

Application Interface Installation Guide for Bare Metal Server Workloads

- Concepts Introduction
 - Interface types
 - Management Interface
 - Application Interface
 - Single or Multiple physical NICs
 - Single pNIC
 - Multiple pNICs
 - Single or Multiple IP address
 - Support Modes in Ansible Playbook
- How to download Ansible Playbook?
- Ansible tools requirement
 - Where to run Ansible Playbook?
 - Version requirement for Ansible Playbook
- How to play with Ansible Playbook?
 - Static Mode
 - DHCP Mode
 - Migration Mode
 - Installation Restoration
- Other Usage
 - Prepare host with dependency package
 - Remove dependency which are installed during prepare phase
- Troubleshooting
 - Config Result Example in Bare Metal Server
 - Interface
 - Migration mode process log
 - Manual Restore Bare Metal Server
 - static/dhcp mode
 - migrate mode

This guide doc is for how to create Application Interface for Bare Metal Server Workloads.

Before start creating Application Interface for Bare Metal Server, please make sure that Bare Metal Server has been registered as Transport Node;

Ref step1-4 from <https://docs.vmware.com/en/VMware-NSX-T-Data-Center/2.3/com.vmware.nsx.install.doc/GUID-31B12D54-69B1-4458-8E96-8F53332EE3FD.html>

Concepts Introduction

Ansible playbook for Bare Metal Server is a set of automated scripts to install Application Interface for Bare Metal Server.

Interface types

Management Interface

Interface in which the Bare Metal Server is managed from; Management interface works for management traffic, such as SSH connection, NSX Manager/Controller connection, etc...

Application Interface

Interface in which the Application on the Bare Metal Server is accessed from; Application interface works for application traffic, include VLAN and Overlay traffic.

Single or Multiple physical NICs

Physical NICs in Bare Metal Server, here could be two main cases:

Single pNIC

Both for Management and Application interface;

Multiple pNICs

Management and Application interfaces have separate pNIC;

Single or Multiple IP address

IP address solution for Bare Metal Server, here could be two main cases:

1. Single IP address
This single IP address is for both Management and Application IP Interfaces;
2. Two IP addresses
Management and Application Interfaces have separate IP address;

Support Modes in Ansible Playbook

1. Static IP mode
Enable static IP configuration on Application Interface; This means Application IP address is static;
2. DHCP mode
Enable DHCP configuration on Application Interface; This means Application IP address is from DHCP;
3. Migration mode
This mode supports Management and Application sharing the same IP;
Before Application Interface installation, original IP address already exists in Management Interface, so after Application Interface be created, Management and Application Interface will share the same Interface, meanwhile, original Management IP address also will be migrated to this Application Interface.
Under this mode, due to the sharing IP address for both Management and Application traffic, Application Interface only support attach to VLAN 0 Logical Switch(VLAN ID is zero);
This mode is also called "VLAN0 Mode" or "underlay mode";

How to download Ansible Playbook?

Git clone or download code package from: <https://github.com/vmware/bare-metal-server-integration-with-nsxt>

Ansible tools requirement

Where to run Ansible Playbook?

Ansible Playbook scripts can be run from any Linux machine, only require reachable to NSX Manager and Bare Metal Server.

Version requirement for Ansible Playbook

Require Ansible Playbook version **equal or greater than 2.4.3.0**.

How to play with Ansible Playbook?

Static Mode

Steps:

1. Open/edit Ansible inventory file:

```
$vi bms-ansible-nsx/hosts
```

2. Example config for inventory file(bms-ansible-nsx/hosts):

```
#####
### Copyright (C) 2018 VMware, Inc. All rights reserved.
### SPDX-License-Identifier: BSD-2-Clause
#####

# Create an group that contains the Bare Metal Servers
[TransportNodes:children]
servers_static
servers_dhcp
servers_migration

# Set variables common for all Bare Metal Servers
[TransportNodes:vars]
# SSH user
ansible_ssh_user=root // Login Username of Bare Metal Server
ansible_ssh_pass=test // Login Password of Bare Metal Server

rpm_dependency=["tcpdump", "boost-filesystem", "PyYAML", "boost-iostreams", "boost-chrono", "python-mako", "python-netaddr", "python-six", "gperftools-libs", "libunwind", "snappy", "boost-date-time", "c-ares", "redhat-lsb-core", "wget", "net-tools", "yum-utils", "lsfd", "libvirt-libs", "python-gevent", "libev", "python-greenlet"]

dpkg_dependency=["libunwind8", "libflags2v5", "libgoogle-perftools4", "traceroute", "python-mako", "python-simplejson", "python-unittest2", "python-yaml", "python-netaddr", "libboost-filesystem1.58.0", "libboost-chrono1.58.0", "libgoogle-glog0v5", "dkms", "libboost-date-time1.58.0", "python-protobuf", "python-gevent", "libsnappy1v5", "libleveldb1v5", "libboost-program-options1.58.0", "libboost-thread1.58.0", "libboost-iostreams1.58.0", "libvirt0"]

# host group for servers
[servers_static]
server1 ansible_ssh_host=1.1.1.1 static_ip=2.2.2.2 netmask=255.255.255.0 ls_name=LogicalSwitch1 // Bare Metal Server Configure Info for static IP mode

[servers_dhcp]

[servers_migration]
server6 ansible_ssh_host= migrate_intf= ls_name=

[servers_restoration]
server6

# NSX Configuration
[NSX]
#=====
# NSX Manager Credential
nsxmanager ip=192.168.1.1 username=test password=test // NSX Manager Credential
#=====
```

3. Run Ansible Playbook:

```
$ansible-playbook -i hosts static_config.yml
```

DHCP Mode

steps:

1. Open/edit Ansible inventory file:

```
$vi bms-ansible-nsx/hosts
```

2. Example config for inventory file(bms-ansible-nsx/hosts):

```
#####
### Copyright (C) 2018 VMware, Inc. All rights reserved.
### SPDX-License-Identifier: BSD-2-Clause
#####

# Create an group that contains the Bare Metal Servers
[TransportNodes:children]
servers_static
servers_dhcp
servers_migration

# Set variables common for all Bare Metal Servers
[TransportNodes:vars]
# SSH user
ansible_ssh_user=root // Login Username of Bare Metal Server
ansible_ssh_pass=test // Login Password of Bare Metal Server

rpm_dependency=["tcpdump", "boost-filesystem", "PyYAML", "boost-iostreams", "boost-chrono", "python-mako", "python-netaddr", "python-six", "gperftools-libs", "libunwind", "snappy", "boost-date-time", "c-ares", "redhat-lsb-core", "wget", "net-tools", "yum-utils", "lsf", "libvirt-libs", "python-gevent", "libev", "python-greenlet"]

dpkg_dependency=["libunwind8", "libgflags2v5", "libgoogle-perftools4", "traceroute", "python-mako", "python-simplejson", "python-unittest2", "python-yaml", "python-netaddr", "libboost-filesystem1.58.0", "libboost-chrono1.58.0", "libgoogle-glog0v5", "dkms", "libboost-date-time1.58.0", "python-protobuf", "python-gevent", "libsnappy1v5", "libleveldb1v5", "libboost-program-options1.58.0", "libboost-thread1.58.0", "libboost-iostreams1.58.0", "libvirt0"]

# host group for servers
[servers_static]
server1 ansible_ssh_host= static_ip= netmask= ls_name=

[servers_dhcp]
server2 ansible_ssh_host=1.1.1.2 ls_name=LogicalSwitch2 // Bare Metal Server Configure Info for DHCP mode

[servers_migration]
server6 ansible_ssh_host= migrate_intf= ls_name=

[servers_restoration]
server6

# NSX Configuration
[NSX]
#=====
# NSX Manager Credential
nsxmanager ip=192.168.1.1 username=test password=test // NSX Manager Credential
#=====
```

3. Run Ansible Playbook:

```
$ansible-playbook -i hosts dhcp_config.yml
```

Migration Mode

Steps:

1.Open/edit Ansible inventory file:

```
$vi bms-ansible-nsx/hosts
```

2.Example config for inventory file(bms-ansible-nsx/hosts):

```
#####
### Copyright (C) 2018 VMware, Inc. All rights reserved.
### SPDX-License-Identifier: BSD-2-Clause
#####

# Create an group that contains the Bare Metal Servers
[TransportNodes:children]
servers_static
servers_dhcp
servers_migration

# Set variables common for all Bare Metal Servers
[TransportNodes:vars]
# SSH user
ansible_ssh_user=root // Login Username of Bare Metal Server
ansible_ssh_pass=test // Login Password of Bare Metal Server

rpm_dependency=["tcpdump", "boost-filesystem", "PyYAML", "boost-iostreams", "boost-chrono", "python-mako", "python-netaddr", "python-six", "gperftools-libs", "libunwind", "snappy", "boost-date-time", "c-ares", "redhat-lsb-core", "wget", "net-tools", "yum-utils", "lsf", "libvirt-libs", "python-gevent", "libev", "python-greenlet"]

dpkg_dependency=["libunwind8", "libgflags2v5", "libgoogle-perftools4", "traceroute", "python-mako", "python-simplejson", "python-unittest2", "python-yaml", "python-netaddr", "libboost-filesystem1.58.0", "libboost-chrono1.58.0", "libgoogle-glog0v5", "dkms", "libboost-date-time1.58.0", "python-protobuf", "python-gevent", "libsnappy1v5", "libleveldb1v5", "libboost-program-options1.58.0", "libboost-thread1.58.0", "libboost-iostreams1.58.0", "libvirt0"]

# host group for servers
[servers_static]
server1 ansible_ssh_host= static_ip= netmask= ls_name=

[servers_dhcp]
server2 ansible_ssh_host= ls_name=

[servers_migration]
server6 ansible_ssh_host=1.1.1.3 migrate_intf=eth0 ls_name=vlan0-ls // Bare Metal Server Configure Info for Migration Mode

[servers_restoration]
server6

# NSX Configuration
[NSX]
#=====
# NSX Manager Credential
nsxmanager ip=192.168.1.1 username=test password=test // NSX Manager Credential
#=====
```

3. Run Ansible Playbook:

```
$ansible-playbook -i hosts migrate.yml
```

Installation Restoration

Restoration is for recovering Bare Metal Server from Application Interface installation;

Steps:

1.Open/edit Ansible inventory file:

```
$vi bms-ansible-nsx/hosts
```

2.Example config for inventory file:

```
#####
### Copyright (C) 2018 VMware, Inc. All rights reserved.
### SPDX-License-Identifier: BSD-2-Clause
#####

# Create an group that contains the Bare Metal Servers
[TransportNodes:children]
servers_static
servers_dhcp
servers_migration

# Set variables common for all Bare Metal Servers
[TransportNodes:vars]
# SSH user
ansible_ssh_user=root // Login Username of Bare Metal Server
ansible_ssh_pass=test // Login Password of Bare Metal Server

rpm_dependency=["tcpdump", "boost-filesystem", "PyYAML", "boost-iostreams", "boost-chrono", "python-mako", "python-netaddr", "python-six", "gperftools-libs", "libunwind", "snappy", "boost-date-time", "c-ares", "redhat-lsb-core", "wget", "net-tools", "yum-utils", "lsf", "libvirt-libs", "python-gevent", "libev", "python-greenlet"]

dpkg_dependency=["libunwind8", "libgflags2v5", "libgoogle-perftools4", "traceroute", "python-mako", "python-simplejson", "python-unittest2", "python-yaml", "python-netaddr", "libboost-filesystem1.58.0", "libboost-chrono1.58.0", "libgoogle-glog0v5", "dkms", "libboost-date-time1.58.0", "python-protobuf", "python-gevent", "libsnappy1v5", "libleveldb1v5", "libboost-program-options1.58.0", "libboost-thread1.58.0", "libboost-iostreams1.58.0", "libvirt0"]

# host group for servers
[servers_static]
server1 ansible_ssh_host= static_ip= netmask= ls_name=

[servers_dhcp]
server2 ansible_ssh_host= ls_name=

[servers_migration]
server6 ansible_ssh_host= migrate_intf= ls_name=

[servers_restoration] // All Bare Metal Servers which need to be restored should be listed here
server1
server2
server6

# NSX Configuration
[NSX]
#=====
# NSX Manager Credential
nsxmanager ip=192.168.1.1 username=test password=test // NSX Manager Credential
#=====
```

3. Run Ansible Playbook:

```
$ansible-playbook -i hosts restore.yml
```

Other Usage

Prepare host with dependency package

This section is for auto detect and install third-party dependencies for Bare Metal Server before Host Agents installation;

command:

```
$ansible-playbook -i hosts prepare.yml
```

This command will auto detect the missed required third-party dependency package, record, then auto install them;

Remove dependency which are installed during prepare phase

This section is for auto remove dependencies which are installed during host prepare phase;

command:

```
$ansible-playbook -i hosts unprepare.yml
```

Troubleshooting

Config Result Example in Bare Metal Server

Interface

Following two new interfaces are created:

```
nsx-eth: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.117.4.167 netmask 255.255.252.0 broadcast 10.117.7.255
inet6 fc00:10:117:7:250:56ff:feb0:9b81 prefixlen 64 scopeid 0x0<global>
inet6 fe80::250:56ff:feb0:9b81 prefixlen 64 scopeid 0x20<link>
ether 00:50:56:b0:9b:81 txqueuelen 1000 (Ethernet)
RX packets 2748 bytes 281525 (274.9 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 365 bytes 81477 (79.5 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

nsx-eth-peer: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet6 fe80::54a8:7fff:fe8e:6648 prefixlen 64 scopeid 0x20<link>
ether 56:a8:7f:8e:66:48 txqueuelen 1000 (Ethernet)
RX packets 365 bytes 81477 (79.5 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 2748 bytes 281525 (274.9 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Migration mode process log

/var/log/migrate.log

Manual Restore Bare Metal Server

static/dhcp mode

```
$/opt/vmware/nsx-bm/static_dhcp_manual_restore.sh
```

migrate mode

```
$/opt/vmware/nsx-bm/migration_manual_restore.sh
```

