LOC-A
for VCF User Guide
(Version 3.3 R2410)

Date: 2024-10-18

Table of Contents

Summary of Statement

This document is intended for both professional services engineers and end users.

For professional service engineers, this document explains how to set up a LOC-A bootstrap environment and how to deploy a Service Provider Domain (SPD) as a management cluster. In addition, it details how to upgrade LOC-A services.

For end users, this document describes how to use LOC-A automation to deploy or expand a workload cluster.

Terminology

Term	Definition
LOC-A	Lenovo Open Cloud Automation
LIS	LOC-A Inventory Service
LMS	LOC-A Hardware Management Service
LCS	LOC-A Configuration Service
LRS	LOC-A Repository Service
LDS	LOC-A Discovery Service
LLS	LOC-A Logging Service
BMC/XCC	Integrated Management Module
LXCA	Lenovo xClarity Administrator
HCI	Hyper-converged Infrastructure
VCF	VMware Cloud Foundation
OOB	Out-of-band
DCIM	Datacenter Information Management
SPD	Service Provider Domain
CWD	Customer Workload Domain

LOC-A Overview

Architecture

Lenovo Open Cloud Automation (LOC-A) is a modular automation framework designed to enable Lenovo's customers to easily deploy and manage cloud solutions and workloads on Lenovo hardware. It is intended to be:

- A lightweight automatic deployment engine, which is **OPEN** to extend support to various cloud providers
- An **Enterprise** solution for cloud lifecycle management

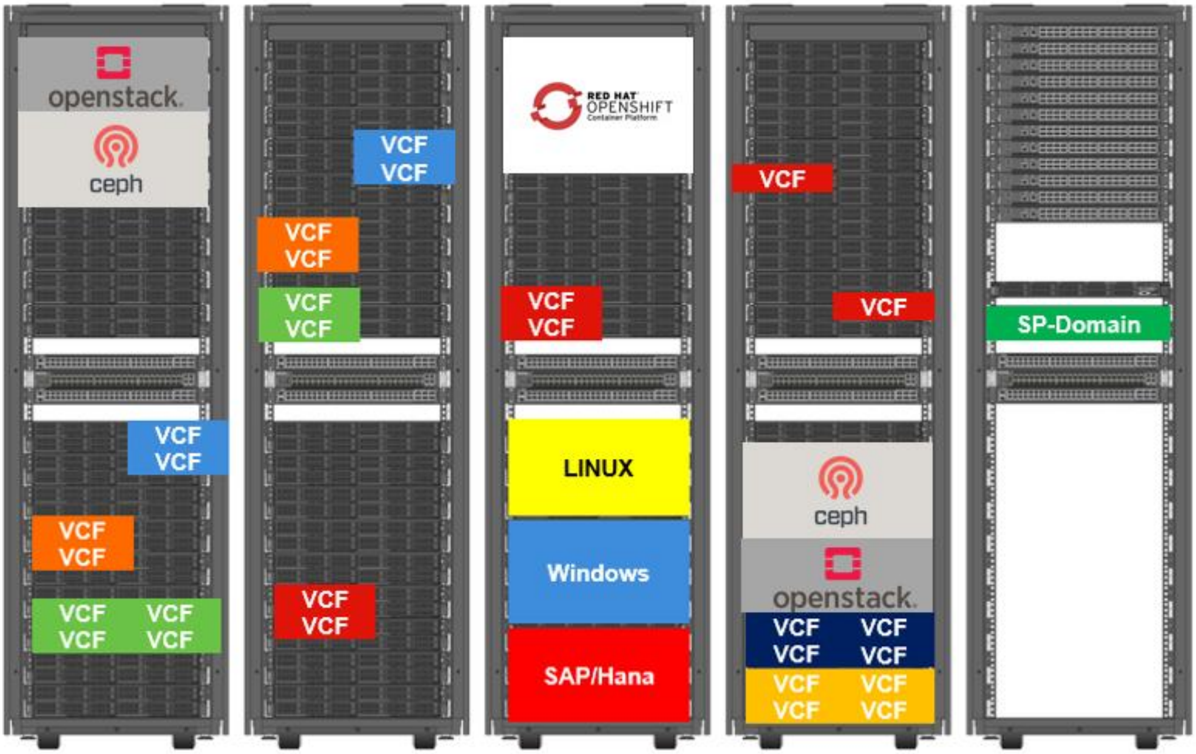


Figure 1: LOC-A cloud for multiple racks/multiple workloads

LOC supports deploying both VMWare-based and Red Hat-based clouds. The VMWare version (LOC-A VCF) supports the deployment of VMWare Cloud Foundation clouds for its tenants while the Red Hat version (LOC-A RH) supports the deployment of OpenShift clouds. As LOC is modular, it is designed to enable the implementation of support for other cloud providers in the future. This document intends to focus on the LOC-A VCF version.

Physical Infrastructure

A typical LOC-A infrastructure in a customer production environment contains two types of clusters:

- **A management cluster:** The management cluster is the Service Provider Domain (SPD), which hosts the LOC-A management services.

- **One or more workload clusters:** The workload clusters, also known as the Customer Workload Domain (CWD) are the customer cloud infrastructures.

In addition, LOC-A requires an initial node as Bootstrap node that runs basic LOC-A services to deploy the SPD and bring up the management services.

Figure 2 shows a typical LOC-A infrastructure:

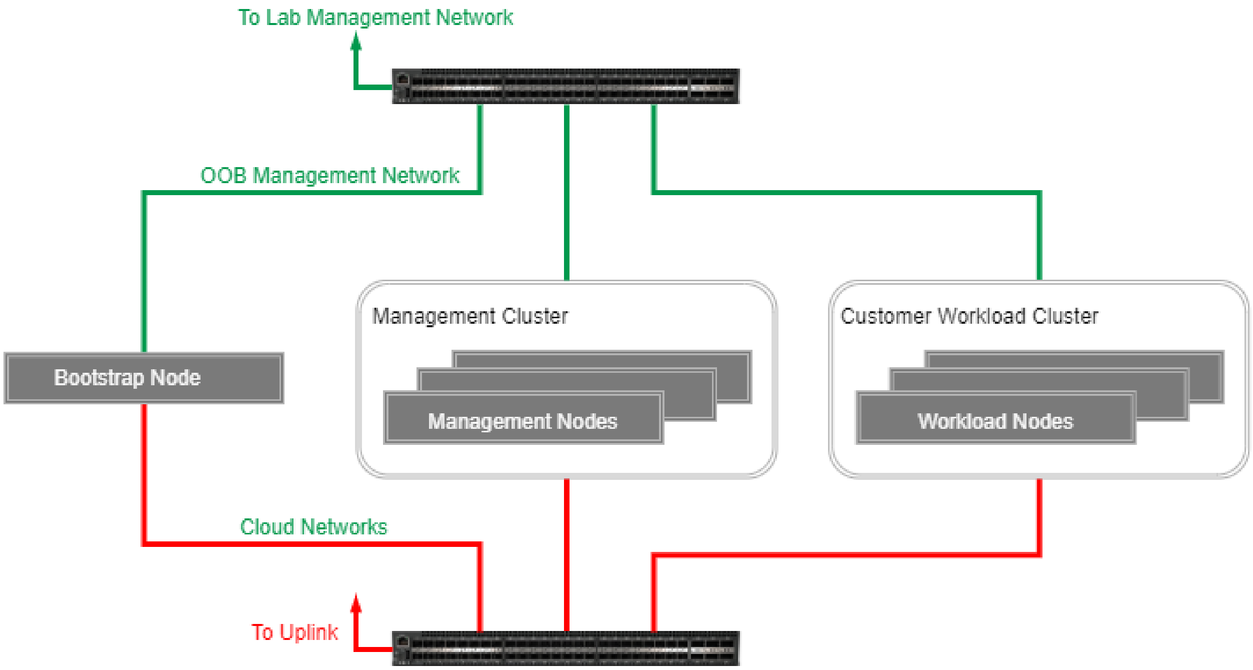


Figure 2: typical LOC-A infrastructure

- **Bootstrap Node:** Initial node with base LOC-A services that is used to bring up the rest of the infrastructure. The Bootstrap is usually a one-off utility, it is not necessarily required to be persistent in customer environment. It can be a laptop, a dedicated Tiny workstation, or a shared utility server in the customer environment that has virtualization technology enabled.
- **Management Cluster (SPD):** The cluster that hosts LOC-A and workload management services. LOC-A VCF currently supports VMware Cloud Foundation as the SPD.
- **Workload Cluster (CWD):** The customer workload cluster deployed and managed by LOC-A. The supported versions of workload clusters will be extended in the future.
- **OOB Management Network:** Out-of-band management network for XCC, as well as for switch discovery and management
- **Cloud Networks:** In-band, cloud-specific data and management networks. The cloud network topology may vary for different cloud versions and should be customized flexibly according to customer requirements. It is usually isolated with multiple VLANs. VLAN technology provides a simple way for logical grouping and isolation of the various networks. Cloud networks can further access into the uplink network.

The components above can be co-located in the same rack in a Datacenter, and LOC-A supports multiple CWDs across multiple racks. Figure 3 shows the rack view of a typical LOC-VCF deployment:

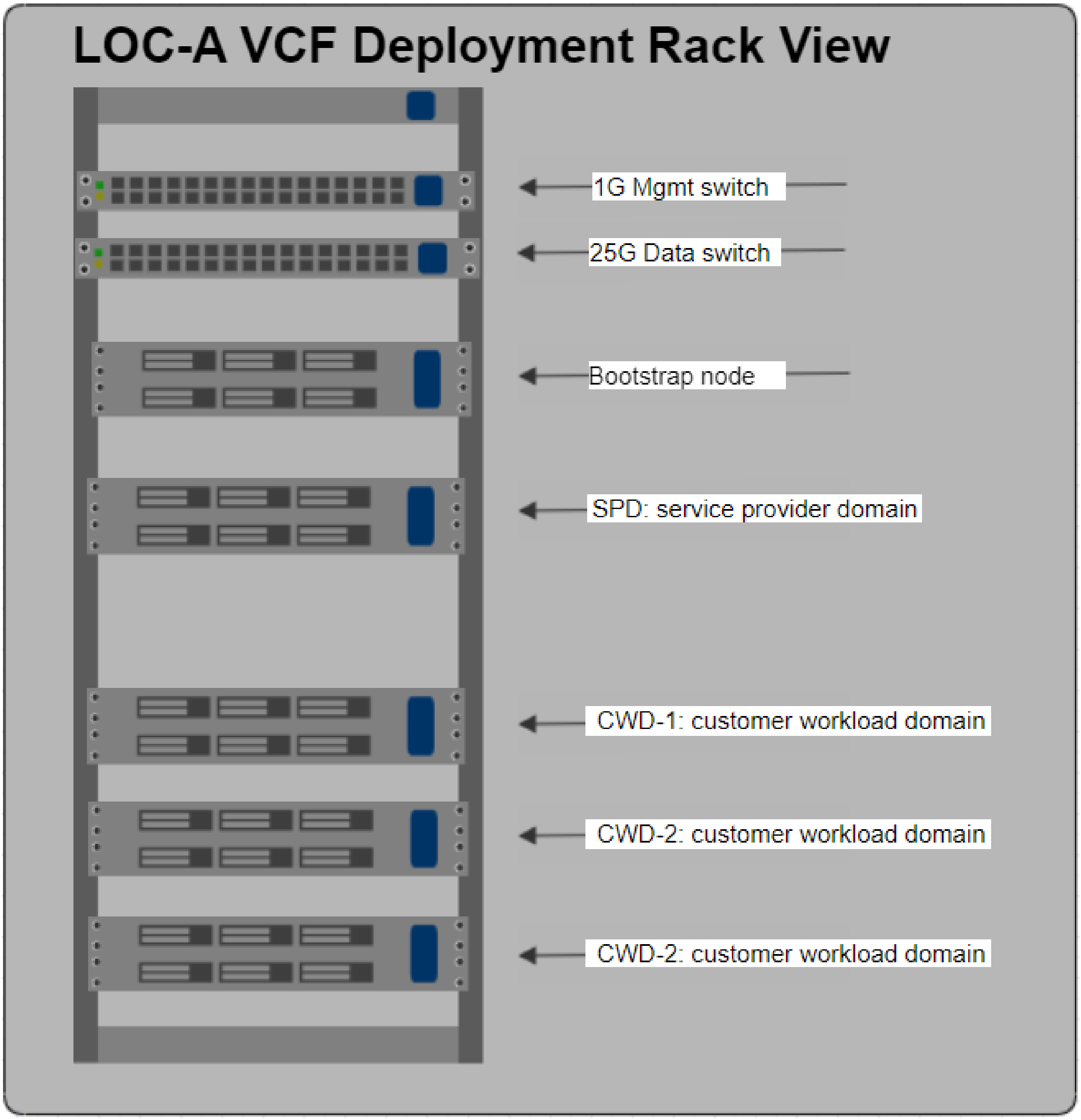

LOC-A VCF Deployment Rack View
1G Mgmt switch
25G Data switch
Bootstrap node
SPD: service provider domain
CWD-1: customer workload domain
CWD-2: customer workload domain
CWD-2: customer workload domain

Figure 3: LOC-A In a Rack View

Software Components

The LOC-A automation adopts the design principle of Infrastructure as code. It leverages the open-source Datacenter Information Management tool as the metadata database for datacenter resource management and cloud version representation, and it uses Ansible to codify the execution of the deployment of the infrastructure. It automates the process of planning, configuration, deployment, and lifecycle management for supported cloud versions.

There are two types of services running on a LOC-A management cluster:

- **LOC-A services:** The modular software components that act as the engine for deployment automation and day-2 management. The services cover a wide range of common features, including server discovery and provisioning, software package management, configuration driven automation, as well as inventory planning and tracking, validation, monitoring, and logging, etc. LOC-A services are installed as virtual appliances on the SPD.
- **Cloud management services:** The cloud workload management services are usually deployed via LOC-A services. They are cloud-specific services to support the deployment or management of various cloud infrastructures supported by LOC-As. Optionally, the services can be a virtualized control plane or installer of the cloud software (eg. VCF NSX edge nodes, VCF Cloud Builder, CWD-specific DNS service appliance, etc..). The services can also be Lenovo proprietary software, such as Lenovo Ceph Dashboard. Workload management services can vary on the system, depending on the current workload cluster being used.

LOC-A Services

Inventory service (LIS)

LIS is the source-of-truth of infrastructure planning that handles Datacenter resources, including devices (servers, switches, etc.), VLAN, IP prefixes and IPs, virtual machines, etc., and the cloud objects, such as tenants and clusters, etc. The metadata of resources is added and allocated automatically in the planning phase.

LOC-A uses NetBox as the inventory service that stores the metadata in the database, and it also provides an API endpoint that is consumed by the configuration service. The automation service uses this API to put system records for the environment into the service, to retrieve information about the systems in inventory, and to update the status of systems in inventory. This allows for LOC-A to automatically consume systems from inventory that are used to deploy cloud or storage nodes and subsequently update their status. It also provides a user-friendly web endpoint for administrators to see the state of their clusters.

From LOC-A 2.4, The version of Netbox has been upgraded from v2.6.7 to v3.1.5.

From LOC-A 2.6, The version of Netbox has been upgraded from 3.1.5 to 3.3.7.

From LOC-A 3.1 R2405, The version of Netbox has been upgraded from 3.3.7 to 3.7.4.

Repository service (LRS)

LRS is where LOC-A stores its configuration files and Ansible playbooks. All published automation scripts are saved and version-controlled by this service. Configuration files and playbooks (LOC-A codes) are stored as git repositories in GitLab. Professional services can update configuration files of the infrastructure via the git command line. The repositories are consumed and executed by configuration service.

From LOC-A 2.6, The version of GitLab has been upgraded from 13.10.5 to 15.3.1 for the security requirement.

Discovery service (LDS)

LDS is the helper service that discovers the server inventories in the datacenter. It is connected to the OOB network to monitor for Lenovo hardware resources. The inventory information can also be added into the LIS as metadata. LOC-A uses Lenovo Confluent as the discovery service.

From LOC-A 2.5, The version of confluent has been upgraded from 3.1.1 to 3.5.0.

From LOC-A 2.8 R2307, The version of confluent has been upgraded from 3.5.0 to 3.9.0.

From LOC-A 3.1 R2405, The version of confluent has been upgraded from 3.9.0 to 3.9.3.

Configuration service (LCS)

LCS is an automation execution orchestrator built on AWX. It supports common workflow patterns, such as sequential, parallel, mixed, and cron. The execution status and outputs are logged and can be tracked through this service.

LCS is configured with a list of pre-defined automation workflows and job templates that make managing the infrastructure easy and efficient. It also allows LOC-A customers to create and manage their own automation playbooks to be executed against an LOC-A based cloud.

From LOC-A 3.2 R2407, The version of awx has been upgraded from 15.0 to 24.0. upgrade from docker to k3s.

Hardware management service (LMS)

LMS helps to provision hardware and performs hardware management operations during the lifecycle of Lenovo servers. LOC-A includes Lenovo XClarity Administrator (LXCA) as its hardware management service. LMS is responsible for:

- Server inventory
- Server power operations
- Server operating system deployment
- Server firmware updates
- Server configurations

From LOC-A 2.4, LXCA has been upgraded to version 3.4.5 to fix the Apache Log4j common vulnerabilities and exposures.

From LOC-A 2.5, The version of LXCA has been upgraded from 3.4.5 to 3.6.0 to support baremetal as a service feature.

From LOC-A 2.9 R2311, The version of LXCA has been upgraded from 3.6.0 to 4.1.

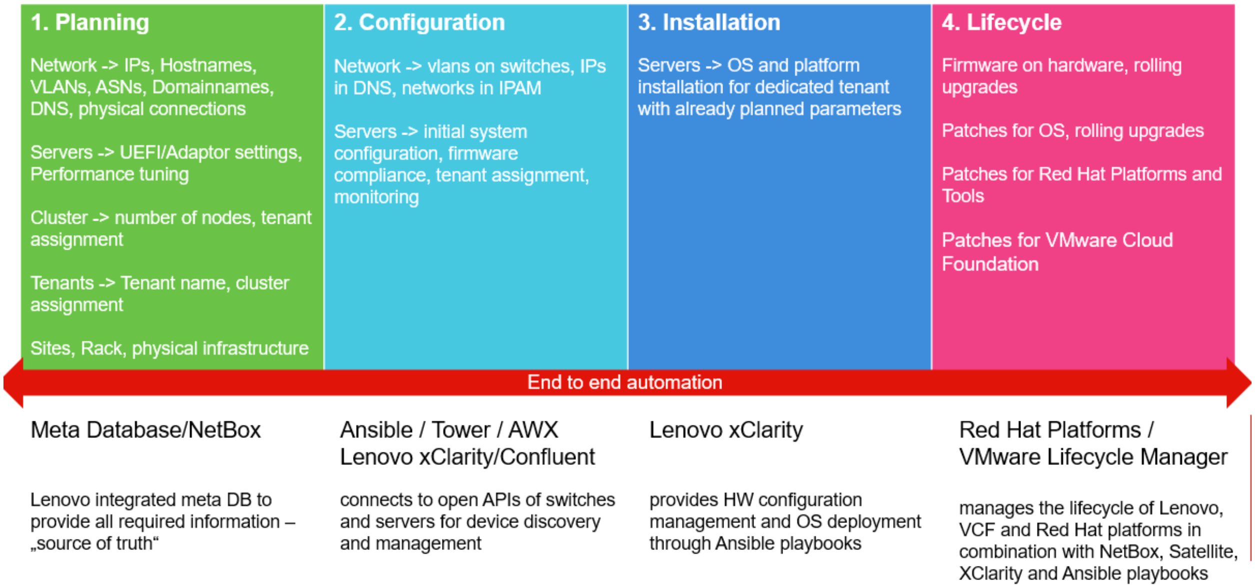
Logging and analysis service (LLS)

LLS is the central endpoint for LOC-A logging. It collects logs generated during cloud deployment and lifecycle and can be extended for further analysis. LOC-A integrates the ELK stack as the logging and analysis service. An ARA instance is also installed on LLS to facilitate log collections and analysis from AWX workflows.

From LOC-A 3.3 R2410, The version of ARA has been upgraded from 1.5.7 0 to 1.7.1

LOC-A services for cloud automation

Via the software stack of LOC-A service components, LOC-A can perform automatic planning, configuration, installation, and lifecycle management for the end-to-end cloud infrastructure lifecycle.



1. Planning

Network -> IPs, Hostnames, VLANs, ASNs, Domainnames, DNS, physical connections

Servers -> UEFI/Adaptor settings, Performance tuning

Cluster -> number of nodes, tenant assignment

Tenants -> Tenant name, cluster assignment

Sites, Rack, physical infrastructure

2. Configuration

Network -> vlans on switches, IPs in DNS, networks in IPAM

Servers -> initial system configuration, firmware compliance, tenant assignment, monitoring

3. Installation

Servers -> OS and platform installation for dedicated tenant with already planned parameters

4. Lifecycle

Firmware on hardware, rolling upgrades

Patches for OS, rolling upgrades

Patches for Red Hat Platforms and Tools

Patches for VMware Cloud Foundation

End to end automation

Meta Database/NetBox

Lenovo integrated meta DB to provide all required information – „source of truth“

Ansible / Tower / AWX
Lenovo xClarity/Confluent

connects to open APIs of switches and servers for device discovery and management

Lenovo xClarity

provides HW configuration management and OS deployment through Ansible playbooks

Red Hat Platforms /
VMware Lifecycle Manager

manages the lifecycle of Lenovo, VCF and Red Hat platforms in combination with NetBox, Satellite, XClarity and Ansible playbooks

Figure 4: LOC-A cloud automation phases

Table 1 lists the software bill of materials (BOM) of LOC-A services:

LOC-A VCF	Package	Sub-pkg files	Version
LMS	LXCA	Lxca4.1.ova	4.1.0
LDS	Confluent	docker_confluent_008_20240708.tar.gz config_confluent_005_20231204.tar.gz	3.9.3
LRS	Gitlab	docker_gitlab_003_20220909.tar.gz config_gitlab_001_20210417.tar.gz config_fileserver_003_20231204.tar.gz	15.3.1
LCS	AWX	config_awx_005_20231206.tar.gz k3s_venv.tar.gz k3s_awx_002_20240513.tar.gz	24.0
LIS	NetBox	docker_netbox_007_20240318_v3.7.4.tar.gz config_netbox_007_20240318_v3.7.4.tar.gz	3.7.4
LLS	ARA	docker_ara_003_20240710.tar.gz	1.7.1

LOC-A VCF	Package	Sub-pkg files	Version
		config_ara_001_20210417.tar.gz	

Table 1: LOC-A services Software BOMs

LOC-A Clouds

VMware Cloud Foundation (VCF)

LOC-A VCF supports VCF 4.x and VCF 5.x.

VCF 3.x has been retired since LOC-A 2.8 R2307.

The VCF can be deployed as an SPD or CWDs in the datacenter. By default, a consolidated architecture model is applied when LOC-A deploys VCF cloud instances.

For more information about VCF, see:

https://docs.vmware.com/en/VMware-Cloud-Foundation/3.9/vcf-39-ovdeploy-guide.pdf

Spine-leaf topology

LOC-A supports a spine-leaf network topology for VCF cloud deployment with either non-stretched or stretched (twin-core) deployment model.

Figure 5 shows the typical high-level spine-leaf network topology for a VCF SPD and multiple tenants' CWDs on multiple racks. Two spine switches on top connect to all leaf switches of all racks. Each leaf is connected to each spine with one cable.

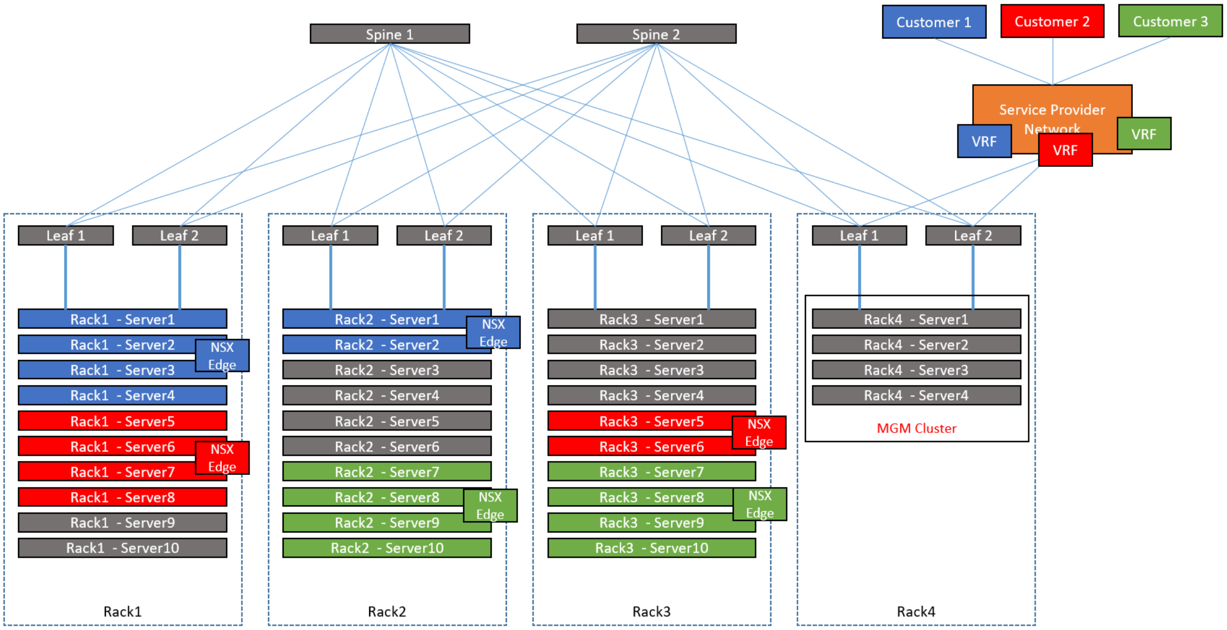


Figure 5: Spine Leaf Topology

Each customer (blue, red, green) has its own dedicated VLANs with overlay networks. By using VLANs only it is not possible i.e., for the blue customer to have server 1-4 in Rack1 and server 1-2 in Rack2 in the same VLAN/Broadcast-Domain. By using VLANs with VXLAN overlay networks, all assigned servers are in the same broadcast domain.

LOC-A supports automated switch configurations for the leaf switches. The supported switches are:

- Lenovo CNOS switches
- Cisco ACI switches
- Arista switches

VCF deployment within the datacenter can be customized via LOC-A configuration files stored in LRS. LOC-A can handle and execute the resource planning and deployment for different configurations of cloud metadata. The following sections within this chapter introduce one typical example of model and network of VCF instance with SPD and CWD in place. **Note that this network topology is a default VCF cloud example and can be modified via proper customization of LOC-A configuration files.**

External Network Services

Various external network services are required for the initial deployment of a VCF-based SPD and CWD, as well as VMware optional components like vRealize Operations or vRealize Automation.

Table 2 lists the required network services:

Service	**Purpose**	**CWD**
Dynamic Host Configuration Protocol (DHCP) server	Automated IP address allocation for VXLAN Tunnel Endpoints (VTEPs) and NSX host VTEPs.	No
Domain Name Services (DNS)	Name resolution for VCF components.	No
Network Time Protocol (NTP)	Provides synchronized time for VCF components.	Yes
AD DNS	Provides name resolution for CWD management NAT IP address to uplink	Yes

Service	Purpose	CWD
L3 router	Required for datacenter multi-CWD deployment. Provides L3 routing for various CWD and SPD VCF components and LOC-A services. This can be either a physical router in a production environment, or a software router like pfSense in a development environment.	No
AVN BGP neighbor	Required as SPD AVN BGP Peer for VMware LogInsight and vRealize Suite	No

Table 2: External Services required by VCF

Management Network

On each rack of the datacenter, there is a 1G switch for the out-of-band management of the XCC and the switches connected to the SP domain directly for Lenovo XClarity Administrator communication. The OOB network can be on a separate VLAN network. The network design for the management network must meet certain requirements:

- All server XCCs are connected to the OOB management network.
- One or more dedicated OOB switches are used for the OOB management network. The management IP address of the OOB switches need to be L3 accessible by LOC-A services in the LOC-A services network (see Table 3 in Cloud Network)
- LCS needs to be in the same L2 domain as the server XCCs and switches OOB management network to enable automatic server node detection and XCC IP assignment.
- LMS needs to be able to route to the OOB management network via an external L3 router to facilitate automated operating system deployment for the servers.

Cloud Network

All servers will need to have at least 2 10Gb or 25Gb network interfaces (in-band) for cloud network data transfer. For any SPD or CWD, Table 3 lists the required or optional VLANs on the cloud network:

VLAN	Description	Global	Associated Nodes/Components
vmanagement	VCF management network	No	Workload nodes
vsan	VCF VSAN network	No	Workload nodes
vmotion	VCF vMotion network	No	Workload nodes
uplink (ccm_tra)	VCF Edge to uplink	Yes	SPD Workload nodes, CWD NSX Management Edge
vtep	VXLAN (NSX VTEP)	No	Workload node

VLAN	Description	Global	Associated Nodes/Components
vrealize_uplink1	VRealize suite uplink1	No	SPD workload nodes, workload node, CWD NSX DLR, CWD NSX Management Edge
vrealize_uplink2	VRealize suite uplink1	No	SPD workload nodes, workload node, CWD NSX DLR, CWD NSX Management Edge
edge_to_wan	cwd_network, for the tenant-access itself, will be connected to some upstream switches from the cloud-provider	No	SPD workload nodes, workload nodes, CWD NSX Management Edge
vsan_witness	For twin-core only	No	CWD Witness VM
LOC-A services	LOC-A services interconnections and uplink access	Yes	LOC-A services
OOB	Server hardware management network	Yes	Server node XCC, LOC-A LDS

Table 3: Cloud Networks for VCF

Datacenter VLAN Planning

Networks are usually isolated via VLANs within a datacenter to contain one instance of SPD (static) and multiple instances of CWD (dynamic). Table 4 shows an example of VLAN planning of one site of a datacenter. For twin-core VCF across two sites, the same sets of VLANs need to be allocated.

They can be customized in the configuration file of LOC-A.

VLAN Group	VLANs	Count	Network Purpose
siteX_switch_management	6	1	Switch OOB management
siteX_device-management	7	1	Server OOB management
siteX_product-data-1	600-611	12	SPD VLANs for VCF Deploy (e.g. (vManagement, vSAN, vMotion, vTEP)
	27-28	2	SPD vRealize uplink VLANS
	612	1	LOC services
	24	1	uplink (ccm_tra)
	613-2612	2000	Reserved for CWD VLAN allocation dynamically. The count can be set on-demand. For OnPrem, this will be in the remote edge site.
siteX_witness_vlan_group	25	1	VMware Witness network

Table 4: VLAN planning for Datacenter

OnPrem Network

OnPrem CWD on remote edge site connected with Datacenter SPD by VPN connection. There is one MGMT GW (management gateway) device on each edge site. All network configurations on the edge site need to be done manually, include VCF requirement network configurations according to the CWD VLAN plan and VPN configuration. VPN configuration on the SPD edge is set by LOC_edge_nem_workflow.

Figure 6 shows the network topology between SPD and OnPrem CWD.

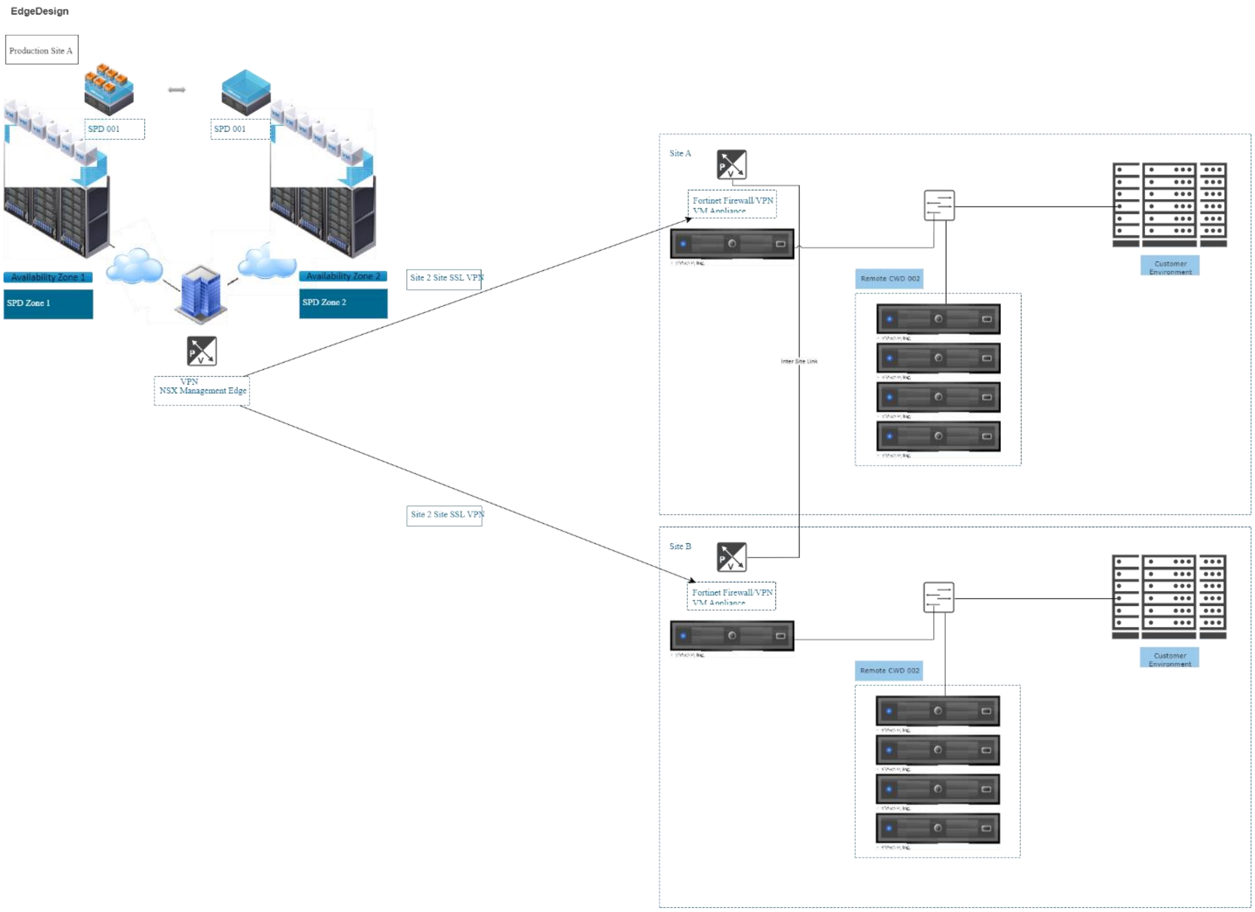


Figure 6: Onprem Network Topology

Table 15 lists the required external network services for OnPrem CWD:

Service	Purpose	OnPrem CWD
Dynamic Host Configuration Protocol	Automated IP address allocation for VXLAN Tunnel Endpoints (VTEPs) and NSX host VTEPs.	No

Service	Purpose	OnPrem CWD
(DHCP) server		
Domain Name Services (DNS)	Name resolution for VCF components.	No
Network Time Protocol (NTP)	Provides synchronized time for VCF components.	Yes
AD DNS	Provides name resolution for CWD management NAT IP address to uplink	Yes
L3 router	Required for datacenter multi-CWD deployment. Provides L3 routing for various CWD and SPD VCF components and LOC-A services. This can be either a physical router in a production environment, or a software router like pfSense in a development environment.	Yes
AVN BGP neighbor	Required as SPD AVN BGP Peer for VMware LogInsight and vRealize Suite	Yes

Table 15: External Services required by OnPrem CWD

LOC-A installation

Prerequisites

- You will need at least the following hardware to perform LOC-A installation.

Device	Count	Minimal HW configuration	Role
Switch	1	1Gb	OOB switch
Switch	1	10/25Gb	Cloud data switch
GP server	1	32 CPU Core, 64GB Memory, 2x25Gb NIC, 2x480GB SSD	Bootstrap
VSAN certified servers	4	192GB Memory, 2x25Gb NIC, 1x1Gb BMC NIC, and must align with VSAN ReadyNodes requirement. Identical configurations are required for the 4 nodes	SPD nodes
VSAN certified servers	4+	192GB Memory, 2x25Gb NIC, 1x1Gb BMC NIC, and must align with VSAN ReadyNodes requirement. Identical configurations are required for the same CWD.	CWD nodes

Table 5: LOC-A Hardware Requirement

If you want to install a twin-core SPD, you will need an additional 4 nodes with identical configuration in Site B for SPD deployment.

- Validate that the hardware is installed and cabled to the network based on the wiring diagram shown in Figure 6.

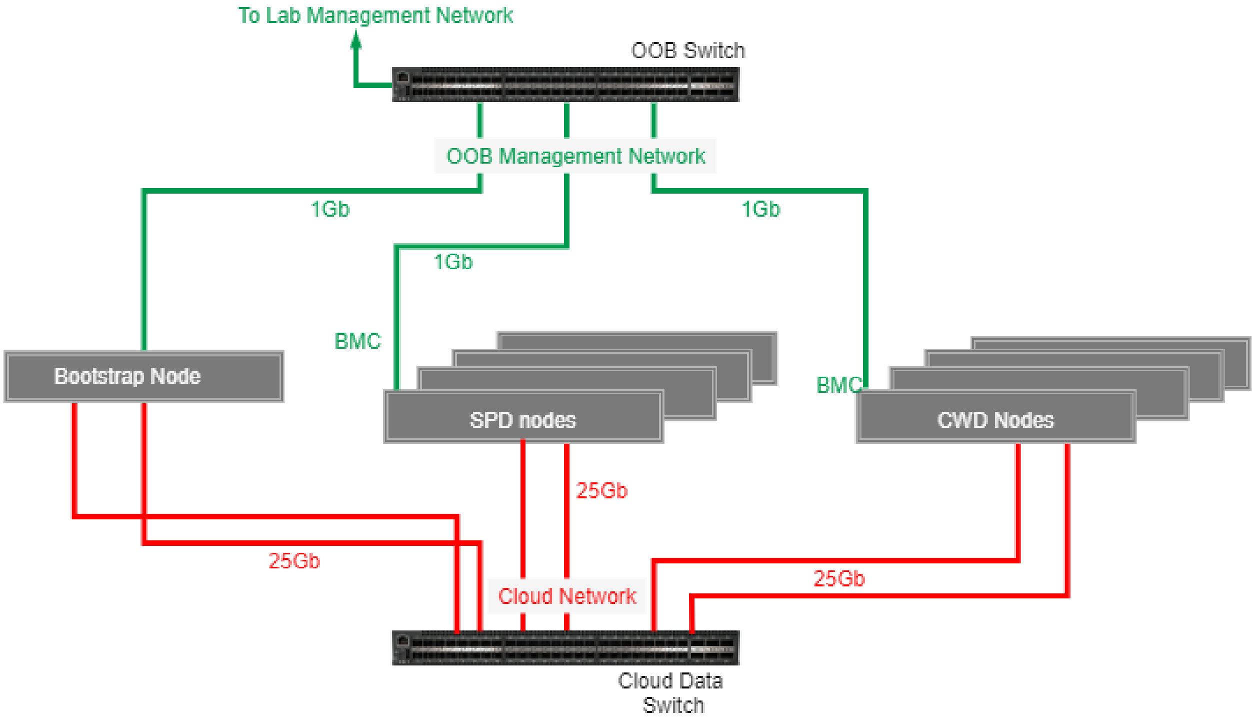


Figure 6: Wiring Diagram

- Obtain all software packages
  You are expected to get LOC-A VCF software packages from the software development team pre-day1. E-mail locautomation_help@lenovo.com to request how to get the software packages.

  By Default, LOC-A software packages contain three parts:
  - loc: contains all LOC-A automation code
  - configs: contains all LOC-A automation configuration files
  - files: contains all LOC-A VM images (ova files) and OS files (iso)

LOC-A automation configuration files layout:

```
├──── configs          // real config file folder, usually at runner /opt/loc/configs, or configs repo of LRS
|   ├──── loc_service_input.yml      //user input parameters for spd
|   ├──── loc_service_bootstrap_input.yml    //user input parameters for bootstrap
|   ├──── mgmt.yml   //auto generated, describe the existing management platform, READ ONLY
|   ├──── netbox-initialization.yml    //unified netbox initialization file
|   ├──── version           //config file version
|   ├─── devices           // contains devices onboard files
|   |   ├─── rack1_device.yml
|   |   ├─── rack2_device.yml
|   ├─── vcf5.2.0.0  // all configuration files for vcf 5.2.0.0. There will be a separate vcf{{version}}
directory likewise for each version of VCF that LOC-A supports
|   |   ├─── default       // default configuration file for VCF
│   │   │   ├─── input.yml         // VCF cloud configuration parameter file
│   │   │   ├─── aci_configs.yml   // Cisco ACIi parameter file for cloud deployment
│   │   │   ├─── firewall_rule.yml // firewall configuration file
│   │   │   ├─── vcf-standard-5.2.0.0.j2   // VCF deployment template file
|   |   ├─── cwd                       // specific configuration files for cwd of a vcf cloud
│   │   │   ├───  inventory-device-allocate.yml     // VCF cloud device allocation planning file
│   │   │   ├───  inventory-vm-create.yml           // VCF cloud vm allocation planning file
│   │   │   ├───  cwd-nsxt-config.j2        // NSXT configuration template file
|   |   ├─── spd                       // specific configuration files for SPD of a VCF cloud
│   │   │   ├───  inventory-device-allocate.yml   // cloud device allocation planning file
│   │   │   ├───  inventory-vm-create.yml         // cloud vm allocation planning file
│   │   │   ├─── spd-nsxt-config.j2  // NSXT configuration template file
|   |   ├─── edge                      // specific configuration files for OnPrem CWD of a VCF cloud
│   │   │   ├─── firewall_rule.yml // firewall configuration file
│   │   │   ├─── edge_aci_config.yml   // Cisco ACIi parameter file for cloud deployment
│   │   │   ├─── nsxt-config.j2   // NSXT configuration template file
│   │   │   ├───  real-site   // cloud device allocation planning file
│   │   │   │   ├───  edge_vm_site1.yml   // stretch VCF cloud vm allocation planning file
│   │   │   │   ├───  edge_vm_site1_nonstretch.yml// non-stretch VCF cloud vm allocation planning
file
│   │   │   │   ├───  init
│   │   │   │   │   ├───  edge_base_site1.j2  // netbox initialization file
│   │   │   │   ├───  onboard
│   │   │   │   │   ├───  edge_rack_site1.j2 // sitea netbox devices onboard files
│   │   │   │   │   ├───  edge_rack_siteb.j2 // siteb netbox devices onboard files
│   │   │   │   ├───  allocate
│   |   │   │   │   ├──── edge_allocate_site1.yml   // sitea VCF cloud device allocation planning file
│   |   │   │   │   ├──── edge_allocate_siteb.yml   // siteb VCF cloud device allocation planning file

|   ├──── vra8.10.0        // all configuration files for vra. There will be a separate vra{{version}} directory
likewise for each version of VRA that LOC-A supports
|   |   ├──── default
│   |   │   ├──── suit.yml           // vcf cloud configuration parameter file
│   |   │   ├──── vrealize_suite_vms.yml      // vra vm allocation planning file
│   |   │   ├──── vrealize_vrni_vms.yml       // vrni vm allocation planning file
|   |   ├──── edge
│   |   │   ├──── vrealize_suite_vms.yml      // onprem vra vm allocation planning file
│   |   │   ├──── vrealize_vrni_vms.yml       // onprem vrni vm allocation planning file
```

- Review VMware documentation

  LOC-A for VCF is based upon VMware Cloud Foundation version 4.x/5.x. Before starting the LOC-A for VCF installation process, it is important to review the VMware VCF documentation and ensure that all of the prerequisites are met VMware VCF production deployment doc.

- Make sure an L3 router exists in the customer environment. This can be either a physical router for the production environment, or a software router like pfSense in the development environment. The L3 router is usually configured manually to provide network routing for the following subnets in the table. Some subnets and component can be combined as one:

Subnet/component	**Accessible to:**
Bootstrap LOC-A services	Bootstrap LOC-A services, Bootstrap Runner, SPD LOC-A services
SPD LOC-A services	Bootstrap Runner, Bootstrap LOC-A services, SPD LOC-A services
CWD uplinks (ccm_tra)	Bootstrap LOC-A services, Bootstrap Runner, SPD LOC-A services, SPD nodes
SPD vRealize uplink 1	SPD LOC-A services, SPD nodes
SPD vRealize uplink 2	SPD LOC-A services, SPD nodes
SPD vmanagement	Bootstrap LOC-A services, Bootstrap Runner, SPD LOC-A services, SPD nodes
SPD Witness	Bootstrap LOC-A services, SPD LOC-A services, SPD nodes
SPD VSAN	Bootstrap LOC-A services, SPD LOC-A services, SPD nodes
SPD VMotion	Bootstrap LOC-A services, SPD LOC-A services, SPD nodes
SPD VTEP	Bootstrap LOC-A services, SPD LOC-A services, SPD nodes
AD DNS server	Bootstrap LOC-A services, Bootstrap Runner, SPD LOC-A services, SPD nodes
NTP server	Bootstrap LOC-A services, Bootstrap Runner, SPD LOC-A services, SPD nodes
OnPrem CWD vRealize uplink 1	SPD LOC-A services by VPN, MGMT GW, OnPrem CWD nodes
OnPrem CWD vRealize uplink 2	SPD LOC-A services by VPN, MGMT GW, OnPrem CWD nodes

OnPrem CWD vmanagement	SPD LOC-A services by VPN, MGMT GW, OnPrem CWD nodes
OnPrem CWD Witness	MGMT GW, OnPrem CWD nodes
OnPrem CWD VSAN	SPD LOC-A services by VPN, MGMT GW, OnPrem CWD nodes
OnPrem CWD VMotion	SPD LOC-A services by VPN, MGMT GW, OnPrem CWD nodes
OnPrem CWD VTEP	SPD LOC-A services by VPN, MGMT GW, OnPrem CWD nodes

Table 6: External L3 Routers configuration

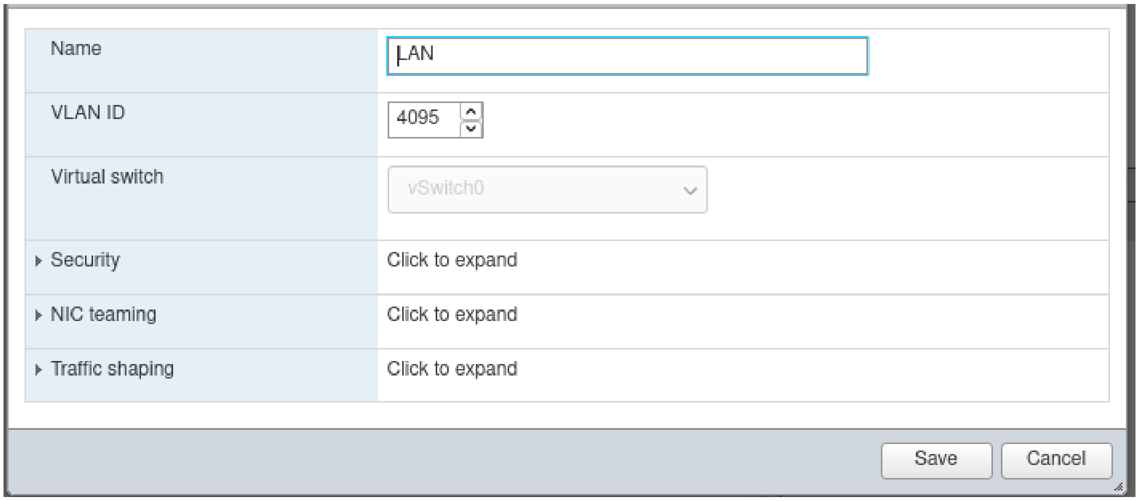
Set up the LOC-A Bootstrap environment

Complete the following steps to set up the bootstrap environment:

1. Configure the XCC management IP address of the Bootstrap node (manual).

2. Install ESXi on the Bootstrap node (manual):
   Note: It is recommended to install the operating system on the M.2 drive (Raid 1)
   a) Log in to the XCC on the Bootstrap node:
      https://<bootstrap xcc ipaddress>
      UserID: *<BMC username>* Password: *<BMC password>*
   b) Launch the Remote Console in the web interface.
   c) Get the ESXi image from the USB drive. Then upload and install ESXi via the Remote Console.
   d) Configure the Management network for IPv4. For example:
      - IPv4 Address: *10.0.0.4 (example)*
      - Subnet Mask: *255.255.254.0*
      - Default Gateway: *10.0.0.1 (example)*

3. Configure the Bootstrap network (manual):
   a) Make sure any onboard NICs are disabled on the nodes and only two 25GB NICs are enabled in the BIOS.
   b) Log in to the Bootstrap operating system::
      https:// <bootstrap exsi ipaddress >
   c) Configure the default standard virtual switch “vSwitch0” as a LAN switch. Add the following port s:
      - Name: LAN
      - VLAN ID: 4095
      - Virtual switch: vSwitch0



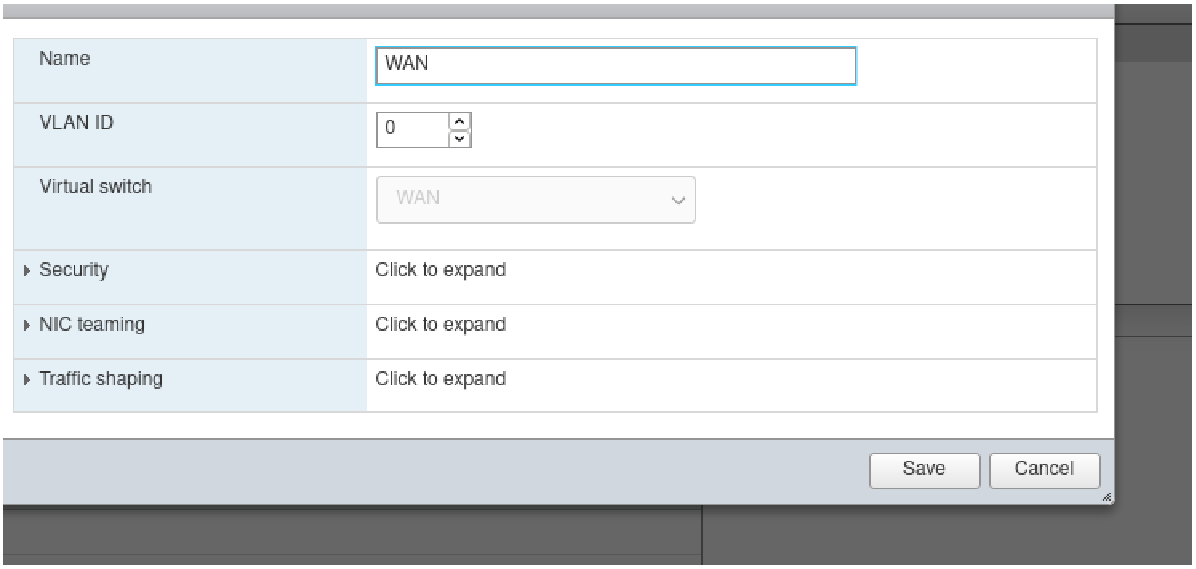
Name	LAN
VLAN ID	4095
Virtual switch	vSwitch0
▸ Security	Click to expand
▸ NIC teaming	Click to expand
▸ Traffic shaping	Click to expand

Figure 7: LAN Port Group on Bootstrap

This port group will be used to communicate to the SPD cloud data network in trunk mode.

d) Add a standard virtual switch named "WAN." Add the following port group:

- Name: WAN
- VLAN ID: 0
- Virtual switch: WAN



Name	WAN
VLAN ID	0
Virtual switch	WAN
Security	Click to expand
NIC teaming	Click to expand
Traffic shaping	Click to expand

Figure 8: WAN Port Group on Bootstrap

This port group is used for the uplink connection for Bootstrap access.

4. Install vCenter on the Bootstrap node (manual):

a) For information about installing vCenter, see the VMware vCenter deployment doc.

b) Configure the network for the installed vCenter instance:

- Network: LAN
- IP Version: IPv4
- IP assignment: static
- IP address: *10.0.0.100 (example)*

c) Ping the Bootstrap operating system IP address on vCenter and make sure that it is reachable (it on the same VLAN as the host).

5. Deploy the “Runner VM” in vCenter on the Bootstrap node and configure the WAN interface.
The Runner VM is designed as LOC-A file server, code repository, and update engine to make Boostrap installation and SPD deployment easier. It stores all LOC-A deployment resources and act as Git server when there is no internet access in the customer environment.

It includes Ansible, tower-cli and AWX command-line tools as the environment for Ansible playbook execution. The major components installed are:

- awx, version: 24.0
- Ansible, version: 2.15.12
- Python, version: 3.9.18

a) Log in to the vCenter instance:
https:// <bootstrap vcenter ipaddress>

b) Launch a new Virtual Machine.

c) Select the port group "VLAN" and make sure it can connect to the customer’s L3 router.

6. Configure the “Runner VM” (manual)
   a) Log in to the vCenter instance:
      https:// <bootstrap vcenter ipaddress>
   b) Edit the settings for the Runner VM to add a Network Adapter and select the LAN.
   c) Log in to the Runner VM and configure the LAN interface as:
      - IPADDR=*10.0.0.150 (example)*
      - NETMASK=255.255.254.0
      - GATEWAY=*10.0.0.1 (example)*
      - DNS1=*10.0.0.10 (example)*

      Configure the VLAN interface as:

      - IPADDR=192.168.1.2 *(example)*
      - NETMASK=255.255.255.0 *(example)*
      - GATEWAY=192.168.1.1 *(example)*
   d) Establish an SSH session to the Runner VM and create the following file folders:
      - /opt/loc/configs
      - /opt/loc/loc
      - /opt/loc/files
   e) Copy the appropriate LOC-A packages to the newly created folders.

7. **Optional:** An NTP server is required for SPD deployment that can be accessed by the Bootstrap node and the SPD via a router. The NTP server can be either an external one provided by the customer or a VM created on the Bootstrap system. Create a VM and enable the NTP service if necessary.

   Optional: If the customer has their own NTP, configure accordingly, and make sure the Runner VM can access it. The NTP IP will be used when preparing the SPD configuration files.

8. **Optional:** An AD DNS server is required that can be accessed by the Bootstrap and SPD via a router. The DNS server can be either an external one provided by customer environment, or via a VM created on the Bootstrap system. Create a Windows VM and enables AD DNS services with proper domain configured if you need.
   Optional: If the customer has their own DNS server, then configure and make sure the Runner VM can access it. The DNS IP will be used when preparing for SPD configuration files.
9. Prepare the configuration files for the Bootstrap node. Configure the vCenter information in *configs/loc_service_bootstrap_input.yml*:

```
loc:
  loc_source_path: /opt/loc/loc              #The directory of the loc code being used
  configs_source_path: /opt/loc/configs  #The directory of the configuration file used
  vcenter:
    hostname: "10.0.0.100"               # example, vcenter ip(domain) used to deploy bootstrap
    user: "administrator@vsphere.local"      #The username of vcenter
    pass: "<replace with your password>"      # example, the password of vcenter you installed
    datacenter: "Bootstrap"                 # example, The target datacenter of vcenter
    datastore: "datastore"                 # example, The target datastore used by bootstrap vm
    resource_pool: "xx_pool"               # example, The target resource pool used by bootstrap vm
    switch_name: "vSwitch0"                # example, The switch name on the esxi connected to
vcenter(This parameter is required if you create a resource pool on bootstrap)
    hosts: "10.0.0.4"                  # example, The ESXi host connected to vCenter (This parameter is
required if you create a resource pool on bootstrap.)
```

You will also need to configure the Bootstrap LOC-A service root password as well as the IP addresses and gateway information for each LOC-A service component expected on the Bootstrap node. Allocate 4 IP addresses for LOC-A services that can be accessed to the Runner via customer’s L3 router and provide proper gateway information.

10. Deploy and configure the Bootstrap service VMs:
    a) Log in to the Runner VM via SSH, and make sure that the Runner VM can access the Bootstrap vCenter instance.
    b) Go to the /opt/loc/loc directory and run the following command:

       ```
       ansible-playbook loc-service-deploy.yml -e "deploy_mode=bootstrap"
       ```

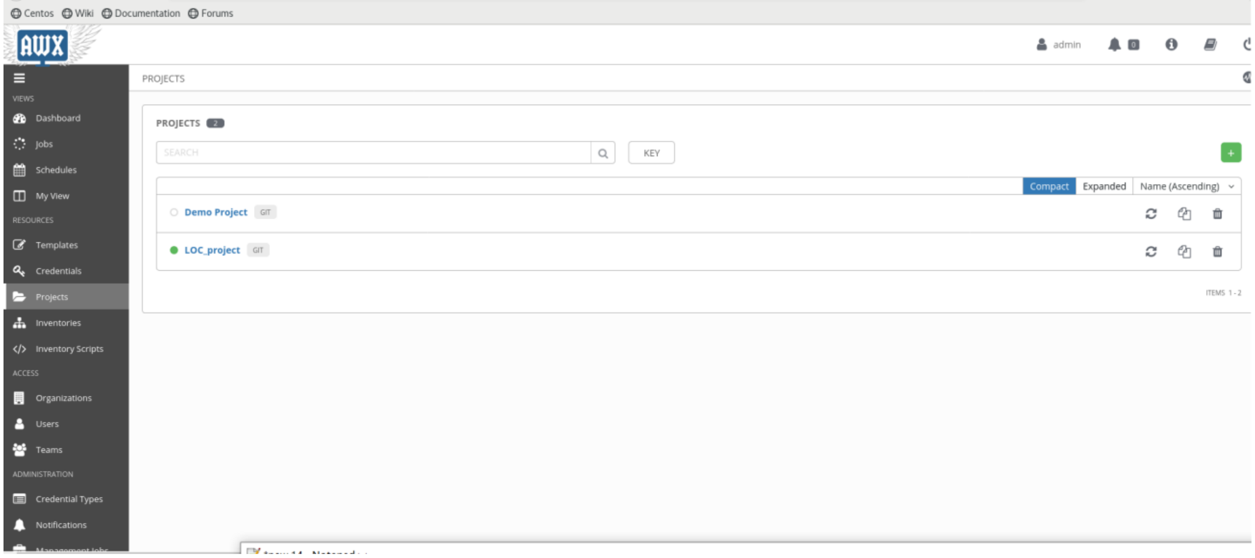
       The playbook automatically deploys the following service VMs on the Bootstrap instance:
       - LRS (Gitlab)
       - LMS (LXCA)
       - LCS (AWX)
       - LIS (NetBox)
    c) Run the following command to configure the LOC-A services on Bootstrap:

       ```
       ansible-playbook loc-bootstrap-service-config.yml
       ```

       The playbook automatically configures the Gitlab, AWX and LXCA services. It performs the following functions:
       1) Copies code from /opt/loc/loc on the Runner VM to Bootstrap Gitlab’s local lenovo open cloud repository (loc).
       2) Copies configuration files from /opt/loc/configs on Runner VM to Bootstrap Gitlab’s configuration repository (configs). The files in the configuration repository are automatically encrypted with the vault password defined in *configs/loc_service_bootstrap_input.yml* to keep sensitive data like passwords safe. Users can use ansible-vault to view or modify them in the future.

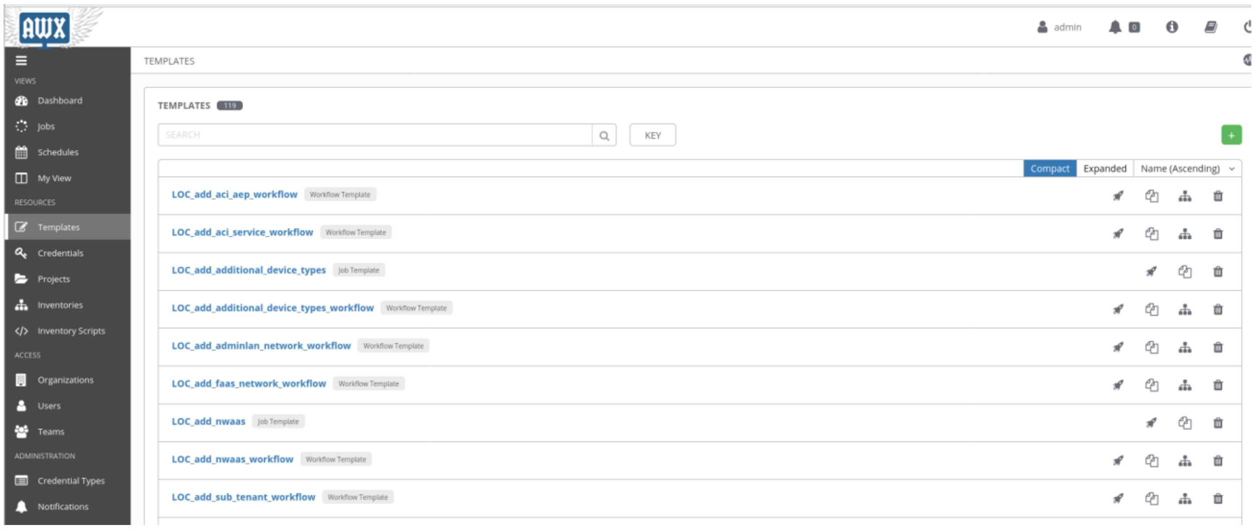
3) Creates a file server on the Bootstrap Gitlab with port: 9000, protocol: http. Then, it copies images from /opt/loc/files on the Runner VM to the file server.
4) Loads predefined VCF automation workflows to AWX.
5) Applies pre-defined patches for LXCA.
6) Imports firmware payload files and policy files for LXCA.
7) Loads LXCA Quantity License, operating system image files, and pre-defined server patterns in LXCA.
8) Automatically configures LXCA to enable VLAN mode in operating system installation.

11. At this point, the Bootstrap node is ready to deploy the Service Provider Domain (SPD) and support further automation steps. If necessary, validate that LOC-A services on the Bootstrap are working properly, using the following steps:
    a) Log in to the AWX Web GUI or CLI. Check that the LOC project is created, and that playbooks and workflows are generated. For example:



○ Demo Project	GIT
● LOC_project	GIT

Figure 9: AWX projects with LOC-A



LOC_add_aci_aep_workflow	Workflow Template
LOC_add_aci_service_workflow	Workflow Template
LOC_add_additional_device_types	Job Template
LOC_add_additional_device_types_workflow	Workflow Template
LOC_add_adminlan_network_workflow	Workflow Template
LOC_add_faas_network_workflow	Workflow Template
LOC_add_nwaas	Job Template
LOC_add_nwaas_workflow	Workflow Template
LOC_add_sub_tenant_workflow	Workflow Template

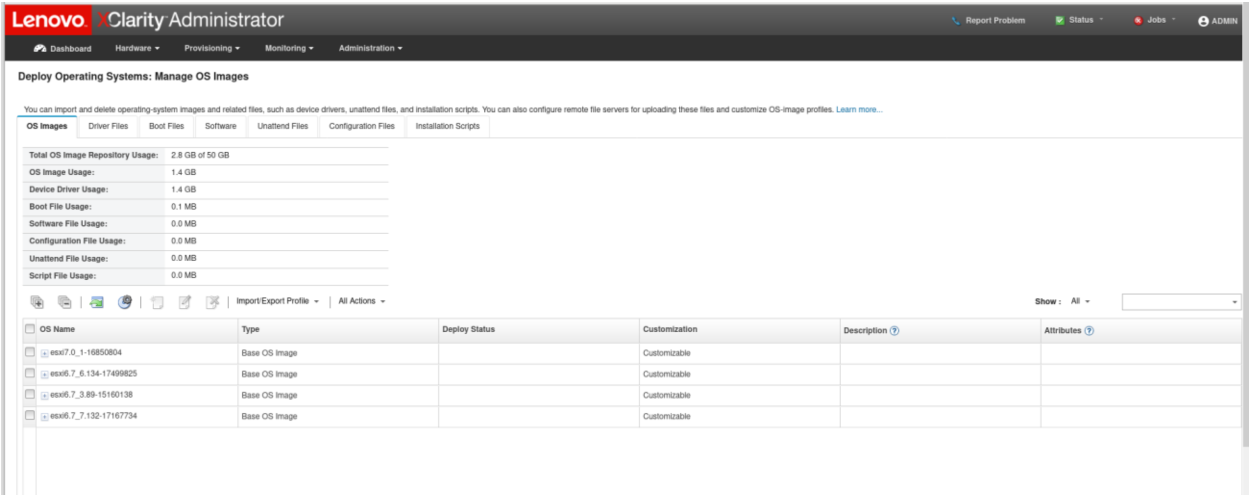
Figure 10: AWX Workflow and Job Templates of LOC-A

b) Log in to AWX host vis SSH. Check that it is accessible to the other LOC-A services via SSH, and that it is accessible to the external DNS server and gateway.

c) Log in to the Gitlab Web interface. Check that two repositories are created: loc and configs, with playbooks and configuration files loaded.

d) Log in to the NetBox Web interface and check that the system is clean (no clusters or devices are loaded).

e) Log in to the LXCA Web interface. Check that the operating system image and server patterns are preloaded, and that VLAN mode is enabled in the OS deployment Global Settings.



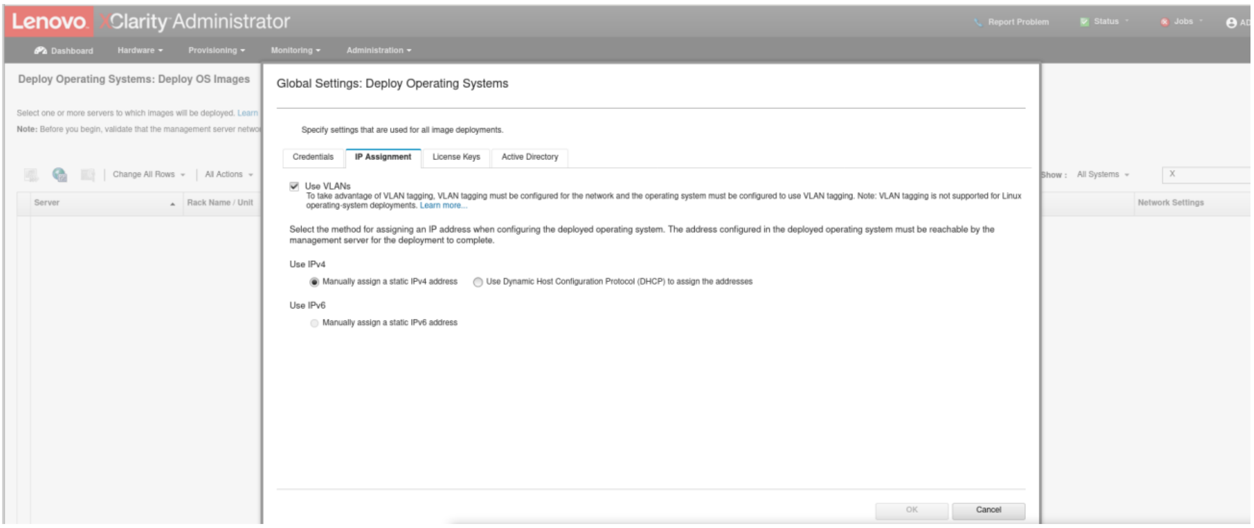
Deploy Operating Systems: Manage OS Images

You can import and delete operating-system images and related files, such as device drivers, unattend files, and installation scripts. You can also configure remote file servers for uploading these files and customize OS-image profiles. Learn more...

Total OS Image Repository Usage:	2.8 GB of 50 GB
OS Image Usage:	1.4 GB
Device Driver Usage:	1.4 GB
Boot File Usage:	0.1 MB
Software File Usage:	0.0 MB
Configuration File Usage:	0.0 MB
Unattend File Usage:	0.0 MB
Script File Usage:	0.0 MB

OS Name	Type	Deploy Status	Customization	Description	Attributes
esxi7.0_1-16850804	Base OS Image		Customizable		
esxi6.7_6.134-17499825	Base OS Image		Customizable		
esxi6.7_3.89-15160138	Base OS Image		Customizable		
esxi6.7_7.132-17167734	Base OS Image		Customizable		

Figure 11: LXCA OS profiles



Deploy Operating Systems: Deploy OS Images

Select one or more servers to which images will be deployed. Learn

Note: Before you begin, validate that the management server netwo

Global Settings: Deploy Operating Systems

Specify settings that are used for all image deployments.

☑ Use VLANs
To take advantage of VLAN tagging, VLAN tagging must be configured for the network and the operating system must be configured to use VLAN tagging. Note: VLAN tagging is not supported for Linux operating-system deployments. Learn more...

Select the method for assigning an IP address when configuring the deployed operating system. The address configured in the deployed operating system must be reachable by the management server for the deployment to complete.

Use IPv4

● Manually assign a static IPv4 address ○ Use Dynamic Host Configuration Protocol (DHCP) to assign the addresses

Use IPv6

○ Manually assign a static IPv6 address

Figure 12: LXCA global settings

For more information, refer to Reference: LOC-A Services and Operational Guide for details on how to log in and operate LOC-A services.

LOC-A SPD (Service Provider Domain) and LOC-A services deployment

Complete the following steps to set up the SPD and deploy LOC-A services:

1. Make sure your SPD nodes are properly wired to the switches according to Figure **6**: Wiring Diagram.
2. Configure the external L3 router to add a route for the SPD networks and SPD LOC-A services network according to Table **6**: External L3 Routers configuration.

   Note:

   a) Make sure your subnet prefixes of SPD networks are contained in the same master prefix, which can be defined in *vcf{{version}}/spd/inventory-device-allocate.yml* configuration file, or can be passed as extra variable during SPD deployment.

   b) You should also set up the firewall rules for each created virtual interface/route to forward IPv4 and IPv6 addresses in the necessary protocol.
3. Make sure the AVN BGP neighbor is available and configure interfaces to connect to SPD Edge Services Gateways (ESGs).
4. Follow the inline instructions in the files to update the following configuration files on the Bootstrap LRS configuration repository to make it align to your datacenter metadata:

Filename	Comment
netbox_initialization.yml	• Update expected sites information. • Perform VLAN planning of the datacenter according to Table **4**: VLAN planning for Datacenter. • Add additional device types if needed. • Update global prefixes to reflect reality.
vcf{{version}}/default/input.yml	• Update expected sites information. • Update BMC and switch information. • Update the NTP server IP address. • Update the AD DNS server IP address and credentials • Update Domain information that aligns with DNS entries • **Optional:** Update default credentials and specs for VCF deployment • Update BGP and BGP neighbours configurations for SPD • **Optional:** Update server config patterns to apply, OS images to use, target device to install OS, and firmware policies to apply if you need to change the default settings • **Optional:** Update custom static DNS entries to be configured on DNS server.
loc_service_input.yml	• Update the SPD LOC-A service root password.
devices/onboard_device.sample	• Follow the instructions in this example file to create a new device file (such as rack1_devices.yml) under this directory to match your SPD and CWD nodes in the rack. LOC-A will onboard the rack device file automatically in the following operations, and use them as the hardware resources for VCF deployment
vra{{version}}/default/suit.yml	• **Optional:** Update the default license/credential information for VRealize Suite deployment
vcf{{version}}/spd/inventory-device-allocate.yml	• Update the tenant prefixes for SPD network to reflect the environment. • **Optional:** Update the default master_prefix if you prefer not to pass the variable in the SPD deployment workflow.
vcf{{version}}/spd/inventory-vm-create.yml	• Update vm_domain to align with DNS entries.

vcf{{version}}/spd/spd-nsxt-config.j2	• NSXT configuration file, add for VCF version >= 4.

Table 7: Configuration file required for SPD deployment

Note: Backup your /opt/loc/configs folder regularly because it contains key environment data in case of an update or reconstruction of the SPD.

5. LOC-A supports upgrading server firmware when a CWD is deployed. See "[Firmware upgrade for servers]" to enable firmware updates with alternative firmware policies.

6. Log in to the Bootstrap AWX, and run the following workflows:

   "LOC_spd_deployment_workflow"(with the following variables):

```
firmware_update: false
netbox_device_config_file: rack1_device.yml
site: shzja
spdData:
  jobId: spddeploy
  noOfHosts: 4
cwdData:
  cwdCredentials:
    nem_admin:
      userAccount: admin
      password: <replace with your password>
    esx_root:
      userAccount: root
      password: <replace with your password>
    ncc_admin:
      password: <replace with your password>
    nmg_admin:
      userAccount: admin
      password: <replace with your password>
    vce_admin:
      userAccount: administrator@vsphere.local
      password: <replace with your password>
    psc_root:
      password: <replace with your password>
    vce_root:
      password: <replace with your password>
    sdc_superuser:
      userAccount: <sdc_superuser_name>
      password: <replace with your password>
    sdc_api_admin:
      userAccount: admin
      password: <replace with your password>
    sdc_root:
      userAccount: root
      password: <replace with your password>
    vli_admin:
      password: <replace with your password>
    vli_root:
      password: <replace with your password>
  licenses:
    VS6-EPL-C:
      lics:
        - '<replace with your licenses>'
    VR-LIS4-CPU-C:
      lics:
        - '<replace with your licenses>'
    NX-DC-EPL-C:
      lics:
        - '<replace with your licenses>'
    CF-SDDC-MAC-C:
      lics:
        - '<replace with your licenses>'
    VCS6-STD-C:
      lics:
        - '<replace with your licenses>'
    ST6-EN-C:
      lics:
        - '<replace with your licenses>'
stretched: false
tenant_type: spd
workflow_operation: cluster_deploy
```

Note:

- The *site* variable needs to align with what you specified in the configuration files.
- If you set the *stretched* variable to true, make sure thatyou have defined site B metadata in the configuration files, and have the proper switches and servers in place at site B.
- If you don't pass in the *credentials* or *licenses* variables, the default values in *vcf{{version}}/default/input.yml* will be used.
- The *netbox_device_config_file* variable is the filename of the device file you created in configuration repository for your SPD and CWD devices in the datacenter.

For more detailed information of LOC-A workflow specification, please refer to document: "LOC-A API Guide".

7. After the SPD is deployed and LOC-A services are installed and configured, the system is ready for further CWD automation deployment.

How to update LOC-A Service environment

Professional Services needs to update LOC-A services, code, or configuration files on the SPD when there is a need to fix LOC-A bugs, or to enable new features. Professional Service engineers can perform an automatic LOC-A service upgrade via the Runner VM. There are two types of scenarios where SPD LOC-A services are upgraded:

- Major upgrade

  A major upgrade is performed when the LOC-A software component version needs to be upgraded, or the base operating system of the VM appliances need to be upgraded. Usually, a major upgrade is required when LOC-A releases a new release.

  During major upgrade, LOC-A will automatically perform the following steps for the LOC-A services:

  - Back up the existing data on service.
  - Remove the application from the VM host.
  - Decommission the old VM from vCenter.

  - Deploy a new VM using the generic VM template with target IP address and name.
  - Deploy the container instances for the service applications.
  - Restore data for the service.

- Minor upgrade
  A minor upgrade is performed when only configuration changes are required (a LOC-A software component upgrade is not needed). A minor upgrade can be run on demand within one LOC-A release across different builds. The minor upgrade process will complete the following tasks according to the extra control variables (if needed):
  - Update the workflow and job templates or configuration changes for AWX.
  - Update the virtual environment for AWX.
  - Update global metadata for NetBox.
  - Update the LOC-A code for the GitLab repository.
  - Update the LXCA server patterns, operating system profiles, and base operating system.
  - Apply patches to fix LXCA issues.
  - Update the file packages on the file server.

Complete the following steps to upgrade LOC-A services:

1. Get latest release package from LOC-A team.

The released packages include two code repositories and an images/files folder.

Copy the file bundles into the /opt/loc/files directory of your runner VM.

Copy your configuration and local source code into any location of your working directory. The file trees on the Runner VM should look like:

```
/opt
  |---  loc
    |---  files      //file bundles including isos, fw payloads, vcf software bundles, etc.
/your_working_directory/
    |---  configs             //config_space repo
    |---  loc                 // major code repo
```

2. Update *configs/ loc_service_input.yml* . The configuration file provides the loc.service_password, loc.service_vmnet, and the loc.oob_vmnet variables, which are used in upgrade process. Before you upgrade the LOC services, make sure these variables comply with your existing SPD LOC service.

3. Run runner-loc-service-major-upgrade.yml with the following parameters:

Parameter	**Required/Optional**	**Descriptions**
target_netbox_url	Required	The NetBox IP address of the SPD LOC services. The SPD information will be retrieved from NetBox.
vcenter_pass	Required	The password for the vCenter of your SPD.

Table 8: LOC-A major upgrade playbook parameters

Example Usage:

```
ansible-playbook runner-loc-service-major-upgrade.yml -e "target_netbox_url=https://10.0.0.66
vcenter_pass=<replace with your password>"
```

4. Prepare extra LXCA patterns, operating system images, and patches for your additional devices on the runner VM (optional for customization)

Path	Description
/opt/loc/files/servers/lxca/{{ target_lxca_version }}/firmwares/	Pre-defined firmware payload files to be imported into LXCA.
/opt/loc/files/servers/lxca/{{ target_lxca_version}}/patches/	Patches required for LXCA (current version 3.4.5)
/opt/loc/files/servers/lxca/{{ target_lxca_version}}/patterns/	Pre-defined server pattern files to be imported into LXCA. If you have extra device types to support, add the server patterns here.
/opt/loc/files/servers/lxca/{{ target_lxca_version}}/policies/	Pre-defined firmware policy to be imported into LXCA.
/opt/loc/files/servers/OS/	Operating system files that need to be imported into LXCA.
/opt/loc/files/servers/profiles/	Operating system profile files that need to be imported into LXCA.
/opt/loc/files/servers/lxca/{{ target_lxca_version}}/license/	LXCA license file that needs to be applied to the LXCA instance.

5. Run runner-loc-service-minor-upgrade.yml with the following parameters

Parameter	Required/Optional	Descriptions
target_netbox_url	Required	The NetBox IP information of the existing loc_services. Information about the loc_services will be retrieved from NetBox.
lxca_update	Optional	Controls whether to configure LXCA to load expected operating system profiles, images, and settings, as well as server patterns, import licenses, load, and apply patches. The values can be true or false. Default value: true **Note:** To update new ISO images or to apply additional LXCA patches, set the update_fileserver to true as well. All ISO files and patches should be copied from /opt/loc/files/ in your Runner VM to the file server before LXCA is updated.
lxca_patches_update	Optional	Controls whether to apply LXCA patches if update_lxca is enabled

Parameter	Required/Optional	Descriptions
		The values can be true or false. Default value: true
gitlab_update	Optional	Controls whether to recreate the code repository with the new codebase. **Note:** This will purge the existing code repository of *loc* and re-generate it. Do **NOT** enable this parameter if you are using other alternatives, such as repository mirroring to update code. config_space (configs/) on SPD gitlab will NOT be updated with this option enabled. The values can be true or false. Default value: false
files_update	Optional	Controls whether to update the SPD file server with the contents of /opt/loc/files in the Runner VM, which can be time consuming. The values can be true or false. Default value: true
awx_venvs_update	Optional	Controsl whether to upgrade the AWX environment. The , the AWX ENV file to be updated should be uploaded to the Runner VM as /opt/loc/files/loc/extra/lnvenv3.tar.gz before running the script. The values can be true or false. Default value: true
awx_workflow_update	Optional	Controls whether to purge and recreate LOC AWX workflows and jobs templates. The AWX workflow and job definition file is located here: /*your_working_directory*/loc/var_files/workflow_cwd.yml. **Important:** This will delete the existing workflow and job templates on SPD AWX and recreate new ones. Set this parameter to false if you decide to keep your customized change. The values can be true or false. Default value: true
netbox_data_update	Optional	Controls whether to apply the NetBox global metadata upgrade to be compatible to current release. **Note:** This will usually only upgrade global or device metadata according to changes across versions. For per-tenant metadata of each tenant CWD, run LOC_per_tenant_upgrade job on demand. The values can be true or false. Default value: true
logging_update	Optional	Controls whether to update the installation and configuration of the ELK and filebeat agents.

Parameter	Required/Optional	Descriptions
		The values can be true or false. Default value: true

Table 9: LOC-A Minor upgrade playbook parameters

Example Usage:

```
ansible-playbook runner-loc-service-minor-upgrade.yml -e "target_netbox_url=https://10.0.0.66
files_update=false awx_workflow_update=false"
```

It is recommended to use the default options for a LOC-A services minor upgrade to guarantee that all LOC-A services are properly updated.

After all steps are performed, LOC-A services on the SPD are up-to-date and are ready for further cloud automation deployment.

6. update LOC-A Service environment on MGMT GW
   The LOC-A Service on MGMR GW only contains NFS, lDS and LMS, run LOC_per_tenant_upgrade in SPD AWX to do update.
   Parameter to do major update for LDS:

   tenant_name: c007

   upgrade_tags:

   - confluent_major

   Parameter to do minor update for LDS:

   tenant_name: c007

   upgrade_tags:

   - confluent_minor

   Parameter to do major update for LMS:

   tenant_name: c007

   upgrade_tags:

   - lxca_major

   Parameter to do major update for LDS:

   tenant_name: c007

   upgrade_tags:

   - lxca_minor

LOC-A cloud automations features

LOC-A CWD (Customer Workload Domain) deployment

Introduction

LOC-A supports the automatic deployment of the customer workload domain with VMWare Cloud Foundation in the consolidated deployment model. VMware requires 4 to 64 nodes for the primary VCF cluster that will be deployed initially for the CWD.

Usage

1. Ensure the following prerequisites for CWD deployment are met:
    - Both the bootstrap node and the SPD nodes can connect to the switch's ports with config “ccm_tra” portgroup and VLAN IDs of loc-service.
    - After the command cwd netbox onboard, verify that a virtual interface is created.
    - The SPD environment and LOC-A services have been deployed successfully.
    - CWD nodes are connected to the switch, and the nodes are healthy. If necessary, manually the XCC for their health status. The status should be "healthy" without any errors.
    - Any onboard NICs are disabled on CWD nodes and only the two reserved 25GB NICs are enabled in BIOS.
    - loc-services and ccm_tra network port groups have been created.

2. Update the rack devices configuration file based on the example file.
3. Log in to the SPD AWX, and run the following workflows:

    “LOC_rack_device_onboard_workflow” (with the following variables)

    firmware_update: true

    netbox_device_config_file: rackX_device.yml

    The proper rack devices should now be ready for CWD deployment.

4. Log in to the SPD AWX, and run the following workflows

    “LOC_cwd_create_metadata_workflow” (with the following variables)

    stretched: false
    cwdData:
    cwdIpRange: 10.50.0.0/20
    jobId: cwd222222-1054-43c8-9c69-e691239b5d5d
    noOfHosts: 4
    site: shzja
    tenant_devices_role: vx-node-d-v1

5. The tenant CWD is now configured. Get the tenant name from NetBox by searching the description with the jobId name you specified. Eg. the tenant name is ‘c006’ in NetBox.
6. Configure the route for the edge network before starting the CWD deployment manually.
    a. Go to NetBox, check for the NAT child prefix for this CWD, and the cwd_ccm_tra_ip which is the IP address of the CWD NSX management edge (VM with prefix nem) on the ccm_tra interface.
    b. Create the route for this CWD edge device on the external physical or software router:

        Route gateway: cwd_ccm_tra_ip
        Virtual Interface: ccm_tra (which is created before SPD deployment)
        Network: nat child prefix

7. Log in to the SPD AWX, and run the following workflows:

    “LOC_cwd_deployment_workflow” (with the following variables)

```
cwdData:
  cwdCredentials:
    esx_root:
      userAccount: root
      password: <replace with your password>
  cwdId: c006
  jobId: cwd222222-1054-43c8-9c69-e691239b5d5d
  mgmtNetwork:
    adminLan:
      bgpFilters:
        - 30.0.0.0/20
      ipSubNet:
      - 10.0.0.0/24
      - 20.0.0.0/24
      - 10.193.254.0/24
    bgp:
      bgpAvnAsn: '65657'
      bgpEdgeAsn: '65656'
      bgpPassphrase: <replace with your BGP passphrase>
      bgpPeerAsn: '64515'
      bgpPeerIp:
        - 10.8.11.1
        - 10.8.11.65
        - 10.80.10.200
        - 10.99.0.100
site: shzja
```

LOC-A supports server firmware upgrades when CWD is deployed. See “Firmware upgrade for servers” for more information

LOC-A CWD Twin-core deployment

Introduction

LOC-A supports the deployment of a CWD in twin-core mode. The VCF cluster is deployed across two sites in a high availability environment.

If the VCF instance is specified as a stretched type, it will be deployed in twin-core mode. If the VCF instance is specified as a non-stretched type, it will be deployed in normal single-core mode. The two sites serve as the resource pools for the two availability zones (AZs) of the VCF clusters and the hardware between the two AZs is mirrored. If there is any failure with one of the AZs, the cloud can continue to run.

Note: One Region supports two AZs.

The network topology between the two AZs needs to follow a spine-leaf configuration mirrored in each AZ with an Inter-Pod Connection. LOC-A supports the automatic configuration of switches with the Cisco ACI type.

The two sites are defined as primary site and secondary site. During the twin-core CWD deployment automation, LOC-A will:

1. Perform metadata planning to automatically allocate proper resources, including server nodes, VLANs, IP prefixes, VMs, etc. from the two sites.
2. Configure the ACI APIC with the proper underlay and overlay network configurations for the VCF deployment on the primary site.
3. Perform the deployment of the stretched VCF cloud.

What is out of scope for this feature:

- Conversion of a non-stretched cluster to a stretched cluster
- Conversion of a stretched cluster to a non-stretched cluster

Usage

1. Ensure the following prerequisites are met:
   - The spine-leaf network topology is set up, and there is underlay network connectivity between the two sites via the Cisco spine and leaf switches. The Cisco ACI APIC and switches have been physically configured and are available via a configured IP address.
   - The LOC-A SPD is deployed as a stretched cluster.
   - The VPN connection to the witness is configured manually, persistent, and is accessible.
   - There are enough devices configured already in NetBox for both sites.

2. Log in to the SPD AWX, and run the following workflows:

   "LOC_cwd_create_metadata_workflow" (with the following variables)

```
stretched: true
cwdData:
cwdIpRange: 10.50.0.0/20
jobId: cwd222222-1054-43c8-9c69-e691239b5d5d
noOfHosts: 4
site: shzja
tenant_devices_role: vx-node-d-v1
```

3. After this is done, the tenant CWD is configured. Get the tenant name from NetBox by searching the description with the jobId name you pass. Eg. the tenant name is 'c006' in NetBox.
4. Log in to the SPD AWX, and run the following workflows:

   "LOC_cwd_deployment_workflow" (with the following variables)

```
cwdData:
 cwdCredentials:
  esx_root:
   userAccount: root
   password: <replace with your password>
 cwdId: c006
 jobId: cwd222222-1054-43c8-9c69-e691239b5d5d
 mgmtNetwork:
  adminLan:
   bgpFilters:
    - 30.0.0.0/20
   ipSubNet:
   - 10.0.0.0/24
   - 20.0.0.0/24
   - 10.193.254.0/24
  bgp:
   bgpAvnAsn: '65657'
   bgpEdgeAsn: '65656'
   bgpPassphrase: <replace with your BGP passphrase>
   bgpPeerAsn: '64515'
   bgpPeerIp:
    - 10.8.11.1
```

```
        - 10.8.11.65
        - 10.80.10.200
        - 10.99.0.100
    site: shzja
```

"LOC_cwd_stretch_create_metadata_workflow" (with the following variables)

```
cwdData:
  cwdId: c006
  noOfHosts: 4
secondary_site: shzjb
site: shzja
```

"LOC_stretched_cwd_deployment_workflow" (with the following variables)

```
cwdData:
  cwdCredentials:
    esx_root:
      password: <replace with your password>
      userAccount: root
  cwdId: c006
  mgmtNetwork:
    adminLan:
      ipSubNet:
        - 10.0.0.0/24
        - 15.0.0.0/24
        - 10.193.254.0/24
site: shzja
secondary_site: shzjb
```

LOC-A CWD (Customer Workload Domain) expansion

Introduction

This feature allows customers to expand a cluster with additional nodes on demand. The automation playbook can add multiple nodes at the same time.

Usage

1. Ensure the following prerequisites are met:

   - After expanding the nodes into the cluster, the nodes in the cluster should not surpass the maximum allowed number of 64.
   - Ensure there are enough free nodes in Inventory status in NetBox that can be allocated. If the VCF cluster is deployed in twin-core mode (stretched cluster type), ensure there are enough free nodes for both sites. Because when you expand N nodes into a stretched cluster, N nodes will be allocated within each site and get commissioned into the cluster.

2. Log in to the SPD AWX, and run the following workflows:

   "LOC_cwd_expand_create_metadata_workflow" (with the following variables)

   ```
   expand_job_id: c008expandcluster01
   tenant_name: c008
   cluster_name: c008-shzj-cl01
   tenant_expand_node_number: 1
   ```

"LOC_cwd_expand_workflow" (with the following variables)

```
cwdData:
  cwdCredentials:
    esx_root:
      password: <replace with your password>
      userAccount: root
  cwdId: c008
  mgmtNetwork:
    adminLan:
      ipSubNet:
        - 10.0.0.0/24
        - 15.0.0.0/24
        - 10.193.254.0/24
site: shzja
secondary_site: shzjb
cluster_name: c008-shzj-cl02
```

LOC-A CWD (Customer Workload Domain) shrink

Introduction

This feature allows customers to reduce cost by removing unused hardware. LOC-A provides the capability to remove nodes from existing customer cluster automatically. The automation playbook can remove multiple nodes at the same time.

What is out of scope for this feature:

- Data on the nodes will not be stored.
- LOC-A will not move hosts into maintenance mode. They must be put into maintenance mode by the customer manually running this workflow.
- The Remove Node function will not handle capacity calculations.

Usage

1. Ensure the following prerequisites are met:

   - The nodes that need to be removed should be in the same cluster.
   - The nodes that need to remove should be set to maintenance mode manually by the customer.
   - After removing the nodes, the VCF environment should be working normally, which means that the reminding number of nodes should meet the minimum requirements for the cluster
     - Primary cluster: 4 nodes
     - Additional cluster: 3 nodes

2. Log in to the SPD AWX, and run the following workflows:

   "LOC_remove_nodes_workflow" (with the following variables)

```
cwdData:
  cwdCredentials:
    sdc_api_admin:
      password: <replace with your password>
      userAccount: admin
    vce_admin:
      userAccount: administrator@vsphere.local
      password: <replace with your password>
tenant_name: c008
job_id: cluster02removenode
cluster_name: c008-shzj-cl02
nodes:
   - esx64001shzjb
   - esx64008shzja
```

LOC-A CWD (Customer Workload Domain) removal

Introduction

This feature allows customer to release its CWD. The LOC-A automation code will:

1. Shut down all the devices in this tenant using the LXCA API.
2. Remove all of the VMs that were created for this tenant, including the VCB, witness, and VTEP edge VMs in the SPD environment if the CWD is stretched.
3. Delete DNS/AD entries for this tenant from the DNS/AD DNS server.
4. Remove the nodes interfaces' CWD VLAN configuration for this tenant from the ACI/CNOS switches.
5. Remove the CWD VLAN configuration for SPD node interfaces from the ACI/CNOS switches
6. Remove the IP/clusters for the CWD node from NetBox, put the devices back to the available devices pool, and put the VLANs to the available VLANs pool.
7. Remove cwd VLANs for the SPD nodes from NetBox.
8. Restore the tenant and set the tenant status to "in-use" to prevent this tenant name from being reused again in a short period of time.
9. Remove the tenant configuration context data.

Usage

Log in to SPD AWX, and run the following workflows:

“LOC_remove_cwd_workflow” (with the following variables)tenant_name: c001

LOC-A CWD additional cluster deployment

Introduction

VMware Cloud Foundation allows for multiple clusters to be created on an existing VCF instance. This feature allows customers to create additional clusters on an existing CWD with non-stretched type.

Usage

1. Ensure the following prerequisite is met:
   - There are enough server nodes available already in NetBox for the additional cluster. The server requirement for an additional cluster is 3 to 64 nodes.

2. Log in to the SPD AWX, and run the following workflows:

   “LOC_cwd_additional_cluster_create_metadata_workflow” (with the following variables)

```
stretched: true
cwdData:
  cwdId: c008
  jobId: c008addcluster02
  noOfHosts: 3
```

```
site: shzja
```

3. The additional cluster is configured. Get the cluster name from NetBox by searching the description with the jobId name you pass. The cluster name is usually named as “{{ tenant_name }}-{{ site }}-cl0*x*”, where *x* is the index. Cl01 is usually the initial cluster and additional clusters start from index 2.
4. Log in to the SPD AWX, and run the following workflows:

   “LOC_cwd_deploy_additional_cluster” (with the following variables)

```
stretched: true
cwdData:
  cwdCredentials:
    esx_root:
      password: <replace with your password>
      userAccount: root
  cwdId: c008
  mgmtNetwork:
    adminLan:
      ipSubNet:
        - 10.0.0.0/24
        - 15.0.0.0/24
        - 10.193.254.0/24
site: shzja
cluster_name: c008-shzj-cl02
```

LOC-A CWD twin-core additional cluster deployment

Introduction

VMware Cloud Foundation allows for multiple clusters to be created on an existing VCF instance. This feature allows customers to create additional clusters on an existing CWD with stretched type.

Usage

1. Ensure the following prerequisite is met:
   - There are enough server nodes available already in NetBox for the additional cluster for both AZs. The server requirement for an additional cluster is 3 to 64 nodes per AZ.

2. Log in to the SPD AWX, and run the following workflows:

   “LOC_cwd_additional_cluster_create_metadata_workflow” (with the following variables)

```
stretched: true
cwdData:
  cwdId: c008
  jobId: c008addcluster02
  noOfHosts: 3
site: shzja
```

3. The additional cluster is configured. Get the cluster name from NetBox by searching the description with the jobId name you pass. The cluster name is usually named as “{{ tenant_name }}-{{ site }}-cl0*x*”, where *x* is the index. Cl01 is usually the initial cluster, and additional clusters start from index 2.
4. Log in to the SPD AWX, and run the following workflows:

   “LOC_cwd_deploy_additional_cluster” (with the following variables)

```
stretched: true
cwdData:
  cwdCredentials:
    esx_root:
      password: <replace with your password>
      userAccount: root
  cwdId: c008
  mgmtNetwork:
    adminLan:
      ipSubNet:
        - 10.0.0.0/24
        - 15.0.0.0/24
        - 10.193.254.0/24
site: shzja
cluster_name: c008-shzj-cl02
```

"LOC_cwd_additional_cluster_stretch_create_metadata_workflow" (with the following variables)

```
cwdData:
  cwdId: c008
cluster_name: c008-shzj-cl02
```

"LOC_cwd_stretch_additional_cluster"(with the following variables)

```
cwdData:
  cwdCredentials:
    esx_root:
      password: <replace with your password>
      userAccount: root
  licenses:
    VS6-EPL-C:
      lics:
        - <replace with your licenses>
  cwdId: c008
  mgmtNetwork:
    adminLan:
      ipSubNet:
        - 10.0.0.0/24
        - 20.0.0.0/24
        - 10.193.254.0/24
site: shzja
secondary_site: shzjb
cluster_name: c008-shzj-cl02
```

LOC-A Virtual Infrastructure (VI) Workload Domain deployment/removal

Introduction

VMware Cloud Foundation allows for multiple clusters to be created on an existing VCF instance.

The SDDC functionality is distributed across multiple workload domains and VMware vSphere® clusters.
A workload domain, whether it is a management or virtual infrastructure domain, is a logical abstraction of compute, storage, and network capacity; it can consist of one or more clusters.

This feature allows customers to create a Virtual Infrastructure Workload Domain in an existing SPD environment.

Usage

1. Ensure the following prerequisite is met:
   - There are enough server nodes available already in NetBox. The server requirement is 3 to 64 nodes.

2. Log in to the SPD AWX, and run the following workflows:

   "LOC_wld_create_metadata_workflow" (with the following variables)

```
firmware_update: false
netbox_device_config_file: rack2_devices.yml
site: shzja
spdData:
  jobId: xlwlddeploy
  noOfHosts: 4
workflow_operation: wld_deploy
product_version: 4.2.1.0
tenant_name: s002
```

3. Log in to the SPD AWX, and run the following workflows:

   "LOC_spd_wld_deployment_workflow" (with the following variables)

```
site: shzja
spdData:
  jobId: xlwlddeploy
  noOfHosts: 4
cwdData:
  cwdCredentials:
    esx_root:
      userAccount: root
      password: <replace with your password>
    vce_admin:
      userAccount: administrator@vsphere.local
      password: <replace with your password>
    vce_root:
      password: <replace with your password>
    sdc_superuser:
      userAccount: vcf
      password: <replace with your password>
    sdc_api_admin:
      userAccount: admin
      password: <replace with your password>
    sdc_root:
      userAccount: root
      password: <replace with your password>
    nmg_admin:
      userAccount: admin
      password: <replace with your password>
tenant_name: s002
workflow_operation: wld_deploy
product_version: 4.2.1.0
```

4. Log in to the SPD AWX, and run the following workflows:

   "LOC_spd_wld_release_workflow" (with the following variables)

```
wld_name: s002wld01
cwdData:
  cwdCredentials:
    sdc_api_admin:
      password: <replace with your password>
      userAccount: admin
    sdc_root:
      password: <replace with your password>
      userAccount: root
    sdc_superuser:
      password: <replace with your password>
      userAccount: vcf
    vce_admin:
      password: <replace with your password>
      userAccount: administrator@vsphere.local
product_version: 4.2.1.0
site: shzja
spdData:
  jobId: xlWldRelease
tenant_name: s002
workflow_operation: wld_release
```

LOC-A OnPrem CWD deployment

Introduction

CWD can be deployed on the remote edge site. VCF builder will be deployed on MGMT GW. All network configurations on the edge site need to be done manually, include VCF requirement network configurations according to the CWD VLAN plan and VPN configuration. VPN configuration on the SPD edge is set by LOC_edge_nem_workflow.

Reference the LOC-A CWD deployment.

Usage

1. Must meet all the prerequisites in Datacenter CWD deployment.
2. Follow the inline instructions in the files to update the following configuration files in the SPD LRS configuration repository to make it align to your OnPrem metadata:

Filename	Comment
vcf{{version}}/edge/real-site/init/edge_base_site1.j2	• sites information. • Perform VLAN planning. • Add additional device types if needed. • Update global prefixes to reflect reality.
vcf{{version}}/edge/real-site/onboard/edge_rack_site1.j2	• Follow the instructions in this example file to create a new device template file (such as edge_rack_sitea.j2) under this directory to match your OnPrem nodes in the rack.
vcf4.4.0.0/edge/real-site/allocate/edge_allocate_site1.yml	• Update the tenant prefixes for CWD network to reflect the environment.

3. Log in to the SPD AWX, and run the following workflows, reference the API guide for detail parameters.

3.1. Run LOC_edge_site_onboard_workflow to create edge site and onboard devices of the site.
example parameters:
site: edgea
netbox_device_config_file: edge_rack_sitea.j2

3.2. Run LOC_edge_cwd_create_metadata_workflow to create meta data.

3.3. Run LOC_edge_nem_workflow to config the networks and NSXT in SPD side.

3.4. Run LOC_edge_service_onboard_workflow to deploy NFS, confluent, DHCP server, DNS on MGMT GW.

3.5. Run LOC_edge_cwd_deployment_workflow

3.6. Run LOC_edge_cwd_lxca_workflow to deploy LMS in MGMT GW.

LOC-A OnPrem CWD Twin-core deployment

Introduction

OnPrem CWD Twin-core deployment on remote site. Each site has one MGMT GW device. witness will be deployed on primary site's MGMT GW.

Reference the LOC-A CWD Twin-core deployment.

OnPrem stretch has been supported since LOC-A 2.9 R2311.

Usage

1. Follow the inline instructions in the files to update the following configuration files in the SPD LRS configuration repository to make it align to your OnPrem metadata:

Filename	**Comment**
vcf{{version}}/edge/real-site/init/edge_base_site1.j2	• sites information. • Perform VLAN planning. • Add additional device types if needed. • Update global prefixes to reflect reality.
vcf{{version}}/edge/real-site/onboard/edge_rack_site1.j2 vcf{{version}}/edge/real-site/onboard/edge_rack_siteb.j2	• Follow the instructions in this example file to create a new device template file (such as edge_rack_sitea.j2) under this directory to match your OnPrem nodes in the rack. • edge_rack_site1.j2 is the default one for primary site, edge_rack_siteb.j2 is the default one for sibling site.
vcf4.4.0.0/edge/real-site/allocate/edge_allocate_site1.yml vcf4.4.0.0/edge/real-site/allocate/edge_allocate_siteb.yml	• Update the tenant prefixes for CWD network to reflect the environment. • edge_allocate_site1.yml is the default one for primary site, edge_allocate_siteb.yml is the default one for sibling site.

2. Log in to the SPD AWX, and run the following workflows, reference the API guide for detail parameters.
   2.1. Run LOC_edge_site_onboard_workflow twice to create two edge sites and onboard devices of each site.
   example parameters for sitea:
   site: edgea
   netbox_device_config_file: edge_rack_sitea.j2
   example parameters for siteb:
   site: edgeb
   primary_site: edgea
   netbox_device_config_file: edge_rack_siteb.j2
   2.2. Run LOC_edge_cwd_create_metadata_workflow**.**

2.3. Run LOC_edge_nem_workflow to config the networks and NSXT in SPD side.
2.4. Run LOC_edge_service_onboard_workflow twice for each site to deploy NFS, confluent, DHCP server, DNS on MGMT GW.
2.5. Run LOC_edge_cwd_deployment_workflow
2.6. Run LOC_edge_cwd_stretch_create_metadata_workflow
2.7. Run LOC_edge_stretched_cwd_deployment_workflow
2.8. Run LOC_edge_cwd_lxca_workflow to deploy LMS in MGMT GW.

LOC-A OnPrem CWD (Customer Workload Domain) removal

Introduction

Remove the OnPrem CWD. Reference the LOC-A CWD removal.

Usage

Log in to SPD AWX, and run the following workflows:

“LOC_edge_remove_cwd_workflow” (with the following variables)tenant_name: c001

vRealize Suite deployment/removal/upgrade/scale

Introduction

VMware vRealize Suite is an optional software component that enables VMware day-2 operations, management, and analysis. LOC-A supports the automatic deployment or removal of a vRealize Suite.

Usage

1. Log in to the SPD AWX, and run the following workflows to deploy a vRealize Suite:

   “LOC_cwd_vra_deploy_workflow” (with the following variables)

```
install_vrops: present
cwdData:
  cwdCredentials:
    vce_admin:
      userAccount: administrator@vsphere.local
      password: <replace with your password>
    vlm_root:
      password: <replace with your password>
    vlm_admin:
      password: <replace with your password>
    vid_root:
      password: <replace with your password>
    vid_cfgadmin:
      userAccount: <replace with your username>
      password: <replace with your password>
    vop_admin:
      password: <replace with your password>
    vop_root:
      password: <replace with your password>
  licenses:
    VR19-ENT-C:
      lics:
        - <replace with your licenses>
  cwdId: c001
```

2. Log in to the SPD AWX, and run the following workflows to upgrade a vRealize Suite:

“LOC_cwd_vra_deploy_workflow” (with the following variables)

```
cwdData:
  cwdCredentials:
    vlm_admin:
      password: <replace with your password>
    vlm_root:
      password: <replace with your password>
    vid_sshuser:
      password: <replace with your password>
    vce_admin:
      userAccount: administrator@vsphere.local
      password: <replace with your password>
    sdc_api_admin:
      password: <replace with your password>
      userAccount: admin
  cwdId: c007
  licenses:
    VR-LIS4-CPU-C:
      lics:
        - <replace with your licenses>
    VR19-ENT-C:
      lics:
        - <replace with your licenses>
install_vrops: upgrade
site: edge2a
skip_upgrade_product_list:
  - vaa
  - vop
  - vrli
vra_upgrade_snapshot: false
vrealize_version: 8.4.1
```

vRealize Network Insights deployment/removal/upgrade

Introduction

VMware vRealize Network Insight (vRNI) is an optional software component that enables VCF cloud network analysis. LOC-A supports the automatic deployment or removal of vRNI.

Usage

1. Ensure the following prerequisite is met:
   - The CWD VCF is in healthy status, and the CWD vRealize Suite is already deployed.

2. Log in to the SPD AWX, and run the following workflows:

   “LOC_cwd_vrni_deploy_workflow” (with the following variables)

```
install_vrni: present
cwdData:
  cwdCredentials:
    vce_admin:
      userAccount: administrator@vsphere.local
      password: <replace with your password>
    vlm_root:
      password: <replace with your password>
    vlm_admin:
      password: <replace with your password>
    vrni_root:
      password: <replace with your password>
  cwdId: c008
```

Virtual Machine as a Service - VMaaS

Introduction

This feature allows customers to dynamically add, delete, or modify one or more VMs in NetBox and configure a VM with a specified network interfaces, VLAN roles, IP addresses, and NAT IP addresses.

Usage

1. Log in to the SPD AWX, and run the following workflows:

   “LOC_vmaas_workflow” (with the following variables)

```
cwdData:
tenant_name: c001
virtual_machines:
  - name: VMaas_test_vm
    device_role: management-vm
    state: present/modify/absent
    cluster: c001-shzj-cl01
    vcpus: 3
    site: shzja
    memory: 3000
    platform: linux
    disk:10
    interfaces:
      - name: eth0
        mode: tagged
        tagged_vlans:
          - vmanagement
        mtu: 9600
        addresses:
          - auto
        nat:
          address: auto
        prefix_role: vmanagement
```

LOC-A additional device type support

Introduction

This feature allows customer to define and add additional device types in NetBox,and configure these device types to be used to deploy the VCF cloud. With additional device type support, LOC-A can:

- Support extra server types to be configured for VCF cloud deployment.
- Support servers with more than 2 NICs, and automatically configure VLANs for the connected NIC ports.

- Support the detection of NIC port MAC addresses, Ethernet adapters, and the connectivity of the NIC ports. If more than one high-speed Ethernet adapters isinstalled, LOC-A will pick the first port of each adapter when deploying VCF (based on high-availability best practice).
- Support the mapping of the selected NICs with the proper vmnic name in ESXi for VCF deployment

Usage

1. Prepare the configuration files for additional devices type, and rack-based workload devices definition.

   To support additional node types, you must prepare the definition files. Follow the example file to define rack-based workload devices in configs/netbox_rack_workload_2.conf.

   This is an example file for rack 2 definition. You can use whatever name preferred. The file name can be passed as parameter when executing the "LOC_rack_device_onboard_workflow" workflow in the following steps.

   After all the configuration files are ready, check in them in SPD Gitlab configs repository under /

2. Create the new device type in NetBox. For example, if user needs to create a 4-NIC SR650 device type, log in to the SPD AWX, and run the following workflows:

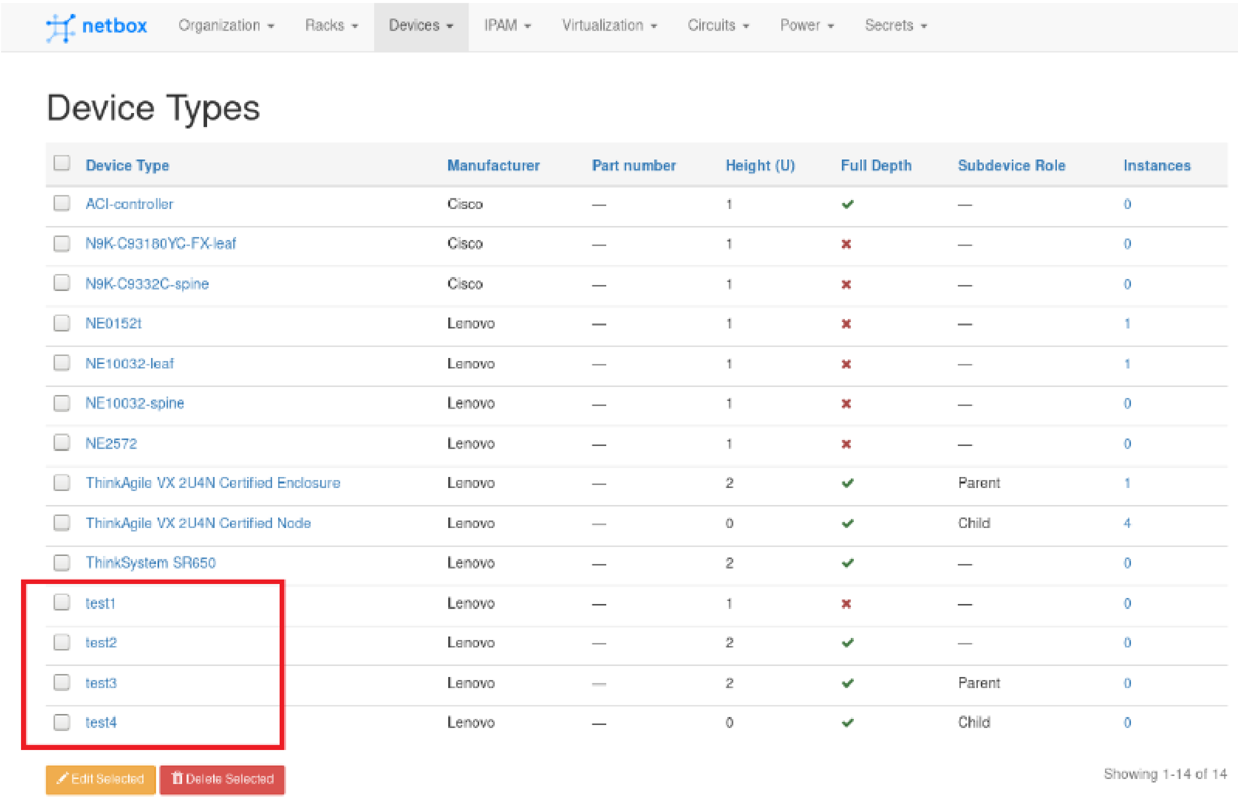
   "LOC_add_additional_device_types_workflow" (with the following variables)

```
device_type_templates:
 - name: SR650
   manufacturer: Lenovo
   model: SR650
   u_height: 2
   is_full_depth: false
   subdevice_role: 'none'
   tags: server
   interfaces:
     - name: ccm_tra
       name_prefix:
       mgmt_only: false
       form_factor: 0
     - name: edge_mgmt
       name_prefix:
       mgmt_only: false
       form_factor: 0
     - name: 0,1,2,3
       name_prefix: eth
       mgmt_only: false
       form_factor: 1350
     - name: inet
       name_prefix:
       mgmt_only: false
       form_factor: 0
     - name: vbackup
       name_prefix:
       mgmt_only: false
       form_factor: 0
     - name: vmanagement
       name_prefix:
       mgmt_only: false
       form_factor: 0
```

```
      - name: vmotion
        name_prefix:
        mgmt_only: false
        form_factor: 0
      - name: vrealize_uplink1
        name_prefix:
        mgmt_only: false
        form_factor: 0
      - name: vrealize_uplink2
        name_prefix:
        mgmt_only: false
        form_factor: 0
      - name: vsan
        name_prefix:
        mgmt_only: false
        form_factor: 0
      - name: vtep
        name_prefix:
        mgmt_only: false
        form_factor: 0
      - name: wan
        name_prefix:
        mgmt_only: false
        form_factor: 0
      - name: xcc
        name_prefix:
        mgmt_only: true
        form_factor: 1100
device_role_templates:
  - name: SR650
    color: ffff00
```

3. After the workflow is executed successfully, there will be a new device type in NetBox instance. Make sure your new device type has the expected eth*X* interfaces defined and that the Tags are set 'server' for VCF deployment usage.

   Example of new device types in Netbox:



Device Types

□	Device Type	Manufacturer	Part number	Height (U)	Full Depth	Subdevice Role	Instances
□	ACI-controller	Cisco	—	1	✓	—	0
□	N9K-C93180YC-FX-leaf	Cisco	—	1	✗	—	0
□	N9K-C9332C-spine	Cisco	—	1	✗	—	0
□	NE0152t	Lenovo	—	1	✗	—	1
□	NE10032-leaf	Lenovo	—	1	✗	—	1
□	NE10032-spine	Lenovo	—	1	✗	—	0
□	NE2572	Lenovo	—	1	✗	—	0
□	ThinkAgile VX 2U4N Certified Enclosure	Lenovo	—	2	✓	Parent	1
□	ThinkAgile VX 2U4N Certified Node	Lenovo	—	0	✓	Child	4
□	ThinkSystem SR650	Lenovo	—	2	✓	—	0
□	test1	Lenovo	—	1	✗	—	0
□	test2	Lenovo	—	2	✓	—	0
□	test3	Lenovo	—	2	✓	Parent	0
□	test4	Lenovo	—	0	✓	Child	0

Showing 1-14 of 14

Figure 13: Device types in Netbox

4. Prepare LXCA configuration patterns and firmware fw files for the additional device types, and configure LXCA:
   a) Log in to your LXCA. Then create two server patterns: one for deployment mode and another for production mode.

      The name of the pattern must follow the rules of *{{device.config_pattern}}-{{device_role}}-{{config_pattern}}*. E.g: sr650-vsan_node-deployment

      Currently, there are two options for the value of the config_pattern variable:
      - deployment: The configuration pattern used during operating system deployment. Usually, secure boot is disabled in this pattern.
      - production: The configuration pattern used after operating system for the production environment. Usually, this pattern will have secure boot enabled.
   b) Log in to your LXCA and create a firmware update policy for the new device types.

      The config_pattern string can also be configured in input.yml file

5. Provision new devices on a per-rack basis. Log in to the SPD AWX, and run the following workflows:

   “LOC_rack_device_onboard_workflow” (with the following variables)

   firmware_update: true
   netbox_device_config_file: netbox_rack_workload_2.conf

6. This LOC-A workflow will use confluent to automatically discover the devices, and automatically assign XCC IP addresses if the OOB switch is supported by Confluent and if mgmt_interface_ip is set to auto for the server in netbox_rack_workload_2.conf.

   It will also detect proper NIC inventory information in server 'description' field in NetBox. LXCA will manage the server nodes and perform a firmware update if firmware_update option is enabled. Check that the rack devices are added into SPD NetBox with NIC cabling information properly, and that the devices are added into LXCA properly.
7. Configure the tenant with devices of the new device types androles in NetBox. Use the same workflow as normal VCF deployment to deploy, expand, or stretch the CWD. Make sure you pass the "tenant_devices_role" as an extra variable with the expected device role.
8. Run the VCF deployment. Use the same workflow as normal VCF deployment to deploy, expand, or stretch the CWD.

LOC-A Network as a Service (NWaaS)

Introduction

This feature allows customers to add additional network configurations to an existing CWD. The network component, including VLAN roles, IP prefixes, VRFs, etc., will be added in Netbox dynamically. During this workflow, the CWD will be locked and will refuse other operations, such as expand, stretch, release, or remove.

Usage

1. add nwaas

a) Ensure the following prerequisite is met:
   The tenant tag should be "in-use" or "add-network".
   All other tags will not succeed and will cause the workflow to exit.
b) Log in to the SPD AWX, and run the following workflows:

“LOC_add_nwaas_workflow”(with the following variables)(with the following variables)site: shzja #required, current site name

state: present #required, present or absent

tenant_name: c001 #required, tenant name

master_prefix: 30.0.0.0/20 #optional, master prefix, if undefined, will use prefix to allocate the special prefix

vrf: t001.c001_vrf #optional, if defined, will use customer defined vrf name, if undefined, will use 1) {{sub_tenant_id}}.{{tenant_name}}_vrf when sub_tenant_id is defined 2) {{tenant_name}}_vrf

net_driver: Cisco #required, only support Cisco ACI drive now

remove workflow.

sub_tenant_id: t001 #optional,

sub_prefixes: #--> required, child prefix item list

- prefix_role: edge_to_wan #--> required, prefix role name, suggest that use "_" as the hyphenation, if we use "-" may cause the netbox response py filter cannot work correctly. name convention: {{tenant_name}}-{{prefix_role}}-{{prefix_role_suffix}}-{{increase_number}}, prefix_role_suffix is controll by parameter prefix_shared, increase_number range is [01， 50]

  vlan_role: edge_to_wan #--> optional, if defined, will allocate the vlan. vlan role name, name convention: {{tenant_name}}-{{vlan_role}}-{{increase_number}}

  prefix_shared: true #--> optional, default is true, only impact stretched cwd env, if == false, will add az1 or aZ2 as the name suffix for parameter prefix_role_suffix base on site to the prefix role

```
  increase_number: false      #--> optional, default is false, if it is true, will add increase_number to the prefix/vlan role, number range is [01，50]

  prefix:                     #--> optional, if not defined, will use master prefix and prefix_length to auto allocate the child prefix, if defined, has high priority than master prefix and prefix_length

  prefix_length: 24           #--> optional, if not defined, will use prefix parameter to allocate the prefix

  description: "xxx"          #--> optional, if not defined, will use "{{ prefix_role }}", if defined, will use the input value

  spd_mapping: yes            #--> required, if yes, will add vlan to spd nodes and create the portgroup in spd environment

  reserved_ips:               #--> optional, the reserved ip lists, if undefined, will not create any reserved ip for this prefix

   - gateway                  #--> optional, the reserved ip item, allocate the gateway ip in this prefix

   - fw1                      #--> optional, the reserved ip item, allocate the fw1 ip in this prefix

   - fw2                      #--> optional, the reserved ip item, allocate the fw2 ip in this prefix

  ACI:                        #--> optional, the ACI settings, if undefined, will do nothing in ACI switch side

   bd_vrf: edge.to.wan_VRF   #mandatory, example: edge.to.wan_VRF

   bd_name:                  #default: prefix_BD or sub_tenant_id.prefix_BD if sub_tenant_id is defined

   EPG:

    - name:                  #default: prefix_EPG or subtenant_id.prefix_EPG

     profile:                #mandatory: EDGE/FAAS/VCF

     bd_epg_vlan:            #get from netbox if not configured

     layer: "L2"             #default: L2

     bd_epg_gw_IP:           #get from netbox,if configured,It will get tag mapping ip,if not configured default is gateway tag.

     bd_epg_gw_type:         #default None(empty), other options: aci/nem;if nem, will config the edge gateway ip in spd nem vm(instead of configure gateway in aci), if aci, will config the gateway according to "layer"; if none, will create EPG as layer=L2 in ACI, no gateway will be configured.

     aep:                    #default: name cwd, mode: regular, all aep options:spd/cwd/MCCP/WSA/BCS.REP/ADMIN, and mode:regular/native

      - name: "spd"      #mean AEP for both site if spd is stretched, otherwise mean aep in site1,that's: Shared:LOC.spd.az1_AEP

       mode: "regular"  #options: regular/native

      - name: "cwd"      #mean AEP for both site if cwd is stretched, otherwise mean aep in site1: {{tenant}}:LOC.cwd.az1_AEP

       mode: "regular"

      - name: "MCCP"

       mode: "regular"
```

2. remove nwaas

a) Ensure the following prerequisite is met:

Tenant tag should be "in-use" or "add-network.", All other tags will not succeed and will cause the workflow to exit.

b) Log in to the SPD AWX, and run the following workflows

“LOC_add_nwaas_workflow” (with the following variables)

#####remove one of the prefixes under this VRF###

state: absent   #required, present or absent

tenant_name: c001   #required, tenant name

vrf: t001.c001_vrf   #required, VRF full name

sub_tenant_id: t001   #optional,

net_driver: Cisco   #required, only support Cisco ACI drive now

release_vrf: false   #optional, default is false, if == true, will delete all master-prefix/child-prefix under this vrf, and if this is customer VRF, will also delete this VRF itself.

#if no prefix could be found in this VRF, then release_vrf == true

master_prefix: 192.168.0.0/20   #optional, if defined, will try to delete it when no child prefix under this master prefix

sub_prefixes: #child prefix list, depends on release_vrf, if release_vrf == true, then this parameter is optional. if release_vrf == false, then this parameter is required

- prefix: 192.168.0.0/24   #depends on release_vrf, if release_vrf == true, then this parameter is optional. if release_vrf == false, then this parameter is required

prefix_role: edge_to_vlan   #optional, if above prefix is defined, then this parameter is no need, if prefix is undefined, then this parameter is required

####remove all of the prefixes under this vrf###

state: absent

tenant_name: c001

vrf: t001.c001_vrf

net_driver: Cisco

release_vrf: true

LOC-A DNS as a Service (DNSaaS)

Introduction

This feature allows customers to create DNS Services based on DNSMASQ, and to create records in DNS (A, CNAME, TXT), etc**.**

Usage

1. Log in to the SPD AWX, and run the following workflows:

“LOC_vmaas_workflow” (with the following variables)

```
dns_ip_entry:
  internal:
      - name: bcy001c002shzjx.c002.shzj.pudong.net
        ip: 172.22.32.17
        type: A
        state: absent
      - name: bcy002c002shzjx.c002.shzj.pudong.net
        ip: 172.22.32.18
        type: A
        state: absent
      - name: bcy001c002shzjx.c002.shzj.pudong.net
        ip: 172.22.32.17
        type: A
        state: present
      - name: bcy002c002shzjx.c002.shzj.pudong.net
        ip: 172.22.32.18
        type: A
        state: present
    external:
      - name: bcy001c002shzjx.c002.shzj.pudong.net
        ip: 25.160.28.36
        type: A
        state: absent
      - name: bcy002c002shzjx.c002.shzj.pudong.net
        ip: 10.160.28.38
        type: A
        state: absent
      - name: bcy001c002shzjx.c002.shzj.pudong.net
        ip: 10.160.28.36
        type: A
        state: present
      - name: bcy002c002shzjx.c002.shzj.pudong.net
        ip: 10.160.28.38
        type: A
        state: present
  tenant_name: c002
```

LOC-A SDDC multiple connections

Introduction

This feature allows customers to add a subtenant to a shared CWD tenant, using NWaaS and VM as a Service to create the network and VM settings for the subtenant.

Usage

1. Follow the inline instructions to configure configs/nwaas/loc_sub_tenant.yml. This file is used to specify subtenant creation settings.
2. Make sure tenant tag is ‘in-use’.
3. Log in to the SPD AWX, and run the following workflows:
   a. To add a subtenant:

   “LOC_add_sub_tenant_workflow” (with the following variables)

```
site: shzja
tenant_name: c001
master_prefix: 30.0.0.0/24
```

vf: t001.c001_vrf
sub_tenant_id: t001
net_driver: Cisco

b. To remove a subtenant:

"LOC_remove_sub_tenant_workflow" (with the following variables)

tenant_name: c001
vrf: t001.c001_vrf
net_driver: Cisco
sub_tenant_id: t001

c. To update a CWD tenant's shared type:

"LOC_update_cwd_shared_type_workflow" (with the following variables)

tenant_name: c001
shared_type: true

Parallel Workflow Deployment

Introduction

Parallel workflow deployment enables workflows to be run in parallel. A user can:

- Deploy multiple CWDs at the same time.
- Deploy multiple clusters for different tenants at the same time.
- Deploy multiple nodes for different tenants at the same time.

This feature enables multiple tenants to request resources at the same time, which reduces deployment times.

What is not supported:

- Job and workflow of NetBox metadata related operations will NOT run in parallel to avoid resource conflicts.
- Multiple workflows **CANNOT** start at the same time for the same tenant. For example, the tenant that is stretching a cluster will not be allowed to create an additional cluster at the same time.

The following table shows the LOC-A SPD workflow that support parallel workflows::

Support parallel	Workflow
No	LOC_spd_expand_create_metadata_workflow
No	LOC_spd_stretch_create_metadata_workflow
No	LOC_spd_additional_cluster_create_metadata_workflow
No	LOC_spd_additional_cluster_expand_create_metadata_workflow
No	LOC_spd_additional_cluster_stretch_create_metadata_workflow
No	LOC_cwd_create_metadata_workflow
No	LOC_cwd_expand_create_metadata_workflow
No	LOC_cwd_stretch_create_metadata_workflow

Support parallel	Workflow
No	LOC_cwd_additional_cluster_create_metadata_workflow
No	LOC_cwd_additional_cluster_expand_create_metadata_workflow
No	LOC_cwd_additional_cluster_stretch_create_metadata_workflow
Yes	LOC_cwd_deployment_workflow
Yes	LOC_cwd_expand_workflow
Yes	LOC_stretched_cwd_deployment_workflow
Yes	LOC_cwd_deploy_additional_cluster
Yes	LOC_cwd_stretch_additional_cluster
Yes	LOC_cwd_vra_deploy_workflow
Yes	LOC_cwd_vra_deploy_workflow
No	LOC_spd_expand_workflow
No	LOC_stretched_spd_deployment_workflow
No	LOC_spd_deploy_additional_cluster
No	LOC_spd_stretch_additional_cluster
Yes	LOC_remove_nodes_workflow
Yes	LOC_remove_cwd_workflow
Yes	LOC_vmaas_workflow
No	LOC_rack_device_onboard_workflow

Table 10: Parallel Workflow Support Matrix

Monitoring LOC-A Services

Introduction

LOC-A comes with utilities to check the health of the services that run in the SPD. These roles and playbooks provide a quick and easy way to validate that the services in the SPD are all up and running properly to support CWD deployments.

The loc_spd_health_check playbook performs the following actions to validate the state of the SPD's services:

- Log in to the SPD's vCenter and check the power state for each service VM.
- Log in to the SPD's vCenter and ensure that the IP address for each VMs IP corresponds to the expected IP address for that service.
- For service VMs that have an SSH connection, log in to the VM using the configured username and password.

- For service VMs that have a web service, access the configured URL to ensure an HTTP 200 is returned.

If any of the checks fail, that service will be reported as in error. The health check is integrated in the deployment workflow of LOC-A services on the SPD.

The following table indicates the checks performed against service:

Service Name	Power State Check	IP Address Check	SSH Check	Service URL Check
Lenovo Repository Service (LRS)	X	X	X	X
Lenovo Configuration Service (LCS)	X	X	X	X
Lenovo Inventory Service (LIS)	X	X	X	X
Lenovo Discovery Service (LDS)	X	X	X	
Lenovo Management Service (LMS)	X	X		X
Lenovo Logging Service (LLS)	X	X	X	X

Table 11: LOC-A Services Health Check

NetBox data backup and restore

Introduction

NetBox data contains critical information about the environment configuration, such as available devices, and tenants. Users should back up NetBox data on a regular basis to protect critical data.

Two playbooks are available to back up and restore NetBox data:

- netbox_db_backup.yml
- netbox_db_restore.yml

LOC-A supports the backup of NetBox data to an external SFTP server or a Gitlab repo. It can also restore the NetBox instance with the data file backed up on an external SFTP server or Gitlab repo.

Usage

For NetBox data backup, call the netbox_db_backup.yml playbook with the following variables:

netbox_backup_mode: Select where data will be back up. Supported options are: sftp or gitlab. Default is sftp.
netbox_db_file: latest,or specify the name for the backup.
netbox_ip: target NetBox IP address.
netbox_user: NetBox user.
netbox_pass: NetBox password (ssh).

# vars for sftp   Parameters required in sftp mode
remote_ip: Remote SFTP server IP address. By default, the Gitlab VM IP address is used.
remote_user: 'root' is the default.
remote_pass: Password for remote SFTP server. By default Gitlab VM password is used
remote_path: Remote folder path to backup the NetBox database. For example, “/opt/loc/”
remote_folder: The subfolder name under remote_path where the NetBox database is stored, default name is: netbox_data.

# vars for gitlab   Parameters required in gitlab mode
git_api_url: The target GitLab URL.
git_api_token: The target GitLab API token.
git_user: GitLab user
git_pass: GitLab password
git_url: Full clone path for GitLab, Eg. git@xx.xxx.com:root/loc_backup.git
git_branch: GitLab branch. By default, the master branch is used.

git_remote_folder: The folder name in the GitLab repository where the NetBox database is stored. By default, the folder is netbox_data.

For NetBox data restoration, call netbox_db_restore.yml playbook with the following variables:

netbox_restore_from: Select how to recover data. Supported options are: sftp or gitlab. Default is sftp.
netbox_db_file: latest or specify the file name to restore.
netbox_ip: target NetBox IP address.
netbox_user: NetBox user (ssh).
netbox_pass: NetBox password (ssh).

# vars for sftp   Parameters required in sftp mode
remote_ip: Remote SFTP server IP address. By default, the GitLab VM IP address is used.
remote_user: 'root' is the default.
remote_pass: Password for the remote SFTP server. By default, the GitLab VM password is used.
remote_path: Remote folder path to backup NetBox database. For example, "/opt/loc/"
remote_folder: The subfolder name under remote_path where the NetBox database is stored. By default, the folder is netbox_data.

# vars for gitlab   Parameters required in gitlab mode
git_api_url: The target GitLab URL.
git_api_token: The target GitLab API token.
git_user: GitLab user.
git_pass: GitLab password.
git_url: Full clone path for GitLab. For example, git@xx.xxx.com:root/loc_backup.git
git_branch: GitLab branch. By default, the master branch is used.
git_remote_folder: The folder name in the GitLab repository where the NetBox database is stored. By default, the folder is netbox_data.

Restoring an LXCA instance

Introduction

If there are issues with the LXCA VM, LOC-A can redeploy the LXCA instance and automatically re-manage the devices that have been managed previously by the broken LXCA instance.

The playbook loc_service_lxca_restoration.yml performs the following actions:

1. Removes the old LXCA VM if it exists.
2. Deploys a new LXCA VM from an .OVA file.
3. Applies patches for LXCA, if necessary.
4. Imports the necessary data into LXCA, such as: QuantityLicense, server pattern, operating system profile, and other pre-defined data.
5. Manages servers for the new LXCA instance.
6. Imports the firmware payload file and firmware policy file to the new LXCA instance.

Usage

To restore an LXCA instance on the SPD, run the workflow loc_spd_lxca_restoration_workflow in the AWX on the SPD.

Firmware upgrade for servers

Introduction

LOC-A supports upgrading the firmware (XCC/UEFI) of servers to a standard version when configuring rack servers.

Usage

Complete the following steps to complete the upgrade.

Step1. Configure LXCA to upload the firmware and policy file

Before deploying or updating LOC-A services, specify the path of the firmware payload files and policy files that are needed to upload in the input.yml of the config space. The location is:

- spd.spd.cwd.nodes.firmware.firmware_file_list
- spd.spd.cwd.nodes.firmware.policy_file_list

The firmware_file_list is an array, and each element of the array is a path of a firmware payload file. The policy_file_list is also an array; each element of the array is a policy file path.

Step2. Set the Firmware information

LOC-A supports using LXCA to upgrade multiple components of server firmware; you can specify which type of firmware will be upgraded. In addition, you must specify the upgrade policy to use.

Specify the policy and components for the upgrade in the input.yml of the configuration space. The location is: spd.spd.cwd.nodes.firmware.with_policy .

There are two attributes that should be specified:

- policy: the name of the policy file uploaded in Step 1.
- device_components: an array containing upgrade components

The device_components array has a fixed format:

```
[
  {
          "Component": "XCC (Primary)"
  },
  {
          "Component": "UEFI (Primary)"
  },
  …
]
```

It is recommended to upgrade both the XCC and UEFI of the servers. The corresponding component values are: "XCC (Primary)" and "UEFI (Primary)". If you want to upgrade only the XCC, only the object corresponding to XCC should be put in the device_components array.

Step3. Enable Firmware update

Firmware upgrades are disabled by default. There are two methods to set the firmware upgrade as true.

- Set the variable that indicates the firmware upgrade in config space input.yml to be true. The location is:
  spd.spd.cwd.nodes.firmware_update

- Pass an extra variable firmware_update for the workflow LOC_rack_device_onboard_workflow and the LOC_spd_deployment_workflow to control whether firmware upgrades are enabled when provisioning the servers.

Automated firmware policy compliance for servers

Introduction

LOC-A supports applying a firmware compliance policy to the datacenter servers that it manages via LXCA. The firmware patches can be downloaded manually.

Usage

1. Upload/Apply firmware patch
   method 1:
   a) Manually download the patch from https://datacentersupport.lenovo.com/cn/en/products/solutions-and-software/software/lenovo-xclarity/downloads/driver-list/component?name=Software%20and%20Utilities
   Make sure to get all 3 files(**.chg/.tgz/.txt/.xml**) and put them into a folder.
   b) Run the playbook
   'ansible-playbook runner-upgrade-lxca.yml -e"lxca_ip=<*your_lxca_ip*> _patch_file_dir=<*your_folder*>" -vvv'
   to upload and apply the patch.

   method 2:
   a) Run the AWX job template 'LOC_patch_auto_apply' to automatically download and apply the latest patch. LXCA must have access to the Internet.

2. Synchronize the policy with NetBox:
   run playbook 'ansible-playbook inventory-policy-sync.yml -vvv ' on Runner to Sync policy from lxca to netbox.

3. Upgrade the firmware:

   a) Use the 'LOC_firmware_upgrade_cron_workflow' to upgrade 'Inventory' servers using a cron job.
   b) Use the 'LOC_firmware_upgrade_workflow' to update any servers including active servers. You can use the following parameters:

```
policy: latest # Policy name, default is latest. Optional

devices: [] # Device name list, when devices is defined,  all status/tenant/rack/type will be ignored. Optional

status: 'Decommission' # For device belonging to a tenant, user should manually put the them in maintenance mode, and set the status to Decommission on NetBox.  After upgrading, manually set the status on NetBox back and  unset maintenance mode.

tenant: c001 # Tenant name. Optional

rack: 001 #optional

type: Lenovo ThinkAgile VX 2U4N Certified Node # Optional

force_update_mode:  false   # Ooptional  To downgrade the firmware version,  set it to true
```

Custom Repository Support

Introduction

This feature enables LOC-A to run customer playbooks, including custom jobs or workflows within LOC-A project, while being able to call loc-a functionalities (Ansible roles, Ansible libraries, Ansible modules, etc.). This feature has been supported since LOC-A 2.3.

Assumptions:

- Custom playbooks live in a separate repository from loc-a and work like an optional plugin.
- Custom jobs and custom workflows are supported.
- Main repository loc functions (Ansible roles, Ansible libraries, Ansible modules, filter plugins) can be reused and executed by custom playbooks. Custom repository code will NOT be executed by main repository loc playbooks.
- There is no separate configuration space repository for custom playbooks; all variables andconfiguration files for custom playbooks are included in custom repo.
- All names for custom workflows and jobs must start with "CUS_"   (LOC-A main APIs start with "LOC_").
- Code in a custom repository can be developed by Professional Services, developers, or end users, as long as it follows LOC-A coding guideline.

Repository layout:

The custom playbooks repository is a submodule of the loc-a project code. The complete repository layout looks like this with custom repo support:

```
loc-a
├── configs                              // submodule of config files
├── custom_playbooks                     // submodule of custom playbooks
|    ├── ansible.cfg                     // custom dir ansible.cfg
|    ├── custom.yml                      // custom playbooks
|    ├── filter_plugins                  // custom filters
|    |    └── custom_filter1.py
|    ├── group_vars
|    |    └── all -> ../../loc/group_vars/all    // soft link to loc group_vars/all for default variables
|    ├── library                         // custom ansible modules
|    |    └── custom_module1.py
|    ├── roles                           // custom roles
|    |    └── custom_role1
|    └── var_files                       // custom variable/config files
└── loc                                  // submodule of loc-a main codes
├── ansible.cfg                          // main loc-a ansible.cfg
├── cwd-deploy.yml                       // main loc-a playbooks
├── filter_plugins                       // main loc-a filters
|    └── plugin1.py
├── group_vars
|    └── all                             // default variables
├── library                              // main loc-a ansible modoules
|    └── module1.py
├── module_utils                         // main loc-a python module utils
└── roles                                // main loc-a roles
   └── lenovo.example.role1
```

Usage

- ### Running custom repository code via ansible-playbook command on the Runner VM:

ansible.cfg file is requried in custom_playbooks directory if you need to run custom repo code via the ansible-playbook command. Or you should specify Ansible settings via virtual environment variables. To support custom playbooks that call loc-a roles/filter/modules, the following settings MUST be set in ansible.cfg:

```
library       = ../loc/library

module_utils   = ../loc/module_utils/

COLLECTIONS_PATHS = /opt/virtual-envs/lnvenv3/lib/collections

roles_path    = ./roles:../loc/roles

filter_plugins     = ./filter_plugins:../loc/filter_plugins
```

# Note: to set via ENV variables, refer to https://docs.ansible.com/ansible/latest/reference_appendices/config.html

- ### Running custom repository code in AWX via Custom Jobs and Workflows:

After a minor upgrade of AWX to the LOC-A 2.3 release, you can create your custom jobs and workflow, and run your code via AWX.

- ### Deploy your custom repository code in the SPD:

Run runner-custom-repo-setup.yml on runner to deploy custom repos

Playbook parameters:

Parameter	**Required / Optional**	**Description**
init	Optional	Value: true or false. Default: false Choose whether to initialize the custom repository on Runner. This will create ../custom_playbooks directory for your future usage.
update_repo	Optional	Value: true or false. Default: true choose to update lrs repo with the custom repo, this will not impact the rest of the repos
lrs_ip	Optional	Specify the LRS IP address. Required if update_repo is set to true.
update_awx	Optional	Value: true or false. Default is true. choose to update awx with the workflow
lcs_ip	Optional	Specify the LCS IP address. Required if update_awx is set to true.
workflow_spec_file	Optional	Specify the workflow and job definition file for the AWX workflow and job creation.

Parameter	Required / Optional	Description
		Default: ../custom_playbooks/var_files/workflow_cus.yml. This creates AWX workflows and jobs for custom repositories. **Note:** All custom job/workflows are by default starting with CUS_xxxx

Table 12: Parameter of custom repo setup job

Example user scenarios in steps:

Complete the following steps to set up and run custom playbooks:

1. Initialize your custom playbooks directory on your Runner. This will create your custom repository with example files:
   ansible-playbook runner-custom-repo-setup.yml -e "init=true update_repo=false update_awx=false"
2. Prepare your custom playbooks inside the directory by adding new playbooks, roles, etc., and prepare your workflow and job definitions in custom_playbooks/var_files/workflow_cus.yml.
3. Upload your custom playbooks into the SPD GitLab and update the SPD AWX with the jobs and workflows you defined:
   ansible-playbook runner-custom-repo-setup.yml -e "lrs_ip=$your_gitlab_ip lcs_ip=$your_awx_ip"
4. Run your custom playbooks in AWX.

Logging

Introduction

In a LOC-A VCF environment, logs coming from different LOC-A services are required for troubleshooting. The most important logs are:

- LOC-A AWX logs: Logs of Ansible playbooks generated when customers run AWX workflows or jobs.
- LXCA(Lenovo XClarity Administrator) logs: Service bundles of LXCA for hardware management.

LOC-A uses ARA for online AWX log troubleshooting, and a standalone Ansible playbook for AWX and LXCA log collection. This solution provides an easy way to aggregate logs for easier analysis and collection when transmission to Lenovo Support is required.

Solution components:

- ARA: A lightweight tool that records Ansible and make it easy to troubleshoot. By default, LOC-A installs ARA as part of LOC-A services, and automatically configures ARA as an AWX callback plugin for log recording. This tool can help customers or professional service engineers to view and analyze logs in a running VCF environment.

- LOC-A log collection playbook and workflow: A LOC-A playbook collects and packs specified AWX logs and LXCA service bundles from a running VCF environment, so that the collection can be transmitted back to Lenovo. This playbook can be run on a utility node that can access LOC-A AWX and LXCA, or it can be run from AWX as a workflow.

Flow

The following figure shows the workflow for ARA:

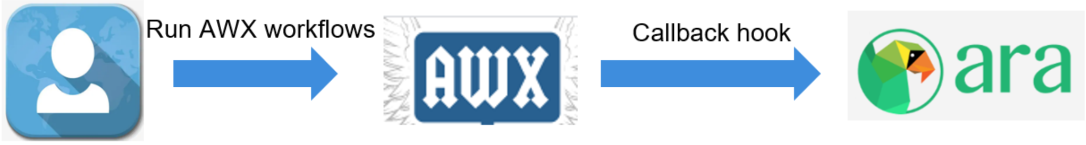


Figure 14: Recording Logs via ARA

Offline log collection can be achieved via the LOC-A Ansible workflow:

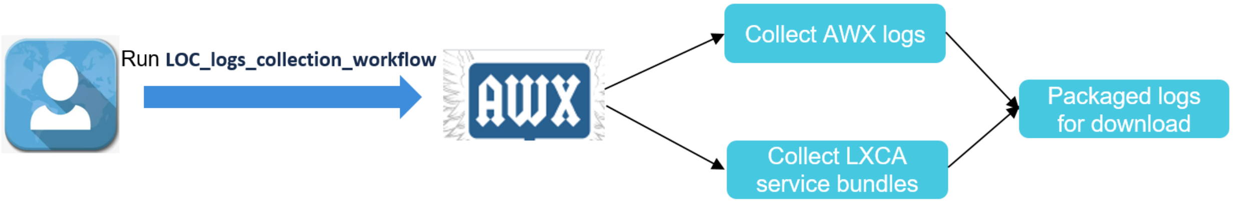

Run LOC_logs_collection_workflow
AWX
Collect AWX logs
Collect LXCA service bundles
Packaged logs for download

Figure 15: Offline Log Collection Flow

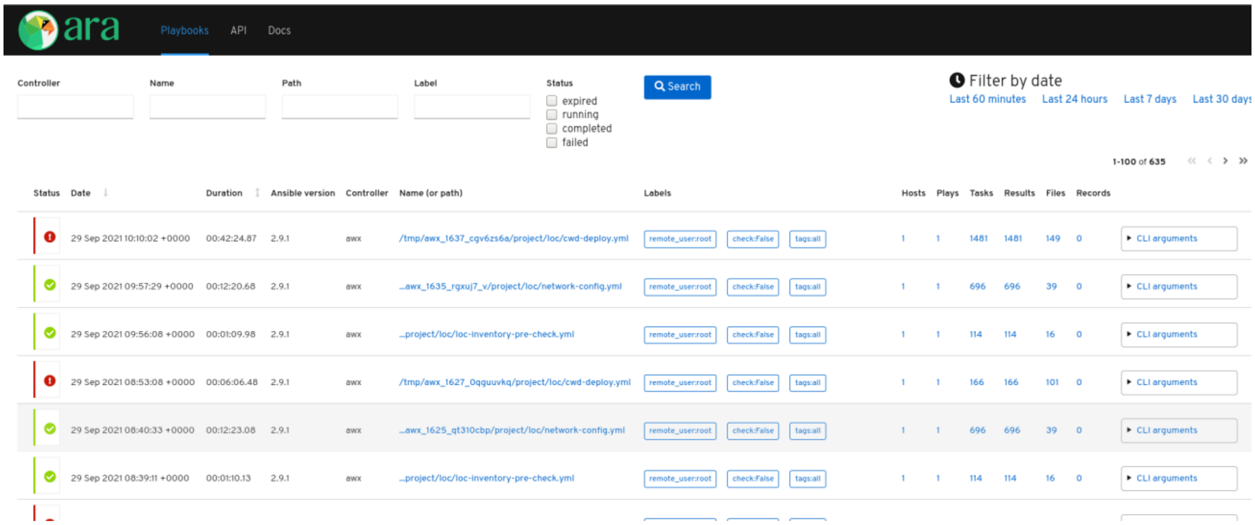
A brief overview of how to interact with each of these tools is included below:

ARA

Logs recorded by ARA can be viewed via ARA web GUI. To access ARA, from a Web browser, go to https://<*IP of ARA VM*>

The default username and password are admin/Passw0rd123.

After you log in to ARA, you will see dashboard with logs collected from AWX instance.



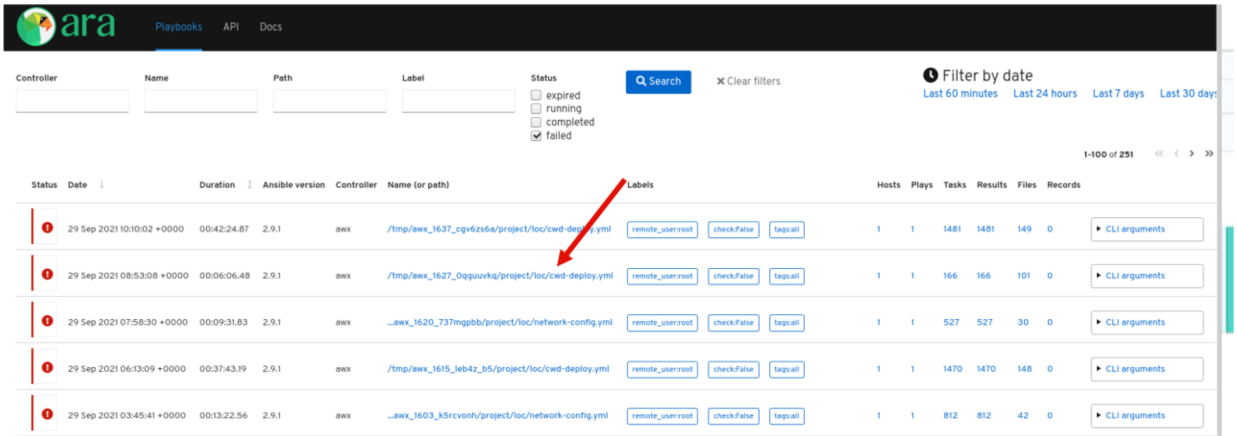
Controller
Name
Path
Label
Status
□ expired
□ running
□ completed
□ failed

Status	Date	Duration	Ansible version	Controller	Name (or path)	Labels			Hosts	Plays	Tasks	Results	Files	Records	
!	29 Sep 2021 10:10:02 +0000	00:42:24.87	2.9.1	awx	/tmp/awx_1637_cgv6zs6a/project/loc/cwd-deploy.yml	remote_user:root	check:False	tags:all	1	1	1481	1481	149	0	▸ CLI arguments
✓	29 Sep 2021 09:57:29 +0000	00:12:20.68	2.9.1	awx	...awx_1635_rgxuj7_v/project/loc/network-config.yml	remote_user:root	check:False	tags:all	1	1	696	696	39	0	▸ CLI arguments
✓	29 Sep 2021 09:56:08 +0000	00:01:09.98	2.9.1	awx	...project/loc/loc-inventory-pre-check.yml	remote_user:root	check:False	tags:all	1	1	114	114	16	0	▸ CLI arguments
!	29 Sep 2021 08:53:08 +0000	00:06:06.48	2.9.1	awx	/tmp/awx_1627_0qguuvkq/project/loc/cwd-deploy.yml	remote_user:root	check:False	tags:all	1	1	166	166	101	0	▸ CLI arguments
✓	29 Sep 2021 08:40:33 +0000	00:12:23.08	2.9.1	awx	...awx_1625_qt310cbp/project/loc/network-config.yml	remote_user:root	check:False	tags:all	1	1	696	696	39	0	▸ CLI arguments
✓	29 Sep 2021 08:39:11 +0000	00:01:10.13	2.9.1	awx	...project/loc/loc-inventory-pre-check.yml	remote_user:root	check:False	tags:all	1	1	114	114	16	0	▸ CLI arguments

Figure 16: ARA Dashboard

To view logs of a specified time period or status, you can use the filters on the top of the screens. For example, to view all failed playbooks, check the ‘failed’ Status and click the ‘Search’ button.

You can also click each playbook to view detailed information about the failed task in this playbook.

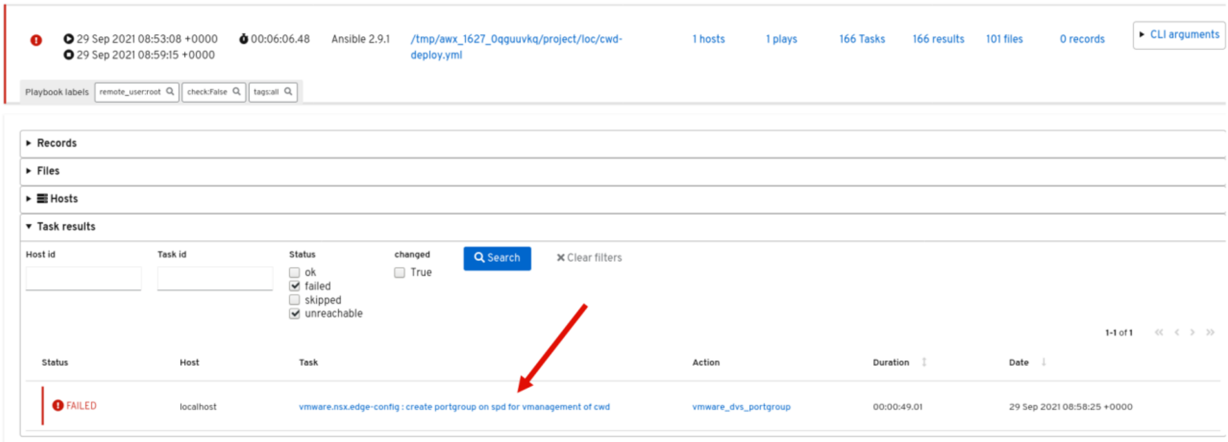


Controller
Name
Path
Label
Status
□ expired
□ running
□ completed
☑ failed

Status	Date	Duration	Ansible version	Controller	Name (or path)	Labels			Hosts	Plays	Tasks	Results	Files	Records	
!	29 Sep 2021 10:10:02 +0000	00:42:24.87	2.9.1	awx	/tmp/awx_1637_cgv6zs6a/project/loc/cwd-deploy.yml	remote_user:root	check:False	tags:all	1	1	1481	1481	149	0	▸ CLI arguments
!	29 Sep 2021 08:53:08 +0000	00:06:06.48	2.9.1	awx	/tmp/awx_1627_0qguuvkq/project/loc/cwd-deploy.yml	remote_user:root	check:False	tags:all	1	1	166	166	101	0	▸ CLI arguments
!	29 Sep 2021 07:58:30 +0000	00:09:31.83	2.9.1	awx	...awx_1620_737mgpbb/project/loc/network-config.yml	remote_user:root	check:False	tags:all	1	1	527	527	30	0	▸ CLI arguments
!	29 Sep 2021 06:13:09 +0000	00:37:43.19	2.9.1	awx	/tmp/awx_1615_leb4z_b5/project/loc/cwd-deploy.yml	remote_user:root	check:False	tags:all	1	1	1470	1470	148	0	▸ CLI arguments
!	29 Sep 2021 03:45:41 +0000	00:13:22.56	2.9.1	awx	...awx_1603_k5rcvonh/project/loc/network-config.yml	remote_user:root	check:False	tags:all	1	1	812	812	42	0	▸ CLI arguments

Figure 17: Check Logs of playbooks

Filter and click into the specified task. Then you can view more details about the output of a task.

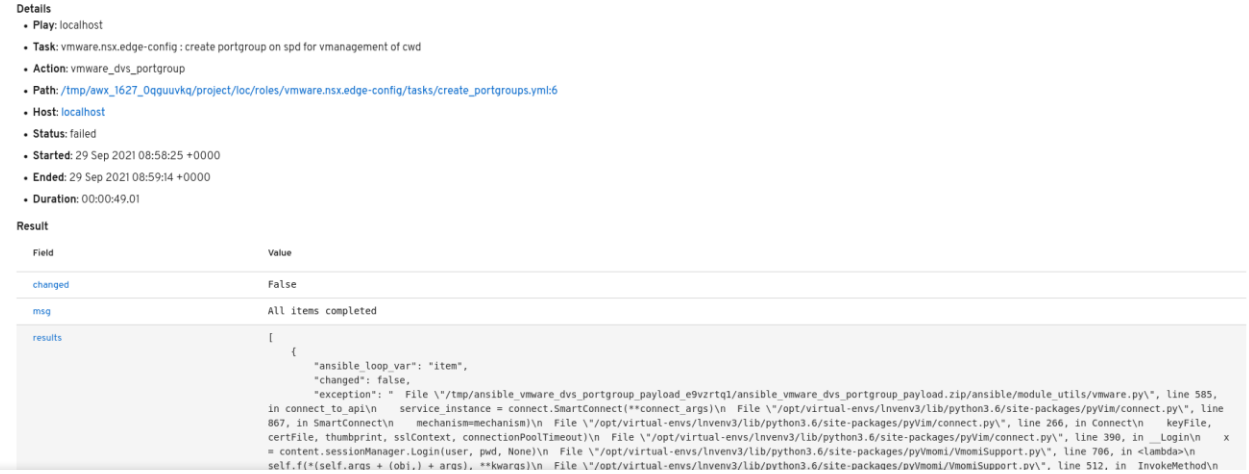


29 Sep 2021 08:53:08 +0000
29 Sep 2021 08:59:15 +0000
00:06:06.48
Ansible 2.9.1
/tmp/awx_1627_0qguuvkq/project/loc/cwd-deploy.yml
1 hosts · 1 plays · 166 Tasks · 166 results · 101 files · 0 records

Status	Host	Task	Action	Duration	Date
FAILED	localhost	vmware.nsx.edge-config : create portgroup on spd for vmanagement of cwd	vmware_dvs_portgroup	00:00:49.01	29 Sep 2021 08:58:25 +0000

Figure 18: Check Logs of playbook tasks

The following example shows detailed output for an Ansible task:



Details

- **Play:** localhost
- **Task:** vmware.nsx.edge-config : create portgroup on spd for vmanagement of cwd
- **Action:** vmware_dvs_portgroup
- **Path:** /tmp/awx_1627_0qguuvkq/project/loc/roles/vmware.nsx.edge-config/tasks/create_portgroups.yml:6
- **Host:** localhost
- **Status:** failed
- **Started:** 29 Sep 2021 08:58:25 +0000
- **Ended:** 29 Sep 2021 08:59:14 +0000
- **Duration:** 00:00:49.01

Result

Field	Value
changed	False
msg	All items completed
results	see below

```
[
    {
        "ansible_loop_var": "item",
        "changed": false,
        "exception": "  File \"/tmp/ansible_vmware_dvs_portgroup_payload_e9vzrtq1/ansible_vmware_dvs_portgroup_payload.zip/ansible/module_utils/vmware.py\", line 585,
in connect_to_api\n    service_instance = connect.SmartConnect(**connect_args)\n  File \"/opt/virtual-envs/lnvenv3/lib/python3.6/site-packages/pyVim/connect.py\", line
867, in SmartConnect\n    mechanism=mechanism)\n  File \"/opt/virtual-envs/lnvenv3/lib/python3.6/site-packages/pyVim/connect.py\", line 266, in Connect\n    keyFile,
certFile, thumbprint, sslContext, connectionPoolTimeout)\n  File \"/opt/virtual-envs/lnvenv3/lib/python3.6/site-packages/pyVim/connect.py\", line 390, in __Login\n    x
= content.sessionManager.Login(user, pwd, None)\n  File \"/opt/virtual-envs/lnvenv3/lib/python3.6/site-packages/pyVmomi/VmomiSupport.py\", line 706, in <lambda>\n
self.f(*(self.args + (obj,) + args), **kwargs)\n  File \"/opt/virtual-envs/lnvenv3/lib/python3.6/site-packages/pyVmomi/VmomiSupport.py\", line 512, in  InvokeMethod\n
```

Figure 19: Detailed Log of a Task

Configuring AWX for logging:

AWX needs to be configured to send logs to ARA. This procedure is done automatically during the initial LOC-A installation and configuration. If you see issues with ARA log recording or if you changed the ARA user, password, or IP address, you can check or update the AWX configuration using the following method:

Select **Settings** → **System.**

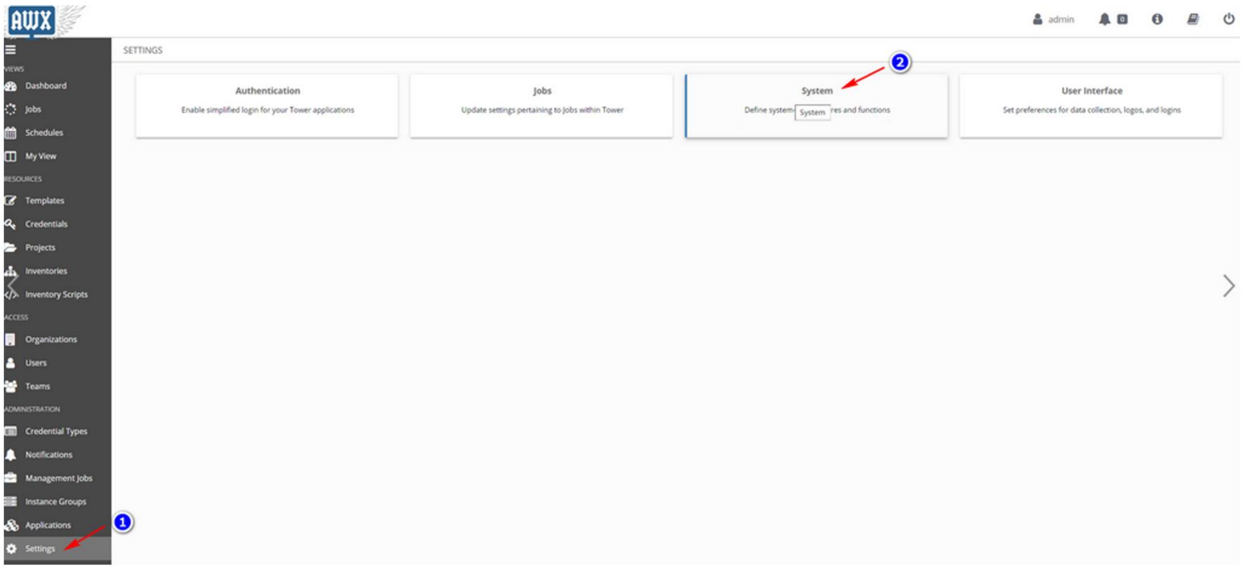


Figure 20: AWX Configuration for Logging

In “EXTRA ENVIRONMENT VARIABLES,” the following variables need to be configured for ARA recording:

```
"ANSIBLE_FILTER_PLUGINS": "./loc/filter_plugins:./custom_playbooks/filter_plugins",
 "ARA_API_CLIENT": "http",
 "ARA_API_SERVER": "https://<your_ara_url>",
 "ARA_IGNORED_ARGUMENTS": "vault_password_files",
 "ARA_API_USERNAME": "awx_user",
 "ARA_API_PASSWORD": "Passw0rd123",
 "ARA_API_INSECURE": "true"
```

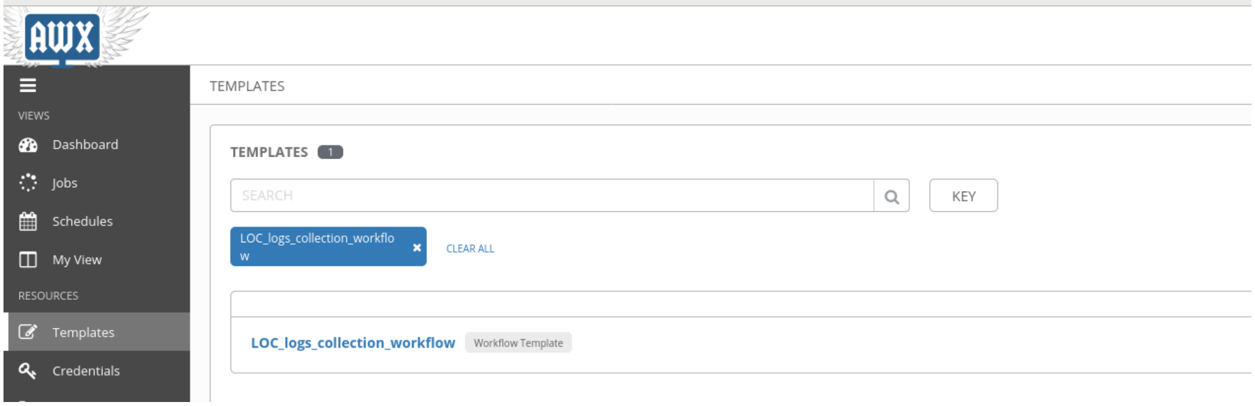
Note: There are other settings in the extra variables section in the JSON format. Make sure you don’t change them by accident.

Sending Logs to Lenovo

Note: Customer logs are not automatically sent to Lenovo by LOC-A. Logs are only sent manually, and this should only be done only with the customer’s consent.

To facilitate debugging, Lenovo has created a playbook to collect, and pack specified AWX logs and LXCA service bundles from a running VCF environment, so that it can be transmitted back to Lenovo. This playbook can be run on a utility node that can access LOC-A AWX andLXCA, or it can be run from AWX as a workflow (preferred method).

From AWX, navigate to ‘Templates,’ and find the workflow template “**LOC_logs_collection_workflow**”:

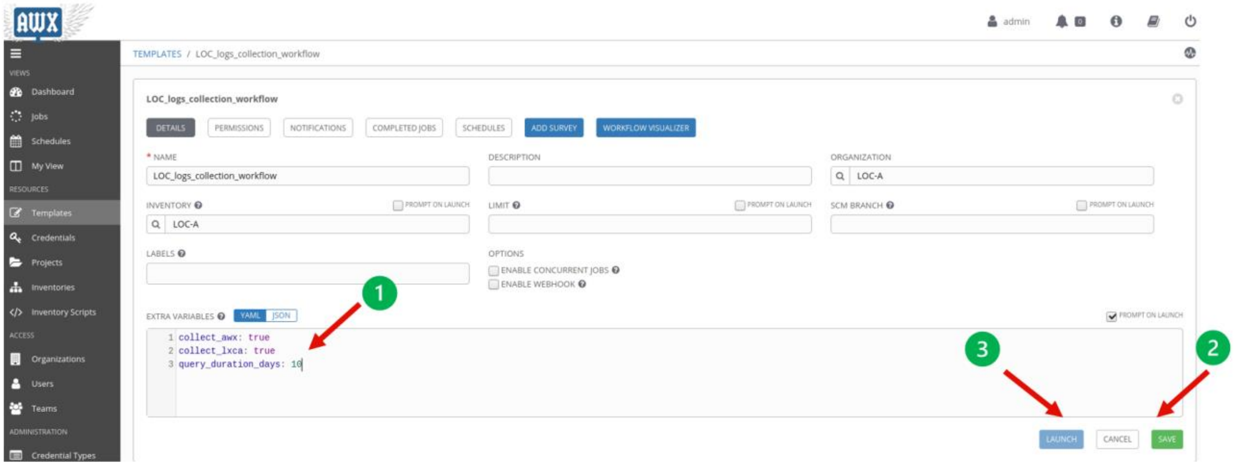


LOC_logs_collection_workflow ✖

LOC_logs_collection_workflow

Figure 21: Sending Logs to Lenovo-1

Fill in the extra variables with parameters you want to specify. Then click **SAVE→LAUNCH** to run the workflow.



LOC_logs_collection_workflow

* NAME: LOC_logs_collection_workflow

DESCRIPTION:

ORGANIZATION: LOC-A

INVENTORY: LOC-A □ PROMPT ON LAUNCH

LIMIT: □ PROMPT ON LAUNCH

SCM BRANCH: □ PROMPT ON LAUNCH

LABELS:

OPTIONS: □ ENABLE CONCURRENT JOBS □ ENABLE WEBHOOK

EXTRA VARIABLES: ☑ PROMPT ON LAUNCH

```
collect_awx: true
collect_lxca: true
query_duration_days: 10
```

Figure 22: Sending Logs to Lenovo-2

The workflow supports the following parameters:

Parameters	Required / Optional	Type	Description
collect_awx	Optional	Boolean	Indicate whether or not to collect AWX logs. This can be one of the following values: • True(default) • False
collect_lxca	Optional	Boolean	Indicate whether to collect LXCA logs or not. This can be one of the following values: • True(default) • False
query_job_id	Optional	Integer	This query parameter is used to filter the job ID of the AWX job logs to collect. If this field is not set, it will not filter the job ID.
query_job_status	Optional	String	This query parameter is used to filter the job status of the AWX job logs to collect. This can be one of the following values.

Parameters	Required / Optional	Type	Description
			• successful • failed If this field is not set, it will not filter the job status.
query_duration_days	Optional	Integer	The query parameter is used to filter the number of days of the AWX job logs to collect. If this field is not set, default value is 7, meaning the logs from last 7 days will be collected.

Table 13: Log Collection Workflow Parameters

After the playbook is executed successfully, you will see output with log file path in the workflow job like:

```
........

CUSTOM STATS:
**********************************************************************************
**********************************************************************************
*****************************



      RUN: { "awx_download_path": "/opt/awx_logs_20210729084310.tar.gz",
"awx_query_duration_days": "100", "lxca_download_path":
"/opt/LXCA_A41FF0706972435C8E5C95FCEF3FF5B3_05-45-53_29-07-2021.tar.gz"}
```

You can then log in to the AWX VM via SSH and download the file from the path. The log files are compressed via gz.

Uploading Files

After the required files have been collected, they need to be uploaded to Lenovo for analysis.

Log uploads occur through Lenovo System Care:
https://servicetools.lenovo.com/customer_logs/

Before any logs are uploaded, an issue should be raised with Lenovo Support. This will result in a Bugzilla being created. A link to the uploaded files needs to be added to the associated Bugzilla entry.

LOC-A Security Enhancement 2.0 (Roles base access in LOC-A)

Introduction

This feature will provide increased security by enabling AD integration and Role Based Access Controls for users as well as log forwarding functionality to SIEM environments. This feature has been supported since LOC-A 2.5.

Assumptions:

- *All Loc-A services(AWX, Netbox, LXCA, Gitlab, confluence, ARA, logging, etc) will use service accounts for application authentication and will allow for user input passwords during environment build*

- *Loc-A will not use “generic” root/admin accounts during normal operations but use application/service accounts*
- *Loc-A will provide a playbook to allow for updating loc-a service passwords.*
- *Loc-A will provide a Password Update playbook will enforce a password policy that can be executed by admin accounts.*
- *Loc-A will provide a playbook to allow for updating loc-a password policy that will be enforced when running the update playbook, that can only run under admin account and not by any other account*
- *Loc-A Password Policy will only affect Local Service/User Accounts*
- *AWX upgrades will keep any existing configuration and manual documentation for enabling LDAP/Active Directory access will be provided.*
- *Netbox will be enabled for authentication via LDAP/Active Directory.*
- *AWX Workflow will be created for enabling LDAP/AD Groups access to Netbox Based on PreDefined roles, roles can be defined in config space.*
- *Loc-A Appliances including DNS appliance will contain the following security measures:*
- *SSH agent forwarding must be disabled.*
- *The following MAC algorithms are allowed and must be configured for SSH daemon:*
  hmac-sha2-512-etm@openssh.com
  hmac-sha2-256-etm@openssh.com
  hmac-sha2-512
  hmac-sha2-256
- *DNS appliances will allow for configurable Syslog destination during CWD build or DNS appliance upgrade/rebuild. Syslog Destinations will be setup as a config space list to allow for one or more destination*
- *AWX Workflow will be created for enabling LDAP/AD Groups access to Gitlab custom_playbooks project*
- *AWX Sync Pipelines will be updated to use new service accounts for gitlab*
- *Documentation will be provided on how to configure log forwarding from central Loc-A syslog server.*

Usage

- Execute LOC_update_loc_services_password_workflow to update LOC-A services password:

Provide the service account and password, and also support multiple account

```
single account:

- username: admin
  password: Lenovo@123


multiple account:

- username: admin
  password: Lenovo@123

- username: loc
   password: Lenovo@123

…
```

# Note: lrs (gitlab) admin account is "root", not "admin"

- Execute LOC_update_loc_services_password_policy_workflow to update LOC-A services password policy:
  After our research, we found only password policy of **LXCA** could be updated among loc services. Please provide the parameters follow the API guide.

- Provide or enable external LDAP/AD service in your environment
- Execute LOC_enable_loc_services_ldap_workflow to enable LDAP service in gitlab and netbox:
  Please provide the parameters follow the API guide.
- Execute LOC_netbox_access_control_workflow to add/delete/view/change access control for LDAP in netbox:
  Please provide the parameters follow the API guide.
- Execute LOC_gitlab_access_control_workflow to add/delete/view/change access control for LDAP in gitlab:
  Please provide the parameters follow the API guide.
- Provide syslog server in your environment.

  - A central syslog server with Linux operation system. centos/rhel/ubuntu…
  - rsyslogd version: v8 or above

- Execute LOC_setup_loc_services_syslog_workflow to enable loc services appliance syslog forwarding.
  Please provide the parameters follow the API guide.
- Below SSH settings will configured during CWD deployment & LOC-A services deployment&major upgrade.
  - *SSH agent forwarding must be disabled.*

- *The following MAC algorithms are allowed and must be configured for SSH daemon:*
  hmac-sha2-512-etm@openssh.com
  hmac-sha2-256-etm@openssh.com
  hmac-sha2-512
  hmac-sha2-256

Flexible Deployment For Multi-Site

Introduction

User problem: Due to Twincore sites are built identically with the same number of hosts , customers do not always want to build twincore, that leaves the B-Side hosts unused and is considered wasted resources.

User value: Increased hardware usage and reduce wasting of resources

This feature will:

Enable flexible allocation of resources to customers. Enable the deployment of CWDs and clusters to be performed directly on the B site and not be locked to Site A only. This will enable a better usage of hardware resources in both sites.

This feature has been supported since LOC-A 2.5.

Assumptions:

- Add Cluster in Second site will not be supported on a CWD if the CWD Management cluster is in the Primary site and is not stretched
- Add Cluster will be enabled to add a cluster to a *CWD with devices from the secondary site by providing the Site attribute*
- *During a Create CWD, when Creating the first cluster the primary site will be designated as a custom field In Netbox on the Cluster and Cluster Group.*
- *Primary Site Attribute will be populated on each Cluster*
- *A CWD can be deployed to Site A or B by providing the site attribute*
- *CWD Network pools will be created based on Cluster Group Site*
- *Primary Site = mucfa à vSAN/Vmotion Prefix for mucfa -> c001-mucfa-networkpool*
- *Primary Site = mucfb à vSAN/Vmotion Prefix for mucfb -> c001-mucfb-networkpool*
- *A Cluster can be stretched from Site A to Site B or from Site B to Site A.*
- *When Stretching a CWD the network pools will be created based on Cluster Group Primary Site Sibling*
- *Primary Site = mucfa à vSAN/Vmotion Prefix for mucfb -> c001-mucfb-networkpool ??*
- *Primary Site = mucfb à vSAN/Vmotion Prefix for mucfa -> c001-mucfa-networkpool ??*
- *When allocating a CWD in site B the same vlans will be allocated in Site A*
- *When Commissioning hosts in SDDC manager, the network pool will be assigned based on device Site*
- *ACI IPGs will be assigned to the correct AEP based on Site Availability Zone value*
- *Sites will be documented with Availability Zone values, mapping to ACI Pod infrastructure*
- *AZ Designated Prefixes will be assigned to the Site based on site Availability Zone Value, Ex, vtep_az1 will be mapped to mucfa if mucfa Availability zone value is set to AZ1*
- *Stretched sites will be part of a Site Group containing only the 2 sites*
- *Stretched Sites will be limited to 2 sites*
- *Secondary site will be automatically chosen from the second site in the Site Group*
- *VLAN Groups for Site Groups will be used for documenting VLANs in use for both sites by using a single VLAN Group for stretched sites*

- *Per Site VLAN Groups will be created and be supported only for clusters that are not stretched and belong to a single site (OnPrem/Edge Scenarios)*
- *When adding a cluster if the MGMT cluster is not stretched and the site attribute for the new cluster does not match the MGMT Cluster Primary site the workflow will fail*
- *When stretching a cluster, the site attribute will be validated against the Primary field in netbox and the workflow will fail if they do not match.*
- *When creating a CWD the <<tenantID>>:MGMT cluster group will be created representing the Workload Domain.*
- *ACI automation will configure the correct prefixes based on the site definition in netbox for the prefixes*
- *ACI AEPs will be allocated to hosts based on Site AZ Value, example if Site is mucfB has AZ value of AZ2 then the hosts in mucfB will be mapped to the CWD AZ2 AEP and mucfA hosts will be mapped to AZ1*
- *Device IPGs will be mapped to AZ1 AEP or AZ2 AEP based on site AZ value, ex if device site is mucfa and mucfa has Availability Zone: AZ1 , hosts in mucfa will be mapped to AZ1 AEP, if device site is mucfb , hosts in mucfb will be mapped to AZ2 AEP*
- *NWaaS will be enabled to specify VLAN Group when defining new VLANs/Prefixes.*
- *SPD vCenter Affinity groups will contain site name for easy filtering by the automation (eric add: edge/vtep/dns):*
- *NSX Management Edge will be mapped to SPD Cluster in Netbox*
- *vTEP Edges will be added to SPD vCenter Affinity group based on Cluster Group Site, during CWD deploy to Cluster Group Primary Site, during stretching based on sibling site of Primary site.*
- *Add Node will take site Variable from Cluster Primary Site variable and will allocate hosts from both sites if cluster is stretched, Secondary site will be defined as Sibling site for the Primary Site.*
- *Cluster Fault Domains and Affinity rules will contain hosts based on cluster Primary Site*
- *Primary AZ = Hosts Matching Primary Site variable on the cluster*
  Secondary AZ = Hosts Matching sibling Site of the Primary Site variable on the cluster
- Primary VRF will be mapped to the Cluster Group, if VRF is not specified, the value will be taken from the Cluster Group Primary VRF value.
- Multiple VRFs will be configurable during tenant onboarding allowing for specifying the primary VRF.
- When selecting a master prefix from a VRF different from the primary VRF, the VRF name will be required
- vCenter name will be mapped to Cluster Group under Controller custom field.

Usage

- update config files:

ID	File Name	Comments
1	netbox_initialization.yml	Separate the original netbox_initialization.yml to [netbox_initialization.yml netbox_initialization.yml will focus on the pre-config the netbox, onboard eg. custom_fields, tags, manufactures, device_roles, device_types, ipam_ro
2	netbox_sites_init.yml	netbox_sites_init.yml will focus on the new site creation in netbox. And allocate the site's resources, this config file should update based on t eg. site/site-group/vlan_group/vlan/global-prefix/global-ip_address

ID	File Name	Comments
3	inventory-device-allocate.yml inventory-vm-create.yml	Create or update the target tenant. And create or update cluster_group/VRF/cluster for the tenant. And allocate the tenant resources for deployment.

- Upgrade the netbox data to new format, please check the LOC-A service minor upgrade item in this user guide.
- Lots of the workflow parameters updated, please follow the latest API guide to execute the workflows.

key updated functions in this feature

Please see the netbox screenshots in this page for some of below items.

- add site-group (2 sites) for VCF stretch scenario, but still support single site non-stretch scenario
- add vlan-group in site-group, the name will looks like {{ site_group }}_product-data-1; in previous design, it is {{ site }}_product-data-1, we should still support
- if vlan is in stretched environment, will remove the site attribute
- if prefix is in stretched environment and prefix is stretched, (e.g. vmanagement) will remove the site attribute
- if prefix is in stretched environment and prefix is non-stretched, (e.g. vsan/vmotion/vtep) will still have site-a or site-b assigned
- prefix will use one same vlan if under a stretched environment, but no site attribute of the vlan
- the cluster will add the custom fields to record the primary_site info
- cwd nem vm will created as before but changed the cluster to spd primary cluster
- SPD vCenter Affinity groups will contain site name for each host-group/vm-group/affinity-rule

original before 2208 release:

```
_host_group_name1: '{{ spd_primary_cluster.name }}_primary-az-hostgroup'
_host_group_name2: '{{ spd_primary_cluster.name }}_secondary-az-hostgroup'

_vm_group_name1: '{{ spd_primary_cluster_info.name}}_primary-az-vmanagement_edge-0_group'
_vm_group_name2: '{{ spd_primary_cluster_info.name }}_secondary-az-vmanagement_edge-1_group'

drs_rule: "{{ spd_primary_cluster_info.name }}_vmanagement_edge-0 must be on Primary Site"

drs_rule: "{{ spd_primary_cluster_info.name }}_vmanagement_edge-1 must be on Secondary Site"
```

after 2208 release:

```
_host_group_name1: '{{ spd_primary_cluster.name }}_{{ spd_primary_site }}-hostgroup'
_host_group_name2: '{{ spd_primary_cluster.name }}_{{ spd_secondary_site }}-hostgroup'

_vm_group_name1: '{{ spd_primary_cluster_info.name}}_{{ primary_site }}-vmanagement_edge-0_group'
_vm_group_name2: '{{ spd_primary_cluster_info.name }}_{{ sibling_site }}-vmanagement_edge-1_group'

drs_rule: "{{ spd_primary_cluster_info.name }}_vmanagement_edge-0 must be on {{ primary_site }} Site"

drs_rule: "{{ spd_primary_cluster_info.name }}_vmanagement_edge-1 must be on {{ sibling_site }} Site"
```

- all deployment workflow will remove the site input

Bare metal as Service

Introduction

Enable automated deployment and configuration of BareMetal Servers.

This will enable customers to order BareMetal servers with various OS flavors. Enable customers to consume Physical servers as BareMetal enabling them to deploy applications that cannot run on virtual infrastructure.

Some applications cannot be deployed in a virtualized infrastructure due to various reasons like cost, licensing or support and require to be deployed on a physical server running a server OS.

This feature has been supported since LOC-A 2.5.

Assumptions:

- *LXCA will be used for OS deployment*
- *LXCA will apply bios profile to all hosts in the deployment*
- *Separate workflow will be available for adding a host*
- *Separate workflow will be available for removing a host*
- *Networks required for devices can be created by using NWaaS & NWaaS Update*
- *Custom Workflow for NWaaS Add and update to create the network zone will be provided.*
- *Custom/Metadata Workflow will allow for customer to specify custom prefix/vlan/description role names for both bare_metal _cus and bare_metal_opt, if duplicate names are identified under the tenant, the name will be appended with _1 , _2 etc as per NWaaS standards.*
- *Device HostName will be updated on deployment based on name input for each device*
- *During removal workflow the hostname in netbox will be set to the device name.*
- *Separate Metadata workflow will be available*
- *NWaaS workflow will create network required for baremetal and allocate Ips as needed*
- *Metadata onboarding workflow will follow the CWD onboarding process allowing for multiple devices to be onboarded and vlans specified as well as ips.*
- *If Device IP defined in input does not belong to the bare metal network , the workflow will error out*
- *Baremetal Metadata workflow will create by default <tenant>_cluster_bm01 with a Bare_metal cluster type and will allow for the name to be sent as an input ex: cluster_name: c001_cluster_bm01/ c004_customer_zone1*
- *Device Allocation will only allocate devices from one site and site will be required as an input.*
- *If Multiple Baremetal servers are deployed they will be configured with the same root password which is required for input*
- *Root User account will be disabled for SSH*
- *All commissioned Baremetal severs will configure public key authentication if key is provided associated with the user sent as input*
- *BareMetal will be compatible with Versioning and will have different configs per OS Flavor*
- *Initial OS flavor supported will be SuSe Linux 15 and RedHat8*
- *OS deploy will configure required packages and setup partitioning according to customer requirements*
- *Packages&Partitioning are part of the deployment profile and will not be customizable for each install, only customizable by OS version.*
- *BareMetal workflow will allow for skipping DNS configuration in CWD DNS via config space.*
- *Each server will be mapped to a specific Netbox Platform based on OS deployed*
- *ESXi hosts in a CWD deployment will also be mapped to a Platform.*

- *OS Deploy will allow for specifying a custom IP the deploy API for the OS being deployed, the IP will be reserved in netbox.*
- *Workflow will allow for multiple hosts to be specified during deployment*
- *API will require an input for hostname to be sent as an array and be equal to the number of hosts.*
- *BareMetal hosts will be first mapped for vmanagement network for staging and install process and will include a nat IP*
- *BareMetal hosts will be moved to the BareMetal tenant vlan the vmanagement and nat Ips will be removed in the process.*
- *LACP Mapping will be documented in Netbox, standard deployments will have all connected ports in the same LACP Group.*
- *Teaming script will configure LACP team with VLAN based on Netbox metadata (Connected NICs and LACP Member) according to customer naming convention*
- *LACP Bond vLAN interfaces will be created for each interface mapped with a VLAN*
- *LACP Bond vLAN Interfaces will be configured with IP or DHCP based on netbox interface configuration*
- *ACI Automation will first map the hosts to the install AEP and will update the port configuration to Production AEP once the servers are moved to the BareMetal VLAN.*
- *Deploy workflow will allow for executing custom commands /scripts based on config space file before moving the host to the BareMetal vlan as well as allowing for a run once script on reboot for post move to BareMetal VLAN.*
- *Deploy workflow will allow for executing custom binaries locally on the servers being deployed.*
- *NWaaS will create 2 new AEPs per tenant one for the Deploy Network (if it does not exist) and one for the BareMetal VLANs*
- *NWaaS will be updated to allow for creating new VLANs without assigning them to any cluster.*
- *NWaaS ACI automation will create all BareMetal VLANS and EPGs and assign them to the AEP.*
- *Removal of Network components will be done via NWaaS*
- *Removal Workflow will only shutdown hosts and revert them back into the pool and reset the hostname back to device name.*
- *BM Removal Workflow will only remove IPGs in ACI so that hosts get removed but networks are not touched*
- *Removal Workflow will remove the baremetal netbox cluster if no more hosts are available in the cluster.*
- *Expanding a baremetal cluster will use the same Baremetal Metadata onboarding workflow.*
- *BareMetal OS deploy will set vlan to 1 in LXCA API call for OS deploy*
- *Remove BareMetal workflow will not remove the prefixes and VLANs, that will be performed via NWaaS once no devices are connected to the network*
- *Remove Workflow will remove any LACP virtual interfaces or BareMetal Virtual interfaces created during build time including IP reservations.*
- *Network as a Service (NWaaS) will be updated to allow for Specifying clusters including No cluster option*
- *Network as a Service (NWaaS Add/Remove) will be updated to allow for specifying a custom Physical Domain and will create it if will not exist, Physical Domain Pool will also be required if a new Physical Domain does not exist already.*
- *Network as a Service (NWaaS Add/Remove) will be updated to allow for removing a custom Physical Domain, and it will pre-check to see if it is still in use before removing the Physical Domain.*

- *Network as a Service (NWaaS add & Update) will be updated to allow for specifying a custom Physical Domain and will create it if will not exist, Physical Domain Pool will also be required if a new Physical Domain does not exist already.*
- *Network as a Service (NWaaS) will also be able to remove a Physical Domain if the domain is not in use.*
- *Remove CWD will error out on pre-check if baremetal hosts are detected inside the CWD.*
- *NWaaS will perform a pre-check before removing the AEP by checking if AEP is still in use by IGPs and EPGs.*
- *AWX Redeploy Workflow will be provided to reset the devices and cluster metadata to deploy state to enable redeploying the devices using the AWX BareMetal Deploy Workflow*
- *NWaaS will perform a pre-check before removing the EPG by checking if there are any MACs discovered in ACI on the EPG.*
- *ACI API Calls will be provided by T-Systems.*

Usage

- Provide bmaas config files
  - config_space
  - ······ ······ baremetal
  - ······ ······ ······ redhat8   # baremetal platform name + version
  - ······ ······ ······ ······ aci_config.yml   # aci config file
  - ······ ······ ······ ······ customize   # customer commands/scripts dir
  - ······ ······ ······ ······ ······ xxx.sh   # command scripts file
  - ······ ······ ······ ······ input.yml   # default password and configs
  - ······ ······ ······ ······ inventory-device-allocate.yml   # allocate devices/interfaces configs
  - ······ ······ ······ ······ inventory-device-deploy.yml   # deploy devices/interfaces configs
  - ······ ······ ······ ······ loc_nwaas_bmaas.j2   # bmaas nwaas prefixes & aci settings
  - ······ ······ ······ suse12
  - ······ ······ ······ ······ aci_config.yml   # aci config file
  - ······ ······ ······ ······ customize   # customer commands/scripts dir
  - ······ ······ ······ ······ ······ xxx.sh   # command scripts file
  - ······ ······ ······ ······ input.yml
  - ······ ······ ······ ······ inventory-device-allocate.yml
  - ······ ······ ······ ······ inventory-device-deploy.yml
  - ······ ······ ······ ······ loc_nwaas_bmaas.j2   # bmaas nwaas prefixes & aci settings
  - ······ ······ devices
  - ······ ······ vcf3.10.1.2
  - ······ ······ vcf3.10.2.2
  - ······ ······ netbox_initialization.yml
- Add new devices type and roles if the device role and type is undefined by the workflow LOC_add_additional_device_types_workflow
- Add new devices if no enough Inventory devices in the netbox by the workflow LOC_rack_device_onboard_workflow
- Deploy bmaas by workflows
  - Add network by LOC_add_nwaas_for_bmaas_workflow
  - Create metadata by LOC_bmaas_create_metadata_workflow

    - The meta data can only be changed with "Staged" status devices. If the baremetal device has been deployed and the status in netbox is "Active", need run LOC_bmaas_reset_metadata_workflow to reset the status to "Stage".
    - Metadata for BM allow for adding/removing of interfaces. Use LOC_bmaas_create_metadata_workflow to do the meta data update. use “state: absent” to remove interface, default is present.
  - Deploy the OS by LOC_bmaas_deployment_workflow
    - OS Management interface will be tagged in netbox with: ‘OS_IP’ tag
- Remove bmaa workflows:
  - Remove nodes/cluster by LOC_remove_bmaas_workflow
  - Remove network settings by LOC_remove_nwaas_for_bmaas_workflow
- Update the bmaas network settings by LOC_update_nwaas_workflow
- LOC_bmaas_reset_metadata_workflow will setup the servers from active to staged status, and update tenant/cluster tag before deploying.
- Expand bmaas metadata should use the same create metadata workflow LOC_bmaas_create_metadata_workflow but update the node number in extra input parameters or config file.
- Expand bmaas deploy should use the same deploy metadata by LOC_bmaas_deployment_workflow.
- Apply the updates to the AEP and IPG mappings by LOC_bmaas_network_update_workflow.
  - The new AEP need to be configurated by LOC_add_nwaas_for_bmaas_workflow.

VRealize Suite Scaling

Introduction

Enable customers to increase the size of disk/cpu or memory for the vRealize Suite Products deployed.

This feature has been supported since LOC-A 2.6.

Assumptions:

- *Scaling will be done via LifeCycle Manager API for all components for storage increase.*
- *Scaling will be done via LifeCycle Manager API for all components CPU/Memory except for Lifecycle Manager*
- *Scaling will require downtime of the suite products.*
- *vRealize Suite Scaling will be integrated into the deployment workflow*
- *vRealize Suite Deployment will ignore components already deployed –done*
- *If vRealize Suite Deployment will be started multiple times with the same scaling settings, the automation will add the disk resources for all the number of jobs started. Ex If workflow runs 3 times to scale with 300 GB Disks, 3 x 300GB disks will be added*
- *vRealize Suite Scaling will check deployment compute size before expanding and skip if the deployment is already on the desired Size.*
- *vRealize Suite Scaling will be used to expand vRealize Lifecycle Manager storage*
- *Current Cluster size will be identified by querying the Environment via vRealize LifeCycle Manager API*
- *vRealize Suite Scaling will support all Loc-A supported products: vRLCM , VIDM , vRA, VROPS, VRLI & vRNI.*

Usage

- VRA each component scale trigger by VRA deployment workflow with install_vrops=scale. The scale will do disk expand and node site scale.
- The scale components need to be passed by the parameter scaling_components with below format:

```
scaling_components:
  - name: vrlcm
    diskgb_expand: 100
  - name: vrops
    size_expand: large
    diskgb_expand: 100
  - name: vra
    size_expand: extra large
    diskgb_expand: 100
  - name: vrli
    size_expand: large
    diskgb_expand: 100
  - name: vidm
    size_expand: large
    diskgb_expand: 100
  - name: vrni
    size_expand: large
    diskgb_expand: 100
```

- vrlcm can only scale disk.
- VRNI scale trigger also by vra deploy workflow with install_vra=scale, and only support same nodesize for controller and platform nodes

OutOfBand Switch Configuration

Introduction

Enable customers to automatically configure out of band switches

This feature has been supported since LOC-A 2.6.

Assumptions:

- *Out of Band Switches will be documented in netbox*
- *Onboarding process will be modified to allow for setting ports to Access mode as well as Tagged mode.*
- *Out of Band Switch configuration will be setup to configure both NAS and regular devices*
- *Out of Band Switch configuration will be skippable in the Device/NAS onboarding workflows based on a common variable: oob_configure: True*
- *Loc-A will support the Cisco NXOS based switches for out of band switch configuration*
- *Loc-A will detect the switch platform and automatically use the correct API for Cisco ACI or Cisco NXOS*
- *Switch OS /Platform will be documented in Netbox.*

- *Workflow will configure only switch ports for the devices being onboarded.*
- *Onboarding will configure dedicated DNS entries for oob with customizable OOB DNS domain if defined.*
- *OOB DNS Domain custom field will be added to Netbox.*
- *Switch port configuration will be defined in netbox with proper vlan tagging type (access/trunk) and MTU size*
- *Switch port configuration will support specifying custom port configurations like enabling STP or BDPU guard. T-Systems is responsible for defining the standard list of options that need to be enabled per port.*
- *Assuming Netbox 3 can be used to upgrade , extra port configuration will be added as a custom field list comma divided on each interface.*

Usage

- The OOB switch configurate only support cisco NXOS switch.
- AWX workflow need to be upgraded.
- Netbox need to be upgraded to 1.32.
- The new device type of the OOB switch, need run LOC_add_additional_device_types_workflow to add it. Example input of LOC_add_additional_device_types_workflow to add new OOB device type:

  ```
  device_types:
    - name: NEXUS9000-C9300V-chassis
      model: NEXUS9000-C9300V-chassis
      manufacturer: Cisco
      u_height: 1
      is_full_depth: false
      subdevice_role: 'parent'
      tags: switch
      interfaces:
        - name:
  1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42
  ,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64
          name_prefix: ethernet1/
          mgmt_only: false
          form_factor: 1000BASE-T (1GE)
        - name: mgmt0
          mgmt_only: true
          form_factor: 1000BASE-T (1GE)
  ```
- OOB switch config process is contained in LOC_rack_device_onboard_workflow. Added variables on device onboard configuration file:

- device.platform   #As oob device, it must be "cisco.nxos" for a NXOS switch
- device.interfaces.mtu
- device.interfaces.mode
- device.interfaces.port_options

Updated config file - device:

```
  platform:(O)             --> optional, str, Defines device Platform, to be used for Network Driver detection
                                 - cisco.nxos  #only for oob switch
```

```
                                          - cisco.aci   #not support as oob switch yet
    mtu(O):                 --> optional, str, define the port MTU size,  used for setting the interface MTU, if empty ,
ignore and skip.
    mode(O):                --> optional, str, define the vlan tagging mode, used for setting the interface to tagged
or access (default tagged). If XCC or SMM(MGMT) interface default to access mode.
                                 - access
                                - tagged
    port_options(O):   --> optional, list, define the list of options(switch dependent) to enable for each port
                                   eg. - spanning-tree portfast
                                        - bdpu
```

- The user name and password of the oob switch need to be defined in input.yml

```
oob:
  cisco_n9k:
    user: "admin"
    pass: "xxxxxxxx"
```

VCF automated upgrade

Introduction

VMware Cloud Foundation provides easy upgrades and automated but there are still tasks that users need to perform manually like uploading packages or vendor provided ISO’s.

This feature has been supported since LOC-A 2.6.

Assumptions:

- *Upgrades will be applied only via SDDC Manager and will follow VMware documentation for upgrading*
- *Automation will upload and configure vendor (Lenovo) provided iso for upgrades*
- *Patch will require an input for the destination version*
- *Operations will perform Precheck and fix any issues that would prevent a successful upgrade such as but not limited to: password updates, free up space, etc.*
- *Workflow will run the VCF Prechecks for each component upgrade and error out if there are errors and provide details around the failure.*
- *Workflow will ignore any error messages related to ESXi ISO*
- *Workflow will allow for seamless job restart once the errors have been fixed*
- *Backups will be performed by the customer before running the upgrade workflow.*
- *Workflow will allow to skip the PreChecks Validation via special flag to be used in special scenarios*
- *Workflow will take snapshots where the option is available. Ex. SDDC Manager, vCenter, NSX-T managers.*
- *Workflow will upgrade all workload domains and clusters.*
- *Parallel cluster upgrades will be disabled to limit the outage risk*
- *Patches will be kept in the same folder as the VCF Version in a subfolder called Patches.*
- *Patch folders will contain both the patch file but also the checksum and .sign files*
- *Automation will loop over all patch files in the patches folder and upload them to SDDC manager*

- *Running 2 upgrade workflows on the same CWD will result in an error and only the first submitted job will succeed.*
- *Both CWD and SPD Upgrades will be supported.*
- *Workflow will require a user account with Admin access to vcenter and SDDC manager to perform the upgrade. User can be administrator@vsphere.local or a service account.*
- *Workflow will ensure passwords are secured*
- *Customer is responsible for following the secure password creation process to secure the input passwords.*
- *All workload domains are on the same VCF Version*

Usage

- Provide vcf_upgrade_bundles.yml config files
- LOC_vcf_upgrade_stage_workflow

Run this workflow to upload bundles and ISOs for the preparations of VCF upgrade.

- LOC_vcf_upgrade_workflow

Run this workflow to upgrade VCF to the target upgrade_version.

License upgrade for VCF/vRA

Introduction

Workflow to update/replace Licenses for all VCF & vRealize Products to enable customers to add more nodes and clusters to a deployment.

This feature has been supported since LOC-A 2.6.

Usage

- Update the latest code and loc services
- Provide the correct API parameters according the API guide.
- Run the workflow LOC_vcf_update_license_workflow
- The SDDC/VRA/VRLI/VRNI licenses cannot remove the old license.
- The License description is not required, the default value is the licenses key such as: "CF-SDDC-MAC-C".

LOC-A version on AWX/Netbox

Introduction

AWX - Login Screen message: LOC-A version: <b>R2203</b>

- Netbox-
    - Config Context - Dump the content of the Version file into a Loc-A Release config context
    - try to use the netbox notification?
- * Applied during any Loc-A upgrade (minor upgrade)

This feature has been supported since LOC-A 2.6.

Usage

- Update the latest code and loc services

- Run the major/minor upgrade for the existed loc sevices.
- For a new deployed loc services, this change can work.

Enabling customer playbook via workflow

Introduction

Custom Playbooks currently get enabled from Runner via playbook , this should be done via AWX Workflow instead to create the custom_playbooks repo in gitlab and allow for creating a user account that has access only to the custom playbooks repo in gitlab.

This feature has been supported since LOC-A 2.6.

Usage

- Update the latest code and loc services
- The latest custom_playbooks repo is upload the gitlab loc repo path: loc/var_files/custom_playbooks_example.tar.gz
- The local custom_playbooks directory has correct layout such as:
-
```
├─── custom_playbooks                    // submodule of custom playbooks
│    ├─── ansible.cfg                     // custom dir ansible.cfg
│    ├─── custom.yml                     // custom playbooks
│    ├─── filter_plugins                 // custom filters
│    │    └─── custom_filter1.py
│    ├─── group_vars
│    │    └─── all -> ../../loc/group_vars/all    // soft link to loc group_vars/all for default
variables
│    ├─── library                        // custom ansible modules
│    │    └─── custom_module1.py
│    ├─── roles                         // custom roles
│    │    └─── custom_role1
│    └─── var_files
│    │    └───workflow_cus.yml             //set up awx jobs/workflows
```
- LOC_update_custom_playbooks_repo_workflow

Run this workflow to Update the custom_playbooks repo.

- LOC_create_customer_account_workflow

Run this workflow to Create a ReadOnly user for custom_playbooks repo.

LOC-A NSXT as a Service (NSXTaaS)

Introduction

configurate NSXT.

This feature has been supported since LOC-A 2.8 R2307.

Usage

A nsxt config file is need for nsxt configuration workflow.

The sample configuration file is at vcf{production}/default/nsxt-config.sample

There are 4 scenarios:

1. input tenant_name

```
tenant_name: c005
cwdData:
  mgmtNetwork:
    adminLan:
      ipSubNet:
        - 10.240.206.0/24
        - 20.0.0.0/23
        - 44.193.254.0/23
    bgp:
      bgpAvnAsn: '65507'
      bgpEdgeAsn: '65503'
      bgpPeerAsn: 85501
      bgpPassphrase: admin-fci
```

   nxstaas workflow read inventory data from netbox, and use the default j2 file to generate the nsxt configuration file according to the tenant type:
   spd/spd-nsxt-config.j2
   cwd/cwd-nsxt-config.j2
   edge/nsxt-config.j2

2. input j2 file and tenant_name

```
tenant_name: c005
nsxt_config_template_file: cwd/witness-vpn-nsxt-config.j2
```

   nxstaas workflow read inventory data from netbox, and use the input j2 file to generate the nsxt configuration file.
   For example, *configure the VPN service for the witness*.

   2.1 new VMS need to be added to netbox by VMaas. run vmaas workflow LOC_vmaas_workflow to add vpn VM:

```
tenant_name: c005
virtual_machines:
  - name: vpn001c005shzjx
    device_role: nsxt-service
    state: present
    domain: c005.shzja.fci.ts-ian.net
```

```
cluster: nem001c005shzjx
comments: NSX-T VPN Service
tags:
  - role_vpn
interfaces:
  - name: ccm_overlay_uplink
    mode: tagged
    mgmt_only: true
    tagged_vlans:
      - ccm_overlay_uplink
    mtu: 9600
    ip_address: auto
    prefix_role: ccm_overlay_uplink
    vlan_role: ccm_overlay_uplink
```

2.2 run nsxtaas workflow LOC_nsxtaas_workflow to configurate NSXT

```
tenant_name: c005
nsxt_config_template_file: cwd/witness-vpn-nsxt-config.j2
vpn:
  helmangroup: dh14
  ike_version: IKEv1
  key: e4ed236fc771bb84d2f719597c6c53d73aa6f520d5e73a83aa120e73
  peer: 192.168.0.151
```

3. input nsxt_config_file, in this case, do not need netbox to get data.

```
nsxt_config_file: ../configs/vcf4.4.0.0/default/nsxt-config.yml
```

4. input nsxt_config

```
cwdData:
  cwdCredentials:
spdnmg_admin:
  password: xxxxxxxx
state: present
nml_host: nml001s003shzjx.s003.shzj.lenovo.loca.cloud
```

```
nsxt_config:
  tier:
    - name: net001c007edge2x
      Interfaces:
        - name: test_interface
          ip_address: 192.168.0.1
          type: EXTERNAL
          segment: "vl2115_test"
```

Note: For tier absent, nsxtaas will remove all the configurations under the tier instead of the detail configs in configuration file, which contains BGP, DHCP, NAT, VPN, firewall rules, static rules.

IPv6 support

Introduction

VMaaS will enable the auto allocation of IPV6 addresses to interfaces

This feature has been supported since LOC-A 2.8 R2308.

Assumptions

- *VMaaS will enable the auto allocation of IPV6 addresses to interfaces*
- *VMaaS will enable add/modify/delete for IPV6 addreses.*
- *VMaaS will allow for an interface to have both IPV6 and IPV4 addresses to be allocated.*
- *VM interfaces in netbox can have IPV6 or IPV4 ips or both*
- *VMaaS will default to interface name for setting the IP name.*
- *VMaaS will allow to define an IP name when using the Auto option for IP allocation.*
- *VMaaS will allow for defining the management IP when multiple Ips are defined on the same interface and the management IP will be added to the DNS server and its NAT to external DNS*
- *NWaaS will be updated to support L3out routing using IPv6 interface Ips.*
- *IPV6 addresses will not be set as management interfaces for a VM*
- *IPV4/IPV6 addresses will be defined as a list under the interface and will allow for specifying a unique prefix role to target the correct prefix. Ex ipv4 will get ip from edge_mgmt, ip6 will get ip from edge_mgmt6*
- *NWaaS Static Route will be updated to filter IP addresses or subnets by role_name, the IP names need to be unique across all tenants and global IPS. Subnets can belong to the tenant/sub_tenant or global and names must be unique between global and tenant prefix roles.*
- *VMaaS will implement a new variable to differentiate between allocating a single IP to the interface and multiple Ips.*
- *VMaaS improvement will be backwards compatible with previous versions and customer will have to update APIs if they want to use the new functionality and allocate multiple Ips on the interface*
- *VMaaS will error out if both single and multiple ip variable is specified for the same interface*
- *VMaaS/NwaaS improvements will apply to both SPD and CWD tenants.*

- *VMaaS/NWaaS modify and delete will support the new functionality.*

Usage

1. run NWaas to add new ipv6 network:

```
tenant_name: c001

site: shzja

sub_prefixes:

  - prefix_role: vmanagement_ipv6

    prefix_shared: false

    increase_number: false

    prefix: 'abcd:ef01:2345:6666::/64'

    spd_mapping: 'yes'

    cwd_mapping: 'yes'

    reserved_ips:

      - gateway

      - fw1
```

2. run VMaas to add ipv6 ip address to vm.

```
tenant_name: c001                          # M --> tenant name

virtual_machines:

  - name: VMaasTestVM                          # M --> VM name displayed in the netbox

    device_role: management-vm                    # M --> VM device role

    state: present                          # O --> VM status, "present" means create, "absent" means delete, "modify" means ch

    cluster: c001-mucf-cl01                      # O --> cluster name in this tenant, by default is cluster 01

    update_dns_nat: false                        # O --> Optional, used to skip DNS & NAT registration

    vcpus: 3                                # O --> CPU for this VM

    site: mucfa                             # O --> site name, will used for site_compact

    comments: Demo VM                           # O --> Comments for the VM to describe its function

    tags:                                  # O --> List of tagsTags for the VM to be used for querying later , DO NOT USE role_tag a

      - Demo                                # O --> List of tagsTags for the VM to be used for querying later , DO NOT USE role_tag

    memory: 3000                              # O --> VM memory
```

```
platform: linux                        # O --> VM platform
disk: 10                               # O --> VM disk, defualt is GB
interfaces:                            # M --> VM interfaces list
  - name: eth0                         # M --> interface name
    mode: tagged                       # O --> interface vlan mode , Access/Tagged
    tags:                              # O --> List of tagsTags for the VM to be used for querying later , DO NOT USE role_tag a
      - Demo                           # O --> List of tagsTags for the VM to be used for querying later , DO NOT USE role_tag
    vrf: c001_vrf                      # O --> global vrf/tenant vrf/sub_tenant vrf. If vrf is not set, will allocate vlan/prefix from
    tagged_vlans:                      # O --> tagged vlans
      - vmanagement                    # O --> vlan role, default vaule is auto get vlan role from netbox
    mtu: 9600                          # O --> interface mtu
    multi_address:
     - ip_address: auto                # M --> auto or input a valid ip, auto means get a ip from the prefix pool (map the i
       mgmt_only: true                 # O --> Used for DNS, Set the interface to primary if setting to "true". If not, will set
       nat_inside:                     # O --> It will assign the nat ip for this interface ip, for spd tenant, should delete this na
         prefix_role: nat              # O --> Auto get the nat ip
         address: 192.168.91.3         # O --> Specify the NAT IP address. If nat_inside.prefix_role is defined, do not defi
       prefix_role: vmanagement        # M --> Prefix role name
     - ip_address: auto                # M --> auto or input a valid ip, auto means get a ip from the prefix pool (map the i
       description: test_ipv6          # O --> Description of the IP will be used as the name of  IP. Default is the interface
       prefix_role: vmanagement_ipv6_az1     # M --> Prefix role name
  - name: eth1
    ip_address: auto
    mgmt_only: true
    vlan_role: vmanagement
    prefix_role: vmanagement
    description: test_ipv4             # O --> Description of the IP will be used as the name of  IP. Default is the interface
```

```
nat_inside:
  prefix_role: nat
tags:
- role_vmanagement
```

3. run NWaas to config static route with ipv6 address.

```
static_routes:
  - name: inet
    tenant: s001
    destination: vmanagement_ipv6_az1
    source: test_ipv6
    aci_route_type: L3Out
    l3out: test_zelin
    node_prof: test_zelin_nodeProfile
    router_leaf_ids: all
```

ESXi TPM Recovery Key Store

Introduction

ESXi TPM Recovery Key Store will allow for automated storing of ESXi TPM Recovery keys during OS deployments.

This feature has been supported since LOC-A 3.1 R2405.

Assumptions:

- *TPM Recovery keys will be stored in Netbox Secret Store for each server individually and linked to each server*
- *The TPM Recovery keys will be read after OS Deployment is performed for any workflow type, for example: NWaaS, CWD Deploy/stretch, Add Node/Cluster, CWD Edge Deploy*
- *The TPM Recovery Keys will be supported only for ESXi OS*
- *The TPM Recovery Keys will be removed once the host is decommissioned*
- *The TPM Recovery Keys will not be saved in any files/logs in clear text and will be saved in the Netbox Secret Store in encrypted format.*
- *Feature will be optional and will be enabled via workflow*
- *Loc-A will generate a self-signed public/private key pair for the admin account if no input is provided to the enable workflow and dump the keys as output at the end of the job*
- *root account will be used to read the keys from ESXi.*

- *Loc-A will provide a workflow to enable access to users and will have as input admin keys and user keys, user keys will be generated using self-signed certificates if not provided and dumped as output at the end of the job*
- *Loc-A will provide a workflow to regenerate/replace user keys*
- *All passwords and keys input will be treated as secure input.*

Usage

- run LOC_enable_secret_store_workflow to add user key for admin or loc user or other user.
- run workflows which contains esxi OS deployment normally. if the esxi host support TPM, the workflow will store the TPM key in netbox after OS deployment.
- check TPM key in netbox
- all the secret will be removed when device be put back to pool.

Generic OVA deployment

Introduction

Deploy vm with custom ova file.

This feature has been supported since LOC-A 3.2 R2407.

Assumptions:

- *VMaaS Workflow will be used to define the VM in netbox and map it to the correct cluster.*
- *OVA Deployment workflow will allow for customizing the input for each OVA*
- *OVA Deployment workflow will use jinja templates in the config space to allow for custom input based on OVA being deployed as well as allow for gathering data from netbox for input(Ips, names, DNS, domain etc.)*
- *OVA Deployment workflow will require providing input for any variables that are specific to the OVA such as portgroup_names, custom options such as ssh keys if the OVA requires them as input,etc that cannot be standardized in the OVA template*
- *OVA Deployment workflow will support the Secure Credentials feature to allow secure input of credentials*
- *OVA Deployment location(vCenter/ esxi) will be based on the VM cluster controller node in netbox, as such VM Cluster mapping will be important.*
- *Loc-A Repo server will be used for storing OVA files and will be synced in all environments to ensure OVA availability*
- *Custom OVA files will have a separate folder for storing all the OVA Files outside of the regular cloud repo folder.*
- *OVA file name will be provided as input in order to be able to specify a particular version of an OVA file.*
- *Lenovo PS will assist with creating the jinja templates for each OVA required*

Usage

- run vmaas to create vm in netbox
- run LOC_deploy_ova_vm_workflow to deploy vm

Device Interface Update

Introduction

Device Interface Update will allow T-Systems to configure devices on day 2 by adding extra interfaces to the device that can be used for extra services for any device in Netbox.

This feature has been supported since LOC-A 3.3 R2410.

Assumptions:

- *Device Interface Workflow will be used to define the new interface in netbox and map it to the correct esxi, cluster or tenant.*
- *Device Interface Workflow will allow for specifying a Device, a cluster or a whole tenant for adding interfaces*
- *Device Interface Workflow will allow for specifying different interface names and prefixes only when specifying the devices in a list, for cluster/tenant allocation the same interface name will be added to all devices.*
- *Device Interface Workflow will allow for adding multiple interfaces at the same time when adding interfaces to a cluster or tenant*
- *Device Interface Workflow will allow filtering for device types when specifying a tenant to avoid adding interfaces to unwanted devices. By default the filter will get all devices.*
- *Device Interface Workflow will be able to add interfaces to any device regardless if the device belongs to a tenant or not*
- *Device Interface Workflow will assign 1 ip per interface assuming the prefix_role specified is global or SPD for non tenant devices*
- *Device Interface Workflow will assign 1 ip per interface for Tenant devices using the ip allocation scheme used by VMaaS where the prefix role can be Global, SPD or Tenant and will be differentiated by VRF.*
- *Device Interface Workflow will support adding , modifying and removing of interfaces*
- *Device Interface Workflow will tag any interface created by it as a service interface (tag extra_interface/Service) that can be removed, the interface can also be designated as permanent so it would not be removed by other workflows (tag: none).*
- *Remove CWD/Node/Cluster will remove the interfaces created by Device Interface and remove any IPs assigned by Device Interface Workflow by checking for tag “extra_interface/Service”.*
- *Device Interface Workflow will support adding both IPv4 and IPv6 ips.*
- *Device Interface Workflow will add NAT entries If they are created on the Management edge*
- *Interface IP allocation will always default to auto when adding interfaces to entire clusters or tenants and specifying a static ip will only be supported if the device list is specified*
- *Modify interfaces will not be available in the initial release due to prioritization of requirements.*

Usage

- run LOC_diaas_workflow, for defail input parameters please check the API doc.

LOC-A software tools

LOC-A includes the following software packages:

- AWX: the execution orchestrator on which LOC-A is built. It is used to control workflows.
- GitLab: Serves as the configuration repository service.
- NetBox: Serves as an inventory service

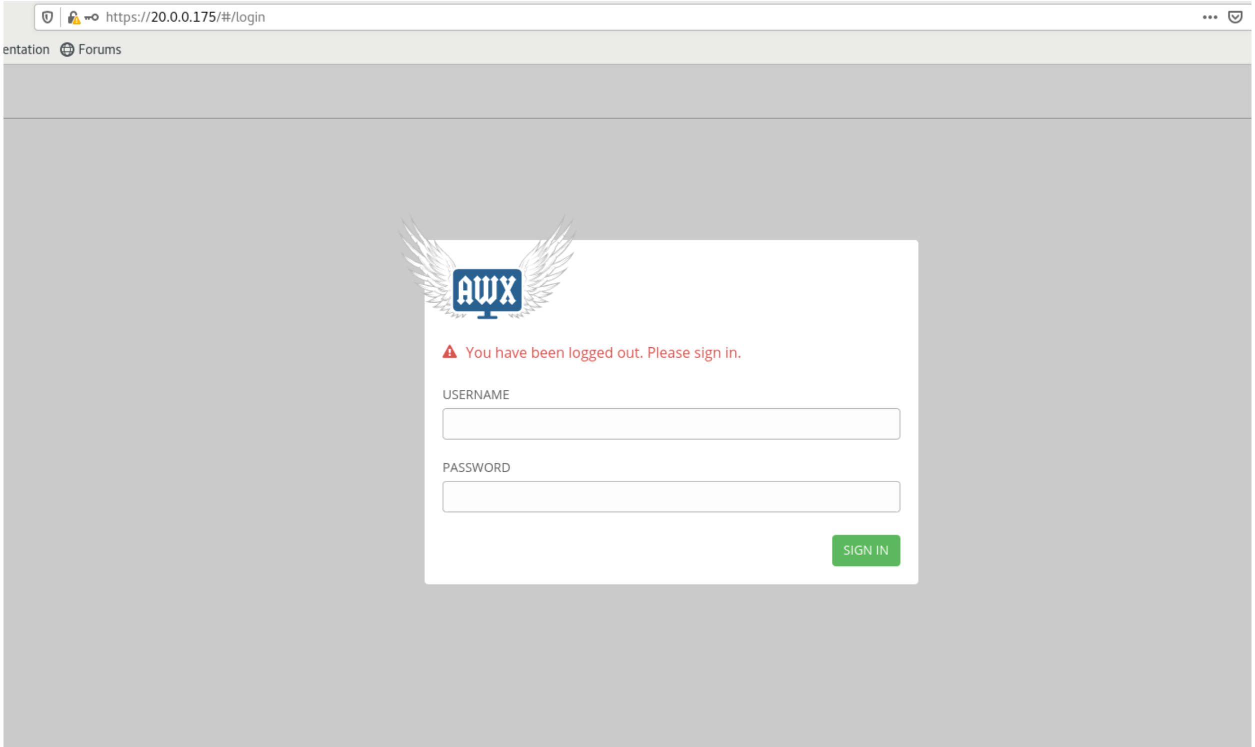
This section provides additional information about using these tools.

AWX

The Professional Services engineer or the System administrator uses the AWX GUI or the AWX CLI to trigger Lenovo LOC-A pre-defined automation workflows.

Method 1: Triggering a workflow from the AWX GUI

1. Log in to
   https://awx_ip/
   The default credentials are admin/Passw0rd.

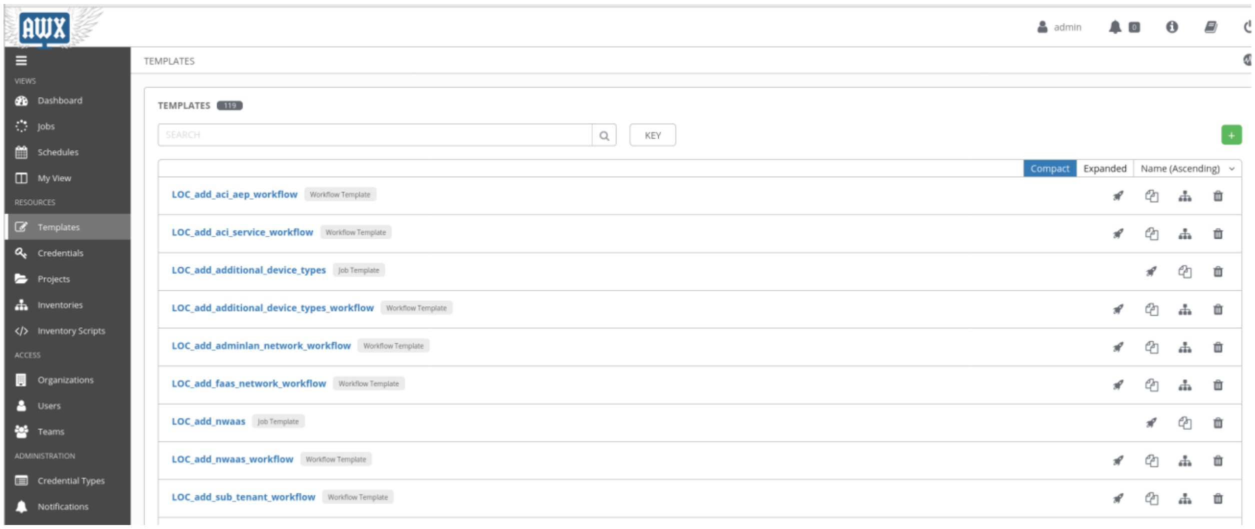


https://20.0.0.175/#/login

You have been logged out. Please sign in.

Figure 23: AWX Login GUI

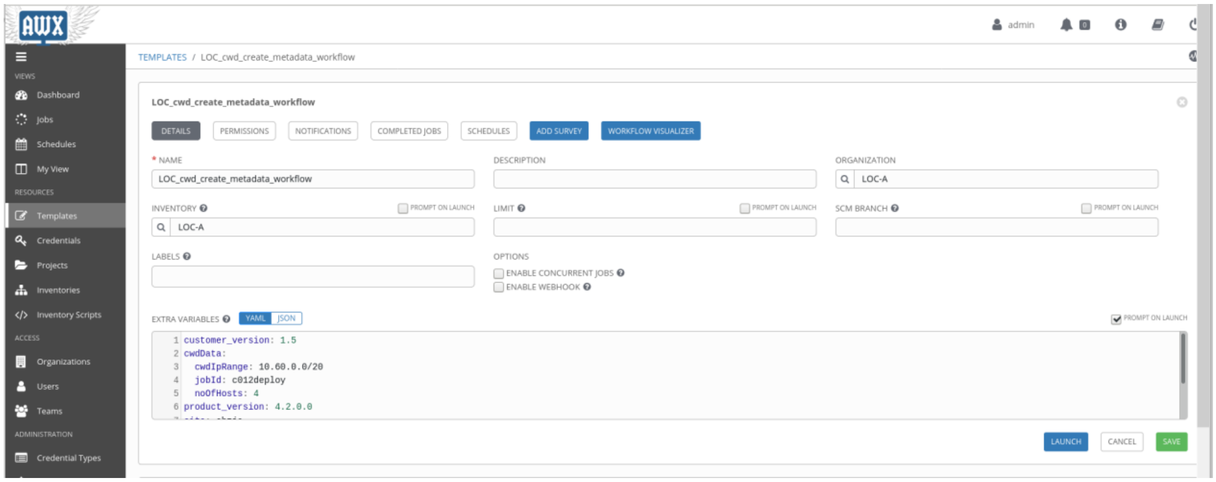
2. Go to the Templates menu and select the workflow template that you want to run. Enter the EXTRA VARIABLES. Then click Save→Launch to run the workflow.



TEMPLATES 119

- LOC_add_aci_aep_workflow Workflow Template
- LOC_add_aci_service_workflow Workflow Template
- LOC_add_additional_device_types Job Template
- LOC_add_additional_device_types_workflow Workflow Template
- LOC_add_adminlan_network_workflow Workflow Template
- LOC_add_faas_network_workflow Workflow Template
- LOC_add_nwaas Job Template
- LOC_add_nwaas_workflow Workflow Template
- LOC_add_sub_tenant_workflow Workflow Template

Figure 24: AWX Job/Workflow Templates



LOC_cwd_create_metadata_workflow

* NAME: LOC_cwd_create_metadata_workflow

DESCRIPTION:

ORGANIZATION: LOC-A

INVENTORY: LOC-A □ PROMPT ON LAUNCH

LIMIT: □ PROMPT ON LAUNCH

SCM BRANCH: □ PROMPT ON LAUNCH

LABELS:

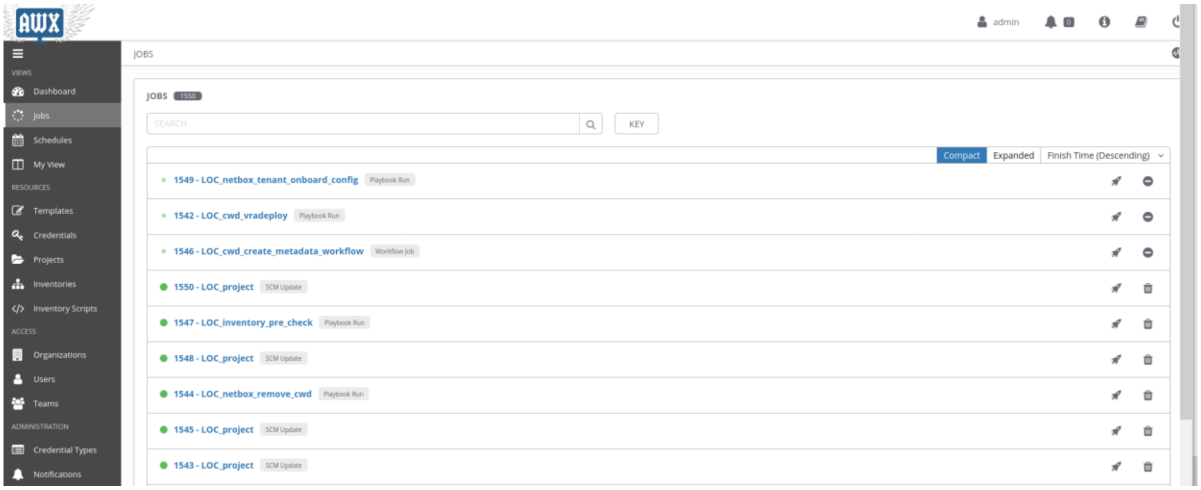
OPTIONS: □ ENABLE CONCURRENT JOBS □ ENABLE WEBHOOK

EXTRA VARIABLES [✓ YAML] JSON ☑ PROMPT ON LAUNCH

```
customer_version: 1.5
cwdData:
  cwdIpRange: 10.60.0.0/20
  jobId: c012deploy
  noOfHosts: 4
product_version: 4.2.0.0
```

Figure 25: AWX Workflow execution with extra variables

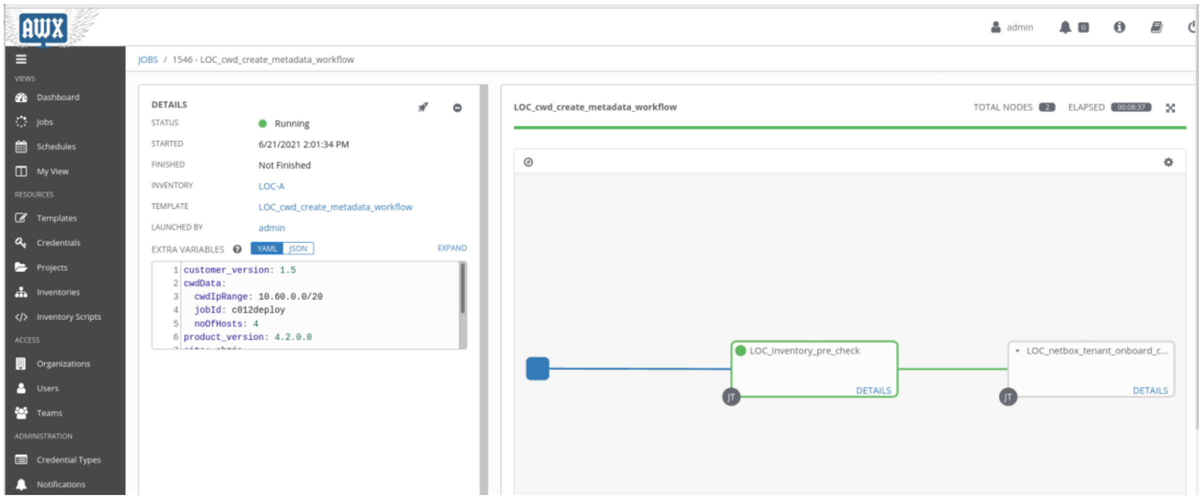
3. You can then view the status of your workflows and jobs. Click a workflow to see additional details about the workflow.



JOBS 1550

- 1549 - LOC_netbox_tenant_onboard_config — Playbook Run
- 1542 - LOC_cwd_vradeploy — Playbook Run
- 1546 - LOC_cwd_create_metadata_workflow — Workflow Job
- 1550 - LOC_project — SCM Update
- 1547 - LOC_inventory_pre_check — Playbook Run
- 1548 - LOC_project — SCM Update
- 1544 - LOC_netbox_remove_cwd — Playbook Run
- 1545 - LOC_project — SCM Update
- 1543 - LOC_project — SCM Update

Figure 26: AWX Workflow status



JOBS / 1546 - LOC_cwd_create_metadata_workflow

DETAILS

STATUS ● Running

STARTED 6/21/2021 2:01:34 PM

FINISHED Not Finished

INVENTORY LOC-A

TEMPLATE LOC_cwd_create_metadata_workflow

LAUNCHED BY admin

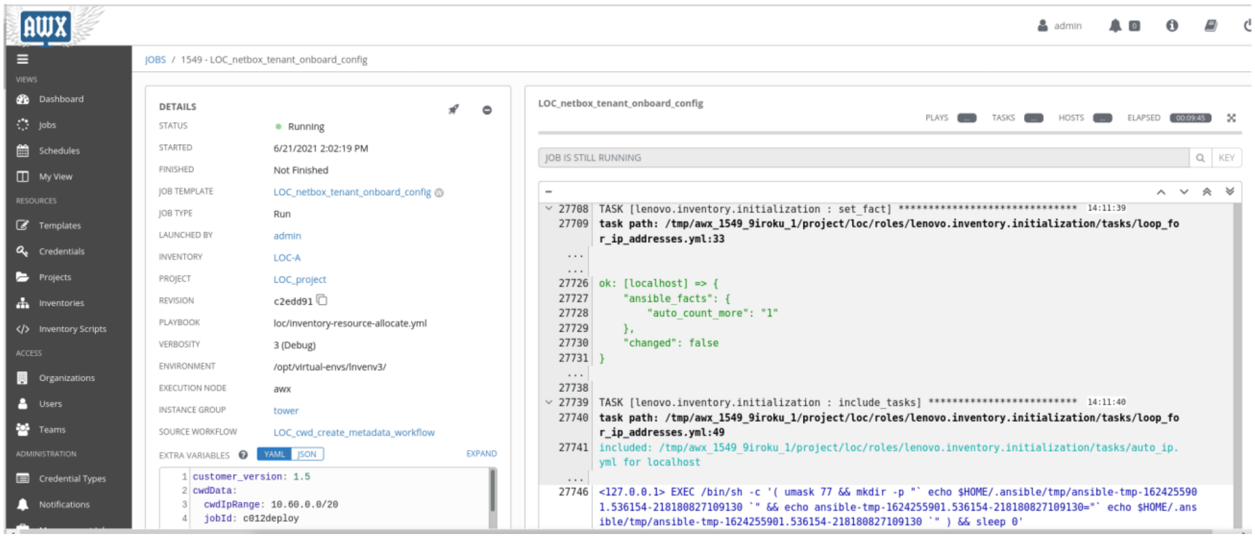
EXTRA VARIABLES YAML JSON EXPAND

```
customer_version: 1.5
cwdData:
  cwdIpRange: 10.60.0.0/20
  jobId: c012deploy
  noOfHosts: 4
product_version: 4.2.0.0
```

LOC_cwd_create_metadata_workflow

Figure 27: AWX Workflow with jobs

4. Click a selected job within the workflow to view the runtime logs of the job.



JOBS / 1549 - LOC_netbox_tenant_onboard_config

DETAILS

STATUS ● Running

STARTED 6/21/2021 2:02:19 PM

FINISHED Not Finished

JOB TEMPLATE LOC_netbox_tenant_onboard_config

JOB TYPE Run

LAUNCHED BY admin

INVENTORY LOC-A

PROJECT LOC_project

REVISION c2edd91

PLAYBOOK loc/inventory-resource-allocate.yml

VERBOSITY 3 (Debug)

ENVIRONMENT /opt/virtual-envs/lnvenv3/

EXECUTION NODE awx

INSTANCE GROUP tower

SOURCE WORKFLOW LOC_cwd_create_metadata_workflow

EXTRA VARIABLES YAML JSON EXPAND

```
customer_version: 1.5
cwdData:
  cwdIpRange: 10.60.0.0/20
  jobId: c012deploy
```

LOC_netbox_tenant_onboard_config

JOB IS STILL RUNNING

```
27708 TASK [lenovo.inventory.initialization : set_fact] ****************************** 14:11:39
27709 task path: /tmp/awx_1549_9iroku_1/project/loc/roles/lenovo.inventory.initialization/tasks/loop_fo
      r_ip_addresses.yml:33
  ...
  ...
27726 ok: [localhost] => {
          "ansible_facts": {
              "auto_count_more": "1"
          },
          "changed": false
      }
27727
27728
27729
27730
27731
  ...
27738
27739 TASK [lenovo.inventory.initialization : include_tasks] ************************ 14:11:40
27740 task path: /tmp/awx_1549_9iroku_1/project/loc/roles/lenovo.inventory.initialization/tasks/loop_fo
      r_ip_addresses.yml:49
27741 included: /tmp/awx_1549_9iroku_1/project/loc/roles/lenovo.inventory.initialization/tasks/auto_ip.
      yml for localhost
  ...
27746 <127.0.0.1> EXEC /bin/sh -c '( umask 77 && mkdir -p "` echo $HOME/.ansible/tmp/ansible-tmp-162425590
      1.536154-218180827109130 `" && echo ansible-tmp-1624255901.536154-218180827109130="` echo $HOME/.ans
      ible/tmp/ansible-tmp-1624255901.536154-218180827109130 `" ) && sleep 0'
```

Figure 28: AWX Job logs

Method 2: Triggering a workflow from the AWX CLI

For example: if you need to run the pre-defined workflow "LOC_rack_device_onboard_workflow" to onboard server and racks.

1. create a file /tmp/vars.conf with content:

```
rack_name: '00'
location: 'rtp'
location_counter: 'f'
availability_zone: 'a'
site: datacenter1
```

2. Run the following commands:

   awx -k workflow_job_templates modify LOC_rack_device_onboard_workflow --extra_vars @/tmp/vars.conf --conf.host https://<your_awx_ip> --conf.username admin --conf.password Passw0rd

   awx -k workflow_job_templates launch LOC_rack_device_onboard_workflow --wait

   Note: The -k or --conf.insecure option is added because AWX HTTPS interface uses self-signed certificates by default. If a user certificate signed by trusted CA is imported, this option is not needed.

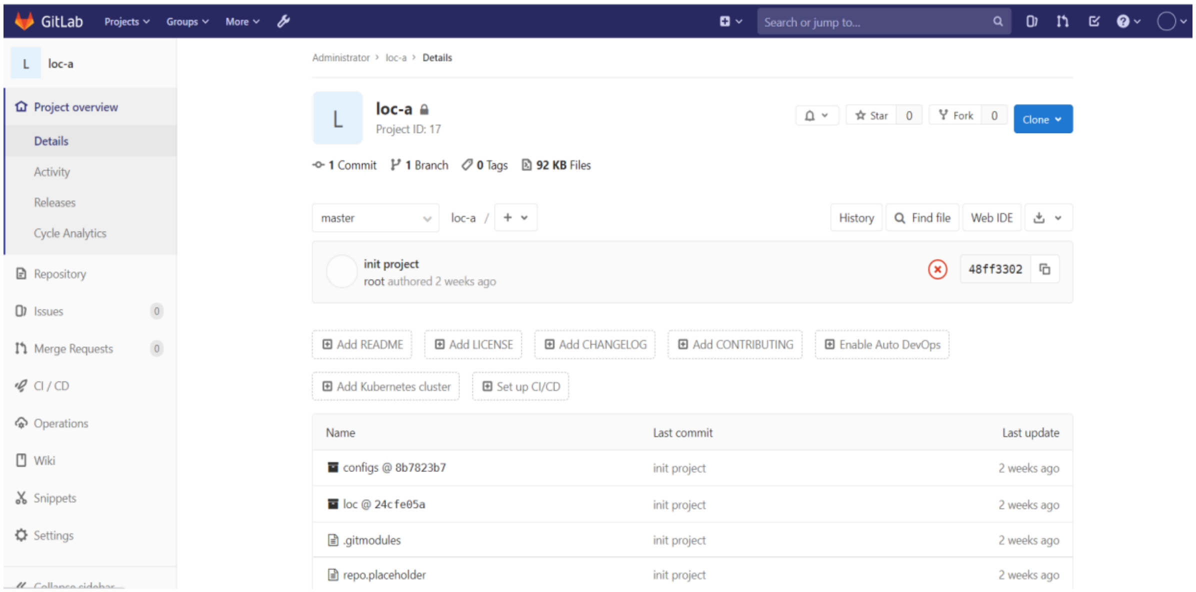
   Refer to https://docs.ansible.com/ansible-tower/latest/html/towercli/reference.html for more information about the AWX CLI.

GitLab

LOC-A uses three repositories on the Gitlab service.

- loc: Contains the Ansible automation playbooks/roles for task automation.
- configs: Contains customer andservice-input configuration files required for task automation. All configuration files are stored encrypted using ansible-vault with a customer pre-defined password. The vault password is defined in the user *loc_services_input.yml* file and will be applied when the Bootstrap and SPD LOC-A services are deployed.

  Important: Make sure you keep the vault password you set in configuration file.
- loc-a: This is the placeholder repository that contains the loc and configs repositories as submodules. This repository is then be associated with AWX for project workflow executions.



Name	Last commit	Last update
configs @ 8b7823b7	init project	2 weeks ago
loc @ 24cfe05a	init project	2 weeks ago
.gitmodules	init project	2 weeks ago
repo.placeholder	init project	2 weeks ago

Figure 29: GitLab Repository Layout

A Professional Service engineer or system administrator can log in to the GitLab GUI to manage and view the contents of the repositories. The default credentials are: root/Passw0rd.

A Professional Service engineer can also view or modify the code via git clone, commit, push operations using HTTPS or SSH on a server that has access to the LOC-A services (eg. on the Runner)

To clone the configs repository:

1. Run one of the following commands:
   git clone https://root:<PASSWORD>@<your_gitlab_ip>/root/configs.git
   or
   git clone git@<your_gitlab_ip>:root/configs.git

   Note:
   - GitLab uses self-signed certificate, you need to disable SSL verification for git clones using https :
   git config --global http.sslVerify false
   - If using git@ via SSH, you need to add your SSH pub key into the GitLab as an authorized key.

2. Edit the following files:
   cd configs/
   ansible-vault edit $your_file_to_edit
   (Input the password: yourvaultpass to view and edit the files. After you have finished, use :wq! to save and quit (similar to vim)
3. Commit the changes:
   git commit -m "any message you want to input for this change"
   git push

NetBox

LOC-A uses NetBox to store the information about hardware, networks, and VMs. It also allocates them during the deployment. A Profession Service engineer or system administrator can use the GUI to view and check the NetBox data.

The GUI of NetBox is accessible via https://<netbox_ip>/. The default credentials are admin/admin.

The Professional Service engineer or system administrator can view or modify the GUI admin user password from in this page after logging in to the website:
netbox-ip:port/admin/auth/user/1/password/

The NetBox metadata is updated via AWX workflows.

Troubleshooting

loc-services cannot connect to switch, cwd / spd tenant cluster

Complete the following steps to resolve the issue:

1. Verify that the switch configuration is correct.
2. Verify that the settings for route, edge and port group are correct.
3. Verify that the configuration of the loc-services gateway is correct.
4. Verify that the ***/etc/resolv.conf*** configuration of loc-services is correct (nameserver should be an external DNS).

Could not update DNS

Complete the following steps to resolve the issue:

Error: 'name or server not known' resulted from DNS cannot resolve domain name of itself.

1. Check the following:
   a) The nameserver in the ***/etc/resolv.conf*** file should be the IP of the DNS itself.
   b) Check the listen-on port 53 in ***/etc/named.conf {ip;}***, The IP here should be the IP of the DNS itself.

c) Check whether there is a domain record for DNS in the ***/var/named/master/s001/c001.xxxx*** configuration file.

If these are correctbut the DNS is still unresolvable, check whether the DNS IP started by the named service is the ***ip ss -nap | grep named*** of the DNS itself. If not, restart the named service.

2. Error 'rc9' is because the zone that maintain the DNS domain record in the file ***/etc/named.conf*** in the DNS host is not configured.

3. To solve the problem of 'update key error,' make sure that the pre-generated update key in the DNS template is consistent.

LXCA could not manage a node

Complete the following steps to resolve the issue:

1. Check the job status of LXCA. If a job has been completed and the node status is normal, but the add host job of AWX is failed, retry the workflow.
   Note: This can occur because of an issue in ***pylxca.***
2. Check the job status of LXCA. The job has been completed but the node status is offline. Check if the node's XCC gateway is or is not an IP in the route, if not, change and retrying.

LXCA could not deploy an operating system

Complete the following steps to resolve the issue:

1. Verify that the node on which the operating system will be deployed has a UUID in NetBox. Make sure the necessary UUID parameters are set when deploying the OS.
2. If the deployment fails with a UUID provided, check the status of the hardware. Ensure that there are no errors or warnings on the node. If node status is normal but not normal in LXCA, try restarting the XCC on the node.
3. If the task of deploying the operating system stops at 25%, open a Remote Console to the node and verify that it is connected to the LXCA timeout. If so, the network configuration is wrong. Check the switch configuration.

Could not deploy the VCF

To resolve the issue, check the log of VCF builder or log in to the GUI to see where the issue happened.

The following logs are generated on the builder during the VCF deployment:

Task	Log File Location
Cloud Builder VM	/opt/vmware/bringup/logs/vcf-bringup.log /opt/vmware/bringup/logs/vcf-bringup-debug.log
JSON generation	/opt/vmware/sddc-support/cloud_admin_tools/logs/ JsonGenerator.log
JSON file validation	/opt/vmware/sddc-support/cloud_admin_tools/logs/ PlatformAudit.log
Bring-up tasks	/var/log/vmware/vcf/bringup/vcf-bringup-debug.log

Table 14: VCF Deployment Logs

During deployment, you can monitor /opt/vmware/bringup/logs/vcf-bringup-debug.log for detailed information

- The Password must meet the following requirements:
  - Password must contain 8-20 characters in length
  - Password must contain at least one upper case letter
  - Password must contain at least one special character [!%@$^#?]
  - Password must contain at least one digit.
- The license must be valid.
- The NTP server must be available. DNS must be normal and the domain record for the VMs has been configured and can be resolved forward and backward.
- If everything is normal, but a vSAN still cannot be created, retry the deployment workflow.

Could not deploy vROps

Because vrops is deployed through the SDDC manager, you can log in to the GUI of SDDC to view the deployment task for any issues.

Verify the following:

- The license must be the ***vrops 7***.
- The vROps password must meet the following requirements:
  - Password must contain 8-20 characters in length
  - Password must contain at least one upper case letter
  - Password must contain at least one special character [!@$^#?]
  - Password must contain at least one digit.
  - Password cannot contain the percent character [%].

Could not expand the VCF cluster

When you try to rerun the expand workflow, it might not succeed because of some nodes are already in the cluster, so you can delete the nodes from cluster and rerun the expand workflow.

Follow below steps:

1. Remove a host from a Cluster in a Workload Domain

1. On the SDDC Manager Dashboard, click **Inventory→ Workload Domains**.
   The Workload Domains page displays information for all workload domains.
2. In the workload domains table, click the name of the workload domain that you want to modify. The detail page for the selected workload domain appears.
3. Click the Clusters tab.
4. Click the name of the cluster from which you want to remove a host.
5. Click the Hosts tab.
6. Select the host to remove and click **Remove Selected Hosts**.
   An alert appears, asking you to confirm or cancel the action. If the removal results in the number of hosts in the cluster being less than the minimum number of required hosts, you must click **Force Remove** to remove the host.
7. Click **Remove** to confirm the action.

   The details page for the cluster appears with a message indicating that the host is being removed. When the removal process is complete, the host is removed from the hosts table.

   The host is removed from the workload domain and added to the free pool.

2.Decommission Hosts

1. On the SDDC Manager Dashboard, click **Inventory→Hosts**.
2. Click **Unassigned Hosts**.
3. In the hosts table, select the one or more hosts to be decommissioned.
4. Click **Decommission Selected Hosts**.
5. Click **Confirm**.

Known Issues and Limitations

- Concurrent workflow execution for VCF deployment is supported only when “UPDATE REVISION ON LAUNCH” is not selected for LOC Project in AWX. Manually configure this on AWX if you need to enable concurrent deployment.
  Note: If you enable concurrent deployment, automatic updates will be disabled if code in the loc repository is updated on GitLab.
- The automatic update of the LOC-A services VM does not support changing network settings (eg. IP address, gateway, etc..) for LOC-A. If the LOC-A services VMs are already created and the network settings need to be updated, the settings must be updated manually.
- When importing devices into NetBox, verify that you do not have duplicate IP addresses. LOC-A does not check for duplicate IP addresses.
- If issues occur during an SPD or CWD deployment workflow, running the workflow again will start the redeployment of the tenant from the beginning (not from where the issue occurred for the last deployment workflow.
- Tenant name of the SPD VCF cluster must be s001 in LOC-A.
- Self-signed certificates are used by default for the HTTPS interfaces of GitLab, NetBox, and Confluent. Automatically importing customer-signed certificates is not supported in LOC-A.
- No automatic validation of password rules is implemented; you must make sure that all passwords adhere to the minimum password requirements for the different components of LOC-A services or VMware.
- The minimum of diskgb_expand for vidm is 4 G. The maximum number of disks is 7. So vrli can scale disk 5 times; vrops can scale disk 4 times; for vidm, the disk_size is separated to 2 disks, so only can scale disk 2 times; vrni can scale disk 6 times.

Appendix

A. End User License Agreement (EULA)

Lenovo License Agreement

L505-0009-06-R2

This Lenovo License Agreement (the “Agreement”) applies to each Lenovo Software Product that You acquire, whether it is preinstalled on or included with a Lenovo hardware product, acquired separately, or downloaded by You from a Lenovo Web site or a third-party Web site approved by Lenovo. It also applies to any updates or patches to these Software Products. This license agreement does not apply to non-Lenovo software that’s either preloaded on or downloaded to your product. This Lenovo License Agreement is available in other languages at https://support.lenovo.com/us/en/solutions/ht100141.

Lenovo will license the Software Product to You only if You accept this Agreement. You agree to the terms of this Agreement by clicking to accept it or by installing, downloading, or using the Software Product.

If You do not agree to these terms, do not install, download, or use the Software Product(s).

- If You acquired the Software Product(s) and paid a license fee, return the Software Product to the party from whom You acquired it to obtain a refund or a credit of the amount You paid.

- If You acquired the Software Product(s) preinstalled on or provided with a Lenovo hardware product, You may continue to use the hardware product, but not the Software Product(s) covered under this Agreement.

"Open Source software" means any computer program, including any modification, improvement, derivative work, release, correction, governed by the terms and conditions of an Open Source license.

"Open Source License" means a license that gives you legal permission to freely use, modify, and share the Open Source software and is

(i) approved by the Open Source Initiative (here after OSI) principles defined in the following website: https://opensource.org/osd and/or
(ii) certified by the OSI (cf. list of such licenses in https://opensource.org/licenses/category) and/or
(iii) compliant with the free software foundation criteria and/or
(iv) that requires the human readable source code of software to be made available to the general public.

"Software Product" includes Lenovo computer software programs (whether preinstalled or provided separately) and related licensed materials such as documentation.

"You" and "Your" refer either to an individual person or to a single legal entity.

1. Entitlement

You must maintain Your original dated sales transaction document, such as a receipt, invoice or similar document, as Your proof of Your right to use the Software Product. The transaction document specifies the usage level acquired. If no usage level is specified, You may install and use a single copy of the Software Product on a single hardware product. Your transaction document also provides evidence of Your eligibility for future upgrades, if any. For Software Products preinstalled on, included with, or distributed at no charge for use on a Lenovo hardware product, Your hardware product sales transaction document is also the proof of Your right to use the Software Product.

2. License

The Software Product is owned by Lenovo or a Lenovo supplier, and is copyrighted and licensed, not sold. Lenovo grants You a nonexclusive license to use the Software Product when You lawfully acquire it.

You may a) use the Software Product up to the level of use specified in Your transaction document and b) make and install copies, including a backup copy, to support such use. The terms of this Agreement apply to each copy You make. You may not remove or alter any copyright notices or legends of ownership.

If You acquire the Software Product as a program upgrade, after You install the upgrade You may not use the Software Product from which You upgraded or transfer it to another party.

You will ensure that anyone who uses the Software Product (accessed either locally or remotely) does so only for Your authorized use and complies with the terms of this Agreement.

You may not a) use, copy, modify, or distribute the Software Product except as provided in this Agreement or in any way that violates any applicable laws including but not limited to copyright laws; b) reverse assemble, reverse compile, or otherwise translate the Software Product except as specifically permitted by law without the possibility of contractual waiver; or c) sublicense, rent, or lease the Software Product.

Lenovo may terminate Your license if You fail to comply with the terms of this Agreement. If Lenovo does so, You must destroy all copies of the Software Product.

Lenovo uses the System Update program to update Software Products on Your computer. By default, critical updates are downloaded and installed automatically. Updates are classified as critical when they

are needed for the computer to function properly. Failure to install critical updates could result in data corruption or loss, a major system malfunction, or a hardware failure. For example, critical updates could include an update to the harddisk-drive firmware, a BIOS upgrade, a device-driver fix, or a fix for the operating system or other preinstalled software. You can disable this automatic feature by changing the settings of the System Update program at any time.

3. Transferability

   You may not transfer or assign the Software Product to any other party, except as permitted in this section.

   Preinstalled Software Products are licensed for use only on the Lenovo hardware product on which they are preinstalled or included with and may be transferred only with that Lenovo hardware product. They may not be transferred independent of the Lenovo hardware product.

4. Open Source and Other Third Party Software Components and Products

   Portion(s) of the Software Products and future updates and patches provided hereunder may include Open Source software licensed under a particular Open Source License. To the extent that the terms of this Agreement conflict with the terms of such Open Source License, then the terms of such Open Source License shall control for such applicable Open Source software. For the sake of clarity, for any portion(s) of the Software Products, which is not governed by such Open Source License, this Agreement shall control.

   Some Lenovo Software Products and future updates and patches may contain third party components, which may include Microsoft Windows Preinstallation Environment. These third party components are provided to You under separate terms and conditions different from this Agreement, typically found in a separate license agreement or in a README (or similarly titled) file. The third party's license terms and use restrictions will solely govern the use of such components.

   Third Party Software Products provided by Lenovo may be governed by the terms of this Agreement but are usually licensed by the Third Party under its own terms and conditions. Third Party Software Products that are not licensed by Lenovo are subject solely to the terms of their accompanying license agreements.

5. Software Product Specifications

   The Software Product specifications and specified operating environment information may be found in documentation accompanying the Software Product, if available, such as a README or similarly titled file, or otherwise published by Lenovo.

6. Privacy

Please review the Lenovo privacy policy statement (http://www.lenovo.com/privacy/software/) that's associated with Your product. Depending on Your particular Lenovo device or software product, the Lenovo privacy statement is located at the point of activation and set-up and/or via "Settings".

7. Charges

   Charges for the Software Product are based on the level of use acquired.

   If You wish to increase the level of use, contact Lenovo or the party from whom You acquired the Software Product. Additional charges may apply.

   If any authority imposes a duty, tax, levy or fee, excluding those based on Lenovo's net income, upon the Software Product, then You agree to pay the amount specified or supply exemption documentation. You are responsible for any personal property taxes for the Software Product from the date that You acquire it.

8. No Warranty

   The Software Product(s) is provided to You "AS IS." SUBJECT TO ANY STATUTORY WARRANTIES WHICH CANNOT BE EXCLUDED, LENOVO MAKES NO WARRANTIES OR CONDITIONS, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE

   IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, REGARDING THE SOFTWARE PRODUCT OR TECHNICAL SUPPORT, IF ANY.

   The exclusion also applies to any of Lenovo's developers and suppliers.

   Suppliers or publishers of non-Lenovo Software Products may provide their own warranties. Lenovo does not provide technical support, unless Lenovo specifies otherwise in writing.

9. Limitation of Liability

   Circumstances may arise where, because of a default on Lenovo's part or other liability, You may be entitled to recover damages from Lenovo. In each such instance, regardless of the basis on which You are entitled to claim damages from Lenovo (including fundamental breach, negligence, misrepresentation, or other contract or tort claim), except and to the extent that liability cannot be waived or limited by applicable laws, Lenovo is liable for no more than the amount of actual direct damages suffered by You, up to the amount You paid for the Software Product. This limit does not apply to damages for bodily injury (including death) and damage to real property and tangible personal property for which Lenovo is required by law to be liable.

   This limit also applies to Lenovo's suppliers and resellers. It is the maximum for which Lenovo, its suppliers and resellers are collectively responsible.

   UNDER NO CIRCUMSTANCES IS LENOVO, ITS SUPPLIERS OR RESELLERS LIABLE FOR ANY OF THE FOLLOWING EVEN IF INFORMED OF THEIR POSSIBILITY: 1) THIRD PARTY CLAIMS AGAINST YOU FOR DAMAGES; 2) LOSS OF, OR DAMAGE TO, YOUR DATA; OR 3) SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, BUSINESS

REVENUE, GOODWILL, OR ANTICIPATED SAVINGS. SOME STATES OR JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

10. Consumer Rights

Nothing in this Agreement affects any statutory rights of consumers that cannot be waived or limited by contract. You may have additional consumer rights under applicable local laws, which this Agreement cannot change.

11. General

a) In the event that any provision of this Agreement is held to be invalid or unenforceable, the remaining provisions of this Agreement remain in full force and effect.

b) You agree to comply with all applicable export and import laws and regulations.

c) Neither You nor Lenovo will bring a legal action under this Agreement more than two (2) years after the cause of action arose unless otherwise provided by local law without the possibility of contractual waiver or limitation.

12. Dispute Resolution

If You acquired the Software Product in Cambodia, Indonesia, Philippines, Vietnam or Sri Lanka, disputes arising out of or in connection with this Software Product shall be finally settled by arbitration held in Singapore and this Agreement shall be governed, construed and enforced in accordance with the laws of Singapore, without regard to conflict of laws. If You acquired the Software Product in India, disputes arising out of or in connection with this Software Product shall be finally settled by arbitration held in Bangalore, India. Arbitration in Singapore shall be held in accordance with the Arbitration Rules of Singapore International Arbitration Center ("SIAC Rules") then in effect. Arbitration in India shall be held in accordance with the laws of India then in effect. The arbitration award shall be final and binding for the parties without appeal and shall be in writing and set forth the findings of fact and the conclusions of law. All arbitration proceedings shall be conducted, including all documents presented in such proceedings, in the English language, and the English language version of this Agreement prevails over any other language version in such proceedings.