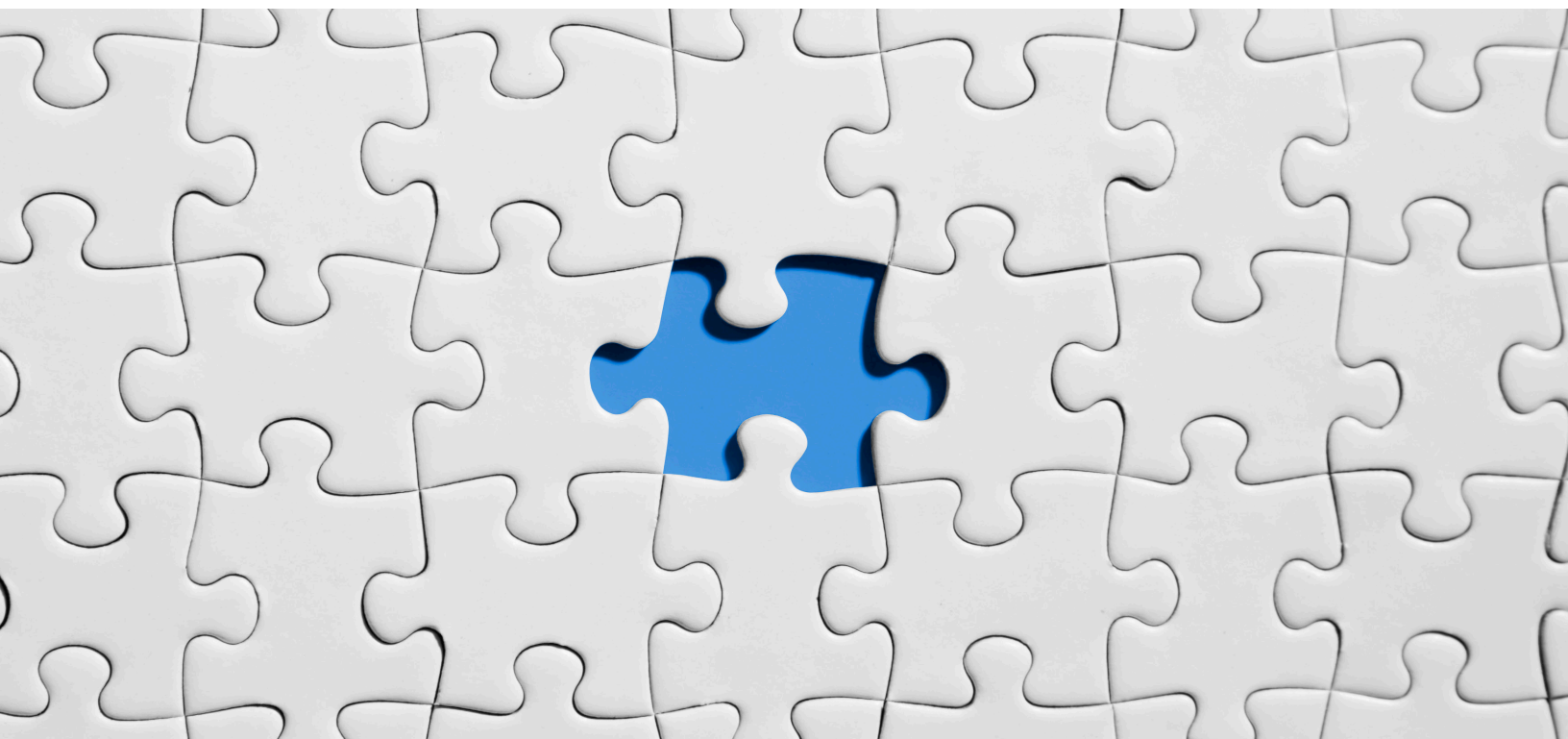


AUTOMATING INSTALL OF SOLUTIONS ENABLER USING ANSIBLE



Ramani Malathy

Consultant, IT Infrastructure

Dell Technologies

Malathybr@yahoo.com

Malathy.ramani@dell.com

Table of Contents

Preface	3
Solutions Enabler and VMAX	4
Ansible for automation	5
Ansible components.....	5
Install and upgrade of Solutions Enabler via Ansible	6
Setup for Solutions Enabler Install and upgrade.....	6
Ansible Requirements and What to do if Python version is behind on target hosts	10
Conclusion	11
Appendix.....	12

Disclaimer: The views, processes or methodologies published in this article are those of the author. They do not necessarily reflect Dell Technologies' views, processes or methodologies.

Preface

Dell EMC Solutions Enabler kit is the software comprising of Symmetrix Command line interface (SYMCLI) and SYMAPI libraries for managing Dell EMC enterprise storage environment supporting VMAX AFA, VMAX3, VMAX family and PowerMax arrays.

Solutions Enabler install or upgrade can be achieved fast and efficiently by leveraging a provisioning tool such as Ansible and automating it. Solutions Enabler is a host software for managing VMAX arrays and needs to be upgraded whenever a new release comes out. This is done by system or storage administrators who need to transfer the code to the servers using a tool such as WinSCP and then manually install or upgrade it on the servers.

Automating the install with Ansible not only saves time, but also eliminates human errors and repetitive manual tasks. The install can be done simultaneously on multiple servers by a single run of Ansible automation playbook.

Solutions Enabler and VMAX

VMAX is an enterprise-class storage array engineered to protect mission critical application environments in the data center with 99.9999% availability. VMAX is geared for high performance, scaling, high availability, and advanced data services. Solutions Enabler is the management software for VMAX storage arrays. The management server on which it is installed is called a control host.

A Solutions Enabler install provides the host with SYMAPI and STORAPI shared libraries as well as the Symmetrix command line interface or SYMCLI for use by storage administrators and systems engineers. SYMCLI is a specialized set of UNIX-formatted commands to perform control operations on devices and data objects toward management of the storage complex. Solutions Enabler software is also used in auditing and performance monitoring of the storage arrays.

Ansible for automation

Ansible is a simple IT automation solution that automates provisioning, configuration management, application deployment, and orchestration. Designed for multi-tier deployments, Ansible works by connecting to other nodes and pushing out modules via Secure Shell (SSH) protocol. These modules are executed on remote systems to take them to the desired state.

Ansible is an agentless tool and uses the “push” concept to manage remote systems over SSH if it is Linux or UNIX and WinRM (Windows Remote Management) for Windows. Ansible leverages the user credentials which are supplied at run time for pushing the modules to the remote servers and resources are consumed on the target systems only when Ansible is managing them. These attributes make it suitable for Ansible to function in high secure and high performing environments.

Ansible components

Inventory

The inventory is where the host information is specified. This is a text file which contains the details of the managed nodes. Host-level variables, roles, and groups are also defined here.

Playbooks

The files created to perform a task. Plays are defined within a playbook, and are executed in a particular order for automation and orchestration. They are written in a format called YAML where instructions are in a certain structure but basically, it is a text file and quite easy to understand.

Play

Plays are contained in a playbook. Each play targets a host or group of hosts and defines a set of tasks to be done.

Tasks

Tasks are actions done by the playbook.

Roles

Roles allow reusability. In the place of creating a long playbook, multiple roles can be created. Roles can be assigned to specific hosts to do a group of tasks.

Handlers

When a “notify” directive happens, handlers are activated. The names defined should be the same in notify and handlers in the playbook for handlers to work.

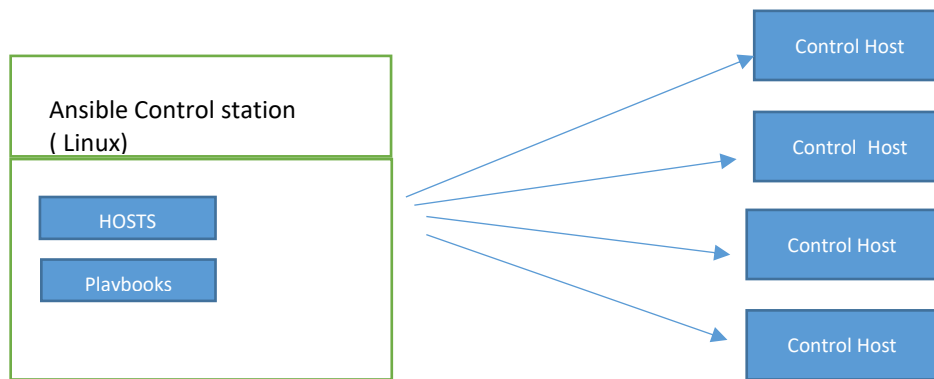
Templates

Templates are based on Python’s Jinja2 engine. It is a file in which configuration parameters are placed. Dynamic values can be given as variables and they are replaced with real values at the time of playbook execution.

Install and upgrade of Solutions Enabler via Ansible

In a large production environment, lots of control hosts with Solutions Enabler managing VMAX arrays are required. Traditionally, Solutions Enabler is installed or upgraded on the control hosts by manually installing or upgrading the executable or binary on each server or running a script. With Ansible, we can simultaneously install or upgrade Solutions Enabler on as many control hosts as we want by simply sitting at the Ansible Control server. No need to also copy the binary manually to each server, Ansible does it for us. Now, we will go through the process of how this can be set up and achieved.

Setup for Solutions Enabler Install and upgrade



Copy the Solutions enabler binary to a directory on Ansible server.

```
[root@ilabansible01 code]# ls
ansible-tower-setup-bundle-3.2.6-1.e17      se8408-Linux-x86_64-ni.tar.gz  test-playbook.retry
ansible-tower-setup-bundle-3.2.6-1.e17.tar  se-playbook.yml                test-playbook.yml
```

Log on to Ansible Tower. Choose Inventory

Add TESTHOST1 which is the target control host.

TESTHOST1

DETAILS PERMISSIONS GROUPS HOSTS SOURCES COMPLETED JOBS

* NAME TESTHOST1 DESCRIPTION

INSIGHTS CREDENTIAL INSTANCE GROUPS ?

VARIABLES ? **YAML** JSON

```
1 ---
```

Edit the /etc/ansible/hosts file, Add

[TESTHOST1]
10.106.16.61

```
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups
#
# Ex 1: Ungrouped hosts, specify before any group headers.
[TESTHOST1]
10.106.16.61
```

Set up ssh connection to the Target Linux host. Ansible 2.6 requires at least Python 2.6 on all servers.

```
[root@ilabansible01 ~]# ansible --version
ansible 2.6.2
config file = /etc/ansible/ansible.cfg
```

Transfer the ssh keys from Ansible server to the target control host.

```
[root@ilabansible01 code]# cd /root/.ssh
[root@ilabansible01 .ssh]# scp id_rsa.pub
rstrusr@10.106.16.61:/root/.ssh/authorized_keys
The authenticity of host '10.106.16.61 (10.106.16.61)' can't be established.
RSA key fingerprint is
SHA256:hsVJPPggXCwievHeClvCRI+KwVHomkx4vFWiM6/5o0OU.
RSA key fingerprint is MD5:bc:77:3a:88:de:b1:d9:b8:22:2e:7d:94:ae:41:13:d1.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.106.16.61' (RSA) to the list of known hosts.
=====
=====
===== WARNING
=====
=====
=====
This system is solely for the use of authorized personnel. Individuals
using this system are subject to having some or all of their activities
monitored and recorded. Anyone using this system expressly consents to
such monitoring and is advised that any unauthorized or improper use of
this system may result in disciplinary action up to and including
termination of employment. Violators may also be subject to civil and/or
criminal penalties.
=====
=====

rstrusr@10.106.16.61's password:
id_rsa.pub
```

Verify connection

```
[root@ilabansible01 .ssh]# ansible -v 10.106.16.61 -m ping -u root
Using /etc/ansible/ansible.cfg as config file
10.106.16.61 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
[root@ilabansible01 .ssh]# cd /root/code
```


Construct the playbook file

```
[root@ilabansible01 code]# cat se-playbook.yml
---
- hosts: TESTHOST1
  tasks:
    - name: This is a test task
      command: mkdir /test
    - name: Extract SE into /test on remote node
      unarchive:
        src: "se8408-Linux-x86_64-ni.tar.gz"
        dest: "/test"
    - name: Execute SE install script
      command: sh "/test/se8408_install.sh" -install -silent
```

The playbook does the following:

- Creates a directory on the control host.
- Transfers the zipped tar file from the ansible server to the newly created directory on remote control host.
- Unzips the file and untars it on the remote control host.
- Upgrades Solutions Enabler from V8.1.2.0 (Edit Level: 2112) to V8.4.0.8 (Edit Level: 2352).
- Sets the path.

Run the playbook. Solutions Enabler gets upgraded on remote control host if Solutions Enabler already is present or installed if not present.

```
[root@ilabansible01 code]# ansible-playbook se-playbook.yml
PLAY [TESTHOST1] *****
TASK [Gathering Facts] *****
ok: [10.106.16.61]
TASK [This is a test task] *****
[WARNING]: Consider using the file module with state=directory rather than running mkdir.  If you need to use command because file is insufficient you can add warn=False to this command task or set command_warnings=False in ansible.cfg to get rid of this message.
changed: [10.106.16.61]
TASK [Extract SE into /test on remote node] *****
changed: [10.106.16.61]
TASK [Execute SE install script] *****
changed: [10.106.16.61]
PLAY RECAP *****
10.106.16.61 : ok=4  changed=3  unreachable=0  failed=0
```

On the Control host:

Before playbook run:

```
-bash-4.1# symcli
```

```
Symmetrix Command Line Interface (SYMCLI) Version V8.1.2.0 (Edit Level: 2112)
built with SYMAPI Version V8.1.2.0 (Edit Level: 2112)
```

After Playbook run:

```
-bash-4.1# symcli
```

**Symmetrix Command Line Interface (SYMCLI) Version V8.4.0.8 (Edit Level: 2352)
built with SYMAPI Version V8.4.0.8 (Edit Level: 2352)**

Solutions Enabler software is upgraded now to 8.4.0.8.

Ansible Requirements and What to do if Python version is behind on target hosts

Ansible uses Python for performing its tasks and therefore needs Python to be running on the target systems. Ansible version being used needs to be compatible with the Python version on managed nodes. Ansible 2.6 requires Python 2.6 or above to be running on managed nodes. Older Linux and UNIX releases such as RH 5 provide only Python 2.4 which poses a challenge when they are managed nodes. To work around this problem, upgrade Python version on the managed nodes to 2.6 by installing it to an alternate path so as not to break the system tools. (reference 4)

```
$ wget
https://archives.fedoraproject.org/pub/archive/epel/5/x86_64/epel-release-5-4.noarch.rpm
$ wget
https://archives.fedoraproject.org/pub/archive/epel/RPM-GPG-KEY-EPEL-5
$ sudo rpm --import RPM-GPG-KEY-EPEL-5
$ sudo yum install epel-release-5-4.noarch.rpm
$ sudo yum install python26
```

Python 2.6 is now installed at an alternative path, /usr/bin/python2.6

We need to tell Ansible to look for Python at a different place, by using the flag, “ansible_python_interpreter” in the inventory file.

In /etc/ansible/hosts file, edit as follows

[TESTHOST1]

```
10.106.16.61 ansible_python_interpreter=/usr/bin/python2.6
```

Conclusion

A great deal of staff resources and time can be saved by using a tool like Ansible which not only transfers the code to the target systems but also installs or upgrades the code in a matter of minutes without manual intervention. This can be achieved on multiple systems simultaneously. All it takes is to code a simple playbook in Ansible and run it. This constitutes a perfect example of a simple infrastructure-as-code solution in a Fiber Channel environment.

Appendix

1. Solutions-Enabler-8.4.0-Installation-and-Configuration-Guide
2. <https://www.dellemc.com/en-us/storage/>
3. <https://www.ansible.com/>
4. <https://www.ansible.com/blog/using-ansible-to-manage-rhel-5-yesterday-today-and-tomorrow>

Dell Technologies believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." DELL TECHNOLOGIES MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying and distribution of any Dell Technologies software described in this publication requires an applicable software license.

Copyright © 2019 Dell Inc. or its subsidiaries. All Rights Reserved. Dell Technologies, Dell, EMC, Dell EMC and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners.