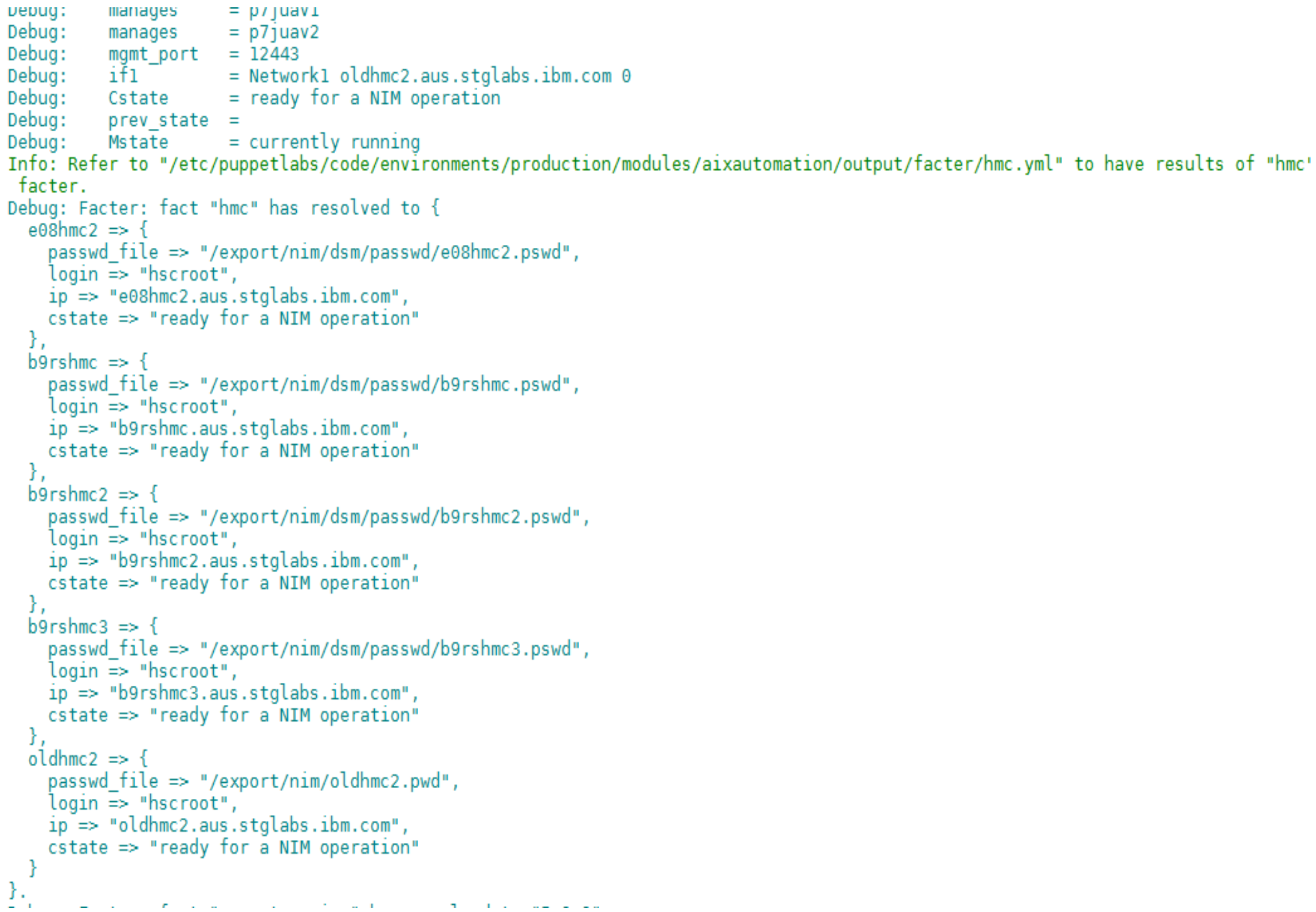
```
Debug: Facter: fact "agent_specified_environment" resolved to null
Info: Computing "applied_manifest" facter
Info: Contents of manifests/init.pp
Debug: Facter: fact "applied_manifest" has resolved to {
  manifest => "class aixautomation {
    vios { 'vios_health_check_only_2':
      ensure     => present,
      actions    => 'health',
      vios_pairs => '(p7jufv1,p7jufv2)',
    }
}",
  targets => [],
  vios => [
    "p7jufv1",
    "p7jufv2"
  ]
}.
```

Facteur ‘**hmc**’ produces following output showing the data related to HMC. As you can see below, facter ‘**hmc**’ is summarized into a yaml file :
/etc/puppetlabs/code/environments/production/modules/aixautomation/output/facter/hmc.yml



```
}.
Info: Computing "hmc" facter
Info: cmd: /usr/sbin/lsnim -t hmc -l
Debug: e08hmc2:
Debug:    class       = management
Debug:    type        = hmc
Debug:    passwd_file = /export/nim/dsm/passwd/e08hmc2.pswd
Debug:    manages     = 8246-L2C_10018FA
Debug:    manages     = 8246-L2C_100194A
Debug:    login       = hscroot
Debug:    keyexch1    = master
Debug:    mgmt_port   = 12443
Debug:    if1         = Network1 e08hmc2.aus.stglabs.ibm.com 0
Debug:    Cstate      = ready for a NIM operation
Debug:    prev_state  =
Debug:    Mstate      = currently running
Debug: b9rshmc:
Debug:    class       = management
Debug:    type        = hmc
Debug:    passwd_file = /export/nim/dsm/passwd/b9rshmc.pswd
Debug:    manages     = 8202-E4B 10DB09P
```

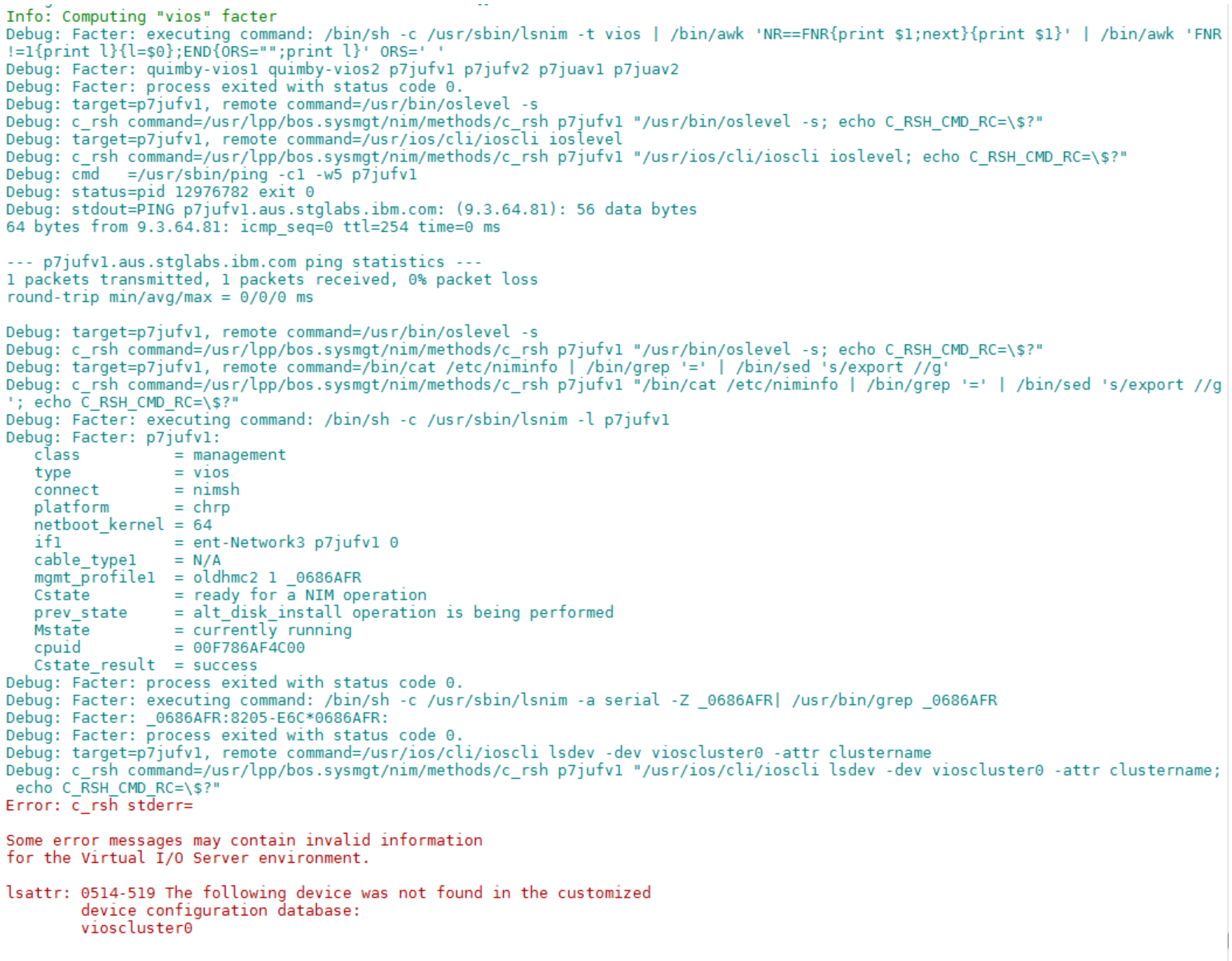
…



```
Debug:    manages     = p7juav1
Debug:    manages     = p7juav2
Debug:    mgmt_port   = 12443
Debug:    if1         = Network1 oldhmc2.aus.stglabs.ibm.com 0
Debug:    Cstate      = ready for a NIM operation
Debug:    prev_state  =
Debug:    Mstate      = currently running
Info: Refer to "/etc/puppetlabs/code/environments/production/modules/aixautomation/output/facter/hmc.yml" to have results of "hmc'
 facter.
Debug: Facter: fact "hmc" has resolved to {
  e08hmc2 => {
    passwd_file => "/export/nim/dsm/passwd/e08hmc2.pswd",
    login => "hscroot",
    ip => "e08hmc2.aus.stglabs.ibm.com",
    cstate => "ready for a NIM operation"
  },
  b9rshmc => {
    passwd_file => "/export/nim/dsm/passwd/b9rshmc.pswd",
    login => "hscroot",
    ip => "b9rshmc.aus.stglabs.ibm.com",
    cstate => "ready for a NIM operation"
  },
  b9rshmc2 => {
    passwd_file => "/export/nim/dsm/passwd/b9rshmc2.pswd",
    login => "hscroot",
    ip => "b9rshmc2.aus.stglabs.ibm.com",
    cstate => "ready for a NIM operation"
  },
  b9rshmc3 => {
    passwd_file => "/export/nim/dsm/passwd/b9rshmc3.pswd",
    login => "hscroot",
    ip => "b9rshmc3.aus.stglabs.ibm.com",
    cstate => "ready for a NIM operation"
  },
  oldhmc2 => {
    passwd_file => "/export/nim/oldhmc2.pwd",
    login => "hscroot",
    ip => "oldhmc2.aus.stglabs.ibm.com",
    cstate => "ready for a NIM operation"
  }
}.
```

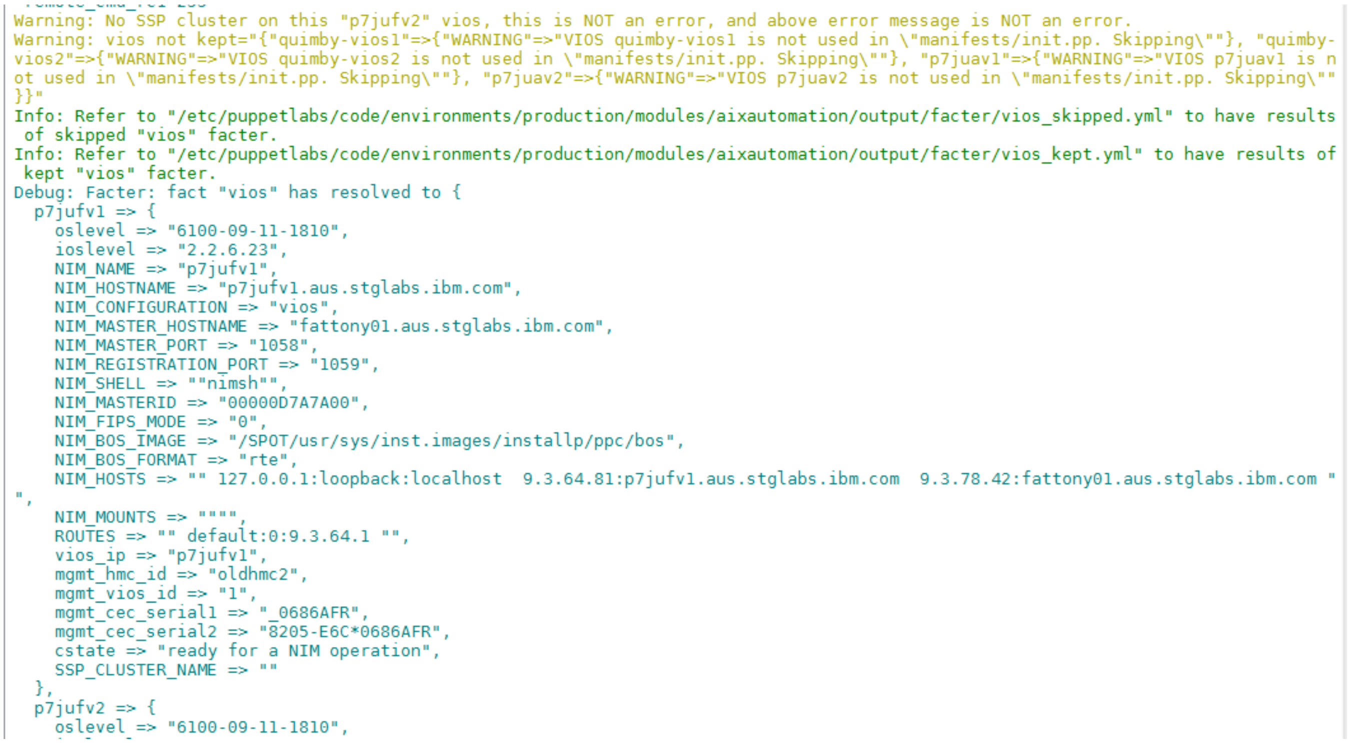
Facter '**vios'** produces following output showing the data related to VIOS. Each VIOS is checked against its belonging to a SSP cluster, and if it is not the case, this generates an error message (in red), which is not really an error. As you can see, '**vios**' facter is summarized into two yaml files, one for the VIOS which are skipped (not responding), and the second one for the VIOS which are kept (responding) :

- /etc/puppetlabs/code/environments/production/modules/aixautomation/output/facter/vios_skipped.yml
- /etc/puppetlabs/code/environments/production/modules/aixautomation/output/facter/vios_kept.yml



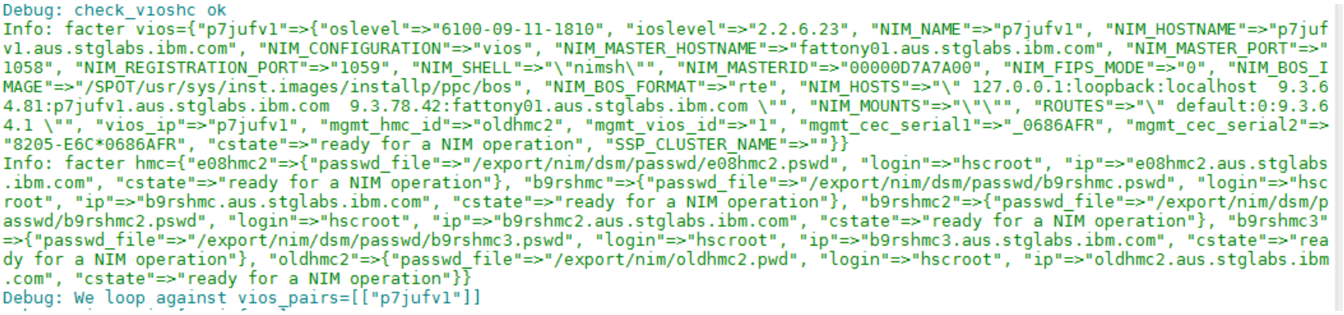
```
Info: Computing "vios" facter
Debug: Facter: executing command: /bin/sh -c /usr/sbin/lsnim -t vios | /bin/awk 'NR==FNR{print $1;next}{print $1}' | /bin/awk 'FNR
!=1{print l}{l=$0};END{ORS="";print l}' ORS=' '
Debug: Facter: quimby-vios1 quimby-vios2 p7jufv1 p7jufv2 p7juav1 p7juav2
Debug: Facter: process exited with status code 0.
Debug: target=p7jufv1, remote command=/usr/bin/oslevel -s
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh p7jufv1 "/usr/bin/oslevel -s; echo C_RSH_CMD_RC=\$?"
Debug: target=p7jufv1, remote command=/usr/ios/cli/ioscli ioslevel
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh p7jufv1 "/usr/ios/cli/ioscli ioslevel; echo C_RSH_CMD_RC=\$?"
Debug: cmd   =/usr/sbin/ping -c1 -w5 p7jufv1
Debug: status=pid 12976782 exit 0
Debug: stdout=PING p7jufv1.aus.stglabs.ibm.com: (9.3.64.81): 56 data bytes
64 bytes from 9.3.64.81: icmp_seq=0 ttl=254 time=0 ms

--- p7jufv1.aus.stglabs.ibm.com ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms

Debug: target=p7jufv1, remote command=/usr/bin/oslevel -s
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh p7jufv1 "/usr/bin/oslevel -s; echo C_RSH_CMD_RC=\$?"
Debug: target=p7jufv1, remote command=/bin/cat /etc/niminfo | /bin/grep '=' | /bin/sed 's/export //g'
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh p7jufv1 "/bin/cat /etc/niminfo | /bin/grep '=' | /bin/sed 's/export //g
'; echo C_RSH_CMD_RC=\$?"
Debug: Facter: executing command: /bin/sh -c /usr/sbin/lsnim -l p7jufv1
Debug: Facter: p7jufv1:
   class          = management
   type           = vios
   connect        = nimsh
   platform       = chrp
   netboot_kernel = 64
   if1            = ent-Network3 p7jufv1 0
   cable_type1    = N/A
   mgmt_profile1  = oldhmc2 1 _0686AFR
   Cstate         = ready for a NIM operation
   prev_state     = alt_disk_install operation is being performed
   Mstate         = currently running
   cpuid          = 00F786AF4C00
   Cstate_result  = success
Debug: Facter: process exited with status code 0.
Debug: Facter: executing command: /bin/sh -c /usr/sbin/lsnim -a serial -Z _0686AFR| /usr/bin/grep _0686AFR
Debug: Facter: _0686AFR:8205-E6C*0686AFR:
Debug: Facter: process exited with status code 0.
Debug: target=p7jufv1, remote command=/usr/ios/cli/ioscli lsdev -dev vioscluster0 -attr clustername
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh p7jufv1 "/usr/ios/cli/ioscli lsdev -dev vioscluster0 -attr clustername;
 echo C_RSH_CMD_RC=\$?"
Error: c_rsh stderr=

Some error messages may contain invalid information
for the Virtual I/O Server environment.

lsattr: 0514-519 The following device was not found in the customized
        device configuration database:
        vioscluster0
```

…



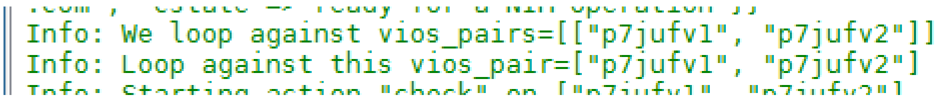
```
Warning: No SSP cluster on this "p7jufv2" vios, this is NOT an error, and above error message is NOT an error.
Warning: vios not kept="{"quimby-vios1"=>{"WARNING"=>"VIOS quimby-vios1 is not used in \"manifests/init.pp. Skipping\""}, "quimby-
vios2"=>{"WARNING"=>"VIOS quimby-vios2 is not used in \"manifests/init.pp. Skipping\""}, "p7juav1"=>{"WARNING"=>"VIOS p7juav1 is n
ot used in \"manifests/init.pp. Skipping\""}, "p7juav2"=>{"WARNING"=>"VIOS p7juav2 is not used in \"manifests/init.pp. Skipping\""
}}"
Info: Refer to "/etc/puppetlabs/code/environments/production/modules/aixautomation/output/facter/vios_skipped.yml" to have results
 of skipped "vios" facter.
Info: Refer to "/etc/puppetlabs/code/environments/production/modules/aixautomation/output/facter/vios_kept.yml" to have results of
 kept "vios" facter.
Debug: Facter: fact "vios" has resolved to {
  p7jufv1 => {
    oslevel => "6100-09-11-1810",
    ioslevel => "2.2.6.23",
    NIM_NAME => "p7jufv1",
    NIM_HOSTNAME => "p7jufv1.aus.stglabs.ibm.com",
    NIM_CONFIGURATION => "vios",
    NIM_MASTER_HOSTNAME => "fattony01.aus.stglabs.ibm.com",
    NIM_MASTER_PORT => "1058",
    NIM_REGISTRATION_PORT => "1059",
    NIM_SHELL => ""nimsh"",
    NIM_MASTERID => "00000D7A7A00",
    NIM_FIPS_MODE => "0",
    NIM_BOS_IMAGE => "/SPOT/usr/sys/inst.images/installp/ppc/bos",
    NIM_BOS_FORMAT => "rte",
    NIM_HOSTS => "" 127.0.0.1:loopback:localhost  9.3.64.81:p7jufv1.aus.stglabs.ibm.com  9.3.78.42:fattony01.aus.stglabs.ibm.com "
",
    NIM_MOUNTS => """",
    ROUTES => "" default:0:9.3.64.1 "",
    vios_ip => "p7jufv1",
    mgmt_hmc_id => "oldhmc2",
    mgmt_vios_id => "1",
    mgmt_cec_serial1 => "_0686AFR",
    mgmt_cec_serial2 => "8205-E6C*0686AFR",
    cstate => "ready for a NIM operation",
    SSP_CLUSTER_NAME => ""
  },
  p7jufv2 => {
    oslevel => "6100-09-11-1810",
```

'**vios**' facter is summarized by the following oputput:



```
Debug: check_vioshc ok
Info: facter vios={"p7jufv1"=>{"oslevel"=>"6100-09-11-1810", "ioslevel"=>"2.2.6.23", "NIM_NAME"=>"p7jufv1", "NIM_HOSTNAME"=>"p7juf
v1.aus.stglabs.ibm.com", "NIM_CONFIGURATION"=>"vios", "NIM_MASTER_HOSTNAME"=>"fattony01.aus.stglabs.ibm.com", "NIM_MASTER_PORT"=>"
1058", "NIM_REGISTRATION_PORT"=>"1059", "NIM_SHELL"=>"\"nimsh\"", "NIM_MASTERID"=>"00000D7A7A00", "NIM_FIPS_MODE"=>"0", "NIM_BOS_I
MAGE"=>"/SPOT/usr/sys/inst.images/installp/ppc/bos", "NIM_BOS_FORMAT"=>"rte", "NIM_HOSTS"=>"\" 127.0.0.1:loopback:localhost  9.3.6
4.81:p7jufv1.aus.stglabs.ibm.com  9.3.78.42:fattony01.aus.stglabs.ibm.com \"", "NIM_MOUNTS"=>"\"\"", "ROUTES"=>"\" default:0:9.3.6
4.1 \"", "vios_ip"=>"p7jufv1", "mgmt_hmc_id"=>"oldhmc2", "mgmt_vios_id"=>"1", "mgmt_cec_serial1"=>"_0686AFR", "mgmt_cec_serial2"=>
"8205-E6C*0686AFR", "cstate"=>"ready for a NIM operation", "SSP_CLUSTER_NAME"=>""}}
Info: facter hmc={"e08hmc2"=>{"passwd_file"=>"/export/nim/dsm/passwd/e08hmc2.pswd", "login"=>"hscroot", "ip"=>"e08hmc2.aus.stglabs
.ibm.com", "cstate"=>"ready for a NIM operation"}, "b9rshmc"=>{"passwd_file"=>"/export/nim/dsm/passwd/b9rshmc.pswd", "login"=>"hsc
root", "ip"=>"b9rshmc.aus.stglabs.ibm.com", "cstate"=>"ready for a NIM operation"}, "b9rshmc2"=>{"passwd_file"=>"/export/nim/dsm/p
asswd/b9rshmc2.pswd", "login"=>"hscroot", "ip"=>"b9rshmc2.aus.stglabs.ibm.com", "cstate"=>"ready for a NIM operation"}, "b9rshmc3"
=>{"passwd_file"=>"/export/nim/dsm/passwd/b9rshmc3.pswd", "login"=>"hscroot", "ip"=>"b9rshmc3.aus.stglabs.ibm.com", "cstate"=>"rea
dy for a NIM operation"}, "oldhmc2"=>{"passwd_file"=>"/export/nim/oldhmc2.pwd", "login"=>"hscroot", "ip"=>"oldhmc2.aus.stglabs.ibm
.com", "cstate"=>"ready for a NIM operation"}}
Debug: We loop against vios_pairs=[["p7jufv1"]]
```

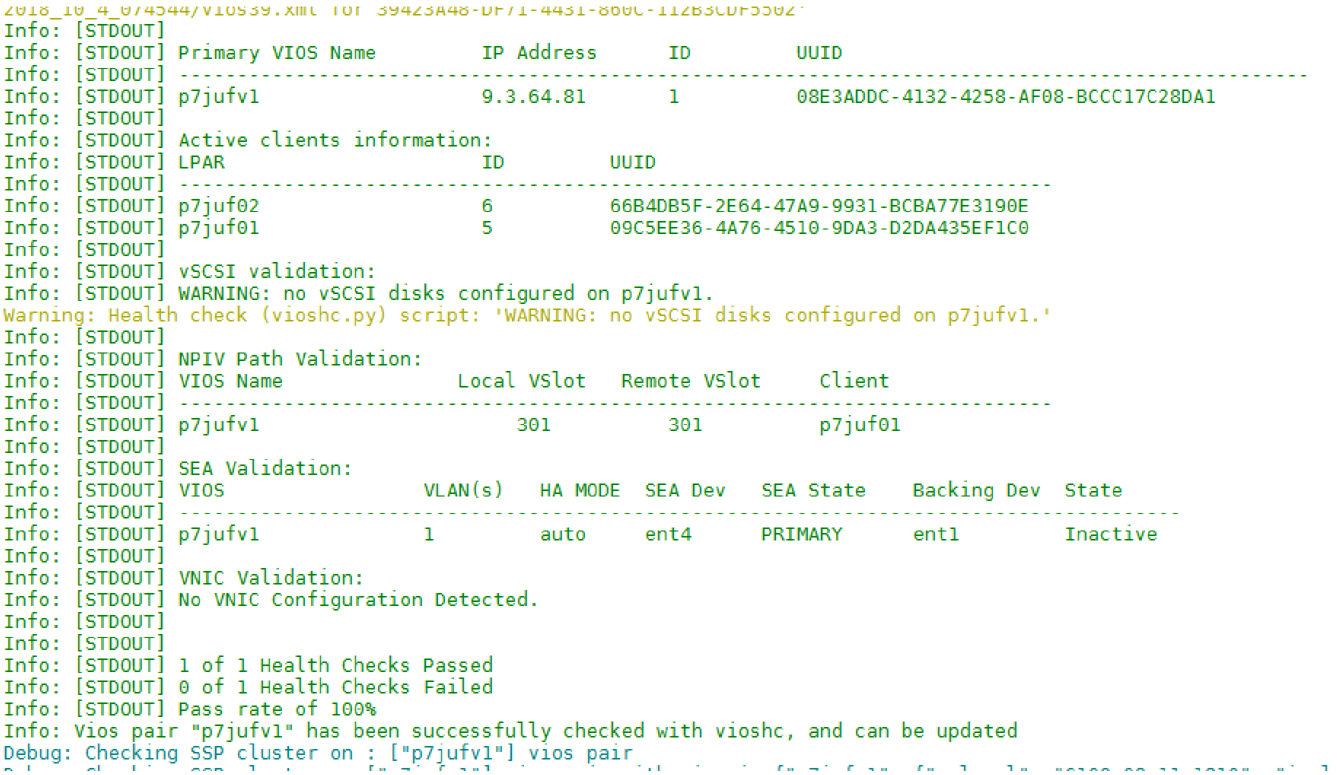
After the facters being done, actions declared into manifests/init.pp file are run. As explained we loop against VIOS pairs, and then for each pair, we loop against the two VIOS.



```
Info: We loop against vios_pairs=[["p7jufv1", "p7jufv2"]]
Info: Loop against this vios_pair=["p7jufv1", "p7jufv2"]
```

The '**health**' action is performed in two steps: health init and health check, which are illustrated below.

Health init retrieves UUID from HMC by using vioshc.py (result is persisted into yaml file). Health check against a single VIOS produces something like that:

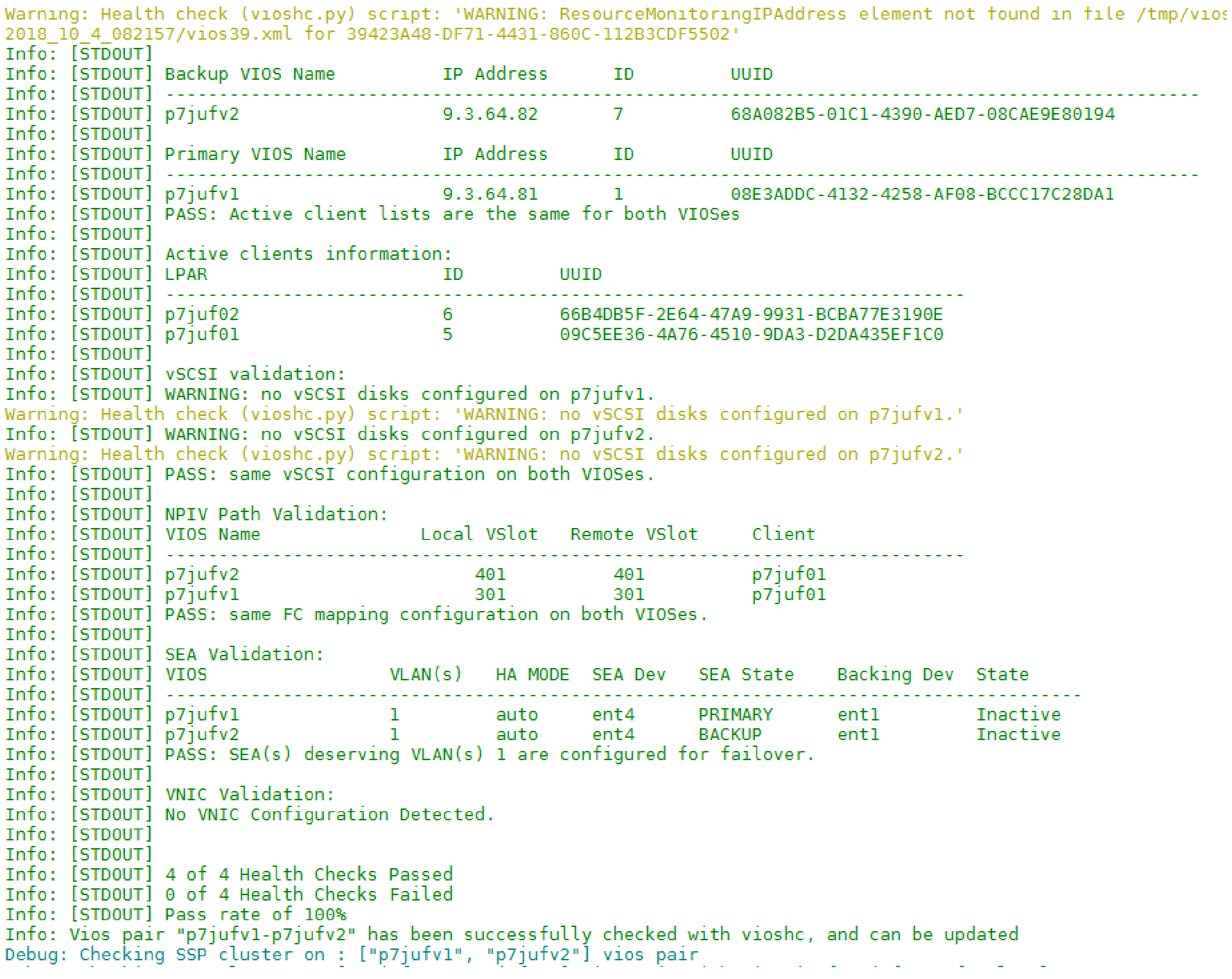


```
2018_10_4_074544/vios39.xml for 39423A48-DF71-4431-860C-112B3CDF5502'
Info: [STDOUT]
Info: [STDOUT] Primary VIOS Name         IP Address      ID         UUID
Info: [STDOUT] ------------------------------------------------------------------------------------------------
Info: [STDOUT] p7jufv1                   9.3.64.81       1          08E3ADDC-4132-4258-AF08-BCCC17C28DA1
Info: [STDOUT]
Info: [STDOUT] Active clients information:
Info: [STDOUT] LPAR                      ID         UUID
Info: [STDOUT] ---------------------------------------------------------------------------
Info: [STDOUT] p7juf02                   6          66B4DB5F-2E64-47A9-9931-BCBA77E3190E
Info: [STDOUT] p7juf01                   5          09C5EE36-4A76-4510-9DA3-D2DA435EF1C0
Info: [STDOUT]
Info: [STDOUT] vSCSI validation:
Info: [STDOUT] WARNING: no vSCSI disks configured on p7jufv1.
Warning: Health check (vioshc.py) script: 'WARNING: no vSCSI disks configured on p7jufv1.'
Info: [STDOUT]
Info: [STDOUT] NPIV Path Validation:
Info: [STDOUT] VIOS Name               Local VSlot   Remote VSlot     Client
Info: [STDOUT] ---------------------------------------------------------------------------
Info: [STDOUT] p7jufv1                      301          301          p7juf01
Info: [STDOUT]
Info: [STDOUT] SEA Validation:
Info: [STDOUT] VIOS                 VLAN(s)   HA MODE  SEA Dev   SEA State    Backing Dev  State
Info: [STDOUT] --------------------------------------------------------------------------------------
Info: [STDOUT] p7jufv1              1         auto     ent4      PRIMARY      ent1         Inactive
Info: [STDOUT]
Info: [STDOUT] VNIC Validation:
Info: [STDOUT] No VNIC Configuration Detected.
Info: [STDOUT]
Info: [STDOUT]
Info: [STDOUT] 1 of 1 Health Checks Passed
Info: [STDOUT] 0 of 1 Health Checks Failed
Info: [STDOUT] Pass rate of 100%
Info: Vios pair "p7jufv1" has been successfully checked with vioshc, and can be updated
Debug: Checking SSP cluster on : ["p7jufv1"] vios pair
```

It might be necessary to unset proxy from your console prior from running health check:

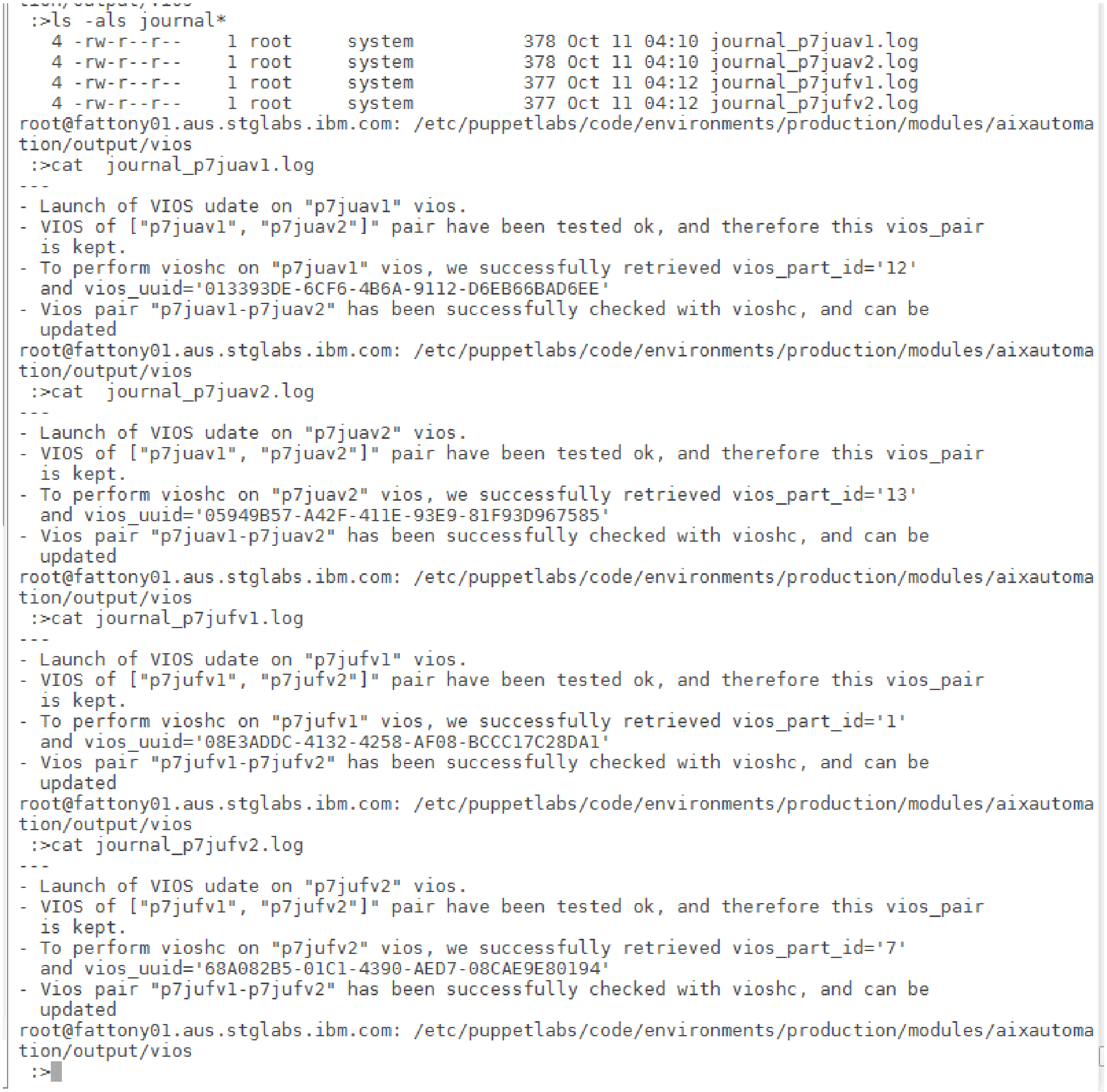
```
# unset http_proxy
# unset https_proxy
# unset HTTP_PROXY
# unset HTTPS_PROXY
```

Health check against a VIOS pair produces following output:



```
Warning: Health check (vioshc.py) script: 'WARNING: ResourceMonitoringIPAddress element not found in file /tmp/vios
2018_10_4_082157/vios39.xml for 39423A48-DF71-4431-860C-112B3CDF5502'
Info: [STDOUT]
Info: [STDOUT] Backup VIOS Name          IP Address      ID         UUID
Info: [STDOUT] -------------------------------------------------------------------------------------------------
Info: [STDOUT] p7jufv2                   9.3.64.82       7          68A082B5-01C1-4390-AED7-08CAE9E80194
Info: [STDOUT]
Info: [STDOUT] Primary VIOS Name         IP Address      ID         UUID
Info: [STDOUT] -------------------------------------------------------------------------------------------------
Info: [STDOUT] p7jufv1                   9.3.64.81       1          08E3ADDC-4132-4258-AF08-BCCC17C28DA1
Info: [STDOUT] PASS: Active client lists are the same for both VIOSes
Info: [STDOUT]
Info: [STDOUT] Active clients information:
Info: [STDOUT] LPAR                      ID         UUID
Info: [STDOUT] ---------------------------------------------------------------------------
Info: [STDOUT] p7juf02                   6          66B4DB5F-2E64-47A9-9931-BCBA77E3190E
Info: [STDOUT] p7juf01                   5          09C5EE36-4A76-4510-9DA3-D2DA435EF1C0
Info: [STDOUT]
Info: [STDOUT] vSCSI validation:
Info: [STDOUT] WARNING: no vSCSI disks configured on p7jufv1.
Warning: Health check (vioshc.py) script: 'WARNING: no vSCSI disks configured on p7jufv1.'
Info: [STDOUT] WARNING: no vSCSI disks configured on p7jufv2.
Warning: Health check (vioshc.py) script: 'WARNING: no vSCSI disks configured on p7jufv2.'
Info: [STDOUT] PASS: same vSCSI configuration on both VIOSes.
Info: [STDOUT]
Info: [STDOUT] NPIV Path Validation:
Info: [STDOUT] VIOS Name               Local VSlot   Remote VSlot     Client
Info: [STDOUT] ---------------------------------------------------------------------------
Info: [STDOUT] p7jufv2                      401           401          p7juf01
Info: [STDOUT] p7jufv1                      301           301          p7juf01
Info: [STDOUT] PASS: same FC mapping configuration on both VIOSes.
Info: [STDOUT]
Info: [STDOUT] SEA Validation:
Info: [STDOUT] VIOS                 VLAN(s)   HA MODE  SEA Dev   SEA State    Backing Dev  State
Info: [STDOUT] --------------------------------------------------------------------------------------
Info: [STDOUT] p7jufv1              1         auto     ent4      PRIMARY      ent1         Inactive
Info: [STDOUT] p7jufv2              1         auto     ent4      BACKUP       ent1         Inactive
Info: [STDOUT] PASS: SEA(s) deserving VLAN(s) 1 are configured for failover.
Info: [STDOUT]
Info: [STDOUT] VNIC Validation:
Info: [STDOUT] No VNIC Configuration Detected.
Info: [STDOUT]
Info: [STDOUT]
Info: [STDOUT] 4 of 4 Health Checks Passed
Info: [STDOUT] 0 of 4 Health Checks Failed
Info: [STDOUT] Pass rate of 100%
Info: Vios pair "p7jufv1-p7jufv2" has been successfully checked with vioshc, and can be updated
Debug: Checking SSP cluster on : ["p7jufv1", "p7jufv2"] vios pair
```

Journalized output files contain summary of what was done on each VIOS:



```
tion/output/vios
 :>ls -als journal*
   4 -rw-r--r--    1 root     system          378 Oct 11 04:10 journal_p7juav1.log
   4 -rw-r--r--    1 root     system          378 Oct 11 04:10 journal_p7juav2.log
   4 -rw-r--r--    1 root     system          377 Oct 11 04:12 journal_p7jufv1.log
   4 -rw-r--r--    1 root     system          377 Oct 11 04:12 journal_p7jufv2.log
root@fattony01.aus.stglabs.ibm.com: /etc/puppetlabs/code/environments/production/modules/aixautoma
tion/output/vios
 :>cat  journal_p7juav1.log
---
- Launch of VIOS udate on "p7juav1" vios.
- VIOS of ["p7juav1", "p7juav2"]" pair have been tested ok, and therefore this vios_pair
  is kept.
- To perform vioshc on "p7juav1" vios, we successfully retrieved vios_part_id='12'
  and vios_uuid='013393DE-6CF6-4B6A-9112-D6EB66BAD6EE'
- Vios pair "p7juav1-p7juav2" has been successfully checked with vioshc, and can be
  updated
root@fattony01.aus.stglabs.ibm.com: /etc/puppetlabs/code/environments/production/modules/aixautoma
tion/output/vios
 :>cat  journal_p7juav2.log
---
- Launch of VIOS udate on "p7juav2" vios.
- VIOS of ["p7juav1", "p7juav2"]" pair have been tested ok, and therefore this vios_pair
  is kept.
- To perform vioshc on "p7juav2" vios, we successfully retrieved vios_part_id='13'
  and vios_uuid='05949B57-A42F-411E-93E9-81F93D967585'
- Vios pair "p7juav1-p7juav2" has been successfully checked with vioshc, and can be
  updated
root@fattony01.aus.stglabs.ibm.com: /etc/puppetlabs/code/environments/production/modules/aixautoma
tion/output/vios
 :>cat journal_p7jufv1.log
---
- Launch of VIOS udate on "p7jufv1" vios.
- VIOS of ["p7jufv1", "p7jufv2"]" pair have been tested ok, and therefore this vios_pair
  is kept.
- To perform vioshc on "p7jufv1" vios, we successfully retrieved vios_part_id='1'
  and vios_uuid='08E3ADDC-4132-4258-AF08-BCCC17C28DA1'
- Vios pair "p7jufv1-p7jufv2" has been successfully checked with vioshc, and can be
  updated
root@fattony01.aus.stglabs.ibm.com: /etc/puppetlabs/code/environments/production/modules/aixautoma
tion/output/vios
 :>cat journal_p7jufv2.log
---
- Launch of VIOS udate on "p7jufv2" vios.
- VIOS of ["p7jufv1", "p7jufv2"]" pair have been tested ok, and therefore this vios_pair
  is kept.
- To perform vioshc on "p7jufv2" vios, we successfully retrieved vios_part_id='7'
  and vios_uuid='68A082B5-01C1-4390-AED7-08CAE9E80194'
- Vios pair "p7jufv1-p7jufv2" has been successfully checked with vioshc, and can be
  updated
root@fattony01.aus.stglabs.ibm.com: /etc/puppetlabs/code/environments/production/modules/aixautoma
tion/output/vios
 :>
```

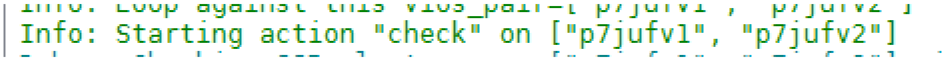
Second example of manifest/init.pp to perform 'save' on a VIOS pair:

Class automation to encapsulate the declaration

```
class aixautomation {
 vios { 'vios_check_save_2':
   actions    => 'check,gc,save',
   vios_pairs => '(p7jufv1, p7jufv2)',
  altinst_rootvg_force => 'yes',
 }
}
```

Attributes of custom type 'vios': this declaration called 'vios_check_save_2' launches the 'check' and the 'save' on a VIOS pair.

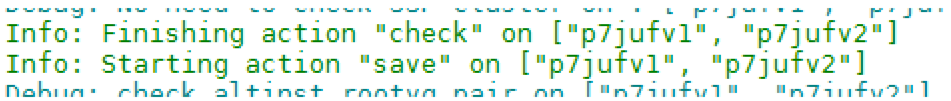
Action 'check' starts with this line



```
Info: Starting action "check" on ["p7jufv1", "p7jufv2"]
```

The 'check' action verifies the presence and states of a potential SSP cluster.

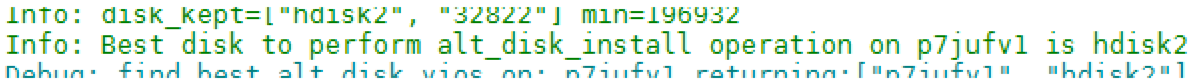
And finishes with this line. Action 'save' starts after:



```
Info: Finishing action "check" on ["p7jufv1", "p7jufv2"]
Info: Starting action "save" on ["p7jufv1", "p7jufv2"]
Debug: check altinst rootvg pair on ["p7jufv1", "p7jufv2"]
```

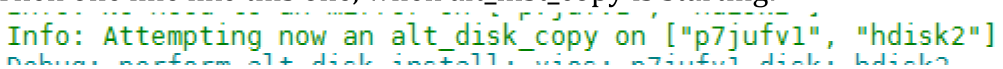
Some logic is run then to find best disk to perform 'save' action: an available disk, large enough to host a copy of rootvg, and if several disks fit, to choose the smallest one of these ones.
Depending on the values of 'altinst_rootvg_force' 'vios_altinst_rootvg' attributes logic run is not the same.
By setting **altinst_rootvg_force =>"yes",** an already existing **"altinst_rootvg"** will be overridden by a new one.

You'll find one line like this one per VIOS, when logic to find best disk to host altinst_rootvg is done:



```
Info: disk_kept=["hdisk2", "32822"] min=196932
Info: Best disk to perform alt_disk_install operation on p7jufv1 is hdisk2
Debug: find best alt disk vios on: p7jufv1 returning:["p7jufv1", "hdisk2"]
```

Then one line like this one, when alt_inst_copy is starting:



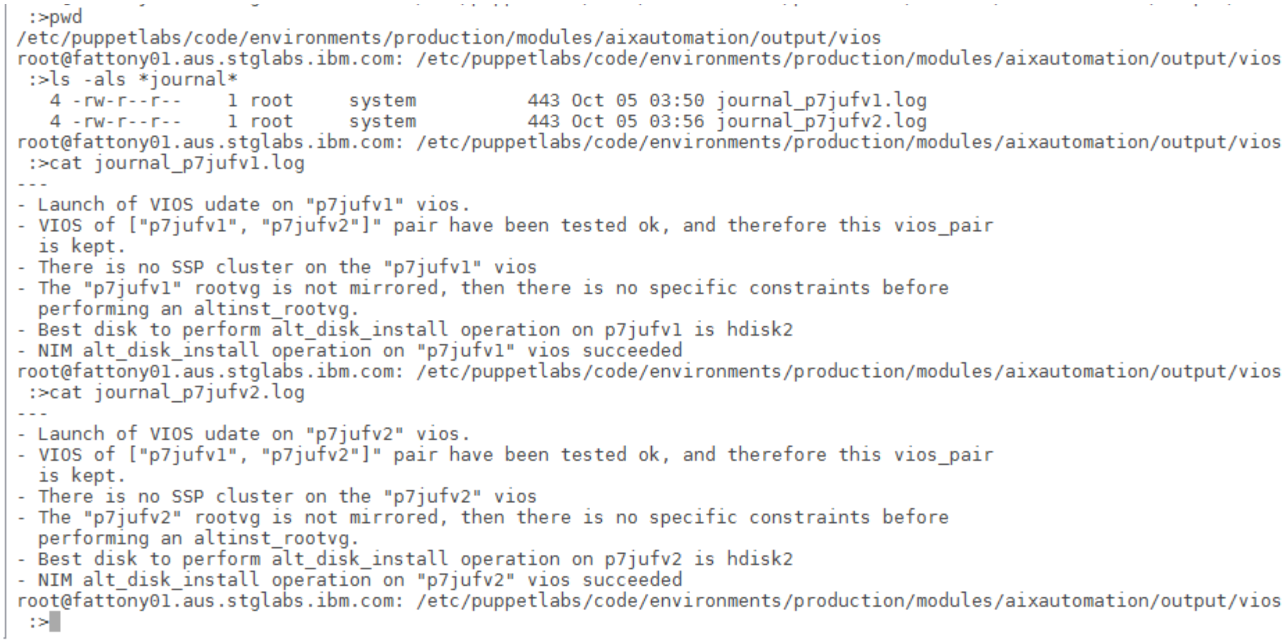
```
Info: Attempting now an alt_disk_copy on ["p7jufv1", "hdisk2"]
Debug: perform alt disk install: vios: p7jufv1 disk: hdisk2
```

Then on line like this one, when alt_inst_copy is done:

```
Info: NIM alt_disk_install operation on "p7jufv1" vios succeeded
```

At the end, all operations on each VIOS are summarized into journal files.
In our example, this journal files are:

- /etc/puppetlabs/code/environments/production/modules/aixautomation/output/vios/journal_p7jufv1.log
- /etc/puppetlabs/code/environments/production/modules/aixautomation/output/vios/journal_p7jufv2.log



```
 :>pwd
/etc/puppetlabs/code/environments/production/modules/aixautomation/output/vios
root@fattony01.aus.stglabs.ibm.com: /etc/puppetlabs/code/environments/production/modules/aixautomation/output/vios
 :>ls -als *journal*
   4 -rw-r--r--    1 root     system          443 Oct 05 03:50 journal_p7jufv1.log
   4 -rw-r--r--    1 root     system          443 Oct 05 03:56 journal_p7jufv2.log
root@fattony01.aus.stglabs.ibm.com: /etc/puppetlabs/code/environments/production/modules/aixautomation/output/vios
 :>cat journal_p7jufv1.log
---
- Launch of VIOS udate on "p7jufv1" vios.
- VIOS of ["p7jufv1", "p7jufv2"]" pair have been tested ok, and therefore this vios_pair
  is kept.
- There is no SSP cluster on the "p7jufv1" vios
- The "p7jufv1" rootvg is not mirrored, then there is no specific constraints before
  performing an altinst_rootvg.
- Best disk to perform alt_disk_install operation on p7jufv1 is hdisk2
- NIM alt_disk_install operation on "p7jufv1" vios succeeded
root@fattony01.aus.stglabs.ibm.com: /etc/puppetlabs/code/environments/production/modules/aixautomation/output/vios
 :>cat journal_p7jufv2.log
---
- Launch of VIOS udate on "p7jufv2" vios.
- VIOS of ["p7jufv1", "p7jufv2"]" pair have been tested ok, and therefore this vios_pair
  is kept.
- There is no SSP cluster on the "p7jufv2" vios
- The "p7jufv2" rootvg is not mirrored, then there is no specific constraints before
  performing an altinst_rootvg.
- Best disk to perform alt_disk_install operation on p7jufv2 is hdisk2
- NIM alt_disk_install operation on "p7jufv2" vios succeeded
root@fattony01.aus.stglabs.ibm.com: /etc/puppetlabs/code/environments/production/modules/aixautomation/output/vios
 :>
```

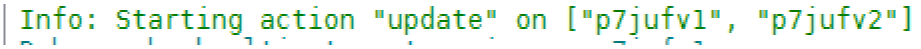
Third example of manifest/init.pp to perform ‘update’ on a VIOS pair:

Class automation to encapsulate the declaration

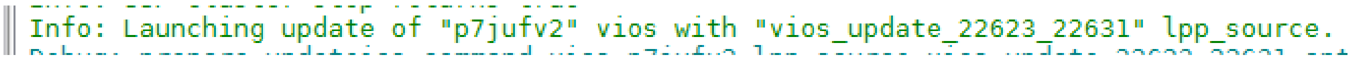
```
class aixautomation {
 vios { 'vios_update_3':
   actions    => 'check,gc,save,unmirror,autocommit,update',
   vios_pairs => '(p7jufv1, p7jufv2)',
   altinst_rootvg_force => 'yes',
   vios_lpp_sources => 'p7jufv1=vios_update_22623_22631,
p7jufv2=vios_update_22623_22631',
  }
}
```

Attributes of custom type ‘vios’: this declaration called ‘vios_update_3’ launches the ‘update’ on one VIOS pair. ‘check’, ‘gc’, ‘save’, ‘unmirror’, ’autocommit’ are necessary. ‘unmirror’ is necessary if VIOS rootvg is mirrored.

Launching ‘VIOS update’ step on VIOS pair, it loops on VIOS:

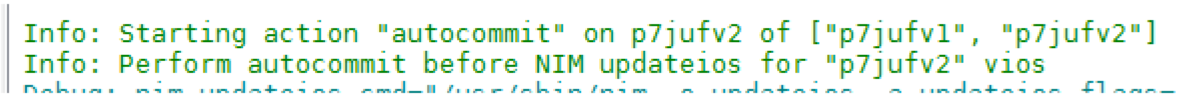


```
Info: Starting action "update" on ["p7jufv1", "p7jufv2"]
```

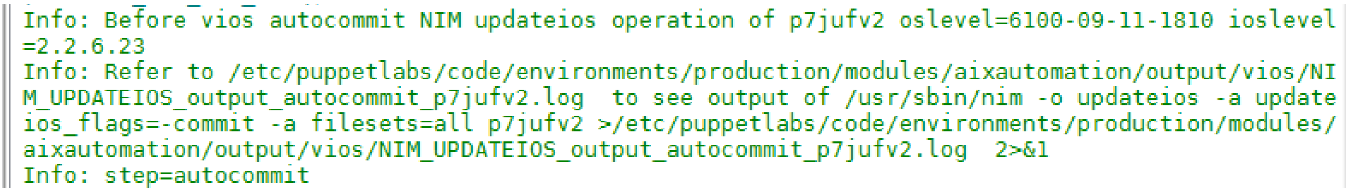


```
Info: Launching update of "p7jufv2" vios with "vios_update_22623_22631" lpp_source.
```

Before ‘autocommit’ step:

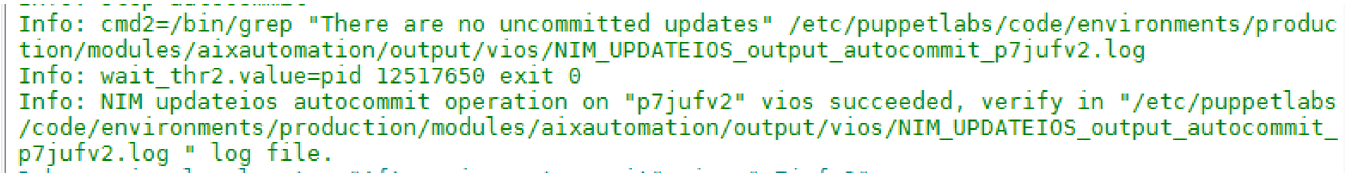


```
Info: Starting action "autocommit" on p7jufv2 of ["p7jufv1", "p7jufv2"]
Info: Perform autocommit before NIM updateios for "p7jufv2" vios
```

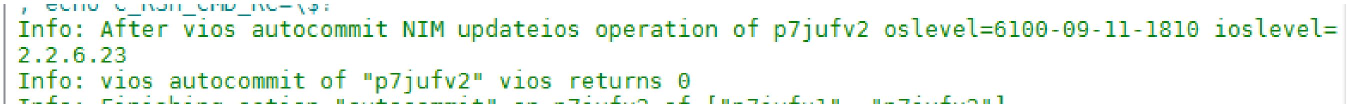


```
Info: Before vios autocommit NIM updateios operation of p7jufv2 oslevel=6100-09-11-1810 ioslevel
=2.2.6.23
Info: Refer to /etc/puppetlabs/code/environments/production/modules/aixautomation/output/vios/NI
M_UPDATEIOS_output_autocommit_p7jufv2.log  to see output of /usr/sbin/nim -o updateios -a update
ios_flags=-commit -a filesets=all p7jufv2 >/etc/puppetlabs/code/environments/production/modules/
aixautomation/output/vios/NIM_UPDATEIOS_output_autocommit_p7jufv2.log  2>&1
Info: step=autocommit
```

After ‘autocommit’ step:

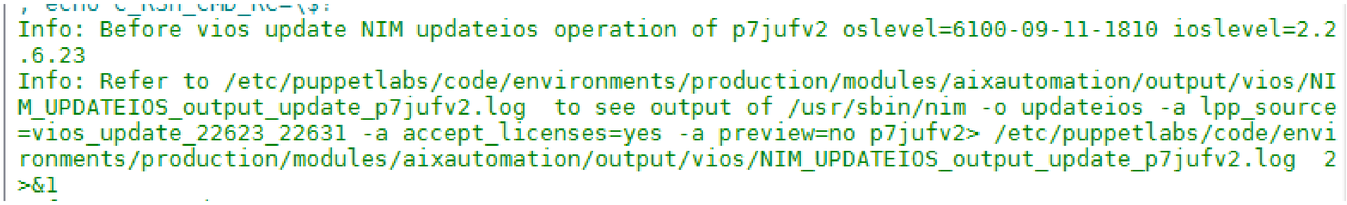


```
Info: cmd2=/bin/grep "There are no uncommitted updates" /etc/puppetlabs/code/environments/produc
tion/modules/aixautomation/output/vios/NIM_UPDATEIOS_output_autocommit_p7jufv2.log
Info: wait_thr2.value=pid 12517650 exit 0
Info: NIM updateios autocommit operation on "p7jufv2" vios succeeded, verify in "/etc/puppetlabs
/code/environments/production/modules/aixautomation/output/vios/NIM_UPDATEIOS_output_autocommit_
p7jufv2.log " log file.
```



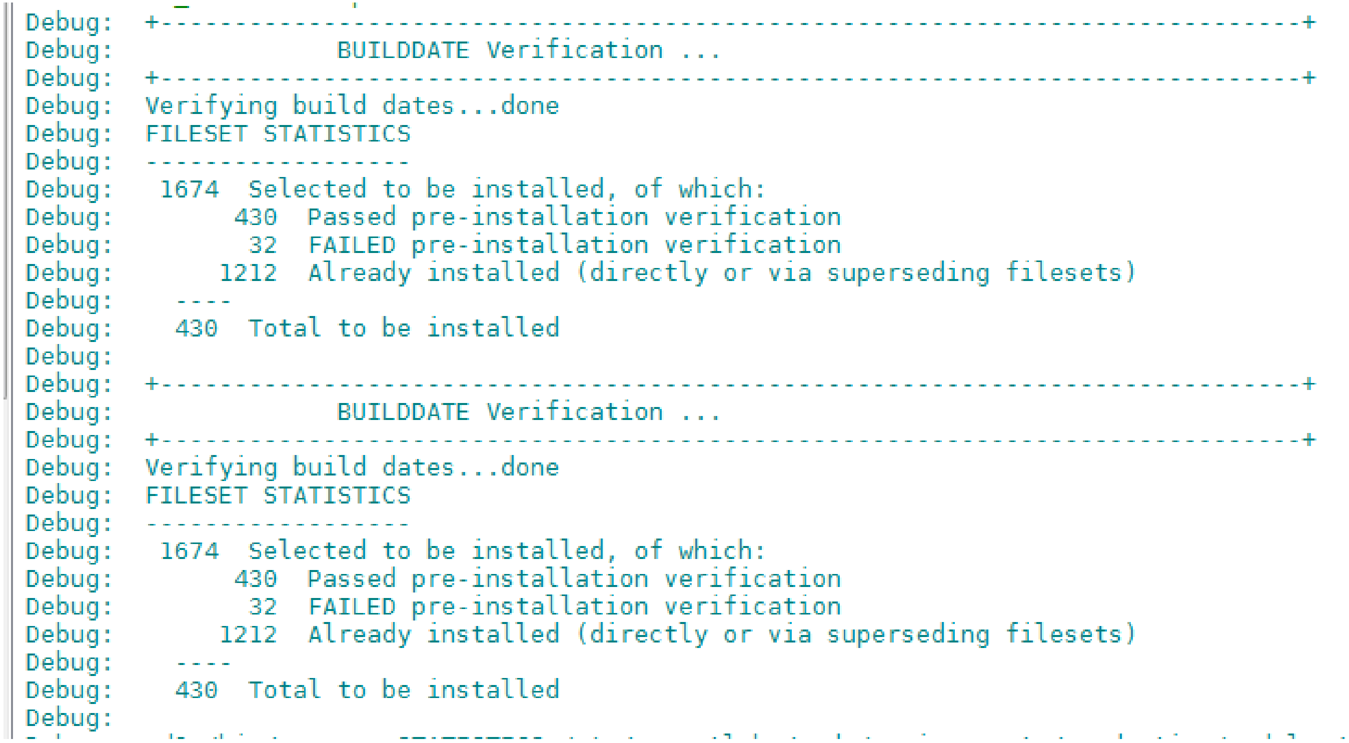
```
Info: After vios autocommit NIM updateios operation of p7jufv2 oslevel=6100-09-11-1810 ioslevel=
2.2.6.23
Info: vios autocommit of "p7jufv2" vios returns 0
```

Before ‘update’ step, ioslevel is displayed, and command run to perform NIM update as well:



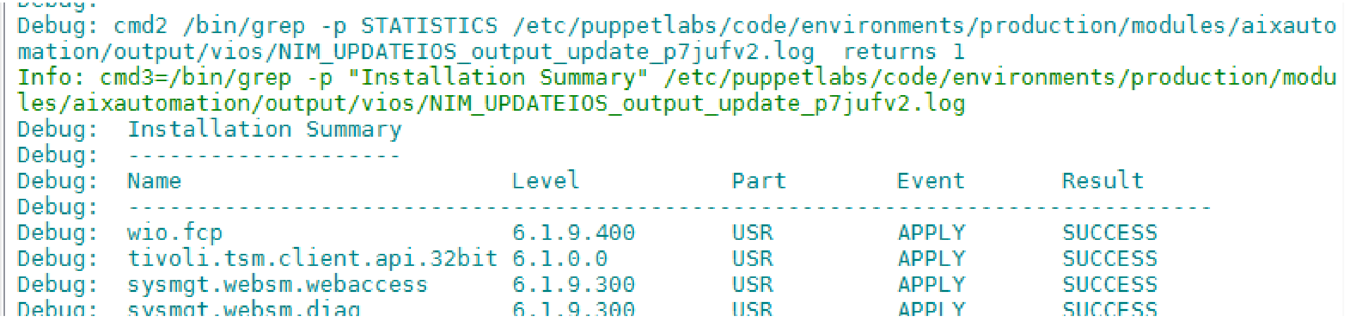
```
; echo C_RSH_CMD_RC=\$?
Info: Before vios update NIM updateios operation of p7jufv2 oslevel=6100-09-11-1810 ioslevel=2.2
.6.23
Info: Refer to /etc/puppetlabs/code/environments/production/modules/aixautomation/output/vios/NI
M_UPDATEIOS_output_update_p7jufv2.log  to see output of /usr/sbin/nim -o updateios -a lpp_source
=vios_update_22623_22631 -a accept_licenses=yes -a preview=no p7jufv2> /etc/puppetlabs/code/envi
ronments/production/modules/aixautomation/output/vios/NIM_UPDATEIOS_output_update_p7jufv2.log  2
>&1
```

A first heuristic applied is to catch if any remaining files needs to be installed:



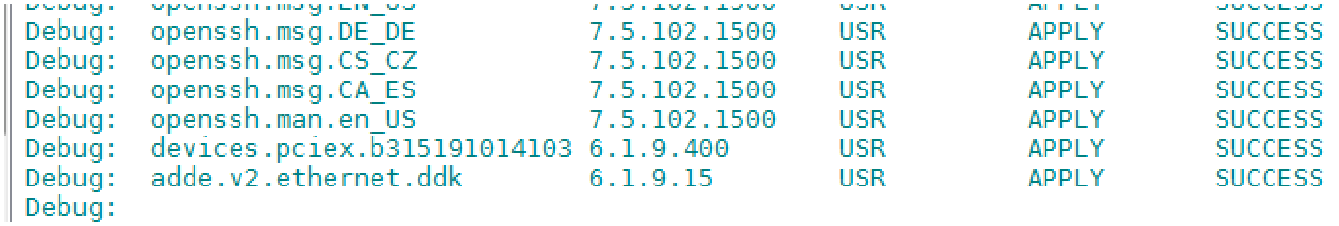
```
Debug:  +-----------------------------------------------------------------------------+
Debug:                    BUILDDATE Verification ...
Debug:  +-----------------------------------------------------------------------------+
Debug:  Verifying build dates...done
Debug:  FILESET STATISTICS
Debug:  ------------------
Debug:   1674  Selected to be installed, of which:
Debug:        430  Passed pre-installation verification
Debug:         32  FAILED pre-installation verification
Debug:       1212  Already installed (directly or via superseding filesets)
Debug:    ----
Debug:    430  Total to be installed
Debug:
Debug:  +-----------------------------------------------------------------------------+
Debug:                    BUILDDATE Verification ...
Debug:  +-----------------------------------------------------------------------------+
Debug:  Verifying build dates...done
Debug:  FILESET STATISTICS
Debug:  ------------------
Debug:   1674  Selected to be installed, of which:
Debug:        430  Passed pre-installation verification
Debug:         32  FAILED pre-installation verification
Debug:       1212  Already installed (directly or via superseding filesets)
Debug:    ----
Debug:    430  Total to be installed
Debug:
```

Installation summary is displayed:



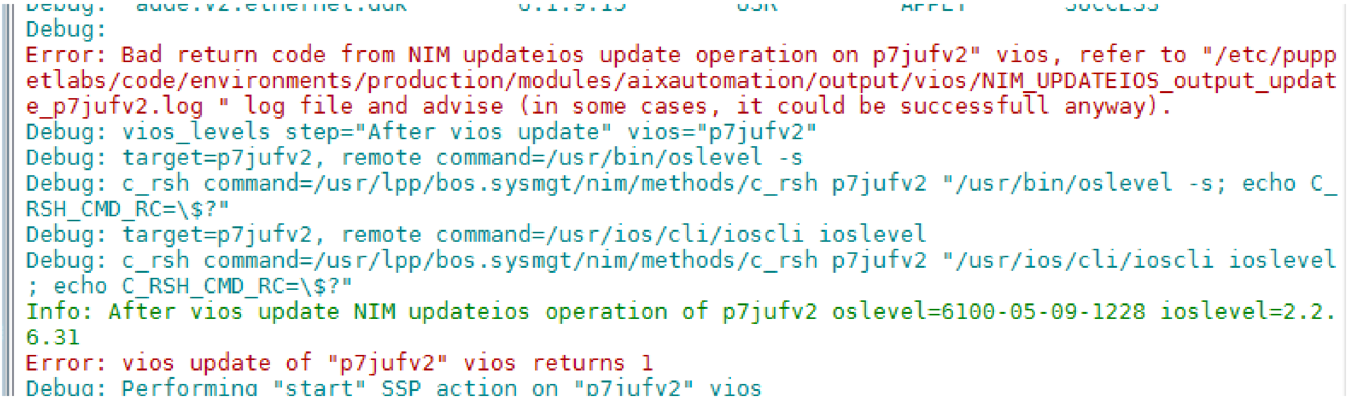
```
Debug:
Debug: cmd2 /bin/grep -p STATISTICS /etc/puppetlabs/code/environments/production/modules/aixauto
mation/output/vios/NIM_UPDATEIOS_output_update_p7jufv2.log  returns 1
Info: cmd3=/bin/grep -p "Installation Summary" /etc/puppetlabs/code/environments/production/modu
les/aixautomation/output/vios/NIM_UPDATEIOS_output_update_p7jufv2.log
Debug:  Installation Summary
Debug:  --------------------
Debug:  Name                        Level           Part        Event       Result
Debug:  -------------------------------------------------------------------------------
Debug:  wio.fcp                     6.1.9.400       USR         APPLY       SUCCESS
Debug:  tivoli.tsm.client.api.32bit 6.1.0.0         USR         APPLY       SUCCESS
Debug:  sysmgt.websm.webaccess      6.1.9.300       USR         APPLY       SUCCESS
Debug:  sysmgt.websm.diag           6.1.9.300       USR         APPLY       SUCCESS
```

…. very long output …



```
Debug:  openssh.msg.EN_US           7.5.102.1500  USR       APPLY     SUCCESS
Debug:  openssh.msg.DE_DE           7.5.102.1500  USR       APPLY     SUCCESS
Debug:  openssh.msg.CS_CZ           7.5.102.1500  USR       APPLY     SUCCESS
Debug:  openssh.msg.CA_ES           7.5.102.1500  USR       APPLY     SUCCESS
Debug:  openssh.man.en_US           7.5.102.1500  USR       APPLY     SUCCESS
Debug:  devices.pciex.b315191014103 6.1.9.400     USR       APPLY     SUCCESS
Debug:  adde.v2.ethernet.ddk        6.1.9.15      USR       APPLY     SUCCESS
Debug:
```

At the end of update, return code and ioslevel are displayed (see below). In that case ioslevel is right, even if return code is wrong (it is wrongly set by NIM to 1, although behavior is correct), an explanation indicates to refer to log file to have status of update. Referring to this file enables to verify that NIM updateios behavior is successful.



```
Debug:  adde.v2.ethernet.ddk        6.1.9.15      USR       APPLY     SUCCESS
Debug:
Error: Bad return code from NIM updateios update operation on p7jufv2" vios, refer to "/etc/pupp
etlabs/code/environments/production/modules/aixautomation/output/vios/NIM_UPDATEIOS_output_updat
e_p7jufv2.log " log file and advise (in some cases, it could be successfull anyway).
Debug: vios_levels step="After vios update" vios="p7jufv2"
Debug: target=p7jufv2, remote command=/usr/bin/oslevel -s
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh p7jufv2 "/usr/bin/oslevel -s; echo C_
RSH_CMD_RC=\$?"
Debug: target=p7jufv2, remote command=/usr/ios/cli/ioscli ioslevel
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh p7jufv2 "/usr/ios/cli/ioscli ioslevel
; echo C_RSH_CMD_RC=\$?"
Info: After vios update NIM updateios operation of p7jufv2 oslevel=6100-05-09-1228 ioslevel=2.2.
6.31
Error: vios update of "p7jufv2" vios returns 1
Debug: Performing "start" SSP action on "p7jufv2" vios
```

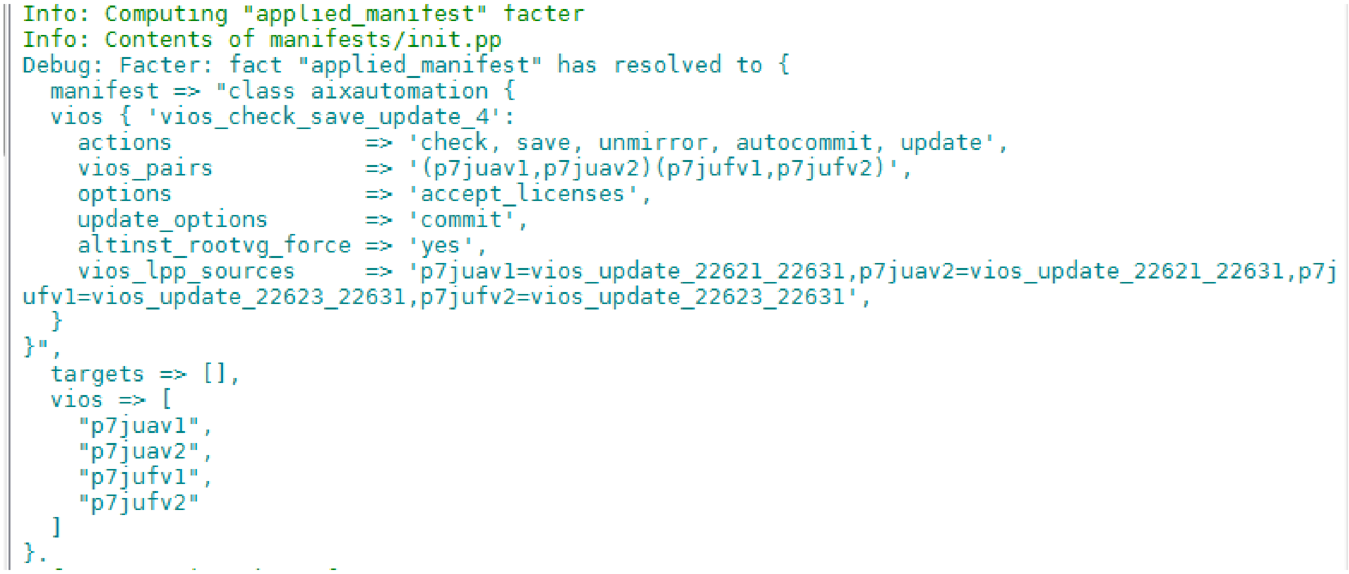
Fourth example of manifest/init.pp to perform 'update' on two VIOS pairs:

Class automation to encapsulate the declaration

```
class aixautomation {
 vios { 'vios_update_4':
actions    => 'check,gc,save,unmirror,autocommit,update',
   vios_pairs => '(p7juav1, p7juav2) ,(p7jufv1, p7jufv2)',
   altinst_rootvg_force => 'yes',
  vios_lpp_sources => 'p7juav1=vios_update_22621_22631,
           p7juav2=vios_update_22621_22631,
           p7jufv1=vios_update_22623_22631,
           p7jufv2=vios_update_22623_22631',
 }
}
```

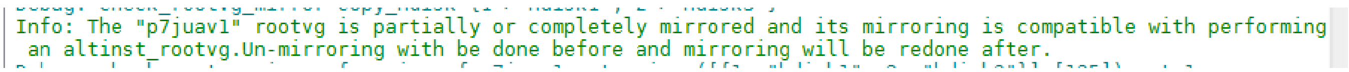
Attributes of custom type 'vios': this declaration called 'vios_update_4' applies on two VIOS pairs: p7juav1 and p7juav2 on one side, and p7jufv1 and p7jufv2 on another side where it launches the 'update' VIOS pair. 'check', 'gc', 'save', 'unmirror', 'autocommit' are necessary. 'unmirror' is necessary if VIOS rootvg is mirrored.

The ‘applied_manifest’ facter displays the contents of manifest/init.pp file which is being run:



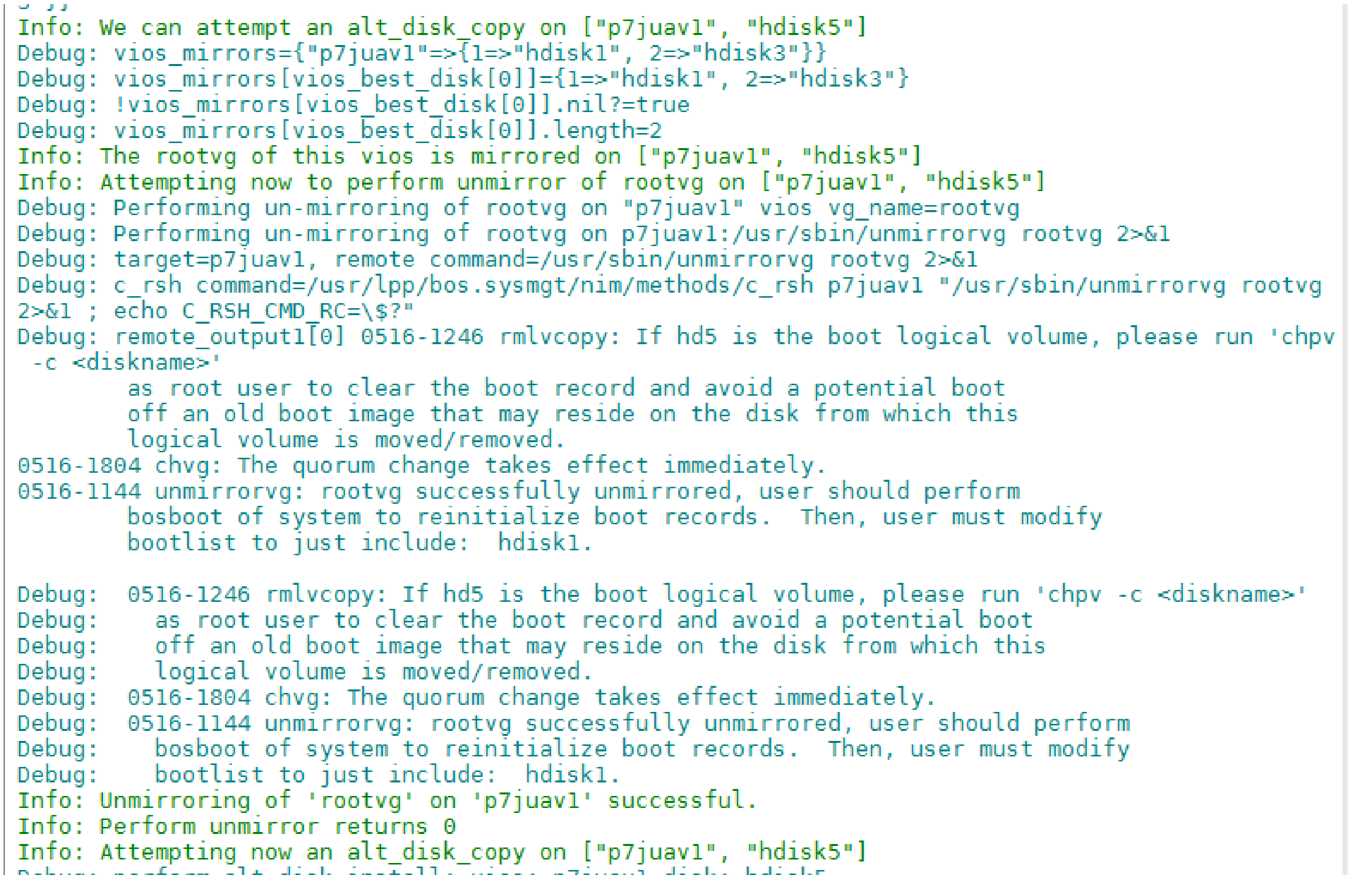
```
Info: Computing "applied_manifest" facter
Info: Contents of manifests/init.pp
Debug: Facter: fact "applied_manifest" has resolved to {
  manifest => "class aixautomation {
  vios { 'vios_check_save_update_4':
    actions               => 'check, save, unmirror, autocommit, update',
    vios_pairs            => '(p7juav1,p7juav2)(p7jufv1,p7jufv2)',
    options               => 'accept_licenses',
    update_options        => 'commit',
    altinst_rootvg_force => 'yes',
    vios_lpp_sources      => 'p7juav1=vios_update_22621_22631,p7juav2=vios_update_22621_22631,p7j
ufv1=vios_update_22623_22631,p7jufv2=vios_update_22623_22631',
  }
}",
  targets => [],
  vios => [
    "p7juav1",
    "p7juav2",
    "p7jufv1",
    "p7jufv2"
  ]
}.
```

The rootvg of both p7juav1 and p7juav2 are mirrored, there it is necessary to perform un-mirror of this ‘rootvg’ while performing ‘alt_disk_copy’.



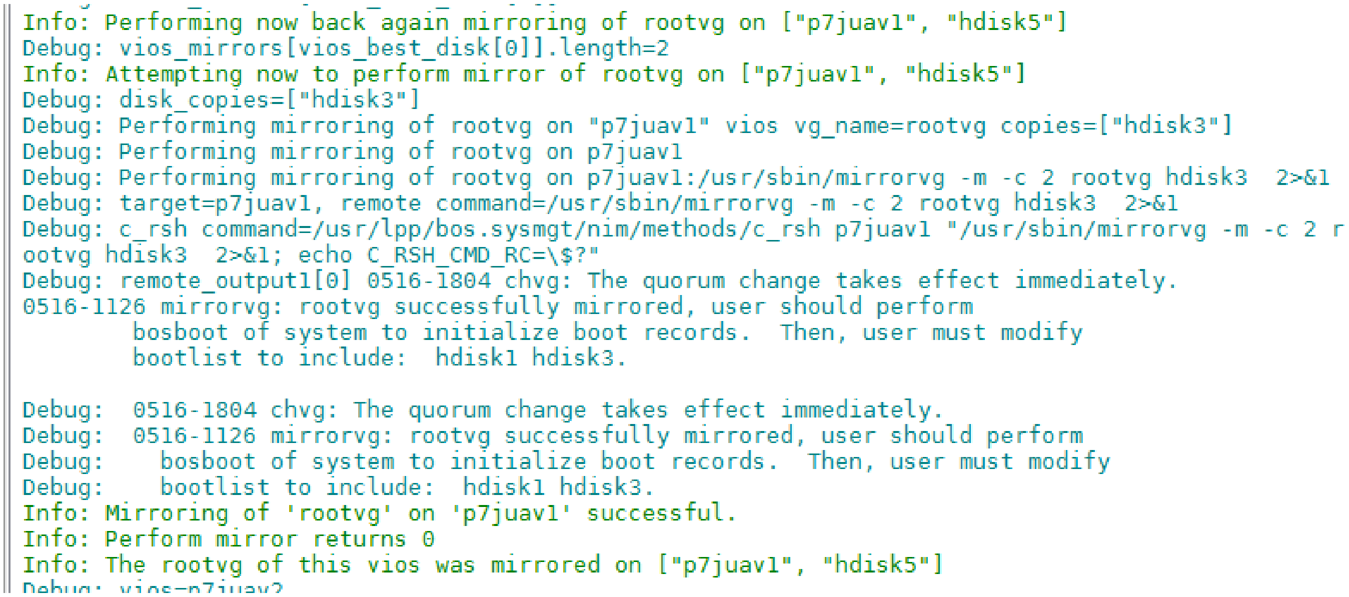
```
Info: The "p7juav1" rootvg is partially or completely mirrored and its mirroring is compatible with performing
 an altinst_rootvg.Un-mirroring with be done before and mirroring will be redone after.
```

This unmirroring has been agreed upon by the user who has set the ‘unmirror’ option into mainfests/init.pp. The following sequence of log items if generated whiel performing unmirroring of rootvg :



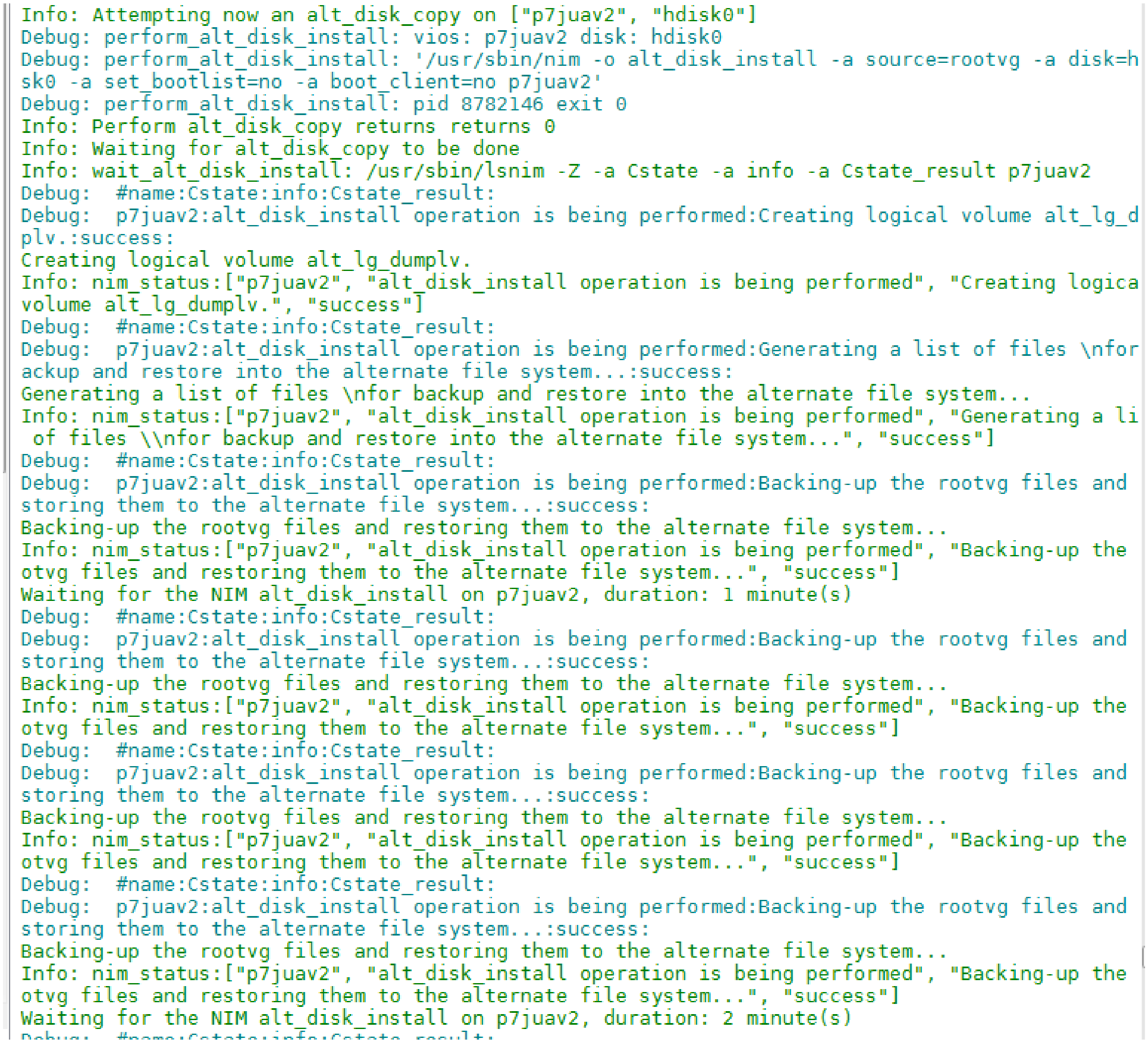
```
Info: We can attempt an alt_disk_copy on ["p7juav1", "hdisk5"]
Debug: vios_mirrors={"p7juav1"=>{1=>"hdisk1", 2=>"hdisk3"}}
Debug: vios_mirrors[vios_best_disk[0]]={1=>"hdisk1", 2=>"hdisk3"}
Debug: !vios_mirrors[vios_best_disk[0]].nil?=true
Debug: vios_mirrors[vios_best_disk[0]].length=2
Info: The rootvg of this vios is mirrored on ["p7juav1", "hdisk5"]
Info: Attempting now to perform unmirror of rootvg on ["p7juav1", "hdisk5"]
Debug: Performing un-mirroring of rootvg on "p7juav1" vios vg_name=rootvg
Debug: Performing un-mirroring of rootvg on p7juav1:/usr/sbin/unmirrorvg rootvg 2>&1
Debug: target=p7juav1, remote command=/usr/sbin/unmirrorvg rootvg 2>&1
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh p7juav1 "/usr/sbin/unmirrorvg rootvg
2>&1 ; echo C_RSH_CMD_RC=\$?"
Debug: remote_output1[0] 0516-1246 rmlvcopy: If hd5 is the boot logical volume, please run 'chpv
 -c <diskname>'
        as root user to clear the boot record and avoid a potential boot
        off an old boot image that may reside on the disk from which this
        logical volume is moved/removed.
0516-1804 chvg: The quorum change takes effect immediately.
0516-1144 unmirrorvg: rootvg successfully unmirrored, user should perform
        bosboot of system to reinitialize boot records.  Then, user must modify
        bootlist to just include:  hdisk1.

Debug:  0516-1246 rmlvcopy: If hd5 is the boot logical volume, please run 'chpv -c <diskname>'
Debug:    as root user to clear the boot record and avoid a potential boot
Debug:    off an old boot image that may reside on the disk from which this
Debug:    logical volume is moved/removed.
Debug:  0516-1804 chvg: The quorum change takes effect immediately.
Debug:  0516-1144 unmirrorvg: rootvg successfully unmirrored, user should perform
Debug:    bosboot of system to reinitialize boot records.  Then, user must modify
Debug:    bootlist to just include:  hdisk1.
Info: Unmirroring of 'rootvg' on 'p7juav1' successful.
Info: Perform unmirror returns 0
Info: Attempting now an alt_disk_copy on ["p7juav1", "hdisk5"]
```

After the alt_disk_copy of rootvg, the rootvg is mirrored back again.

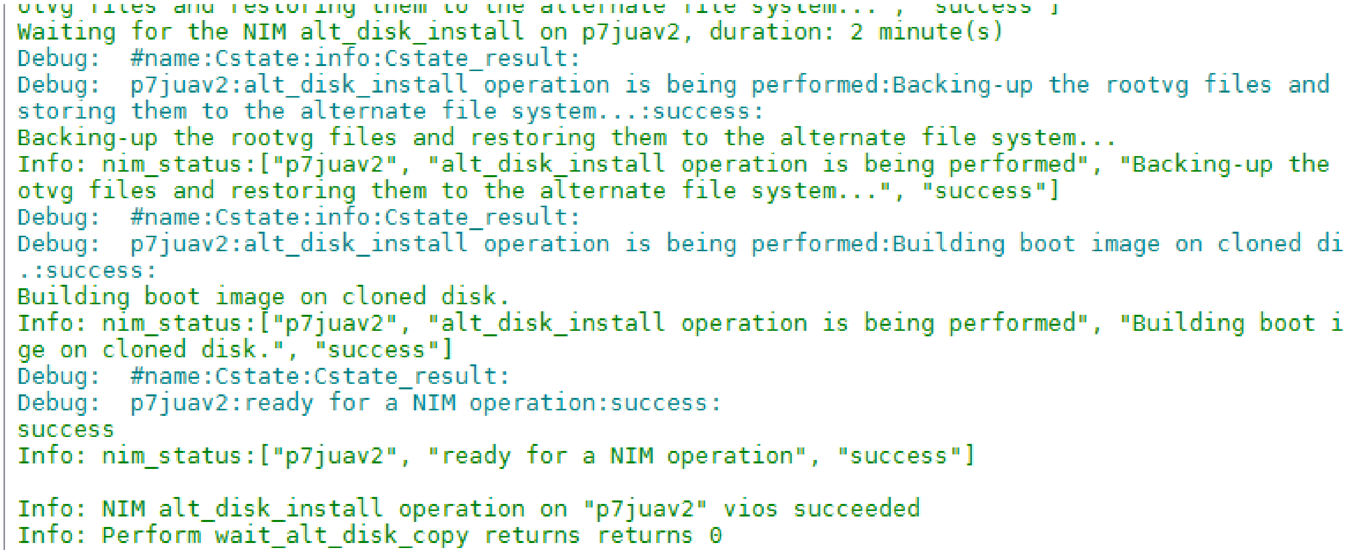


```
Info: Performing now back again mirroring of rootvg on ["p7juav1", "hdisk5"]
Debug: vios_mirrors[vios_best_disk[0]].length=2
Info: Attempting now to perform mirror of rootvg on ["p7juav1", "hdisk5"]
Debug: disk_copies=["hdisk3"]
Debug: Performing mirroring of rootvg on "p7juav1" vios vg_name=rootvg copies=["hdisk3"]
Debug: Performing mirroring of rootvg on p7juav1
Debug: Performing mirroring of rootvg on p7juav1:/usr/sbin/mirrorvg -m -c 2 rootvg hdisk3  2>&1
Debug: target=p7juav1, remote command=/usr/sbin/mirrorvg -m -c 2 rootvg hdisk3  2>&1
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh p7juav1 "/usr/sbin/mirrorvg -m -c 2 r
ootvg hdisk3  2>&1; echo C_RSH_CMD_RC=\$?"
Debug: remote_output1[0] 0516-1804 chvg: The quorum change takes effect immediately.
0516-1126 mirrorvg: rootvg successfully mirrored, user should perform
        bosboot of system to initialize boot records.  Then, user must modify
        bootlist to include:  hdisk1 hdisk3.

Debug:  0516-1804 chvg: The quorum change takes effect immediately.
Debug:  0516-1126 mirrorvg: rootvg successfully mirrored, user should perform
Debug:    bosboot of system to initialize boot records.  Then, user must modify
Debug:    bootlist to include:  hdisk1 hdisk3.
Info: Mirroring of 'rootvg' on 'p7juav1' successful.
Info: Perform mirror returns 0
Info: The rootvg of this vios was mirrored on ["p7juav1", "hdisk5"]
Debug: vios=p7juav2
```

Performing the alt_disk_copy generates following output:

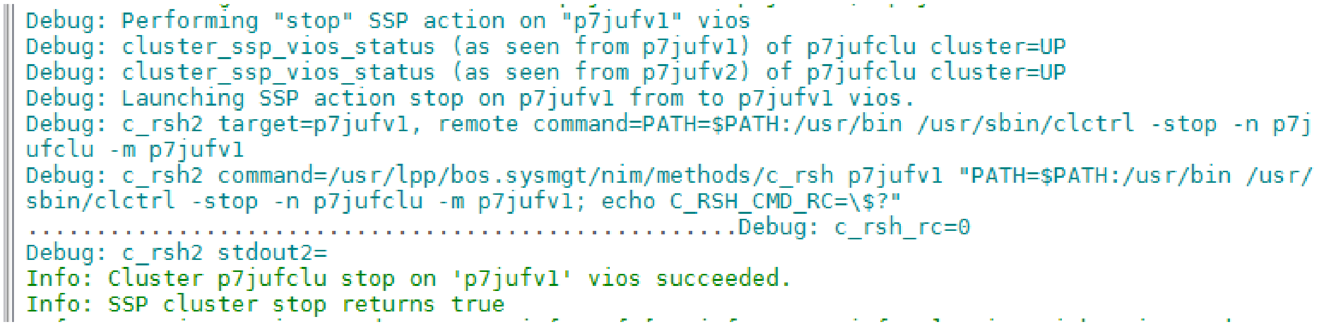


```
Info: Attempting now an alt_disk_copy on ["p7juav2", "hdisk0"]
Debug: perform_alt_disk_install: vios: p7juav2 disk: hdisk0
Debug: perform_alt_disk_install: '/usr/sbin/nim -o alt_disk_install -a source=rootvg -a disk=h
sk0 -a set_bootlist=no -a boot_client=no p7juav2'
Debug: perform_alt_disk_install: pid 8782146 exit 0
Info: Perform alt_disk_copy returns returns 0
Info: Waiting for alt_disk_copy to be done
Info: wait_alt_disk_install: /usr/sbin/lsnim -Z -a Cstate -a info -a Cstate_result p7juav2
Debug:  #name:Cstate:info:Cstate_result:
Debug:  p7juav2:alt_disk_install operation is being performed:Creating logical volume alt_lg_d
plv.:success:
Creating logical volume alt_lg_dumplv.
Info: nim_status:["p7juav2", "alt_disk_install operation is being performed", "Creating logica
volume alt_lg_dumplv.", "success"]
Debug:  #name:Cstate:info:Cstate_result:
Debug:  p7juav2:alt_disk_install operation is being performed:Generating a list of files \nfor
ackup and restore into the alternate file system...:success:
Generating a list of files \nfor backup and restore into the alternate file system...
Info: nim_status:["p7juav2", "alt_disk_install operation is being performed", "Generating a li
 of files \\nfor backup and restore into the alternate file system...", "success"]
Debug:  #name:Cstate:info:Cstate_result:
Debug:  p7juav2:alt_disk_install operation is being performed:Backing-up the rootvg files and
storing them to the alternate file system...:success:
Backing-up the rootvg files and restoring them to the alternate file system...
Info: nim_status:["p7juav2", "alt_disk_install operation is being performed", "Backing-up the
otvg files and restoring them to the alternate file system...", "success"]
Waiting for the NIM alt_disk_install on p7juav2, duration: 1 minute(s)
Debug:  #name:Cstate:info:Cstate_result:
Debug:  p7juav2:alt_disk_install operation is being performed:Backing-up the rootvg files and
storing them to the alternate file system...:success:
Backing-up the rootvg files and restoring them to the alternate file system...
Info: nim_status:["p7juav2", "alt_disk_install operation is being performed", "Backing-up the
otvg files and restoring them to the alternate file system...", "success"]
Debug:  #name:Cstate:info:Cstate_result:
Debug:  p7juav2:alt_disk_install operation is being performed:Backing-up the rootvg files and
storing them to the alternate file system...:success:
Backing-up the rootvg files and restoring them to the alternate file system...
Info: nim_status:["p7juav2", "alt_disk_install operation is being performed", "Backing-up the
otvg files and restoring them to the alternate file system...", "success"]
Debug:  #name:Cstate:info:Cstate_result:
Debug:  p7juav2:alt_disk_install operation is being performed:Backing-up the rootvg files and
storing them to the alternate file system...:success:
Backing-up the rootvg files and restoring them to the alternate file system...
Info: nim_status:["p7juav2", "alt_disk_install operation is being performed", "Backing-up the
otvg files and restoring them to the alternate file system...", "success"]
Waiting for the NIM alt_disk_install on p7juav2, duration: 2 minute(s)
Debug:  #name:Cstate:info:Cstate_result:
```



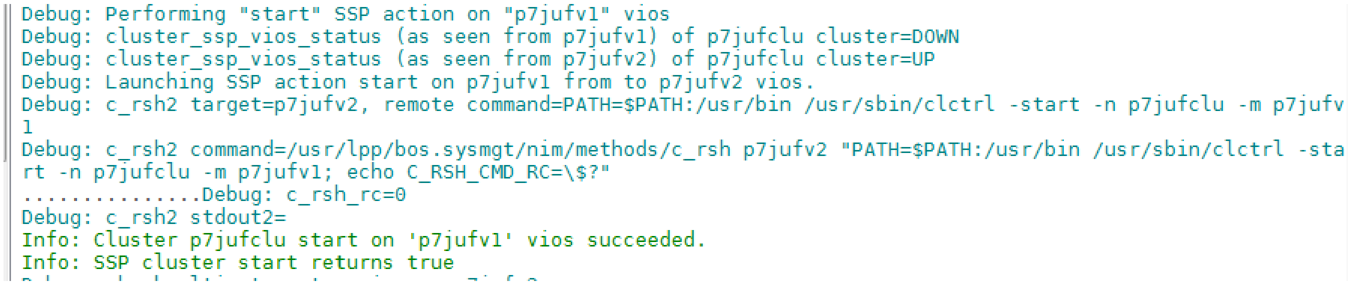
```
otvg files and restoring them to the alternate file system...", "success"]
Waiting for the NIM alt_disk_install on p7juav2, duration: 2 minute(s)
Debug:  #name:Cstate:info:Cstate_result:
Debug:  p7juav2:alt_disk_install operation is being performed:Backing-up the rootvg files and
storing them to the alternate file system...:success:
Backing-up the rootvg files and restoring them to the alternate file system...
Info: nim_status:["p7juav2", "alt_disk_install operation is being performed", "Backing-up the
otvg files and restoring them to the alternate file system...", "success"]
Debug:  #name:Cstate:info:Cstate_result:
Debug:  p7juav2:alt_disk_install operation is being performed:Building boot image on cloned di
.:success:
Building boot image on cloned disk.
Info: nim_status:["p7juav2", "alt_disk_install operation is being performed", "Building boot i
ge on cloned disk.", "success"]
Debug:  #name:Cstate:Cstate_result:
Debug:  p7juav2:ready for a NIM operation:success:
success
Info: nim_status:["p7juav2", "ready for a NIM operation", "success"]

Info: NIM alt_disk_install operation on "p7juav2" vios succeeded
Info: Perform wait_alt_disk_copy returns returns 0
```

Stopping the node of SSP cluster before doing update generates following output:



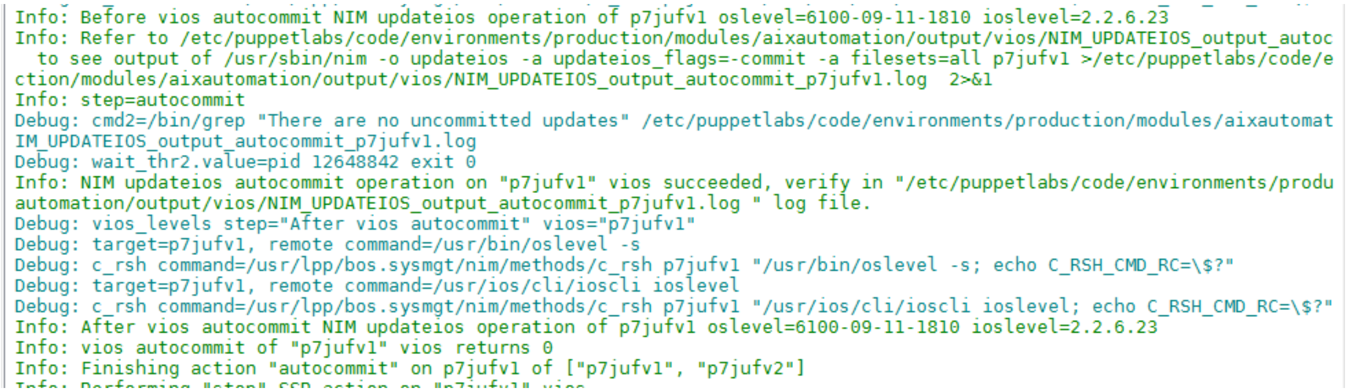
```
Debug: Performing "stop" SSP action on "p7jufv1" vios
Debug: cluster_ssp_vios_status (as seen from p7jufv1) of p7jufclu cluster=UP
Debug: cluster_ssp_vios_status (as seen from p7jufv2) of p7jufclu cluster=UP
Debug: Launching SSP action stop on p7jufv1 from to p7jufv1 vios.
Debug: c_rsh2 target=p7jufv1, remote command=PATH=$PATH:/usr/bin /usr/sbin/clctrl -stop -n p7j
ufclu -m p7jufv1
Debug: c_rsh2 command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh p7jufv1 "PATH=$PATH:/usr/bin /usr/
sbin/clctrl -stop -n p7jufclu -m p7jufv1; echo C_RSH_CMD_RC=\$?"
.....................................................Debug: c_rsh_rc=0
Debug: c_rsh2 stdout2=
Info: Cluster p7jufclu stop on 'p7jufv1' vios succeeded.
Info: SSP cluster stop returns true
```

Restarting the node of SSP cluster after doing update generates following output:



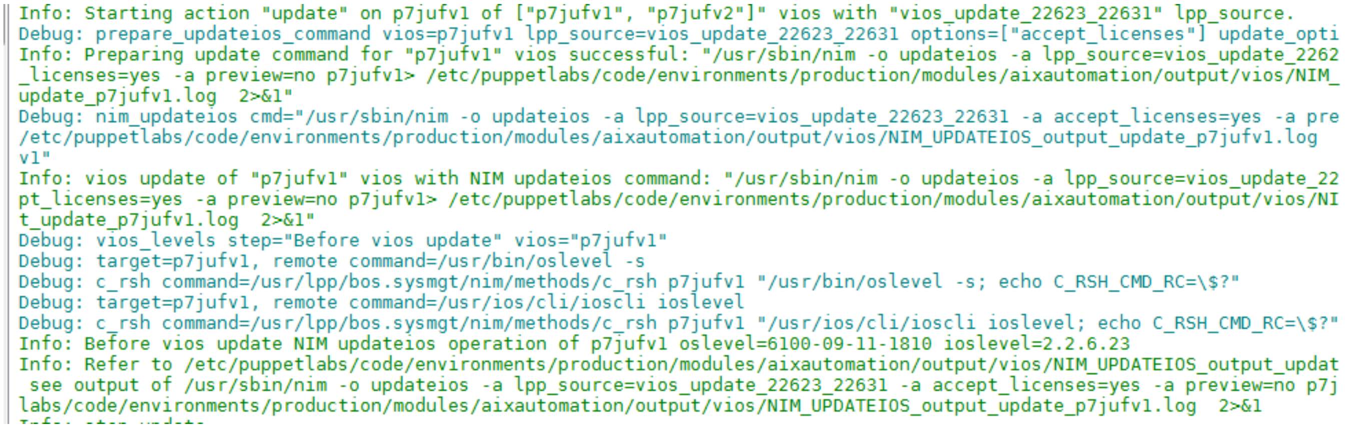
```
Debug: Performing "start" SSP action on "p7jufv1" vios
Debug: cluster_ssp_vios_status (as seen from p7jufv1) of p7jufclu cluster=DOWN
Debug: cluster_ssp_vios_status (as seen from p7jufv2) of p7jufclu cluster=UP
Debug: Launching SSP action start on p7jufv1 from to p7jufv2 vios.
Debug: c_rsh2 target=p7jufv2, remote command=PATH=$PATH:/usr/bin /usr/sbin/clctrl -start -n p7jufclu -m p7jufv
1
Debug: c_rsh2 command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh p7jufv2 "PATH=$PATH:/usr/bin /usr/sbin/clctrl -sta
rt -n p7jufclu -m p7jufv1; echo C_RSH_CMD_RC=\$?"
..............Debug: c_rsh_rc=0
Debug: c_rsh2 stdout2=
Info: Cluster p7jufclu start on 'p7jufv1' vios succeeded.
Info: SSP cluster start returns true
```

Updating VIOS themselves produces following output, first for the autocommit phase:

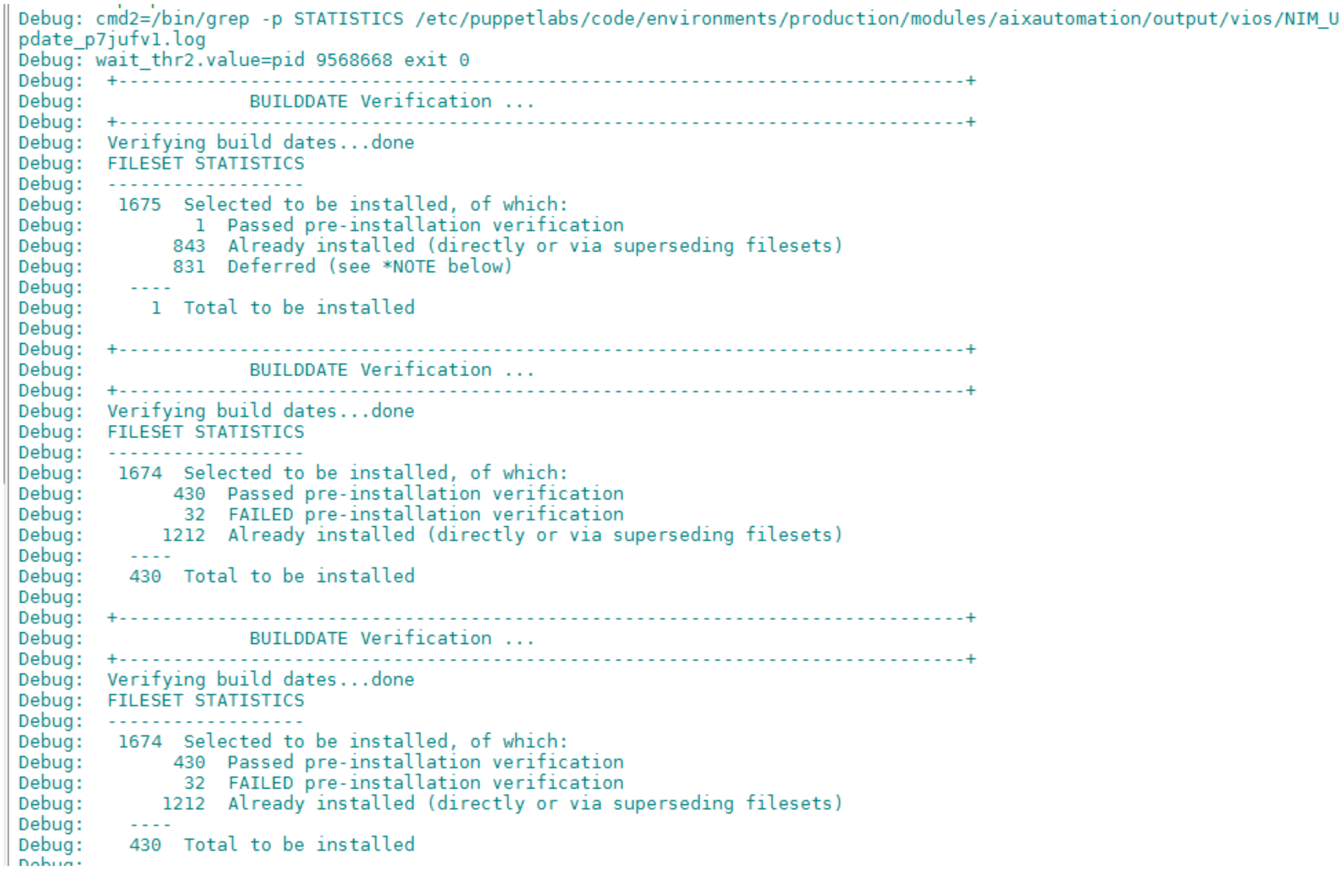


```
Info: Before vios autocommit NIM updateios operation of p7jufv1 oslevel=6100-09-11-1810 ioslevel=2.2.6.23
Info: Refer to /etc/puppetlabs/code/environments/production/modules/aixautomation/output/vios/NIM_UPDATEIOS_output_autoc
  to see output of /usr/sbin/nim -o updateios -a updateios_flags=-commit -a filesets=all p7jufv1 >/etc/puppetlabs/code/e
ction/modules/aixautomation/output/vios/NIM_UPDATEIOS_output_autocommit_p7jufv1.log  2>&1
Info: step=autocommit
Debug: cmd2=/bin/grep "There are no uncommitted updates" /etc/puppetlabs/code/environments/production/modules/aixautomat
IM_UPDATEIOS_output_autocommit_p7jufv1.log
Debug: wait_thr2.value=pid 12648842 exit 0
Info: NIM updateios autocommit operation on "p7jufv1" vios succeeded, verify in "/etc/puppetlabs/code/environments/produ
automation/output/vios/NIM_UPDATEIOS_output_autocommit_p7jufv1.log " log file.
Debug: vios_levels step="After vios autocommit" vios="p7jufv1"
Debug: target=p7jufv1, remote command=/usr/bin/oslevel -s
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh p7jufv1 "/usr/bin/oslevel -s; echo C_RSH_CMD_RC=\$?"
Debug: target=p7jufv1, remote command=/usr/ios/cli/ioscli ioslevel
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh p7jufv1 "/usr/ios/cli/ioscli ioslevel; echo C_RSH_CMD_RC=\$?"
Info: After vios autocommit NIM updateios operation of p7jufv1 oslevel=6100-09-11-1810 ioslevel=2.2.6.23
Info: vios autocommit of "p7jufv1" vios returns 0
Info: Finishing action "autocommit" on p7jufv1 of ["p7jufv1", "p7jufv2"]
Info: Performing "stop" SSP action on "p7jufv1" vios
```

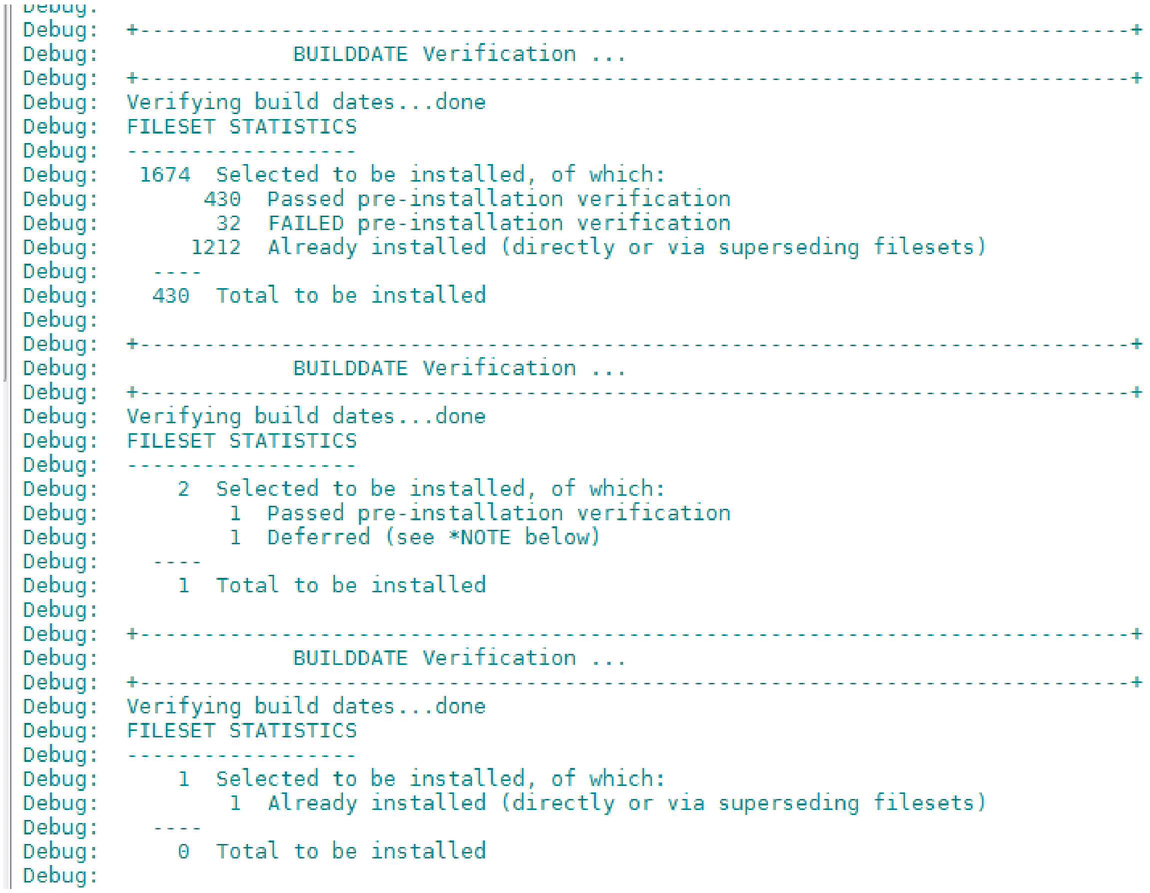
Updating VIOS themselves produces following output, then when launching the update phase, the ioslevel at the beginning is displayed, it is set in this case to 2.2.6.23:



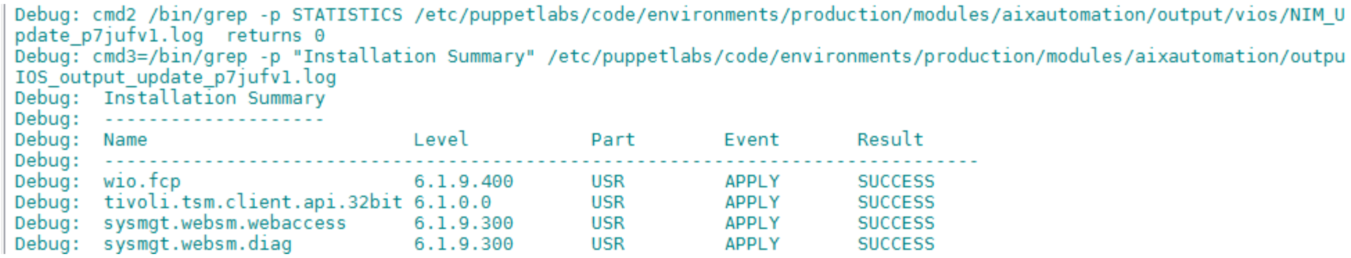
```
Info: Starting action "update" on p7jufv1 of ["p7jufv1", "p7jufv2"]" vios with "vios_update_22623_22631" lpp_source.
Debug: prepare_updateios_command vios=p7jufv1 lpp_source=vios_update_22623_22631 options=["accept_licenses"] update_opti
Info: Preparing update command for "p7jufv1" vios successful: "/usr/sbin/nim -o updateios -a lpp_source=vios_update_2262
_licenses=yes -a preview=no p7jufv1> /etc/puppetlabs/code/environments/production/modules/aixautomation/output/vios/NIM_
update_p7jufv1.log  2>&1"
Debug: nim_updateios cmd="/usr/sbin/nim -o updateios -a lpp_source=vios_update_22623_22631 -a accept_licenses=yes -a pre
/etc/puppetlabs/code/environments/production/modules/aixautomation/output/vios/NIM_UPDATEIOS_output_update_p7jufv1.log
v1"
Info: vios update of "p7jufv1" vios with NIM updateios command: "/usr/sbin/nim -o updateios -a lpp_source=vios_update_22
pt_licenses=yes -a preview=no p7jufv1> /etc/puppetlabs/code/environments/production/modules/aixautomation/output/vios/NI
t_update_p7jufv1.log  2>&1"
Debug: vios_levels step="Before vios update" vios="p7jufv1"
Debug: target=p7jufv1, remote command=/usr/bin/oslevel -s
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh p7jufv1 "/usr/bin/oslevel -s; echo C_RSH_CMD_RC=\$?"
Debug: target=p7jufv1, remote command=/usr/ios/cli/ioscli ioslevel
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh p7jufv1 "/usr/ios/cli/ioscli ioslevel; echo C_RSH_CMD_RC=\$?"
Info: Before vios update NIM updateios operation of p7jufv1 oslevel=6100-09-11-1810 ioslevel=2.2.6.23
Info: Refer to /etc/puppetlabs/code/environments/production/modules/aixautomation/output/vios/NIM_UPDATEIOS_output_updat
 see output of /usr/sbin/nim -o updateios -a lpp_source=vios_update_22623_22631 -a accept_licenses=yes -a preview=no p7j
labs/code/environments/production/modules/aixautomation/output/vios/NIM_UPDATEIOS_output_update_p7jufv1.log  2>&1
Info: step=update
```



```
Debug: cmd2=/bin/grep -p STATISTICS /etc/puppetlabs/code/environments/production/modules/aixautomation/output/vios/NIM_U
pdate_p7jufv1.log
Debug: wait_thr2.value=pid 9568668 exit 0
Debug:  +-----------------------------------------------------------------------------+
Debug:                BUILDDATE Verification ...
Debug:  +-----------------------------------------------------------------------------+
Debug:  Verifying build dates...done
Debug:  FILESET STATISTICS
Debug:  ------------------
Debug:   1675  Selected to be installed, of which:
Debug:          1  Passed pre-installation verification
Debug:        843  Already installed (directly or via superseding filesets)
Debug:        831  Deferred (see *NOTE below)
Debug:    ----
Debug:      1  Total to be installed
Debug:
Debug:  +-----------------------------------------------------------------------------+
Debug:                BUILDDATE Verification ...
Debug:  +-----------------------------------------------------------------------------+
Debug:  Verifying build dates...done
Debug:  FILESET STATISTICS
Debug:  ------------------
Debug:   1674  Selected to be installed, of which:
Debug:        430  Passed pre-installation verification
Debug:         32  FAILED pre-installation verification
Debug:       1212  Already installed (directly or via superseding filesets)
Debug:    ----
Debug:    430  Total to be installed
Debug:
Debug:  +-----------------------------------------------------------------------------+
Debug:                BUILDDATE Verification ...
Debug:  +-----------------------------------------------------------------------------+
Debug:  Verifying build dates...done
Debug:  FILESET STATISTICS
Debug:  ------------------
Debug:   1674  Selected to be installed, of which:
Debug:        430  Passed pre-installation verification
Debug:         32  FAILED pre-installation verification
Debug:       1212  Already installed (directly or via superseding filesets)
Debug:    ----
Debug:    430  Total to be installed
Debug:
```

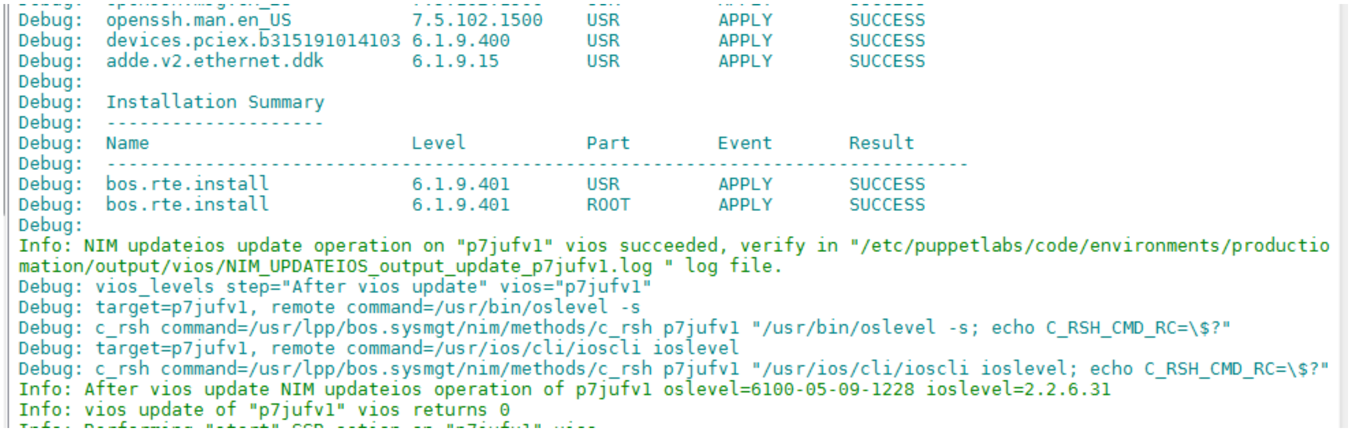


```
Debug:
Debug:  +-----------------------------------------------------------------------------+
Debug:               BUILDDATE Verification ...
Debug:  +-----------------------------------------------------------------------------+
Debug:  Verifying build dates...done
Debug:  FILESET STATISTICS
Debug:  ------------------
Debug:   1674  Selected to be installed, of which:
Debug:        430  Passed pre-installation verification
Debug:         32  FAILED pre-installation verification
Debug:       1212  Already installed (directly or via superseding filesets)
Debug:    ----
Debug:    430  Total to be installed
Debug:
Debug:  +-----------------------------------------------------------------------------+
Debug:               BUILDDATE Verification ...
Debug:  +-----------------------------------------------------------------------------+
Debug:  Verifying build dates...done
Debug:  FILESET STATISTICS
Debug:  ------------------
Debug:      2  Selected to be installed, of which:
Debug:          1  Passed pre-installation verification
Debug:          1  Deferred (see *NOTE below)
Debug:    ----
Debug:      1  Total to be installed
Debug:
Debug:  +-----------------------------------------------------------------------------+
Debug:               BUILDDATE Verification ...
Debug:  +-----------------------------------------------------------------------------+
Debug:  Verifying build dates...done
Debug:  FILESET STATISTICS
Debug:  ------------------
Debug:      1  Selected to be installed, of which:
Debug:          1  Already installed (directly or via superseding filesets)
Debug:    ----
Debug:      0  Total to be installed
Debug:
```



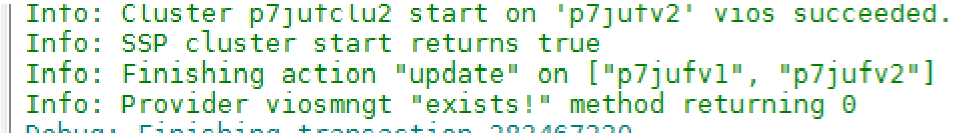
```
Debug: cmd2 /bin/grep -p STATISTICS /etc/puppetlabs/code/environments/production/modules/aixautomation/output/vios/NIM_U
pdate_p7jufv1.log  returns 0
Debug: cmd3=/bin/grep -p "Installation Summary" /etc/puppetlabs/code/environments/production/modules/aixautomation/outpu
IOS_output_update_p7jufv1.log
Debug:  Installation Summary
Debug:  --------------------
Debug:  Name                        Level           Part        Event       Result
Debug:  -------------------------------------------------------------------------------
Debug:  wio.fcp                     6.1.9.400       USR         APPLY       SUCCESS
Debug:  tivoli.tsm.client.api.32bit 6.1.0.0         USR         APPLY       SUCCESS
Debug:  sysmgt.websm.webaccess      6.1.9.300       USR         APPLY       SUCCESS
Debug:  sysmgt.websm.diag           6.1.9.300       USR         APPLY       SUCCESS
```

Output is truncated until the bottom, where you can see that update succeeded, and that new ioslevel is 2.2.6.31



```
Debug:  openssh.man.en_US           7.5.102.1500    USR         APPLY       SUCCESS
Debug:  devices.pciex.b315191014103 6.1.9.400       USR         APPLY       SUCCESS
Debug:  adde.v2.ethernet.ddk        6.1.9.15        USR         APPLY       SUCCESS
Debug:
Debug:  Installation Summary
Debug:  --------------------
Debug:  Name                        Level           Part        Event       Result
Debug:  -------------------------------------------------------------------------------
Debug:  bos.rte.install             6.1.9.401       USR         APPLY       SUCCESS
Debug:  bos.rte.install             6.1.9.401       ROOT        APPLY       SUCCESS
Debug:
Info: NIM updateios update operation on "p7jufv1" vios succeeded, verify in "/etc/puppetlabs/code/environments/productio
mation/output/vios/NIM_UPDATEIOS_output_update_p7jufv1.log " log file.
Debug: vios_levels step="After vios update" vios="p7jufv1"
Debug: target=p7jufv1, remote command=/usr/bin/oslevel -s
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh p7jufv1 "/usr/bin/oslevel -s; echo C_RSH_CMD_RC=\$?"
Debug: target=p7jufv1, remote command=/usr/ios/cli/ioscli ioslevel
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh p7jufv1 "/usr/ios/cli/ioscli ioslevel; echo C_RSH_CMD_RC=\$?"
Info: After vios update NIM updateios operation of p7jufv1 oslevel=6100-05-09-1228 ioslevel=2.2.6.31
Info: vios update of "p7jufv1" vios returns 0
```

At the end, you have a message indicated that update is finished on the VIOS pair.



```
Info: Cluster p7jufclu2 start on 'p7jufv2' vios succeeded.
Info: SSP cluster start returns true
Info: Finishing action "update" on ["p7jufv1", "p7jufv2"]
Info: Provider viosmngt "exists!" method returning 0
```

For more information…

For more information about AIX Automation with Puppet and to know all the attributes available for each custom type, refer to README.md file from Github project repository:

https://github.com/aixoss/aix-puppet/aixautomation/blob/master/README.md

Table of contents