Description:

Puppet is an open source product for configuration management and automation. The NetApp Service Level Manager (NSLM) device module is designed to add support for your storage management using Puppet and its Network Device functionality. The NSLM device module has been written and tested against NSLM 1.0.

This Puppet module will add one provider for every NSLM object that supports PUT, POST and DELETE request through REST APIs. This module enables Puppet users to create new Manifest files for their storage management or change their existing Manifest files to automate storage management operations along with other IT management operations.

Note: This readme does not provide any information or tutorial about how to use Puppet. It gives information about how users can use it to manage their data using NSLM. It is just an integration of Puppet with NSLM. For any further tutorials or information on Puppet, please refer to the https://puppet.com/ website.

Prerequisites:

- Puppet Master Version: Puppet-enterprise-2017…3-el-7
- Puppet Agent Version: Puppet-agent-1.10.5-1.el7
- NetApp Service Level Manager version: NSLM 1.0 RC3

NSLM Puppet Module Configuration in existing Puppet setup:

1. Download the pack “NSLM Puppet Integration Solution” from automationstore.netapp.com -> Integrations.
2. After installation of Puppet master, and Puppet agent, put the unzip the file at /etc/puppetlabs/code/environment/production/modules/
3. Add cluster details to /etc/hosts:

   <cluster-ip> <cluster-name>

e.g.

10.226.207.25 fas8040-206-21-23

Configuration of NSLM server as a Puppet proxy device:

1. Go to Puppet Agent at the following location:

   /etc/puppetlabs/puppet/

2. Create a <file-name>.conf file to add the NSLM-Server as a managed node in the Puppet device.conf file template:

   [<node name>]

   [type netapp]

   [url <nslm-server uri along with user name and password>]

   Example of device.conf file:

   [nslm-server]

   [type netapp]

   [url https://admin:Netapp1!@scspr0303277001.gdl.englab.netapp.com:8443/]

3. List all the NSLM puppet modules and attribute the details of each module:

   a) slo_cifsshare : Module to manage cifsshare

Post:

Mandatory:

name,	or	name,
file_share_key,		file_share_name,
storage_vm_key		file_share_storage_vm_name,
		file_share_storage_vm_storage_system_name,
		storage_vm_name,
		storage_vm_storage_system_name

Optional: directory

Del:

Mandatory:

key	or	file_share_storage_vm_name,
		file_share_storage_vm_storage_system_name,
		file_share_name
		name

b) slo_cifsshareacl : Module to manage cifsshare acls

Post:

Mandatory:

storage_vm_key,	or	storage_vm_name,
cifs_share_key		storage_vm_storage_system_name,
		cifs_share_name,
		cifs_share_file_share_storage_vm_name,
		cifs_share_file_share_storage_vm_ storage_system_name,
		cifs_share_file_share_name

Optional: permission,

user_or_group

Put:

Mandatory: key,	or	storage_platform_type,
		storage_vm_name,
		cifs_share_file_share_storage_vm_name,
		storage_vm_storage_system_name,
		cifs_share_name,
		cifs_share_file_share_storage_vm_ storage_system_name,
		user_or_group,
		cifs_share_file_share_name

Optional: permission

Del:

Mandatory: key	or	storage_platform_type,
		storage_vm_name,
		cifs_share_file_share_storage_vm_name,
		storage_vm_storage_system_name,
		cifs_share_name,
		cifs_share_file_share_storage_vm_ storage_system_name,
		user_or_group,
		cifs_share_file_share_name

c) slo_exportpolicy : Module to manage export polices

Post

Mandatory:

storage_vm_key, name	or	storage_vm_storage_system_name, storage_vm_name, name

Optional:

root ,
rw_hosts,
security_type,
ro_hosts

Put

Mandatory:

key	or	storage_vm_name, storage_vm_storage_system_name, name,

Optional:

root ,
rw_hosts,
security_type,
ro_hosts

Del

Mandatory:

key	or	storage_vm_name, storage_vm_storage_system_name, name

d) slo_fileshare : Module to manage File Shares

Post:

Mandatory:

storage_service_level_key,	or	storage_vm_name,
storage_vm_key,		storage_vm_storage_system_name,
size,		storage_service_level_name,
name		name,
		size

Optional:

storage_pool_key	or	storage_pool_name,
		storage_pool_storage_system_name,

Put:

Mandatory:

key	or	storage_vm_name,
		storage_vm_storage_system_name
		name

Optional:

Size,

operational_state

Del

Mandatory:

key	or	storage_vm_name,
		storage_vm_storage_system_name
		name

e) slo_initiatorgroup : Module to manage initiator groups

Post

Mandatory:

initiator_os_type,	or	initiator_os_type
storage_vm_key,		storage_vm_name,
name		storage_vm_storage_system_name,
		name

Optional: initiators

Put

Mandatory:

key	or	storage_vm_name,
		name,
		storage_vm_storage_system_name

Optional:

initiator_os_type,

initiators

Del

Mandatory:

key	or	storage_vm_name,
		name,
		storage_vm_storage_system_name

f) slo_lun: Module to manage LUNs

Post

Mandatory:

storage_service_level_key,	or	storage_service_level_name,
storage_vm_key,		storage_vm_name,
size,		storage_vm_storage_system_name,
name		size,
		name

Optional:

storage_pool_key,	or	storage_pool_name,
host_usage		storage_pool_storage_system_name,
		host_usage

Put

Mandatory:

key	or	storage_vm_name,
		name,
		storage_vm_storage_system_name

Optional:

size,

operational_state

Del

Mandatory:

key	or	storage_vm_name,
		name,
		storage_vm_storage_system_name

g) slo_lunmap: Module to manage LUN mapping

Post

Mandatory:

initiator_group_key,	or	initiator_group_storage_vm_name,
lun_key,		initiator_group_storage_vm_ storage_system_name,
		initiator_group_name,

lun_storage_vm_
storage_system_name,

lun_storage_vm_name,

lun_name

Optional: lun_id

Del

Mandatory:

key or initiator_group_storage_vm_name,

initiator_group_storage_vm_
storage_system_name,

lun_storage_vm_
storage_system_name,

lun_storage_vm_name,

lun_name,

initiator_group_name

h) slo_nfsshare: Module to manage NFS shares

Post

Mandatory:

export_policy_key or storage_vm_name,

file_share_key storage_vm_storage_system_name

storage_vm_key export_policy_storage_vm_
storage_system_name,

export_policy_storage_vm_name,

export_policy_name,

file_share_storage_vm_name,

file_share_storage_vm_
storage_system_name,

file_share_name

Optional: directory

Put

Mandatory:

key	or	file_share_storage_vm_name, file_share_storage_vm_storage_system_name, file_share_name, directory

Optional:

export_policy_key	or	storage_vm_name, name, storage_vm_storage_system_name

Del

Mandatory:

key	or	file_share_storage_vm_name, file_share_storage_vm_ storage_system_name, file_share_name, directory

i) slo_snapshot: Module to manage snapshots

Post

Mandatory:

name, file_share_key	or	name, file_share_storage_vm_name, file_share_name file_share_storage_vm_storage_system_name

Optional:

retention_type ,

comment

Del

Mandatory:

key, or file_share_storage_vm_name,
file_share_storage_vm_storage_system_name,
file_share_name
name

j) slo_storageservicelevel: Module to manage NSLM service definitions

Post

Mandatory:

expected_iops_per_tb
peak_latency
name
peak_iops_per_tb

Optional: description

Put

Mandatory:

key or name

Optional:

description,
peak_latency,
peak_iops_per_tb,
expected_iops_per_tb

Del

Mandatory:

key or name

Sample manifest file

The following sample of a Manifest file will create a LUN and an igroup, and then map the LUN to the igroup:

```
node 'nslm-server'{

  slo_lun{'lun-management':
   ensure => present,
   name => "finance_lun",
   size => 10737418240,
   storage_service_level_name => "Value",
   storage_vm_name => "Audit_Server",
   storage_vm_storage_system_name => "Finance-Cluster"
   }

  slo_initiatorgroup{'igroup-management':
   ensure => present,
   name => "finance_igroup",
   storage_vm_name => "Audit_Server",
   storage_vm_storage_system_name => "Finance-Cluster",
   initiator_os_type => "default"
   }
 slo_lunmap{'lunmap-management':
   ensure => present,
   initiator_group_storage_vm_name => "Audit_Server",
   initiator_group_storage_vm_storage_system_name => "Finance-Cluster",
   initiator_group_name => "finance_igroup",
   lun_storage_vm_name => "Audit_Server",
   lun_storage_vm_storage_system_name => "Finance-Cluster",
   lun_name => "finance_lun"
```

```
    }
}
```