SX-Aurora TSUBASA

Installation Guide (with OSS)

ChangeLog

Edition	Date of Issue	Chapter/Section	New/ Change	Changes
1	March 2018	All	New	
2	March 2018	2.3. Preparation of the software packages to install for the SX-Aurora TSUBASA system	Change	Download URL was changed.
		3.1. Installing software programs	Change	The reference URL was deleted in 3.1.1. Installing Cobbler and 3.1.2. Installing Ansible.
		4.3.2.1. The configuration of a kickstart file	Change	The reference URL was deleted.
		5.1.1. Files for Ansible	Change	vh-set-bmc.yml was deleted.
		5.1.2. Tuning Ansible performance	Change	The reference URL was deleted.
		6.1. Controlling power with BMC	Change	The explanation of the BMC operation user was added. 6.1.1.1. Checking BMC installation and 6.1.1.2. BIOS Configuration on VHs were deleted.
		A.3.1. VH power control using the powerctrl commands fails.	Change	Cause and Solution were changed.
3	May 2018	2.3. Preparation of the software packages to install for the SX-Aurora TSUBASA system	Change	NEC MPI was excluded from the install target which Ansible is used. NEC Parallel Debugger was added.
		3.2. Creating a Yum repository for SX-Aurora TSUBASA	Change	"4. Update the group definition file." was deleted.
		5.1.1. Files for Ansible	Change	roles/nec-mpi was deleted. roles/parallel-debugger was added.
		5.2. Installing SX-Aurora TSUBASA system software	Change	NEC MPI was excluded from the install target which Ansible is used.
		5.2.1. Editing Playbook	Change	nec-mpi was deleted. parallel-debugger was added.
4	May 2018	2.3. Preparation of the software packages to install for the SX-Aurora TSUBASA system	Change	Description of RHEL7.4 support for ScaTeFS was added. NEC MPI was re-supported as install target with Ansible.

Edition	Date of Issue	Chapter/Section	New/ Change	Changes
		5.1.1. Files for Ansible	Change	roles/nec-mpi was re-supported.
		5.2. Installing SX-Aurora TSUBASA system software	Change	NEC MPI was re-supported as install target with Ansible.
		5.2.1. Editing Playbook	Change	nec-mpi was re-supported.
5	Jan 2019	1.1. Overview	Change	"Software installation and environment setup" functions by Ansible suspend support.

Trademarks

- Linux is a registered trademark of Linus Torvalds in the United States and other countries.
- Red Hat and Red Hat Enterprise Linux are registered trademarks of Red Hat, Inc. in the United States and other countries.
- Apache is a registered trademark of Apache Software Foundation.
- InfiniBand is a trademark or service mark of InfiniBand Trade Association.
- Mellanox is trademark or registered trademark of Mellanox Technologies in Israel and other countries.
- Ansible is a registered trademark of Red Hat, Inc. in the United States and other countries.
- Python is a registered trademark of the Python Software Foundation.
- All other product, brand, or trade names used in this publication are the trademarks or registered trademarks of their respective trademark owners.

Copyright

The information in this document is subject to change at any time, without notice.

Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1. Overview

The SX-Aurora TSUBASA system allows configuring a system with more than hundreds or thousands of VH nodes mounting the Vector Engine (VE). As the system configuration grows larger, huge amount of time is required to install software programs on VHs and set up the environments for the software programs.

The SX-Aurora TSUBASA system provides procedures (this document) for setting up software environments using open source software (OSS), samples of OSS configuration files, and tools used in setting up environments, which automate installation of software programs on VHs and environment setup in a large scale system, leading to improved efficiency.

This document describes how to set up your SX-Aurora TSUBASA system environment by using OSS.

The table below shows OSS used to automate provisioning of the TSUBASA system environment.

Table 1: Used OSS

What is automated.	OSS
OS installation to VHs	Cobbler
Software installation and environment setup	Ansible

Notice

The "Software installation and environment setup" function by Ansible suspend support at Jan 2019.

This document includes the following contents and the description related to Ansible, but you cannot execute as the document. Please use only a reference.

- " *2.3. Preparation of the software packages to install for the SX-Aurora TSUBASA system* on page 12 "
- " *2.4. Preparing packages SX-Aurora TSUBASA system software programs depend on* on page 15 "
- " *3.1.2. Installing Ansible* on page 17 "
- " *3.1.5. Installing createrepo* on page 17 "
- " *3.2. Creating a Yum repository for SX-Aurora TSUBASA* on page 18 "
- " *3.3. Setting up the Mellanox OFED Yum repository* on page 18 "
- " *Chapter 5 Setting up of the SX-Aurora TSUBASA environment* on page 33 "

1.2. Glossary

The table below lists terms used in this document.

Terms	Description
VE	Engine based on NEC's vector architecture and developed to run vector programs

Terms	Description
VH	Linux server host mounting VE

1.3. System Architecture

The SX-Aurora TSUBASA system uses Cobbler and Ansible on the management server to automate environment setup of VHs. Because of this, the management server needs to connect to all VHs of the system through a network. Install Cobbler and Ansible on the management server. Install the HTTP server and the DHCP server on the management server as well to automatically install the OS on VHs using Cobbler.

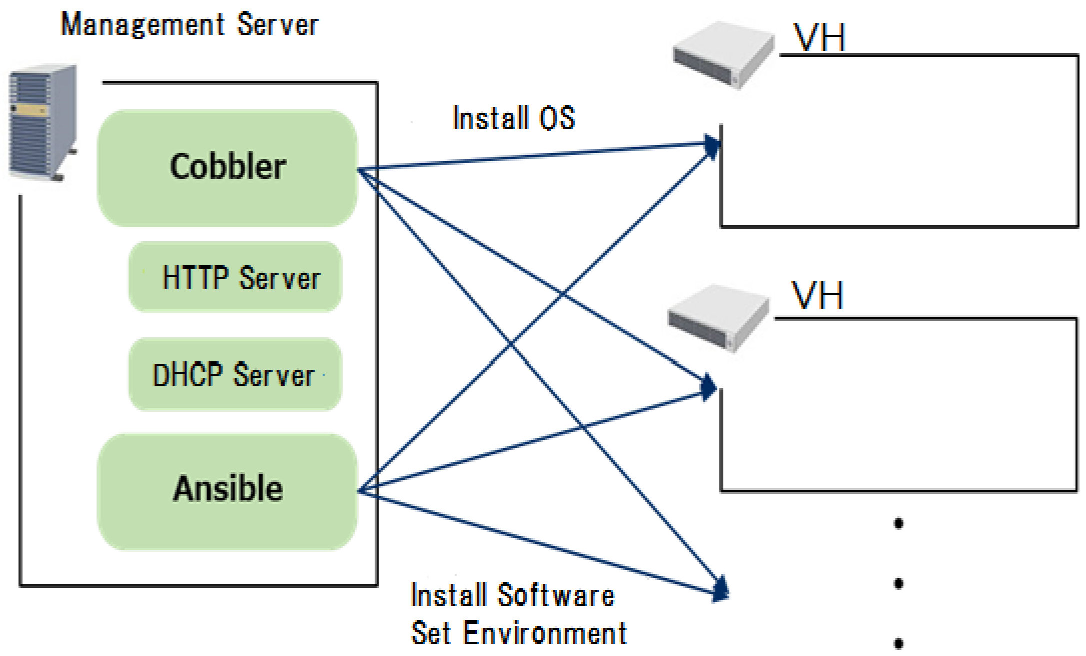

Figure 1: System Architecture

You may need to boot or power off VHs from the management server while setting up the VH environment. Configure your network to allow access from the management server to the BMC of VHs.

Using the same network for management and BMC:

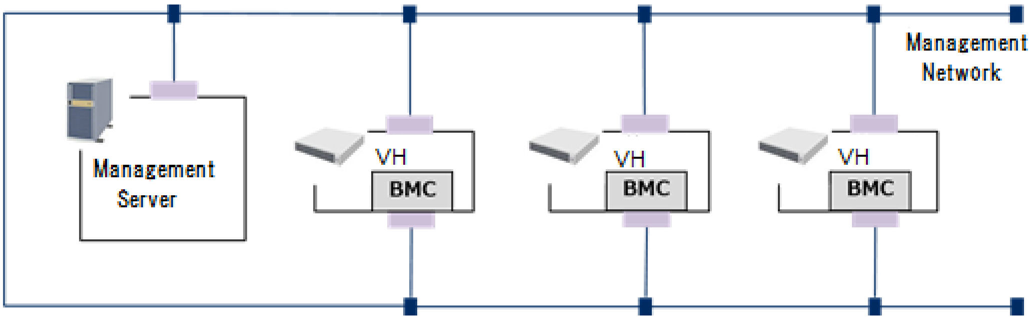

Figure 2: Network Layout 1

Using different networks for management and BMC:

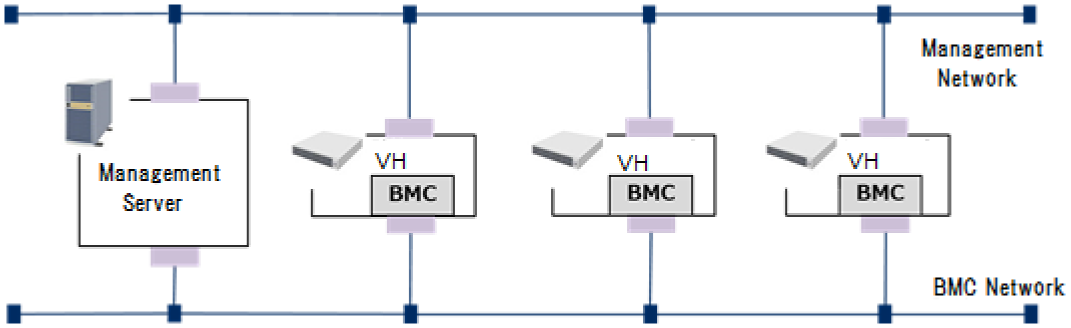

Figure 3: Network Layout 2

1.4. Operating Environment

The following environment supports automation of VH environment setup described in this document.

[Management Server]

H/W	x86_64 Architecture machine
OS	Red Hat Enterprise Linux 7
OSS	Cobbler 2.8.0 / 2.8.3
	Ansible 2.3.0.0
	IPMItool 1.8.15

[VH]

H/W	The models listed in the SX-Aurora TSUBASA product catalog.
OS	Please refer to *SX-Aurora TSUBASA Installation Guide*.

1.5. Sequence of environment setup

Follow the steps below to set up an environment for the SX-Aurora TSUBASA system.

1. Preparation

   Configure BIOS settings of VHs and prepare packages required for installation.
2. Set up the management server.

   Configure the management server environment, which includes installation of OSS used to set up an SX-Aurora TSUBASA environment on VHs.
3. Install the OS on VHs.

   Install the OS on VHs by using Cobbler.
4. Set up the SX-Aurora TSUBASA execution environment.

   Install software programs required for operation of the SX-Aurora TSUBASA system and start their services by using Ansible.
5. (Optional) Prepare a power control tool.

   Prepare a power control tool. If you do no use any power control tool, skip this step.

Chapter 2 Preparation

This chapter describes settings and files to be prepared before starting tasks described in this document.

2.1. BIOS settings of VHs

Please use UEFI mode as factory default at BOOT mode in BIOS settings. System boot or OS installation might be possible if you change the BIOS settings, but this equipment does not support it except factory default.

2.2. Preparation of the ISO file to install OS

Prepare the ISO image file of the OS to be installed on VHs. The IOS file will be used in " *4.3.3. Importing ISO image* on page 30 ".

2.3. Preparation of the software packages to install for the SX-Aurora TSUBASA system

Prepare the software packages of the SX-Aurora TSUBASA system and the yum repository's group definition file (TSUBASA-groups.xml). These files will be used in " *3.2. Creating a Yum repository for SX-Aurora TSUBASA* on page 18 " and " *5.2. Installing SX-Aurora TSUBASA system software* on page 37 ".

Some of the packages are released separately for RHEL versions. So download and prepare the packages for using system.

This document does not cover installing of C/C++ compiler and Fortran compiler. For installing the software, please refer to *SX-Aurora TSUBASA Installation Guide*.

- List of the Package files

  The following packages are included in the SX-Aurora TSUBASA system software.

Products	Package files	Free/ Non-free
License access library	aurlic-lib.x86_64	Free
VEOS Application Runtime	coreutils-ve.x86_64	Free
	gdb-ve.x86_64	
	libsysve-musl.x86_64	
	libved.x86_64	
	musl-libc-ve.x86_64	
	procps-ng-ve.x86_64	
	psacct-ve.x86_64	
	psmisc-ve.x86_64	

Products	Package files	Free/ Non-free
	strace-ve.x86_64	
	sysstat-ve.x86_64	
	time-ve.x86_64	
	util-linux-ve.x86_64	
	ve-memory-mapping.x86_64	
	ve_drv-kmod.x86_64	
	velayout.x86_64	
	veos.x86_64	
	veos-libveptrace.x86_64	
	veosinfo.x86_64	
	vesysinit.noarch	
	vesysinit-udev.noarch	
	vp-kmod.x86_64	
VEOS Application Development	autoconf-ve.noarch	Free
	automake-ve.noarch	
	gdb-ve.x86_64	
	libsysve-musl.x86_64	
	libsysve-musl-devel.x86_64	
	libtool-ve.x86_64	
	libved.x86_64	
	musl-libc-ve.x86_64	
	musl-libc-ve-devel.x86_64	
	vedebuginfo.noarch	
	velayout.x86_64	
	veos-libveptrace.x86_64	
	veos-musl-headers.x86_64	
InfiniBand for SX-Aurora TSUBASA	libibverbs-ve-musl.x86_64	Free
	libvedma-ve-musl.x86_64	
	libmlx5-ve-musl.x86_64	
	libveib.x86_64	
	ve_peermem.x86_64	
	ve_peermem.src	
MMM	ftmon.x86_64	Free
	libsignature.x86_64	

Products	Package files	Free/ Non-free
	mmm.x86_64	
	mmm-analysis.x86_64	
	mmm-msl.x86_64	
	rtmon.x86_64	
	ve-firmware.noarch	
	ve-power.x86_64	
ScaTeFS Client	[RHEL 7.3/CentOS 7.3]	Non-free
	scatefs-client-libscatefsib.x86_64	
	scatefs-client-libscatefsib_ve.x86_64	
	scatefs-client-modules-mlnx_ofed.x86_64	
	scatefs-client-mount-utils.x86_64	
	scatefs-client-rcli-utils.x86_64	
	scatefs-client-utils.x86_64	
	[RHEL 7.4 or later/CentOS 7.4 or later]	
	kmod-scatefs-client-modules-mlnx_ofed.x86_64	
	scatefs-client-libscatefsib.x86_64	
	scatefs-client-libscatefsib_ve.x86_64	
	scatefs-client-mount-utils.x86_64	
	scatefs-client-rcli-utils.x86_64	
	scatefs-client-utils.x86_64	
NEC MPI	nec-mpi-devel-1-0-0.x86_64	Non-free
	nec-mpi-libs-1-0-0.x86_64	
	nec-mpi-utils-1-0-0.x86_64	
	nec-mpi-runtime.x86_64	
Tuning Tool	nec-veperf.x86_64	Non-free
	nec-ftraceviewer.x86_64	
NEC Parallel Debugger	nec-paralleldebugger.x86_64	Non-free
NQSV/JobServer	NQSV-JobServer.x86_64	Non-free
NQSV/Client	NQSV-Client.x86_64	Non-free
Numeric Library Collection	nec-asl-ve-1.0.0.x86_64	Non-free
	nec-aslfftw-ve-1.0.0.x86_64	
	nec-blas-ve-1.0.0.x86_64	
	nec-heterosolver-ve-1.0.0.x86_64	

Products	Package files	Free/ Non-free
	nec-lapack-ve-1.0.0.x86_64	
	nec-nlc-base-1.0.0.noarch	
	nec-nlc-doc-1.0.0.noarch	
	nec-sblas-ve-1.0.0.x86_64	
	nec-scalapack-ve-1.0.0.x86_64	
binutils	binutils-ve.x86_64	Non-free
C/C++ compiler	nec-nc++.x86_64	Non-free
	nec-nc++-musl-inst.noarch	
	nec-nc++-doc.noarch	
Fortran compiler	nec-nfort.x86_64	Non-free
	nec-nfort-musl-inst.noarch	
	nec-nfort-doc.noarch	

These files can be got from the following places.

Files	Places
Non-free software	NEC Internet delivery
TSUBASA-groups.xml License access library Free software	" *https://jpn.nec.com/hpc/aurora/ve-software/* " or " *https://www.nec.com/en/global/prod/hpc/aurora/ve-software/* "

There are difference in the contents of the group definition file (TSUBASA-groups.xml) for each OS version. Please download the file of the corresponding version.

2.4. Preparing packages SX-Aurora TSUBASA system software programs depend on

When you install SX-Aurora TSUBASA InfiniBand on VHs, you need to install Mellanox OFED as well. Please download ISO image, Mellanox OFED for Linux User Manual and Mellanox OFED for Linux Release Notes from following website of Mellanox Technologies.

" *http://www.mellanox.com/page/products_dyn?product_family=26* "

The correspondence of the OS version of VHs and version of Mellanox OFED to be installed is as follows.

OS	Mellanox OFED
RHEL/CentOS 7.3	Mellanox OFED 3.4-2.1.9.0.1
RHEL/CentOS 7.4	Mellanox OFED 4.2-1.2.0.0
RHEL/CentOS 7.5	Mellanox OFED 4.3-3.0.2.1

The ISO file will be used in " *3.3. Setting up the Mellanox OFED Yum repository* on page 18 ".

If you are not installing SX-Aurora TSUBASA InfiniBand, the procedures in this section are not necessary.

2.5. Preparing of the template package

The template package, TSUBASA-sysmng-soft-X.X- Y.noarch.rpm, provides sample files and tools described in this document. Download the latest TSUBASA-sysmng-soft-X.X- Y.noarch.rpm and save it in any directory you wish on the management server.

Chapter 3
Setting up the management server

3.1. Installing software programs

Install the following software programs used for setting up the VH environment.

3.1.1. Installing Cobbler

Install Cobbler on the management server. For information about verified versions, see " *1.4. Operating Environment* on page 11 ". Please refer to the official website of Cobbler for the installation sequence.

3.1.2. Installing Ansible

Install Ansible on the management server. For information about verified versions, see " *1.4. Operating Environment* on page 11 ". Please refer to the official website of Ansible for the installation sequence.

3.1.3. Installing Python

The conversion tool of the VH information file, " *7.2. create-hostlist.py* on page 46 " uses python. Check Python 2.7 or later of the Python 2.x series is installed.

The xlrd package of the python-excel library is used. Install and run the `pip` command to install the xlrd package.

Example:

```
# yum install python-pip
# pip install xlrd
```

3.1.4. Installing Apache HTTP Server

As described in " *Chapter 4 OS installation on VHs* on page 22 " and " *Chapter 5 Setting up of the SX-Aurora TSUBASA environment* on page 33 ", you use the HTTP server on the management server to download files to each VH. Install the Apache HTTP Server on the management server and start the HTTP server.

```
# yum -y install httpd
# systemctl start httpd.service
```

3.1.5. Installing createrepo

As described in " *Chapter 5 Setting up of the SX-Aurora TSUBASA environment* on page 33 ", the SX-Aurora TSUBASA system software programs are installed on VHs from the Yum repository for SX-Aurora TSUBASA on the management server. Please install the tool to create Yum repository, 'createrepo' on the management server.

```
# yum -y install createrepo
```

3.2. Creating a Yum repository for SX-Aurora TSUBASA

SX-Aurora TSUBASA system software programs are installed from the SX-Aurora TSUBASA Yum repository as described in " *Chapter 5 Setting up of the SX-Aurora TSUBASA environment* on page 33 ".

Please create the SX-Aurora TSUBASA Yum repository on the management server by the following procedure.

1. Create a directory for the repository.

   Create a directory for the repository in the DocumentRoot directory of the HTTP server.

   ```
   # mkdir -p /var/www/html/repos/TSUBASA/rpms
   ```

2. Store the SX-Aurora TSUBASA system software packages.

   Place the SX-Aurora TSUBASA system software packages you have prepared as instructed in " *2.3. Preparation of the software packages to install for the SX-Aurora TSUBASA system* on page 12 " in the directory created in the previous step.

   ```
   # ls -1 /var/www/html/repos/TSUBASA/rpms/
   libsignature-0.3.6-1.x86_64.rpm
   mmm-0.3.6-1.x86_64.rpm
    :
   ```

3. Store the group definition file.

   Place the group definition file of the SX-Aurora TSUBASA Yum repository (TSUBASA-groups.xml) you have prepared as instructed in " *2.3. Preparation of the software packages to install for the SX-Aurora TSUBASA system* on page 12 " in `[ /var/www/html/repos/TSUBASA ]`.

   ```
   # ls /var/www/html/repos/TSUBASA
   TSUBASA-groups.xml  rpms
   ```

4. Create a Yum repository.

   Run the createrepo command to create a Yum repository.

   ```
   # createrepo -v -g TSUBASA-groups.xml /var/www/html/repos/TSUBASA
   ```

   This command creates 'repodata' directory under `[ /var/www/html/repos/TSUBASA ]` and also creates repomd.xml' file and database files under the 'repodata' directory.

   Caution
   Do not grant the write permission on the directories and files created in /var/www/html through the steps above to group and others. Set 755 for the directory permission and 644 for the file permission.

   Caution
   If you add or replace any package file, run the createrepo command to update the repository.

3.3. Setting up the Mellanox OFED Yum repository

In this section, How to setup the Yum repository for Mellanox OFED required by SX-Aurora TSUBASA InfiniBand is described. In " *5.2. Installing SX-Aurora TSUBASA system software* on page 37 " Mellanox OFED is installed on each VHs from the Yum repository of Mellanox OFED.

Please skip this section if you don't install SX-Aurora TSUBASA InfiniBand.

1. Mount the ISO image.
   Mount the ISO image file you have prepared as instructed in " *2.4. Preparing packages SX-Aurora TSUBASA system software programs depend on* on page 15 " on any directory of the management server. The following is an example of mounting the ISO image file on /mnt/mlnx-ofed.

```
# mkdir /mnt/mlnx-ofed
# mount -o ro,loop MLNX_OFED_LINUX-X.X-n.n.n.n-rhel7.3-x86_64.iso /
mnt/mlnx-ofed
```

2. Create a link to the directory mounted on the DocumentRoot directory of the HTTP server in step 1 to allow referencing the directory by HTTP.

```
# ln -s /mnt/mlnx-ofed /var/www/html/repos/mlnx-ofed
```

3.4. Creating an administrative user

Create an account for an administrative user admin. And it is required to give sudo privilege and to disable requiretty for admin user. An administrative user admin is a user who runs Ansible commands for setting up the VH environment.

Example

```
# useradd -m admin
# passwd admin
 Changing password for user admin.
 New UNIX password: (Input password)
Retype new UNIX password: (re-input password)
passwd: all authentication tokens updated successfully.
# visudo
     :
admin ALL=(ALL) NOPASSWD: ALL
Defaults:admin !requiretty
```

When you are using Ansible, an administrative user admin remotely logs in VHs from the management server over SSH. Because of this, you need to create the administrative user admin on the VHs as well. Create the administrative user for VHs when you install the OS on VHs by Cobbler. Creating the administrative user for VHs will be performed at the installation of the OS on VHs by Cobbler. Please refer to " *Chapter 4 OS installation on VHs* on page 22 " for details.

3.5. Setting administrative user's SSH public and private keys

Ansible used for provisioning the VH environment first logs into VHs over SSH for further operations. Generate public and private keys for the administrative user admin to log in VHs over SSH, and change the SSH configuration.

- Creating a public/private key pair

```
# su - admin
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/admin/.ssh/id_rsa):
Enter passphrase (empty for no passphrase): (Inout passphrase)
Enter same passphrase again: (Re-input passphrase)
Your identification has been saved in /home/admin/.ssh/id_rsa.
Your public key has been saved in /home/admin/.ssh/id_rsa.pub.
The key fingerprint is:
      :
```

  The public key is created in the ~admin/.ssh/id_rsa.pub file. This public key file is distributed to each VHs at the OS installation by Cobbler. Details are described in " *Chapter 4 OS installation on VHs* on page 22 ".

- Changing the SSH configuration

Change the SSH configuration of administrative user admin, `[ ~admin/.ssh/config ]`, as follows to prevent the warning message from appearing when connecting to a VH over SSH.

```
$ cat ~/.ssh/config
Host *
   StrictHostKeyChecking  no
$ chmod 600 ~/.ssh/config
```

3.6. Installing the template package

Install the template package for the sample files and tools described in this document on the management server.

The template package is provided as TSUBASA-sysmng-soft-*X.X-Y*.noarch.rpm. Get the latest version of the TSUBASA-sysmng-soft package and install it on the management server.

```
# rpm -i TSUBASA-sysmng-soft-X.X-Y.noarch.rpm
```

The sample files and tools are installed under `[ /opt/nec/sysmng-soft/ ]`.

3.7. Creating the VH information file

You need information such as host names, IP addresses, and MAC addresses of VHs to install the OS by Cobbler and set up the environment and control power of VHs by Ansible. Before starting the installation, create a VH information file containing information about VHs you use.

For the information file, use the `[ /opt/nec/sysmng-soft/etc/host-list.xlsx ]` template file installed by the template package, and enter information related to all VHs you use. Save the `[ /opt/nec/sysmng-soft/etc/host-list.xlsx ]` file on the management server.

The VH information file is an Excel file and it includes the following information.

Table 2: Items in the VH information file

Items	category	Tools by which the item is used.
Host Name	Required	Cobbler, Ansible, Power management tool
Group Name	Optional	Ansible
IP address	Required	Cobbler
IP address for BMC	Optional	Power management tool
MAC address	Required	Cobbler, Power management tool
Netmask of IP address	Required	Cobbler
Gateway	Optional	Cobbler

- Host name, IP address netmask, and MAC address must be registered.
- For IP address, specify the IP addresses assigned to VHs. If IP addresses of VHs are assigned by the DHCP server, specify the same IP addresses the DHCP server assigns in the VH information file.
- Specify group names if you do not install the same software programs on all VHs but want to select software programs to be installed based on VH roles. For details, see " *5.2.3. Specifying software according to roles of VHs.* on page 39 ".
- Specify BMC IP addresses if your power control tool uses BMC for power control.

Caution

- Do not change the file format of the template file.
- Use the AA:BB:CC:DD:EE:FF format to enter MAC addresses.

- For optional items you do not use, leave their cells blank.
- This guide does not provide information about setting up the DHCP server even if the DHCP is used to assign IP addresses to VHs.

The VH information file created here is converted to files according to entry formats of tools described in " *3.8. Creating host list file* on page 21 ".

3.8. Creating host list file

On the management server, create all host list files described in this document by using the VH information file created in " *3.7. Creating the VH information file* on page 20 ". The host list files contain information about VHs referenced by Cobbler, Ansible and the power control tool in the entry format of each tools.

Execute the following command as the administrative user admin.

```
$ /opt/nec/sysmng-soft/bin/setup-hostlist
```

This command creates the following files.

`[ /opt/nec/sysmng-soft/etc/ansible/vh-hosts ]`

Host list file in the inventory file format defining hosts to be managed when Ansible is used to manage the VH configuration.

`[ /opt/nec/sysmng-soft/etc/cobimport.txt ]`

The Host list file for the cobimport.sh tool, which registers systems at once when Cobbler is used to install the OS on VHs.

`[ /opt/nec/sysmng-soft/etc/vh_host.conf ]`

The host list file defining those controlled by the powerctrl tool when BMC is used to control power.

`[ /opt/nec/sysmng-soft/etc/vh_host_w.conf ]`

The host list file defining those controlled by the powerctrl tool when WOL is used to control power.

Caution
When the VH information file is modified, be sure to re-create the host list files by `setup-hostlist` command.

Chapter 4
OS installation on VHs

This chapter describes how to automatically install the OS on VHs by using Cobbler.

The OS is installed on VHs in the following sequence.

1. Making preparations to use Cobbler

   Start services required to use Cobber on the management server and disable SELinux and the firewall.

2. Configuring Cobbler

   Configure Cobbler on the management server.

3. Preparing for OS installation

   Register information required for OS installation with the Cobbler server.

4. Starting OS installation

4.1. Making preparations to use Cobbler

To use Cobbler, change the management server settings and start required services.

4.1.1. Configuration of xinetd server

Cobbler uses the tftp service to distribute the OS boot kernel and the initrd image to VHs. Enable the tftp service on the xinetd server.

1. Edit the tftp service configuration file.

   Change *disable* in the `[ /etc/xinetd.d/tftp ]` file to *no*.

```
service tftp
{
    disable = no
    :
}
```

2. Restart xinetd server.

   Execute the following command to restart xinetd server.

```
# systemctl restart xinetd.service
```

4.1.2. Configuration of rsync server

When Cobbler copies a package file to VHs, it uses rsync to synchronize the file and directory. Start up the rsync server.

```
# systemctl start rsyncd.service
# systemctl enable rsyncd.service
# systemctl list-unit-files -t service
# systemctl status rsyncd.service
```

4.1.3. Disabling SELinux and the firewall

If SELinux and the firewall are enabled on the management server, temporarily disable them before you start using Cobbler.

> **Caution**
> Restore the SELinux and firewall configurations after installing the OS.

- Disabling SELinux

```
# setenforce Permissive
```

- Disabling the firewall

```
# systemctl stop firewalld.service
# systemctl disable firewalld.service
# systemctl is-enabled firewalld.service
```

4.2. Configuration of Cobbler

This section describes about the configuration of Cobbler on the management server.

4.2.1. Editing the Cobbler configuration file

In the Cobbler configuration file, `[ /etc/cobbler/settings ]`, specify the IP address of the Cobbler server and how the DHCP server is managed.

- IP address of the Cobbler server

  Specify the IP address of the machine where the Cobbler server runs; specify the IP address of the management server.

```
server: management-server-ipaddr
```

- Setting for DHCP server

  When Installing the VH machine by Cobbler, Cobbler server get the IP address information from DHCP server. If a DHCP server already exists and the information of target machine is registered, set "manage_dhcp" to "0". Otherwise, set "manage_dhcp" to "1".

```
manage_dhcp: 0
or
manage_dhcp: 1
```

  When "manage_dhcp" is "1", a DHCP server for installation is executed by Cobbler server. Once you have installed the OS on VHs, stop DHCP server. Plase refer " *4.5. Post OS installation tasks* on page 31 ".

- IP address of tftp server

  Specify the IP address of the machine where the tftp server runs; specify the IP address of the management server.

```
next_server: management-server-ipaddr
```

- Password of the root user on VHs

  Determine the root password of VHs and set the encrypted string of the password.

```
default_password_crypted: crypted-password-string
```

  To obtain *crypted-password-string* described above, run the `openssl` command on the management server and enter the root password of VHs.

```
# openssl passwd -1
Password: (Input root password)
Verifying - Password: (Re-input root password)
crypted-password-string
```

4.2.2. Configuration of DHCP server

When you use the existing DHCP server, please register information of target machine to a DHCP server. And, to install the OS in UEFI boot mode, edit 'filename' in the '#else' to specify "grub/grub-x86_64.efi" on the group block of `[ /etc/dhcp/dhcpd.conf ]`.

When you execute a DHCP server by Cobbler server, edit the template file `[ /etc/cobbler/dhcp.template ]`. Cobbler creates a configuration file `[ /etc/dhcp/dhcpd.conf ]` for DHCP server from the template file.

4.2.2.1. Basic settings of dhcp.template

Edit the subnet block of the template file, `[ /etc/cobbler/dhcp.template ]`, according to the VH environment.

The below is the default settings of the `[ /etc/cobbler/dhcp.template ]`.

```
subnet 192.168.1.0 netmask 255.255.255.0 {
     option routers             192.168.1.5;
     option domain-name-servers 192.168.1.1;
     option subnet-mask         255.255.255.0;
     range dynamic-bootp        192.168.1.100 192.168.1.254;
     default-lease-time         21600;
     max-lease-time             43200;
     next-server                $next_server;
     (omitted below)
}
```

`subnet <subnet-number> netmask <mask-number>`	Specify network address and its netmask.
`option routers`	Specify the router's IP address.
`option domain-name-servers`	Specify a DNS server's IP address.
`option subnet-mask`	Specify sub-netmask of client hosts
`range dynamic-bootp`	The range of IP addresses dynamically allocated to the clients. Remove this item.
`default-lease-time`	The default lease period of IP address in seconds.
`max-lease-time`	The maximum lease period of IP address in seconds.
`next-server`	Don't touch this item.

4.2.2.2. Installation in UEFI boot mode

To install the OS in UEFI boot mode, edit the group block of `[ /etc/cobbler/dhcp.template ]`. Edit 'filename' in the '#else' clause of the conditional statement '#if $iface.enable_gpxe:' to specify "grub/grub-x86_64.efi".

```
group {
        (omitted)
    host $iface.name {
        (omitted)
        #if $iface.enable_gpxe:
        if exists user-class and option user-class = "gPXE" {
            filename "http://$cobbler_server/cblr/svc/op/gpxe/system/$iface.owner";
        } else if exists user-class and option user-class = "iPXE" {
            filename "http://$cobbler_server/cblr/svc/op/gpxe/system/$iface.owner";
        } else {
            filename "undionly.kpxe";
        }
        #else
        ##filename "$iface.filename";
        filename "grub/grub-x86_64.efi";
        #end if
        (omitted)
}
```

4.2.2.3. The management server and VH machines on different subnets

By default, Cobbler installs the OS only on clients on the same subnet where the Cobbler machine resides. When the management server where the Cobbler server is located, and VH machines reside on different subnets, use the DHCP relay agent function of the router. The DHCP relay agent function allows communicating messages between a DHCP server and DHCP clients residing in different subnets. See the manual of your router to set the DHCP relay agent function.

The subnet block in [`/etc/cobbler/dhcp.template`] should be set by each subnet.

Example when installing VHs on 2 subnets:

```
 subnet 192.168.2.0 netmask 255.255.255.0 {
     option routers             192.168.2.5;
     option domain-name-servers 192.168.1.1;
     option subnet-mask         255.255.255.0;
     range dynamic-bootp        192.168.2.100 192.168.2.254;
     default-lease-time         21600;
     max-lease-time             43200;
     next-server                $next_server;
     (Omitted)
}

subnet 192.168.3.0 netmask 255.255.255.0 {
     option routers             192.168.3.5;
     option domain-name-servers 192.168.1.1;
     option subnet-mask         255.255.255.0;
     range dynamic-bootp        192.168.3.100 192.168.3.254;
     default-lease-time         21600;
     max-lease-time             43200;
     next-server                $next_server;
     (Omitted)
}
```

4.2.3. Starting the Cobbler server

Follow the steps below to start the Cobbler server to check the configuration.

1. Starting http server

   ```
   # systemctl start httpd.service
   ```

2. Starting Cobbler server

   ```
   # systemctl start cobblerd.service
   ```

3. Check the Cobbler server configuration

   ```
   # cobbler check
   ```

   The following message appears when no configuration problems are found.

   No configuration problems found. All systems go.

When there is any configuration problem, an error message appears. Use the error message to solve the problem and repeat " *Check the Cobbler server configuration* ". Repeat this process until all errors are fixed.

Example of error message

1 : SELinux is enabled. Please review the following wiki page for details on ensuring cobbler works correctly in your SELinux environment:

If this message is displayed, disable SELinux described as " *4.1.3. Disabling SELinux and the firewall* on page 23 ".

4.3. Preparation of OS installation

Register the information required for OS installation with the Cobbler server.

4.3.1. Obtaining and storing bootloaders

Cobbler distributes network bootloaders to VHs from the management server. Obtain the network bootloaders Cobbler provides from their official website.

Run `get-loaders`, a sub command of the `cobbler` command, to obtain the network bootloaders.

```
# cobbler get-loaders
```

When a proxy is set on the management server, configure the proxy server in `[ /etc/cobbler/settings ]` and run the command above.

```
proxy_url_ext: "proxy-server-ipaddr:port-number"
```

4.3.2. Editing a kickstart file

Cobbler uses the kickstart templating function to customize installation of the OS on VHs. This section describes how to edit a kickstart file for this function.

Copy the sample file of the kickstart file, '`[ /var/lib/cobbler/kickstarts/sample_end.ks ]` to the actual file.

```
# cd /var/lib/cobbler/kickstarts
# cp sample_end.ks kickstart.ks
```

Use the explanation in this section as reference to edit the *kickstart.ks* file.

4.3.2.1. The configuration of a kickstart file

A kickstart file broadly consists of the following three sections where you can specify your customization.

Command section
Write commands describing the settings for installation at the top of the file.

Package section(%packages)
Specify packages to install by starting the section with the %packages line.

Script section(%pre, %post)

- **The section started with %pre** Specify a script to run before the installation starts.
- **The section started with %post** Specify a script to run after the installation.

This document provides examples of customization only for basic items. Make sure to configure the following to use Ansible for provisioning an environment for the SX-Aurora TSUBASA system.

- Registration of Ansible administrative user.
- Grant sudo permissions to Ansible administrative user.
- Disabling requiretty for Ansibleadministrative user.
- Registration of Ansible administrative user's SSH public key.

For customization of other items, visit Cobbler's official website. For the details of the syntax of kickstart file, please refer to the installation guide of Linux OS.

4.3.2.2. Command section

In the command section, specify items such as installation method, time zone, and partitioning scheme. Register an Ansible administrative user admin as well.

1. Settings of keyboard

   Specify the type of keyboard to use.

   When using the Japanese keyboard:

```
# System keyboard
keyboard jp106
```

2. Settings of language

   Specify the language to use.

   When using the Japanese UFT-8:

```
# System language
lang ja_JP.UTF-8
```

3. Settings of timezone

   Specify the time zone to use. To use the hardware clock time for UTC, specify --isUtc.

   When using the Japan Standard Time:

```
# System timezone
timezone  Asia/Tokyo --isUtc
```

4. Partitioning

   Change partitioning scheme of VH machines according to their configurations.

   The following is an example of setting partitioning without using the auto partition function (autopart).

   Create a /boot/efi partition for using UEFI boot mode.

```
# Allow anaconda to partition the system as needed
part /boot/efi --fstype vfat --size=value --ondisk=sda
part /boot --fstype ext3 --size=value --ondisk=sda
part swap --size=value --ondisk=sda
part pv.id --size=value --ondisk=sda
volgroup volname pv.id
logvol /    --vgname=volname --size=value --name=logvolname-root
logvol /var --vgname=volname --size=value --name=logvolname-var
logvol /tmp --vgname=volname --size=value --name=logvolname-tmp
logvol /home --vgname=volname --size=value --grow --name=logvolname-home
```

5. Creating an Ansible administrative user admin.

Add the following user command, and specify the password encrypting method in the auth command.

```
# System authorization information
auth --useshadow --passalgo=sha512
  (Omitted)
# Add the admin user for Ansible
user --name=admin --homedir=homedirpath --shell=shellpath --uid=userid --
gid=groupid  --iscrypted --password=cryptedpassword
```

Use the user command above to add an administrative user admin, as a user belonging to the group admin (group ID: *groupid*).

Manage the home directory specified for --homedir as a local directory and do not share the directory by shared file systems. Use the following command to create *cryptedpassword*, the encrypted string of the password specified in the --password option. The following command is an example of using SHA512 for hashing.

```
$ python
>>> import crypt, getpass
>>> print(crypt.crypt(getpass.getpass(), "$6$salt"))
Password: (Input password)
cryptedpassword
>>> exit()
```

4.3.2.3. Package section

In the package section, specify packages to install. Specify packages by group names or package names. For information about groups that can be specified, see the installation guide of the OS.

You do not need to specify Core and Base groups because they are selected by default. Because OpenSSH and python, which are required for using Ansible, are included in these groups, no change is necessary in the package section for OpenSSH and python.

If you want to install any additional package, refer to the following for customization.

The %packages section of the kickstart sample file uses the SNIPPET() function.

```
%packages
$SNIPPET('func_install_if_enabled')
%end
```

Edit `[ /var/lib/cobbler/snippets/func_install_if_enabled ]`, the file the SNIPPET function imports, and specify packages to be installed. The following describes how to edit.

- Specifying a package group

  Enter one group per line. Start a line with the at sign (@) followed by the group name. The following is an example of installing packages belonging to X Windows System and the network file system client.

```
@X Window System
@network-file-system-client
```

- Specifying a package

  Enter one package per line. You may use an asterisk (*) for a wildcard character. The following is an example of installing sqlite, curl, and aspell packages as well as any package starting with docbook.

```
sqlite
curl
aspell
docbook*
```

4.3.2.4. Script section

In the script section, specify the processes to run before or after the OS installation as described below.

1. Settings not to use the Yum repository of the Cobbler server.
   For VHs configured not use the Yum repository of the Cobbler server, comment out the "$yum_config_stanza" specification.

```
%post
:
# Start yum configuration
#$yum_config_stanza
# End yum configuration
:
%end
```

2. Adding processes to use Ansible.

   To use Ansible, specify the following processes as scripts to run after the OS installation.

   - Grant sudo privilege to Ansible administrative user.
   - Disabling requiretty for Ansibleadministrative user.
   - Registration of Ansible administrative user's SSH public key.

   A sample of the script file is provided. Edit the sample by the following sequence.

   a. Copy the sample script file.

      Copy the sample of the script file installed in " *3.6. Installing the template package* on page 20 " into the Cobbler environment.

```
# cp /opt/nec/sysmng-soft/etc/vh_admin_conf /var/lib/cobbler/snippets/
```

   b. Edit the script file.

      Edit the [/var/lib/cobbler/snippets/vh_admin_conf] file.

      Content of the sample file:

```
# Add sudo privileges and disable requiretty for the admin user
chmod +w /etc/sudoers
echo 'admin ALL=(ALL) NOPASSWD:ALL' >> /etc/sudoers
echo 'Defaults:admin !requiretty' >> /etc/sudoers
chmod -w /etc/sudoers

# Set the ssh public key for admin user
cd ~admin
mkdir --mode=700 .ssh
cat >> .ssh/authorized_keys << PUBLIC_KEY
ssh-rsa AAAAB3NzaC1yc2E ... xPzomFvVn/g9QOJ admin@serverhostname
PUBLIC_KEY
chown admin.admin -R .ssh
chmod 600 .ssh/authorized_keys
```

      Replace the here document section using PUBLIC_KEY as a delimiter with the content of [/home/admin/.ssh/id_rsa.pub] obtained in " *3.5. Setting administrative user's SSH public and private keys* on page 19 ".

   c. Register the script in the kickstart file.

      Add the process to import [vh_admin_conf] by using the SNIPPET() function to the %post section of the kickstart file.

```
%post
$SNIPPET('vh_admin_conf')
$SNIPPET('log_ks_post')
:
%end
```

To apply the changes in the kickstart file, specify the kickstart file by using the --kickstart option of the `cobbler import` command that registers profiles, and run the command. Details of the cobbler import command are provided in " *4.3.3. Importing ISO image* on page 30 ".

4.3.3. Importing ISO image

Import the ISO image file of the OS to be installed on VHs to Cobber, define the distribution to be provided, and register profiles. A profile associates a system name and the distribution to be provided.

1. Mounting ISO image

On the management server, mount the ISO image file of the OS to be installed. The following is an example of acquiring the ISO image from a CD or DVD.

a. Insert the CD or DVD into the CD-ROM drive.

b. Mount the CD-ROM drive.

```
# mount -t iso9660 -o ro /dev/cdrom /media/cdrom
```

2. Importing the ISO image

Import the ISO image file to Cobbler by using the `import` sub command of the `cobbler` command.

```
# cobbler import --name=rhelN.N-x86_64 --arch=x86_64 \
 --path=/media/cdrom --breed=redhat \
 --kickstart=/var/lib/cobbler/kickstarts/kickstart.ks
```

`--name` Specify the name of the distribution to be provided.

`--arch` Specify the architecture type.

`--path` Specify the path to the location where the ISO image file is stored.

`--breed` Specify the OS type of kernel and other parameters applied for OS installation. Specify *redhat*.

`--kickstart` Specify the path name of the kickstart file created in " *4.3.2. Editing a kickstart file* on page 26 ".

Please refer to the manual of `cobbler` command for the details of these options.

3. Check the distribution.

When you run the `cobbler import` command, the specified distribution is registered. Run either of the following commands to check the distribution is registered successfully.

```
# cobbler distro list
```

```
# cobbler distro report --name=rhelN.N-x86_64
```

4. Check profiles

When you run the `cobbler import` command, profiles are registered. Run either of the following commands to check the profiles are registered successfully.

```
# cobbler profile list
```

```
# cobbler profile report --name=rhelN.N-x86_64
```

4.3.4. Registration of system

Cobbler installs the OS according to profiles registered for each of machines targeted for the installation. Because of this, information specific to each targeted machine (VH) (such as MAC address, IP address) must be registered with Cobber as system records.

Use the `system` sub command of the `cobbler` command for the registration.

1. Register system

   You may use `cobimport.sh`, a bulk registration tool, to register systems with Cobbler. Run the following command.

   ```
   # /opt/nec/sysmng-soft/bin/cobimport.sh \
   --hostlist=/opt/nec/sysmng-soft/etc/cobimport.txt \
   --profile=profilename --interface=interface --static=1
   ```

   The host list file cobimport.txt specified for the --hostlist option is the file created in " *3.8. Creating host list file* on page 21 ". For details about the tool, see " *7.1. cobimport.sh* on page 45 ".

   Configure the --static option and choose whether to specify the IP addresses set to the VHs as static addresses or assign addresses from the DHCP server. Specify --static=1 to assign the IP addresses of the VHs as static IP addresses. Omit the --static option to assign IP addresses from the DHCP server.

   This tool allows registering all the VHs on the host list file at once. In this case, VH host names are registered as system names.

   You may register VHs individually without using the bulk registration tool. In this case run the `cobbler system` command.

   ```
   # cobbler system add --name=systemname --profile=profilename \
    --hostname=hostname --interface=interface --static=1 \
    --ip-address=ipaddr --subnet=subnetmask \
    --gateway=gateway --mac=macaddress
   ```

   For details of each options, please refer to manual page of the `cobbler` command.

2. Check the system

   Information about VHs is registered as systems. Run the following command to check systems are registered successfully.

   ```
   # cobbler system list
   ```

   ```
   # cobbler system report --name=systemname
   ```

4.4. Starting OS installation

Apply the information you have registered in the Cobbler server.

```
# cobbler sync
```

Now you are ready to install the OS on VHs.

Important

When you have updated any file related to Cobbler settings, make sure to apply the latest status in the Cobbler server using `cobbler sync` command.

When you manually power on each VH and network boot starts, the OS is automatically installed. After the OS installation, VHs are started.

4.5. Post OS installation tasks

Once you have installed the OS on VHs, stop the Cobbler server and restore the SELinux and firewall configurations. Stop rsync, xinetd, and DHCP servers Cobbler used. You do not need to stop the HTTP server because you will use it for " *Chapter 5 Setting up of the SX-Aurora TSUBASA environment* on page 33 ".

1. Stop Cobbler server

```
# systemctl stop cobblerd.service
```

2. **Restore the firewall configuration.**
   If you temporarily disabled the firewall to use Cobber in " *4.1.3. Disabling SELinux and the firewall* on page 23 ", restore the configuration.
3. **Restore the SELinux configuration.**
   If you temporarily disabled the SELinux to use Cobber in " *4.1.3. Disabling SELinux and the firewall* on page 23 ", restore the configuration.
4. **Stop rsync server**
   If you do not use the rsync server on the management server, stop the rsync server.

   ```
   # systemctl stop rsync.service
   ```
5. **Stop xinetd server**
   If you do not use the xinetd server on the management server, stop the xinetd server.

   ```
   # systemctl stop xinetd.service
   ```
6. **Stop DHCP server**
   If you set "manage_dhcp" to 1, DHCP server is started from Cobbler server. Stop this DHCP server started on the management server.

   ```
   # systemctl stop dhcpd.service
   ```

Chapter 5
Setting up of the SX-Aurora TSUBASA environment

Use Ansible to install software programs necessary for the SX-Aurora TSUBASA system and to set up VH environments.

Set up the SX-Aurora TSUBASA environment in the following sequence.

1. Preparing for using Ansible
   a. Tuning Ansible performance

      Modify Ansible settings to reduce time for Ansible to run playbooks.
   b. Starting ssh-agent

      Register an SSH private key for remotely logging in VHs without entering a passphrase.
   c. Registering VHs

      Register VHs to be managed by Ansible.
   d. Checking VH connectivity

      Check that VHs are operable from the management server.
2. Installing SX-Aurora TSUBASA Software programs
   a. Editing playbooks

      Select software programs to be installed on VHs and register values specified in software environment configurations.
   b. Running playbooks

      Install and set up software programs on VHs.
3. Restarting VHs

   Restart VHs and start the SX-Aurora TSUBASA service.

Perform the operations described in this chapter with an administrative user on the management server.

5.1. Preparing for using Ansible

This section mainly explains the files Ansible uses and VH registration.

5.1.1. Files for Ansible

The following files are used in the Ansible's operation.

Inventory file	Text file with which managed machines are registered.
Playbook files	Text files describing how to configure managed machines in the YAML format.

Ansible guarantees idempotence by running playbooks that describe desired state of managed machines. Idempotence is the concept that "the same result is obtained even if an operation is performed multiple times", which is achieved by the configuration management tool. All steps written in playbooks are performed against new machines while only differential steps are performed against running machines.

The playbooks described in this document are provided as sample files, which are installed under [/opt/nec/sysmng-soft/etc/ansible] by the procedures in " *3.6. Installing the template package* on page 20 ". The following table shows directory layout and roles of sample playbooks.

Table 3: Structure of Sample Playbooks

File Names	Descriptions
vh.yml	Master playbook configuring VHs; this playbook calls each role
group_vars/all	File defining common variables used across all playbooks
roles/TSUBASA-local-repo/<roles-dir>	Role configuring the settings for Ansible to reference the Yum repository for the SX-Aurora TSUBASA system; [/etc/yum.repos.d/TSUBASA-local.repo] and the GPG public key for the SX-Aurora TSUBASA software programs are configured with this role
roles/aurlic-lib/<roles-dir>	Role installing and configuring License access library.
roles/veos-apprun/<roles-dir>	Role installing VEOS Application Runtime.
roles/veos-appdev/<roles-dir>	Role installing VEOS Application Development.
roles/TSUBASA-InfiniBand/<roles-dir>	Role installing SX-Aurora TSUBASA InfiniBand.
roles/mmm/<roles-dir>	Role installing MMM.
roles/ScaTeFS-Client/<roles-dir>	Role installing ScaTeFS Client.
roles/nec-mpi/<roles-dir>	Role installing NEC MPI.
roles/tuningtool/<roles-dir>	Role installing tuning tool.
roles/parallel-debugger/<roles-dir>	Role installing NEC Parallel Debugger.
roles/NQSV-JobServer/<roles-dir>	Role installing NQSV/JobServer.
roles/NQSV-Client/<roles-dir>	Role installing NQSV/Client.
roles/NumericLibraryCollection/<roles-dir>	Role installing Numeric Library Collection.
roles/binutils/<roles-dir>	Role installing binutils.

Caution

In case of the setting which Ansible was used, VMC Firmware package is installed when updating VMC Firmware.

Playbooks contain steps and variables related to software programs. Put playbooks in <roles-dir> described in the above table. Playbook directory layout described in the above table comply with the ones recommended by the Ansible official document. Main layout is as follows. Some files may not exist according to software program configurations.

Table 4: Structure of <roles-dir>

Files	Description
tasks/main.yml	File defining configuration steps
meta/main.yml	File defining dependent roles

Files	Description
handler/main.yml	File defining steps that are performed according to state changes
vars/main.yml	Variable definitions
templates/*	Template files written in Jinja2 style.

5.1.2. Tuning Ansible performance

When Ansible manages a large number of VHs, running playbooks takes time. Run the Ansible performance tuning to reduce playbook run time. This section describes typical methods of performance tuning. For details, see the Ansible documents.

Ansible connects to VHs over SSH and runs tasks in parallel; changing configurations of SSH connections and parallelism level helps reduce time for Ansible to run playbooks.

Edit the following items to the specified values in the Ansible configuration file, `[ /etc/ansible/ansible.cfg ]`,for performance tuning.

Table 5: Settings of ansible.cfg

Values	Description	Default	etc.
[defaults]			
forks = 100	Number of parallel tasks	5	Adjust the value according to the management server performance and the number of VHs.
timeout = 180	Timeout of SSH (in sec.)	10	
gathering = explicit	Gathering information of managed hosts implicit: explicit: disabled smart: gathering information only when the cache is disabled	implicit	
gather_subset = facter	Type of information items to gather from target hosts	all	
strategy = free	Task running methods linear: Ansible waits for all hosts to complete running tasks. free: Ansible doesn't wait.	linear	
[ssh_connection]			
pipelining = True	Using the pipeline function	False	VHs' requiretty (/etc/sudoers) needs to be disabled.

Ansible uses the SSH control master function by default as defined by the default ssh_args (SSH command arguments) values in the [ssh_connection] section. The specified values are as follows.

```
ssh_args = -C -o ControlMaster=auto -o ControlPersist=60s
```

Caution
To specify SSH command arguments for configuring Ansible, add arguments after the ones specified in the above command line. If the ControlMaster and ControlPersist are not defined, running playbooks takes time.

You need to disable the requiretty(/etc/sudoers) of VH administrative users to use the pipeline function. When administrative users are added to VHs, this setting is disabled by default. For details, refer to " *4.3.2.4. Script section* on page 29 " .

5.1.3. Starting ssh-agent

Ansible used for setting the VH environment first logs in VHs over SSH for further operations. Start the `ssh-agent` and run the `ssh-add` command to register a SSH private key before running the Ansible command. Private key registration enables remote login without entering a passphrase. The private key to be used is created in " *3.5. Setting administrative user's SSH public and private keys* on page 19 ".

```
$ ssh-agent bash
$ ssh-add  ~/.ssh/id_rsa
Enter passphrase for /home/admin/.ssh/id_rsa: (Input passphrase)
Identity added: /home/admin/.ssh/id_rsa (/home/admin/.ssh/id_rsa)
```

5.1.4. Registering VHs

Ansible manages machines registered with the inventory file. The inventory file is a text file where you can specify target machines with host names or IP addresses and you can group machines as necessary.

Use vh-hosts, the Inventory file created in " *3.8. Creating host list file* on page 21 ", and change the Ansible's default inventory file to vh-hosts. Edit inventory on the Ansible configuration file `[ /etc/ansible/ansible.cfg ]` as follows.

```
inventory = /opt/nec/sysmng-soft/etc/ansible/vh-hosts
```

5.1.5. Checking VH connectivity

Use the `ping` module of the `ansible` command to check that the OS installation on a target VH has been completed, the VH has started, and remotely logging in the VH is possible. Check that nothing appears after running the following command.

```
$ ansible all -m ping -o | grep UNREACHABLE
```

When the `ping` module fails, the following message will be displayed.

```
vh-name01 | UNREACHABLE!
```

To see details of the error message, run the `ping` module without specifying -o against the host that failed to respond to the ping.

An example when a target host is not started:

```
$ ansible vh-name01 -m ping
vh-name01 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh.",
    "unreachable": true
}
```

An example when SSH authentication is not configured:

```
$ ansible vh-name01 -m ping
vh-name01 | UNREACHABLE! => {
    "changed": false,
    "msg": "ERROR! SSH encountered an unknown error during the
 connection. We recommend you re-run the command using -vvvv, which
 will enable SSH debugging output to help diagnose the issue",
    "unreachable": true
}
```

5.1.6. Starting Apache HTTP Server

SX-Aurora TSUBASA system software packages are downloaded from the HTTP server on the management server to each VHs in " *5.2. Installing SX-Aurora TSUBASA system software* on page 37 " . So start HTTP server on the management server.

```
# systemctl start httpd.service
```

5.2. Installing SX-Aurora TSUBASA system software

Use sample playbooks under `[ /opt/nec/sysmng-soft/etc/ansible ]` to install software programs necessary for the SX-Aurora TSUBASA system and to set up their environments.

For installing SX-Aurora TSUBASA system software, it is necessary to prepare the Yum repository to be able to install or update packages for the operating system by yum.

Note that this installation using Ansible does not cover C/C++ compiler and Fortran compiler. For installing these software, please refer to the guides of the software.

5.2.1. Editing Playbook

This section explains the sample playbooks that need to be updated according to users' environments, and explains how to install same software programs on all VHs. To select software programs according to VH roles, see " *5.2.3. Specifying software according to roles of VHs.* on page 39 ".

1. Master Playbook

   Specify roles of software programs to be installed in roles:. Delete the roles of software programs that do not need to be installed.

   `[ vh.yml ]`

   ```
   ---
   - name: Set VH machine
     hosts: all
     become: yes
     roles:
       - veos-apprun
       - veos-appdev
       - TSUBASA-InfiniBand
       - mmm
       - ScaTeFS-Client
       - tuningtool
       - parallel-debugger
       - NQSV-JobServer
       - NQSV-Client
       - NumericLibraryCollection
       - binutils
       - nec-mpi
   ```

   If connections via a proxy server are required to obtain packages on VH machines, add the following setting to configure the proxy server.

   ```
     environment:
       http_proxy: http://proxy.example.com:8080
   ```

2. Common variables used across all playbooks

   Specify the Yum repository directory for the SX-Aurora TSUBASA system created on the HTTP server (the management server) according to the procedures in " *3.1.5. Installing createrepo* on page 17 ". The Yum repository directory is the path from the DocumentRoot directory of the HTTP server to the directory where repodata is located.

`[ group_vars/all ]`

```
---
httpserver: http-server-ipaddr
reposdir: repos/TSUBASA
```

3. Role of each software

Set the following variables in `[ vars/main.yml ]` files of each software roles. Please refer to the software's document for the details of the variables.

a. aurlic-lib

Set the following parameters to access the license server when you install ScaTeFS Client or NEC MPI on the VHs.

`[ roles/aurlic-lib/vars/main.yml ]`

Variables	Values
aurlic_serverhost	Specify the license server's host name.
aurlic_serverport	Specify the port number that the license server is using to wait license request.

b. veos-apprun
No variables to bet set.

c. veos-appdev
No variables to bet set.

d. TSUBASA-InfiniBand
`[ roles/TSUBASA-InfiniBand/vars/main.yml ]`

Variables	Values
tsubasa_ib_ofedpkgpath	Specify the path name where the Mellanox OFED ISO image file is extracted. Specify a path from the DocumentRoot directory of the HTTP server on the management server.

e. mmm
No variables to bet set.

f. ScaTeFS-Client
No variables to bet set.

g. tuningtool
No variables to bet set.

h. parallel-debugger
No variables to bet set.

i. NQSV/JobServer
No variables to bet set.

j. NQSV/Client
No variables to bet set.

k. NumericLibraryCollection
`[ roles/NumericLibraryCollection/vars/main.yml ]`

Variables	Values
nlc_version	The version of the group ID for NumericLibraryCollection package. The group ID means the 'X-X-X' part of the NumericLibraryCollection package nec-nlc-X-X-X.

l. binutils
No variables to bet set.

m. nec-mpi

`[ roles/nec-mpi/vars/main.yml ]`

Variables	Values
mpi_inst_vers	The version of the group ID for NEC MPI library group package. The group ID means the 'X-Y-Z' part of the NEC MPI package nec-mpi-X-Y-Z.

5.2.2. Running playbooks

Specify target VHs in the `ansible-playbook` command and run the playbook to install same SX-Aurora TSUBASA software programs on target VHs and configure their environments. If you would like to install different software by the roles of VHs, see " *5.2.3. Specifying software according to roles of VHs.* on page 39 ".

- To install and configure software programs on all VHs

```
$ cd /opt/nec/sysmng-soft/etc/ansible
$ ansible-playbook vh.yml
```

- To install and configure software programs on specified VHs

```
$ ansible-playbook vh.yml -l vh-name01 vh-name02
$ ansible-playbook vh.yml -l 'vh-name0*'
$ ansible-playbook vh.yml -l group-name
```

When playbook execution fails, an error message like the one below appears and host names of failed VHs are recorded in the `[ playbookname.retry ]` file.

```
TASK [veos-apprun : Install VEOS Application Runtime] ***
fatal: [vh-name01]: FAILED! => { ... }
 :
  to retry, use: --limit @/opt/nec/sysmng-soft/etc/ansible/vh.retry
```

To run the command again only against failed VHs, run the following command.

```
$ ansible-playbook vh.yml \
   --limit @/opt/nec/sysmng-soft/etc/ansible/vh.retry
```

5.2.3. Specifying software according to roles of VHs.

While " *5.2.1. Editing Playbook* on page 37 " and " *5.2.2. Running playbooks* on page 39 " explain how to install same software programs in all VHs, this section explains how to specify software programs to be installed according to roles of VHs. The following is an example case where software programs are installed according to the configuration below.

Table 6: Example of software for VH groups

hostname of VHs	VH Group	Software to be installed
vh-name000 vh-name001 vh-name002	group0	VEOS Application Runtime mmm
vh-name100 vh-name101	group1	VEOS Application Runtime VEOS Application Development mmm

1. Create an inventory file with group settings

Create groups on a VH role basis and register them in a VH information file. Follow the procedures in " *3.8. Creating host list file* on page 21 " or " *7.2. create-hostlist.py* on page 46 " to create an inventory file, and follow the procedures in " *5.1.4. Registering VHs* on page 36 " to set the created file as the default inventory file.

2. Create playbooks for groups

   Create playbooks for each group, [group0.yml] and [group1.yml], and put them in the top directory (where a sample playbook, vh.yml, resides). Set hosts to a group name and set roles to roles according to software programs to be installed. The roles described below are created in " *5.2.1. Editing Playbook* on page 37 ".

   [group0.yml]

```
---
- name: Set VH machine of group0
  hosts: group0
  become: yes
  roles:
    - veos-apprun
    - mmm
```

   [group1.yml]

```
---
- name: Set VH machine of group1
  hosts: group1
  become: yes
  roles:
    - veos-apprun
    - veos-appdev
    - mmm
```

3. Updating master playbook

   To setup the entire SX-Aurora TSUBASA system with group0 and group1, modify the sample master playbook [vh.yml] as below.

   [vh.yml]

```
- include: group0.yml
- include: group1.yml
```

4. Execution of master playbook

   Run the master playbook [vh.yml] by using the ansible-playbook command to install only the software programs intended for the VHs registered in group0 and group1.

```
$ ansible-playbook vh.yml
```

5.3. Reboot VHs

Use the ansible command to reboot all VHs.

```
$ ansible all  --become -a "/sbin/reboot"
```

So the services for SX-Aurora TSUBASA system will be started.

The ansible command execution will output the following error message. However the message can be ignored.

```
vh-name01 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: Shared connection to vh-name01 closed.\r
\n",
    "unreachable": true
}
```

5.4. Check the settings of VHs

After rebooting the VHs, check that VE cards are up and in ONLINE state. To check the VE state, execute `vecmd state get` on all VHs using `ansible` as follows.

```
$ ansible all --become -a "/opt/nec/ve/bin/vecmd state get"
vh-name01 | SUCCESS | rc=0 >>
Vector Engine MMM-Command v0.5.13
Command:
state -N 0,1,4,5 get
--------------------------------------------------------------------------------
VE0 [04:00.0] [ ONLINE             ] Last Modif:2017/12/19 14:19:49
VE1 [06:00.0] [ ONLINE             ] Last Modif:2017/12/19 14:19:51
VE4 [0c:00.0] [ ONLINE             ] Last Modif:2017/12/19 14:19:53
VE5 [0e:00.0] [ ONLINE             ] Last Modif:2017/12/19 14:19:55
--------------------------------------------------------------------------------
Result: Success

  :
```

Chapter 6
Power Control

You can control VH power (on and off) by:

- Powering on and off VHs manually
- Powering on and off VHs from the management server with the Baseboard Management Controller (BMC).
- Powering on and off VHs from the management server with Wake-On-LAN(WOL)

When manually powering on and off VHs, skip this chapter.

When powering on and off VHs from the management server, control power with `powerctrl` and `powerctrl_w`, the tools provided in the template package, TSUBASA-sysmngsoft-X.X-Y.noarch.rpm. The `powerctrl` uses the BMC to control power while the `powerctrl_w` uses the WOL. Only the rootuser can perform these power control tools. Use the `powerctrl` or the `powerctrl_w` according to VHs you use.

6.1. Controlling power with BMC

This section explains how to use the Baseboard Management Controller (BMC) to set up environments for controlling VH power from the management server and to power on and off VHs.

The after explanation premises that the BMC operation user is initialized.

6.1.1. Configuration for controlling power with BMC

6.1.1.1. Installing IPMITool on the management server

Install IPMITool on the management server.

Example:

```
$ su
# yum -y install ipmitool
```

6.1.1.2. Configuration for Power control tool `powerctrl`

`powerctrl` refers the host list file which includes the list of target VH hosts, `[ /opt/nec/sysmng-soft/etc/vh_host.conf ]`. This file has been already created in " *3.8. Creating host list file* on page 21 ".

6.1.2. Powering on/off VHs with BMC

Powering on VHs

Run the `on` command to power on the hosts listed in the host list file `[ /opt/nec/sysmng-soft/etc/vh_host.conf ]`.

```
# /opt/nec/sysmng-soft/bin/powerctrl on host001
Total host count is 1.
Do you want to execute "powerctrl on host001"? [y/n] y
BMC user name: ADMIN
BMC password: ************
send power control command to host001
```

You can specify multiple hosts to be powered on. When sequential numbers are assigned at the end of hostnames, you can use a range of sequential numbers to specify hostnames to control their powers. See " *7.3. powerctrl* on page 47 " for details.

>Powering off VHs

In general, use the configuration management function to shut down and power off VHs. When you need to forcibly power off VHs for any reasons, you can do so from the management server.

Run the `off` command to power off the host machines registered in the host list file, `[ /opt/nec/sysmng-soft/etc/vh_host.conf ]`.

```
# /opt/nec/sysmng-soft/bin/powerctrl off host001
Total host count is 1.
Do you want to execute "powerctrl off host001"? [y/n] y
BMC user name: ADMIN
BMC password: ************
send power control command to host001
```

As with powering on, you can specify multiple hosts to be powered off.

6.2. Powering on/off VHs with WOL

This section explains the configuration of the power control function with Wake-On-LAN (WOL) and how to use the tool to control VH power from the management server.

6.2.1. Configuration of controlling power with WOL

6.2.1.1. Check the network card

Check that VH machines' network cards which connect the machines to the management server support Wake-On-LAN (WOL). If network cards do not support the WOL, the functions described in this section are not available.

6.2.1.2. Configuration for the power control tool `powerctrl_w`

`powerctrl_w` refers the host list file which contains the target VH hosts, `[ /opt/nec/sysmng-soft/etc/vh_host_w.conf ]`. This file has been already created in " *3.8. Creating host list file* on page 21 ".

6.2.2. Powering on/off VHs with WOL

Powering on VHs

`powerctrl_w` tool can power on the hosts listed in the host list file `[ /opt/nec/sysmng-soft/etc/vh_host_w.conf ]`.

```
# /opt/nec/sysmng-soft/bin/powerctrl_w host001
Total host count is 1.
Do you want to execute "powerctrl_w host001"? [y/n] y
send power control command to host001
```

You can specify multiple hosts to be powered on. When sequential numbers are assigned at the end of hostnames, you can use a range of sequential numbers to specify hostnames to control their powers. See " *7.4. powerctrl_w* on page 49 " for details.

Powering off VHs

`powerctrl_w` cannot power off VHs. Use the configuration management function to shut down and power off VHs.

Chapter 7
Reference manual

7.1. cobimport.sh

NAME

cobimport.sh -- Cobbler's tool for registering system records at once

SYNOPSIS

cobimport.sh --hostlist=*HPATH* --profile=*PROFILE* --interface=*INTERFACE*
[--static=1] ...

DESCRIPTION

The `cobimport.sh` registers Cobbler system records at once; system records are host information listed in the host list file specified in HPATH of the --hostlist option.

Run the `cobbler` command for system record registration. Options except the --hostlist option are directly passed to the `cobbler` command. However, the options (--name, --hostname, --mac, --ip-address, --subnet, --gateway) that specify information defined in the host list file are ignored.

Specify --static=1 if IP addresses of VHs are static. Omit the -static option when IP addresses are assigned by the DHCP server. For details of other options, see the `cobbler` command reference. Make sure to specify the `cobimport.sh` options that are used across hosts and are necessary for the system record registration.

Use the `create-hostlist.py` tool to create the host list file to be specified in the --hostlist option in advance.

The host list file is a text file where the following items are listed in each row; a hostname, MAC/IP/netmask/ address of a network interface, and gateway address used for the network connection.

Example:

```
vh-host01 00:1D:7D:7F:D7:01 192.168.1.1 255.255.255.0 192.168.1.100
vh-host02 00:1D:7D:7F:D7:02 192.168.1.2 255.255.255.0 192.168.1.100
vh-host03 00:1D:7D:7F:D7:03 192.168.1.3 255.255.255.0 192.168.1.100
vh-host04 00:1D:7D:7F:D7:04 192.168.1.4 255.255.255.0 192.168.1.100
```

RETURN VALUE

Zero is returned for successful system record registration and one is returned for failed registration.

FILES

/opt/nec/sysmng-soft/etc/host-list.xlsx	VH information file

SEE ALSO

" *create-hostlist.py* "

7.2. create-hostlist.py

NAME

create-hostlist.py -- Tool for converting VH information files at once

SYNOPSIS

create-hostlist.py --type=*TYPE* [--infile=*INPUTFILE*] [--outfile=*OUTPUTFILE*]

DESCRIPTION

`create-hostlist.py` creates host list files using the VH information file `[ /opt/nec/sysmng-soft/etc/host-list.xlsx ]`. Specify types of host list files to be created in *TYPE* of the --type option. The types of host list files you can specify in *TYPE* are:

ansible	The tool creates the host list file in the inventory file format defining hosts to be managed when Ansible is used to manage the VH configuration.
cobbler	The tool creates the host list file for the cobimport.sh tool, which registers system records at once when Cobbler is used to install the OS on VHs.
power_bmc	The tool creates the host list file defining those controlled by the powerctrl tool when BMC is used to control power.
power_wol	The tool creates the host list file defining those controlled by the powerctrl_w tool when WOL is used to control power.

A created file is displayed on the console window as standard output by default. If the --outfile option is specified, a file is created as the file specified in *OUTPUTFILE*. If the specified file already exists, file creation fails.

Similarly, a VH information file that is used as an input source can be changed to the file specified in *INPUTFILE* of the --infile option.

VH information file

The following data is included in the VH information file.

hostname	Host names of VHs
host-group	Group names when VHs are grouping
IP-address	IP addresses of VHs
IP-address(BMC)	BMC IP address of VHs
MAC-address	MAC address (AA:BB:CC:DD:EE:FF style) of VHs
netmask	Netmask of the VH IP address
gateway	Gateway's address

The following data are used for each *TYPE*.

	hostname	host-group	IP-address	IP-address (BMC)	MAC-address	netmask	gateway
ansible	*	*					
cobbler	*		*		*	*	*
power_bmc	*			*			

power_wol	*				*		

The VH information file is provided as a sample file. Do not change the file type of the sample file, and leave unused cells blank.

RETURN VALUE

Zero is returned for a successful host list file creation and any number other than zero is returned for a failed creation.

FILES

/opt/nec/sysmng-soft/etc/host-list.xlsx VH information file

NOTE

`create-hostlist.py` uses Python. Make sure that Python 2.7 or higher version is installed. And python-excel library is required. Also make sure xlrd package is installed.

SEE ALSO

" *cobimport.sh* " , " *powerctrl* ", " *powerctrl_w* ", " *setup-hostlist* "

7.3. powerctrl

NAME

powerctrl -- Power control tool for VH (with BMC)

SYNOPSIS

powerctrl [*options*] *command target*

powerctrl { --all | -a } *command*

powerctrl { --help | -h }

DESCRIPTION

The `powerctrl` uses the ipmitool commands to control power of the hosts specified in *target*. It performs power control operations such as booting, shutting down, and rebooting hosts according to a value specified in *command*. Only the rootuser can run this tool.

Behavior of the `powerctrl` command and information displayed after the execution comply with the `chassis power` sub-commands of the `ipmitool`.

OPTIONS

--user=*user*

-u *user*

Specify the BMC management username.

--password=*password*

-p *password*

Specify the password of the BMC management user.

When the -user (or -u) or -password (or -p) options are omitted, the interactive mode prompts you for your next input. Enter the username and password of the BMC management user in the interactive mode.

```
BMC user name:
BMC password:
```

--yes

-y

The `powerctrl` displays the entered command and the number of target hosts before running the command, and asks users whether or not to run the command. When this option is specified, "y" is automatically entered.

When you omit this option, if there is no problem, enter "y". Entering "n" cancel running the command.

--all

-a

A power control operation specified in *command* runs on all the hosts listed in the host information file `[ vh_host.conf ]`. When this option is specified, you do not need to specify *target*.

--toolpath=*ipmitool_path*

-t *ipmitool_path*

Specify the absolute path of the `ipmitool` commands. If it is not specified, `[ /usr/bin/ipmitool ]` is used.

--help

-h

Help message is displayed.

--file=*log_path*

-f *log_path*

Specify the file where execution results are written. When this option is omitted, results are displayed on the console window as standard output.

The `powerctrl` displays target hostnames and operation results as execution results.

command

status	Displays power status of target hosts.
on	Powers on target hosts.
off	Powers off target hosts.
reset	Cold reboot is performed on target hosts.

Specifying host(*target*)

Specify the names of the hosts targeted for power control. The hostnames specified here need to be registered in the host information file `[ vh_host.conf ]`.

When hostnames end with an integer greater than zero, you can enter a range of integers to specify hosts to be operated.

See below to specify hostnames.

(1)	Specifying a single host	host0
(2)	Specifying multiple hosts	host0 host1
(3)	Specifying a range of hosts	host[0-100]
(4)	Combining (2) and (3)	host[0-100] host200 host300

Example

```
# /opt/nec/sysmng-soft/bin/powerctrl on host00[1-8]
Total host count is 8.
```

```
Do you want to execute "powerctrl on host00[1-8]" ? [y/n] y
BMC user name: ADMIN
BMC password: ************
send power control command to host001
send power control command to host002
send power control command to host003
send power control command to host004
send power control command to host005
send power control command to host006
send power control command to host007
send power control command to host008
```

NOTE

Use the same BMC administrative user and password on all hosts.

FILES

`[ /opt/nec/sysmng-soft/etc/vh_host.conf ]` Host information file that defines the relation between host name and BMC IP address of VHs.

SEE ALSO

ipmitool, " *powerctrl_w* ", " *vh_host.conf* "

7.4. powerctrl_w

NAME

powerctrl_w -- Power control tool for VH (with WOL)

SYNOPSIS

powerctrl_w [*options*] *target*

powerctrl_w { --all | -a }

powerctrl_w { --help | -h }

DESCRIPTION

The `powerctrl_w` uses the `ether-wake` commands to control power (power on) of the hosts specified in *target*. Only the rootuser can run this tool.

Behavior of the `powerctrl_w` command and information displayed after the execution comply with the `ether-wake` commands.

OPTIONS

--yes

-y

The `powerctrl_w` displays the entered command and the number of target hosts before running the command, and asks users whether or not to run the command. When this option is specified, "y" is automatically entered.

When you omit this option, if there is no problem, enter "y". Entering "n" cancel running the command.

--all

-a

Power-on is performed on all the hosts listed in the host information file `[ vh_host_w.conf ]`. When this option is specified, you do not need to specify *target*.

--help

-h

Help message is displayed.

--file=*log_path*

-f *log_path*

Specify the file where execution results are written. When this option is omitted, results are displayed on the console window as standard output.

The `powerctrl_w` displays target hostnames and operation results as execution results.

Specifying host(*target*)

Specify the names of the hosts targeted for power control. The hostnames specified here need to be registered in the host information file `[ vh_host_w.conf ]`.

When hostnames end with an integer greater than zero, you can enter a range of integers to specify hosts to be operated.

See below to specify hostnames.

(1)	Specifying a single host	host0
(2)	Specifying multiple hosts	host0 host1
(3)	Specifying a range of hosts	host[0-100]
(4)	Combining (2) and (3)	host[0-100] host200 host300

Example

```
# /opt/nec/sysmng-soft/bin/powerctrl_w host00[1-2]
Total host count is 2.
Do you want to execute "powerctrl_w host00[1-2]" ? [y/n] y
send power control command to host001
send power control command to host002
```

FILES

`[ /opt/nec/sysmng-soft/etc/vh_host_w.conf ]` Host information file that defines the relation between host name and MAC address of VHs.

SEE ALSO

" *powerctrl* ", " *vh_host_w.conf* "

7.5. setup-hostlist

NAME

setup-hostlist -- Tool for converting VH information files at once

SYNOPSIS

setup-hostlist

DESCRIPTION

`setup-hostlist` executes `create-hostlist.py` to creates all host list files using the VH information file `[ /opt/nec/sysmng-soft/etc/host-list.xlsx ].`

File types and paths of created files are as follows:

`[ /opt/nec/sysmng-soft/etc/ansible/vh-hosts ]`

Host list file in the inventory file format defining hosts to be managed when Ansible is used to manage the VH configuration

`[ /opt/nec/sysmng-soft/etc/cobimport.txt ]`

Host list file for the cobimport.sh tool, which registers system records at once when Cobbler is used to install the OS on VHs

`[ /opt/nec/sysmng-soft/etc/vh_host.conf ]`

Host list file defining those controlled by the powerctrl tool when BMC is used to control power

`[ /opt/nec/sysmng-soft/etc/vh_host_w.conf ]`

Host list file defining those controlled by the powerctrl_w tool when WOL is used to control power

For the information of data to register in VH information file, see " *7.2. create-hostlist.py* on page 46 ".

RETURN VALUE

Zero is returned for a successful host list file creation and one is returned for a failed creation.

FILES

/opt/nec/sysmng-soft/etc/host-list.xlsx	VH information file

SEE ALSO

" *create-hostlist.py* ", " *cobimport.sh* " , " *powerctrl* ", " *powerctrl_w* "

7.6. vh_host.conf

NAME

vh_host.conf -- Host information file for powerctrl tool

SYNOPSIS

`[ /opt/nec/sysmng-soft/etc/vh_host.conf ]`

DESCRIPTION

`[ vh_host.conf ]` is the file where VHs controlled by the power control tool, powerctrl, using BMC are registered. This file is created by `create-hostlist.py` tool.

Register one host in one line in the following format.

```
    Hostname  BMC IP address
```

Separate entry fields with space(s) or tab(s) and write VH hostnames in the first field and BMC IP addresses of VHs in the second field.

(Example)

```
host001  192.168.0.1
host002  192.168.0.2
host003  192.168.0.3
host004  192.168.0.4
```

In the above format, strings between the character "#" and the end of a row are treated as comments and are ignored.

FILES

`[ /opt/nec/sysmng-soft/etc/vh_host.conf ]`	Host information file where VH hostnames and their BMC IP addresses are listed

SEE ALSO

" *create-hostlist.py* ", " *powerctrl* "

7.7. vh_host_w.conf

NAME

vh_host_w.conf -- Host information file for powerctrl_w tool

SYNOPSIS

`[ /opt/nec/sysmng-soft/etc/vh_host_w.conf ]`

DESCRIPTION

`[ vh_host_w.conf ]` is the file where VHs controlled by the power control tool `powerctrl_w` using WOL are registered. This file is created by `create-hostlist.py` tool.

Register one host in one line in the following format.

```
    Hostname         MAC address
```

Separate entry fields with space(s) or tab(s) and write VH hostnames in the first field and MAC addresses of VHs' network card in the second field.

(Example)

```
host001  00:13:D3:4B:83:72
host002  00:13:D3:4B:83:96
host003  00:13:D3:4B:84:94
host004  00:13:D3:4B:84:62
```

FILES

`[ /opt/nec/sysmng-soft/etc/vh_host_w.conf ]`	Host information file where VH hostnames and their MAC addresses are listed

SEE ALSO

" *create-hostlist.py* ", " *powerctrl_w* "

Appendix A Trouble Shooting

A.1. OS Installation on VHs

A.1.1. A cobbler command failed.

Problem

The following error occurs when running a `cobbler` command.

```
cobblerd does not appear to be running/accessible: error(111, 'Connection
 refused')
```

Cause

The Cobbler server may not be started.

Solution

Check that the Cobbler server is started. If not, start the Cobbler server.

```
# systemctl status cobblerd.service
# systemctl start cobblerd.service
```

Related information

" *4.2.3. Starting the Cobbler server* on page 25 "

A.1.2. The cobbler command fails to register system records.

Problem

The following error occurs when running the `cobbler system` command or the `cobimport.sh` command.

```
exception on server: 'invalid profile name: xxxxx'
```

Cause

A wrong profile may be specified.

Solution

A profile name is defined in the cobbler import --name=*profilename* in profile registration. Specify this profile name in the --profile option of the failed command and run the command again.

Related information

" *4.3.3. Importing ISO image* on page 30 "

A.1.3. IP addresses are not automatically assigned after OS installation starts.

Problem

The following message appears on the console window after starting a VH machine for the OS installation and the installation process waits for the DHCP server to respond.

```
CLIENT MAC ADDR: xx xx
DHCP..
```

Cause

Possible causes are as follows:

- The DHCP server is not started.

- [/etc/cobbler/dhcp.template] is not correctly configured when the Cobbler server machine and a VH machine are located on the network of the same subnet mask.
- [/etc/cobbler/dhcp.template] is not correctly configured when the Cobbler server machine and a VH machine are located on networks of different subnet masks.

Solution

- Run the `cobbler sync` command and start the DHCP server.
- See " *4.2.2.1. Basic settings of dhcp.template* on page 24 " and check that the setting is correct.
- See " *4.2.2.3. The management server and VH machines on different subnets* on page 25 " and check that the setting is correct.

If you modify the `[ /etc/cobbler/dhcp.template ]`, run the `cobbler sync` command and reboot the DHCP server.

Related information

" *4.4. Starting OS installation* on page 31 "

" *4.2.2.1. Basic settings of dhcp.template* on page 24 "

" *4.2.2.3. The management server and VH machines on different subnets* on page 25 "

A.1.4. OS is not installed as intended.

Problem

The OS is not installed as configured in the kickstart file or kickstart script file.

Cause

After the kickstart file or kickstart script file were edited, modified settings may not be applied in the Cobbler server.

Solution

Run the `cobbler sync` command to apply settings in the Cobbler server.

Related information

" *4.4. Starting OS installation* on page 31 "

A.2. Setting the SX-Aurora TSUBASA environment

A.2.1. Entry of the SSH private key's password is required.

Problem

Entry of the SSH private key's password is required for each target machine when running `ansible-playbook` or `ansible` command.

```
$ ansible-playbook vh.yml

PLAY [Install SX-Aurora TSUBASA Software on VH machine]
 ***********************
Enter passphrase for key '/home/admin/.ssh/id_rsa':
Enter passphrase for key '/home/admin/.ssh/id_rsa':
Enter passphrase for key '/home/admin/.ssh/id_rsa':
 :
```

```
$ ansible all -m ping
Enter passphrase for key '/home/admin/.ssh/id_rsa':
Enter passphrase for key '/home/admin/.ssh/id_rsa':
Enter passphrase for key '/home/admin/.ssh/id_rsa':
 :
```

Cause

The `ssh-agent` is not started or the private key is not registered by the `ssh-add` command after starting the `ssh-agent`.

Solution

Start the `ssh-agent` and use the `ssh-add` to register the private key on the terminal machine where you run `ansible-playbook` or the `ansible` command.

```
$ ssh-agent bash
$ ssh-add  ~/.ssh/id_rsa
Enter passphrase for /home/admin/.ssh/id_rsa: (Input
 passphrase)
Identity added: /home/admin/.ssh/id_rsa (/home/admin/.ssh/
id_rsa)
```

Related information

" *5.1.3. Starting ssh-agent* on page 36 "

A.2.2. Playbook execution fails with "sudo: sorry, you must have a tty to run sudo"

Problem

The error message "sudo: sorry, you must have a tty to run sudo" appears when running the `ansible-playbook` or the `ansible` command, and playbook execution fails.

```
$ ansible-playbook vh.yml

PLAY [Install SX-Aurora TSUBASA Software on VH machine]
 ******************************
TASK [TSUBASA-local-repo : Install SX-Aurora TSUBASA Software GPG-KEY]
 ****************
fatal: [vh-name01]: FAILED! => {"changed": false, "failed": true,
  "module_stderr": "sudo: sorry, you must have a tty to run sudo\n",
  "module_stdout": "", "msg": "MODULE FAILURE", "rc": 1}
        to retry, use: --limit @/tmp/vh.retry
```

```
$ ansible all --sudo -a 'command'
vh-name01 | FAILED! => {
    "changed": false,
    "failed": true,
    "module_stderr": "sudo: sorry, you must have a tty to run sudo\n",
    "module_stdout": "",
    "msg": "MODULE FAILURE",
    "rc": 1
}
```

```
$ ansible all --sudo -a 'command'
vh-name01 | FAILED! => {
    "changed": false,
    "failed": true,
    "module_stderr": "sudo: sorry, you must have a tty to run sudo\n",
    "module_stdout": "",
    "msg": "MODULE FAILURE",
    "rc": 1
}
```

Cause

requiretty in /etc/sudoers is enabled on the VH.

Solution

Add the following setting to /etc/sudoers on the VH to disable requiretty.

```
# visudo
 :
Defaults:admin !requiretty
```

Related information

" *5.1.2. Tuning Ansible performance* on page 35 "

" *4.3.2.4. Script section* on page 29 "

A.2.3. Package download fails with "urlopen error [Errno 113] No route to host"

Problem

When performing tasks for software installation or update, the error message "urlopen error [Errno 113] No route to host" appears, and the playbook execution fails.

```
$ ansible-playbook vh-set.yml
 :
TASK [zabbix-agent : Install repository for zabbix-agent]
 ********************
fatal: [vh-name01]: FAILED! => {"changed": false, "failed": true,
  "msg": "Failure downloading http://xxx/xxx.rpm,
  Request failed: <urlopen error [Errno 113] No route to host>"}
             to retry, use: --limit @/tmp/vh.retry
```

Cause

A proxy server is not set up.

Solution

Check the proxy server to be used and add the following setting to vh.yml.

```
  environment:
    http_proxy: http://proxy.example.com:8080
```

Related information

" *5.2.1. Editing Playbook* on page 37 "

A.2.4. Yum repository update fails with "[Errno 14] HTTP Error 502 - Bad Gateway"

Problem

When executing the "Update metadata for yum" task in the TSUBASA-local-repo role to set the SX-Aurora TSUBASA Yum repository access configuration, playbook fails with "http://*x.x.x.x*/repos/TSUBASA/repodata/repomd.xml:[Errno 14] HTTP Error 502 - Bad Gateway".

```
$ ansible-playbook vh.yml
 :
TASK [TSUBASA-local-repo : Update metadata for yum]
 *************************
 :
fatal: [vh-name01]: FAILED! => {"changed": true, "cmd": "yum", "--
setopt=http_caching=packages", "makecache"],
 :
"stderr": "http://x.x.x.x/repos/TSUBASA/repodata/repomd.xml:
[Errno 14] HTTP Error 502 - Bad Gateway\n
 :
```

Cause

The proxy server is not allowed to access to the HTTP server on the management server.

Solution

Check that proxy server is required to access to the HTTP server on the management server.

- When proxy server is required:

  Configure the proxy server to be able to access to the HTTP server on the management server.
- When proxy server is not required:

Follow the procedure below to access to the HTTP server on the management server without proxy server..

1. Edit the following files on the management server to add "proxy=_none_" line.

   `[ /opt/nec/sysmng-soft/etc/ansible/roles/TSUBASA-local-repo/templates/TSUBASA-local.repo.j2 ]`

   ```
   [TSUBASA-local]
   name=SX-Aurora TSUBASA Software Local Repository
   baseurl=http://{{ httpserver }}/{{ reposdir }}
     :
   proxy=_none_
   ```

   `[ /opt/nec/sysmng-soft/etc/ansible/roles/TSUBASA-InfiniBand/templates/mlnx_ofed.repo.j2 ]`

   ```
   [mlnx_ofed]
   name=MLNX_OFED Repository
   baseurl=http://{{ httpserver }}/{{ tsubasa_ib_ofedpkgpath }}/RPMS
     :
   proxy=_none_
   ```

   Caution
   If you don't install packages for SX-Aurora TSUBASA InfiniBand on VHs, editing the TSUBASA-InfiniBand role is not necessary.

2. Run the Playbook again.

   ```
   $ cd /opt/nec/sysmng-soft/etc/ansible
   $ ansible-playbook vh.yml
   ```

Related information

" *5.2. Installing SX-Aurora TSUBASA system software* on page 37 "

A.3. Power Control

A.3.1. VH power control using the powerctrl commands fails.

Problem

The following operations using the `powerctrl` command fails: powering on/off VHs, information display, or resetting VHs.

Cause

Transmitting packets between the BMCs of the management server and VHs fails. The routing of the BMC network between the management server and VHs is not configured.

Solution

Configure the routing of the BMC network between the management server and VHs.

Run the `chassis status` sub-command of `ipmitool` and check that power status of VH machines is displayed.

```
$ ipmitool -H hostname -U username -P password chassis status
```

If the power status is displayed, packets are sent to the BMC on the VH machine.

Related information

" *1.3. System Architecture* on page 10 "

" *6.1. Controlling power with BMC* on page 42 "

A.3.2. VH power control using the powerctrl_w commands fails.

Problem

`powerctrl_w` fails to power on VHs.

Cause

The following things might be the cause.

- WOL is not enabled on the VH machines.
- The management server and VHs reside on different subnets.

Solution

- Enable WOL on VHs.
- Configure your router so that multicast packets should be transferred from the management server to VH machines.

Related information

" *6.2. Powering on/off VHs with WOL* on page 43 "

Appendix B Ansible Playbook

B.1. How to check Ansible playbooks

This appendix explains how to perform checks when you create or edit Ansible playbooks before running playbooks.

- Checking syntax

  Specify --syntax-check option. This option performs only syntax check of a playbook and does not run the playbook.

  ```
  $ ansible-playbook xxx.yml --syntax-check
  ```

- Checking target hosts

  Specify --list-hosts option. This option displays a list of target hosts and does not run the playbook.

  ```
  $ ansible-playbook xxx.yml --list-hosts
  ```

- Checking tasks

  Specify --list-tasks option. This option displays a list of all tasks and does not run the playbook.

  ```
  $ ansible-playbook xxx.yml --list-tasks
  ```

- Checking modifications made by running a playbook

  Specify --check option. This option displays how execution results will be, and does not make any modifications to target hosts. In addition, run this command with the --diff option to check modifications made in target hosts' files.

  ```
  $ ansible-playbook xxx.yml --check --diff
  ```

SX-Aurora TSUBASA System Software
SX-Aurora TSUBASA
Installation Guide (with OSS)

January, 2019 5th. Edition
NEC Corporation