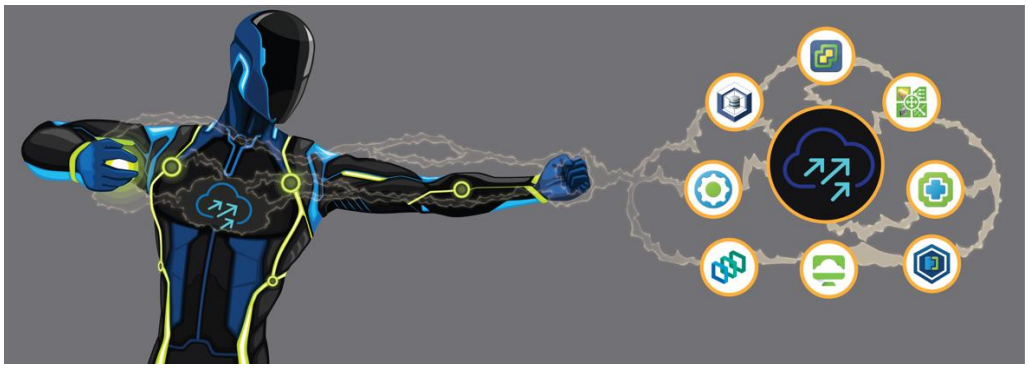


Document Version 4.5 Rev 1 Created by Kevin Tebear, Ben Sier @datareload and Heath Johnson @heathbarj

Support Options

Questions about VCF ask @sddccommander
Support for VLC on Slack vlc-support.slack.com
Public join link for Slack: https://tiny.cc/getVLCSlack

----------<Disclaimer>----------

If you are reading this that means you want to test VMware Cloud Foundation (VCF) on Non-Certified Hardware in a nested environment. Just a word of caution before we start. While this does work for testing and demo's, it is not supported by VMware GSS. Use of this software is at your own risk, we are not responsible for damage to any systems. This software should only be used in a non-production lab environment. Also, because we're nesting three layers deep this will impact performance. So, if you are planning on showing any of this as a demo to a customer. Please communicate to all customers that it will be slower than a physical environment.

----------</Disclaimer>----------

Table of Contents

Welcome to VMware Cloud Foundation Lab Constructor

New Features for Version 4.5

- Supernet/Summary route for NSX traffic
- External DNS Field
- Dynamic naming of Temp Directory and generated ISO
- Headless mode

Fixes

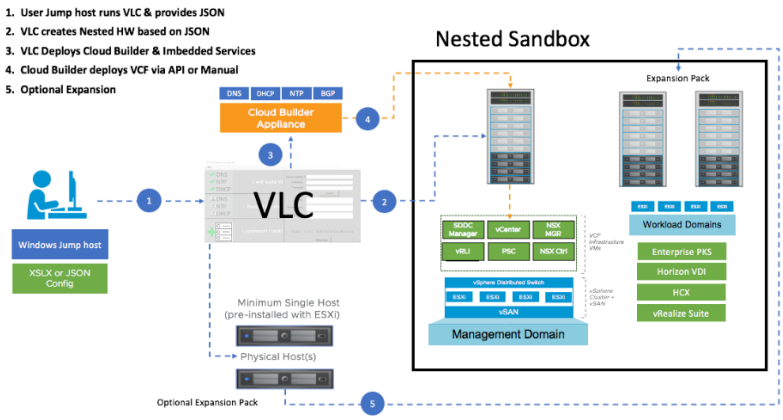
- UI Navigation Fixes
- UI Validation Fixes
- vSphere HA network Gateway setup in Automated mode
- Increased default add_3_hosts RAM to 32gb to allow NSX Edge deployment for a WLD

Overview

VCF Lab Constructor (VLC) is a utility intended to automate the deployment of VCF Cloud Foundation (VCF) in a nested environment. Although this is not a supported or recommended method for deploying VCF, this allows individuals to learn about VCF with a greatly reduced set of resource requirements.

In addition to deploying the nested infrastructure needed to support a VCF deployment, VMware Lab Constructor can also automate the deployment of VCF and configure required services, if desired.

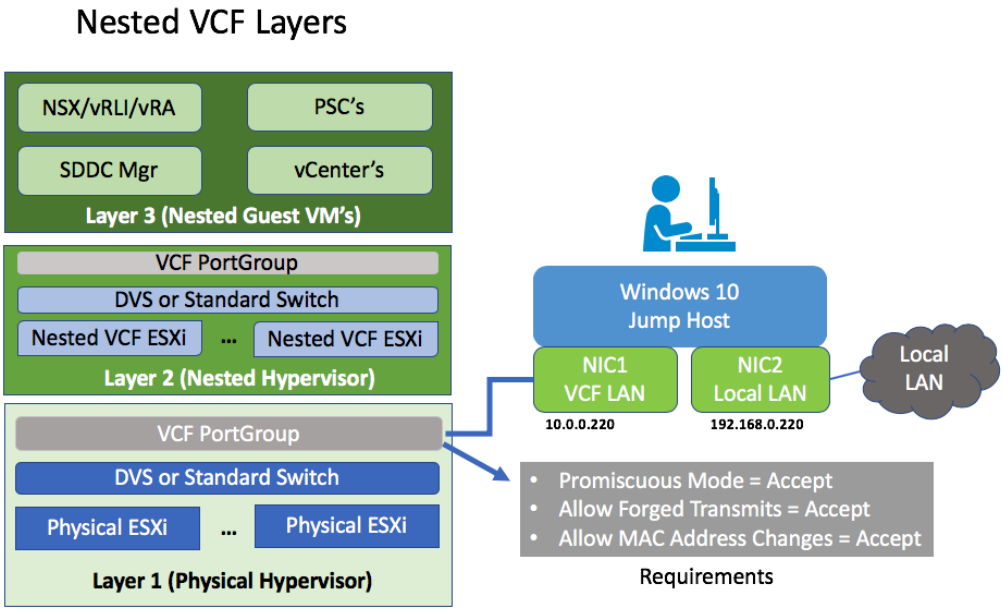
The general flow of operations in a standard VLC enabled environment is depicted in the diagram below:



VCF Lab Constructor Overview

This document will not explain what VCF is. Please see the VMware Documentation for that. This will only cover how this nested version works, how to set it up, and access it.

Let's start by explaining how we are virtualizing a full hardware set for VCF. Below is a physical to logical view of the setup.



Creating Layer 1

Layer 1 is your physical lab equipment. This can be one host or many hosts setup in a cluster with a vCenter. This can also run on a physical vSAN cluster. Providing the physical equipment to run this on is up to you. The requirements for the physical layer are listed below in the detailed requirements section.

Creating Layer 2 – Learning the 3 Methods

Layer 2 and layer 3 are installed by the VLC script. VLC will begin by setting up layer 2. (i.e., this will create a nested virtual version of the hardware requirements for VCF.)

Method 1 – Automated

Using Method 1 means you want this VLC software to provide the required services for you.

Method 2 - Manual (Expert Mode)

Using Method 2 assumes you will provide the required services as spelled out in the VCF documentation. This includes DNS, NTP, and potentially DHCP and BGP routing services. Refer to the official VMware VCF documentation for exactly what is required.

Method 3 – Expansion Pack

Using this method assumes that you have already built a complete lab using either Method 1 or 2 and now you want to add more nested hosts to this existing environment or, you'd just like to create some nested vSAN hosts for another purpose or lab.

Building your first four Hosts with Method 1 or 2

Selecting Method 1 or 2 will build your first four hosts for the Management Domain. This is done by creating four virtual nested ESXi hosts. These nested hosts are automatically sized and created for you. You are able to configure the hostnames and IP addresses to be used within the configuration file that you provide the VCF Lab Constructor.

The VCF Lab Constructor package comes with one sample file:

- NOLIC-45-vcf-ems-public.json

This file not only defines the information for the nested hosts to be created, but it also will be used to complete the VCF bring-up process.

If you would like to change the size of CPU, Memory or Storage of the initial management domain hosts edit the following file:

- <VLC Dir>\conf\default_mgmt_hosthw.json

You can modify these files to suit your particular lab or preferences. It's advisable to create duplicates of the included files before you modify them.

You are required to insert your license keys into the configuration file to be used.
Look for the "licenseFile": "<INSERT LIC>", sections in the JSON file.

Creating Layer 3 - Cloud Builder – Bring-up

The next phase is called the "Bring-up process". This is a fully documented process in the installation of VCF. Using the VCF Lab Constructor will allow you to manually do this so you can follow the steps of the official VMware Documentation, or if you check the box in the GUI the VCF Lab Constructor will complete Bring-up for you automatically.

During the bring up process VCF will use the four nested hosts and build them into a vSAN cluster and create the Management Domain.

Once bring up is complete, you will have a fully deployed VCF Instance in your nested environment.

Accessing the VCF UI

Because the VCF components are installed on Layer 3 you may be thinking "How do I get network access to it?". To gain access, your deployment will need a router to span a portgroup with internet access, or a jump host with multiple NICs. This is diagramed below in the jump host requirements. Be aware that because everything is nested inside Layer 2 all network traffic is being broadcast back up to Layer 1 port groups. Simply having your jump host on this subnet or port group and listening on the default VCF subnet i.e. (192.168.0.0) will allow you to access everything in layer 3. The jump host can also be nested at layer 1 or a physical desktop that has access to the same subnet. Nesting it at Layer 1 has the best performance.

Headless Mode

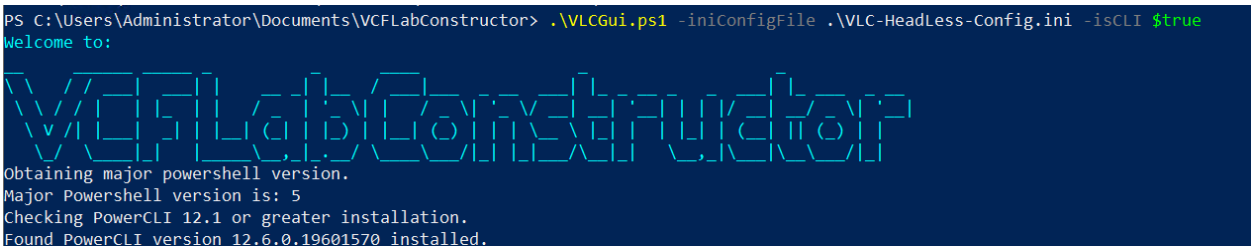
Headless mode is a new mode that allows you to run VLCGui.ps1 without the UI. At this time only the Automated deployment, meaning all infrastructure services live on CloudBuilder, is supported. This is useful for quickly deploying or redeploying an environment with no changes to configuration.

To create an INI file to use for headless mode, simply launch the UI and configure everything you want, then save the config using the File -> Save menu. The INI that's produced can be used for both headless runs, or future UI runs to save time.

*Note: If you have a INI file from a previous version of VLC, you will need to recreate it in this version due to file structure changes.

Headless mode can be utilized by running VLCGUI.ps1 with the following parameters

```
VLCGui.ps1 -iniConfigFile .\VLC-HeadLess-Config.ini -isCLI $true
```



This concludes the overview of the VCF Lab Constructor.

Environment Prerequisites

ESXi or vCenter

You need a single physical host running ESXi 7.0+ with 12 cores, 128 GB RAM and 800 GB SSD. This is the minimum requirement for using VLC, however the more resources the better, especially if you will create WLD's and use integrations like vRA/vROps or Tanzu.

VLC supports the following configurations, most of the documentation will illustrate a standalone ESXi host:

1. Standalone ESXi (no vCenter) – Using VSS
2. ESXi host with vCenter – Using VSS
3. Single ESXi host in a Cluster - Using VDS
   (Note: Services are required to be stopped)
4. Multiple ESXi hosts in a Cluster – Using VDS
5. Multiple ESXi hosts in a vSAN Cluster – Using VDS
   (Note: This requires the use of extra ESX CLI commands. See below for more details.)

Set the vSwitch to MTU = 9000

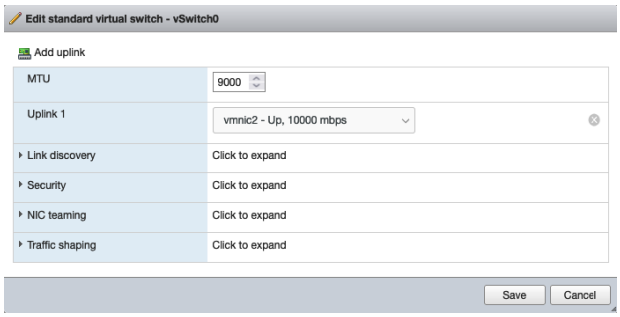


Figure 1: Depicted from standalone ESXi Host

On the vSwitch, create a portgroup for VCF. On the port group (not the switch) set the following VLAN ID and security settings:

- VLAN ID = 4095 (Set portgroup Trunk if DVS)
- Promiscuous Mode = Accept
- Allow Forged Transmits = Accept
- Allow MAC Address Changes = Accept

Make sure you do not have a DHCP server running anywhere that the trunk port above can access unless you are using manual deployment and intending to use it for TEP IP's on a specific VLAN.

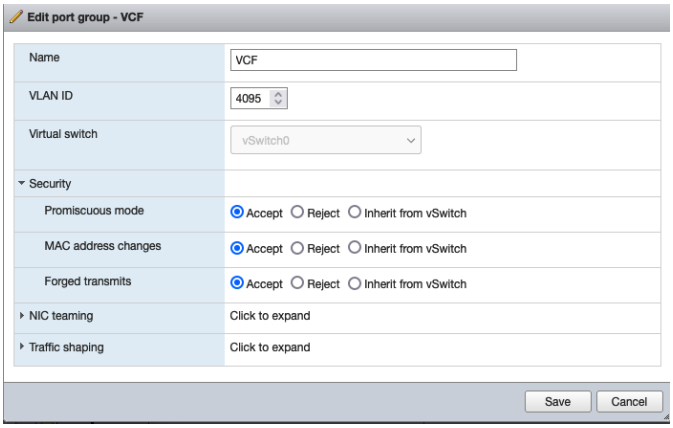


Figure 2: Port group example from a standalone host

If in a cluster configuration, disable all HA and DRS and vMotion on the **physical host(s).**

Jump Host

Build a Windows-based jump host on this ESXi host as a VM on the infrastructure to which you are deploying with the following configurations:

- Windows 10/2016/2019 (Older versions are not supported)
- Powershell 5.1+
- PowerCLI 12.1+

- OVFTool 4.4+
- .Net Framework
- One or Two VMXNET3 NICs
    - MTU 1500

If you build a jump host with one NIC you will need a router to provide outbound connectivity to the internet as well as inbound connectivity to with NAT and FW rules to access RDP on the jump host. While not documented here you can find examples of router setup on https://datareload.com. Your single NIC would need to be connected to the Portgroup created above and set to the VLAN of your VCF Management Network.

On a two NIC jump host, attach two virtual NICs.

- Attached one NIC to your local LAN Network so you can RDP to it.
- Attach the second NIC to the VCF port group created in Step 1 and configure it with the IP 10.0.0.220. Set the DNS on the second NIC to 10.0.0.221. The 10.0.0.221 address will be the address assigned to the Cloud Builder appliance, by default. VLC will modify the Cloud Builder appliance so that it provides specific services, like DNS, for the nested environment. Thus, using this IP for DNS will allow you to access the nested VCF environment when using the default configuration file in Automated mode.

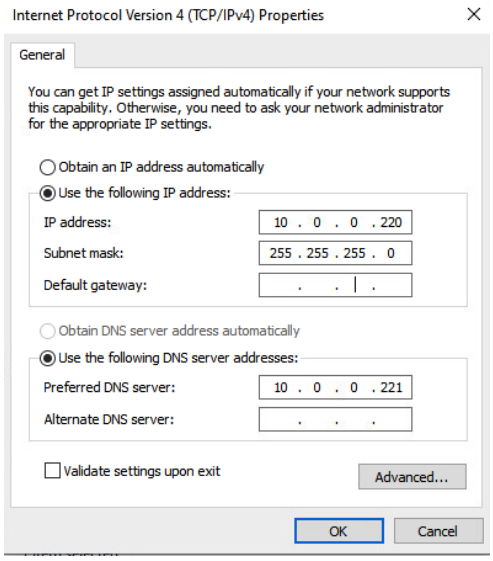


Figure 3: IP settings on Windows jump host

*If you change the config files you will need to change the DNS setting here to match.

- This second NIC will also need to be configured to use the VLAN of your management network, in the default Automated VLC configuration this is VLAN 10:

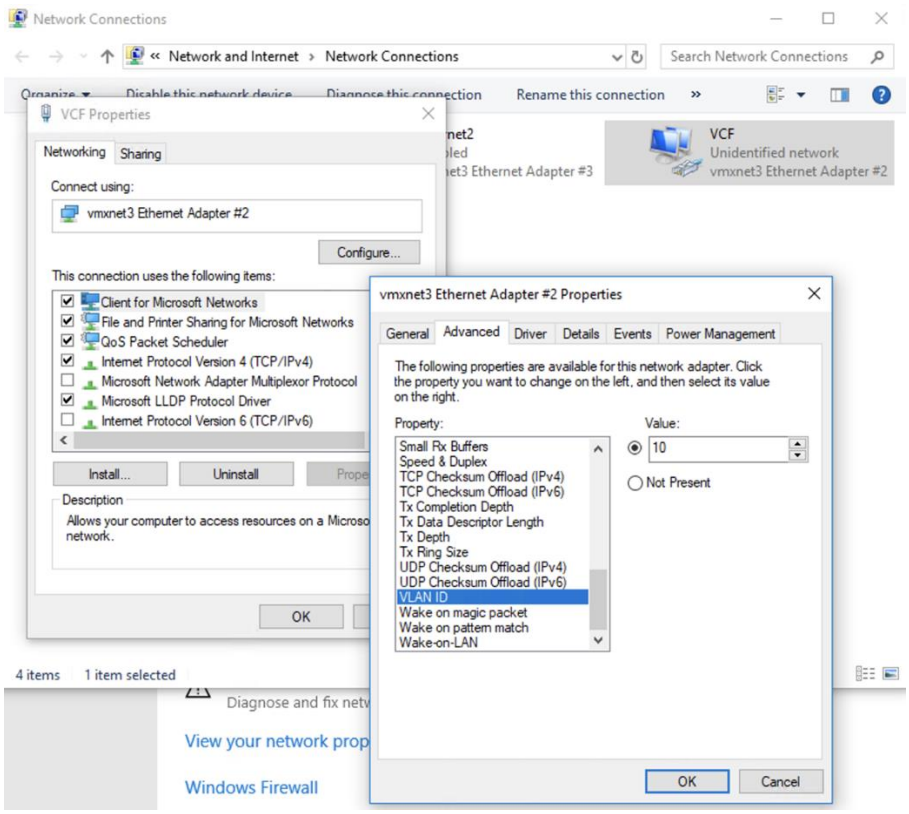


Figure 4: Setting VLAN ID on Windows jump host

Note: Use of a jump host with a single NIC will work, but proper routing must be configured for all networks that VLC uses. In the above example the gateway in Figure 3 could be set to 10.0.0.1 assuming the physical network had that available on VLAN 10. If you choose to use the dual nic setup as we have shown above you may need to create static routes for any logical networks deployed and point them to the CloudBuilder IP address, an example here:

```
route ADD 10.50.0.0 MASK 255.255.255.0 10.0.0.221
```

Disable Windows Firewall.

Turn off Windows Defender Real-time Scanning. Note: this has a habit of resetting after reboots of the Windows VM.

Software

On the Windows jump host, create a local disk folder for VLC. This must be a local attached disk (i.e. “C:\VLC\ ”) as mapped Network drives will fail.

Download the VCF Software (Cloud Builder OVA) into this folder.

(Optional) Download the vSphere ESXi ISO that matches the version required for VCF. The easiest method to do this is to simply copy the .iso file located on the Cloud Builder appliance.

To make this even easier, VLC now provides an option where it will download this file directly from the Cloud Builder appliance that it deploys.

Download and extract the VLC package to this folder as well

Optional software to download on the jump host:

- Putty - or your favorite SSH tool
- Notepad++ - or your favorite text editor
- WinSCP - for easy file transfers

Deployment JSON

In 4.5 there is a single json file to edit for the purpose of completing a basic deployment:

- **Insert your license keys into the NOLIC-45-vcf-ems-public.json file.**
- **Look for all of the lines in the JSON file containing:**
  - **"licenseFile": "<INSERT LIC>"**
- **Replace the quoted text with a valid license key**
- **Do not edit anything else in the JSON**

Quick Start Guide – using "Automated"

Utilizing the "Automated" approach will provide the fastest experience to deploy VCF using VLC.

Basic Deployment Steps

Either open a Powershell window (as Administrator) and execute the VLC PowerShell Script "C:\VLC\VLCGUi.ps1" or right click on the VLCGUI.ps1 and select 'Run with PowerShell'.

VLC UI will Launch

Select the "Automated" Button

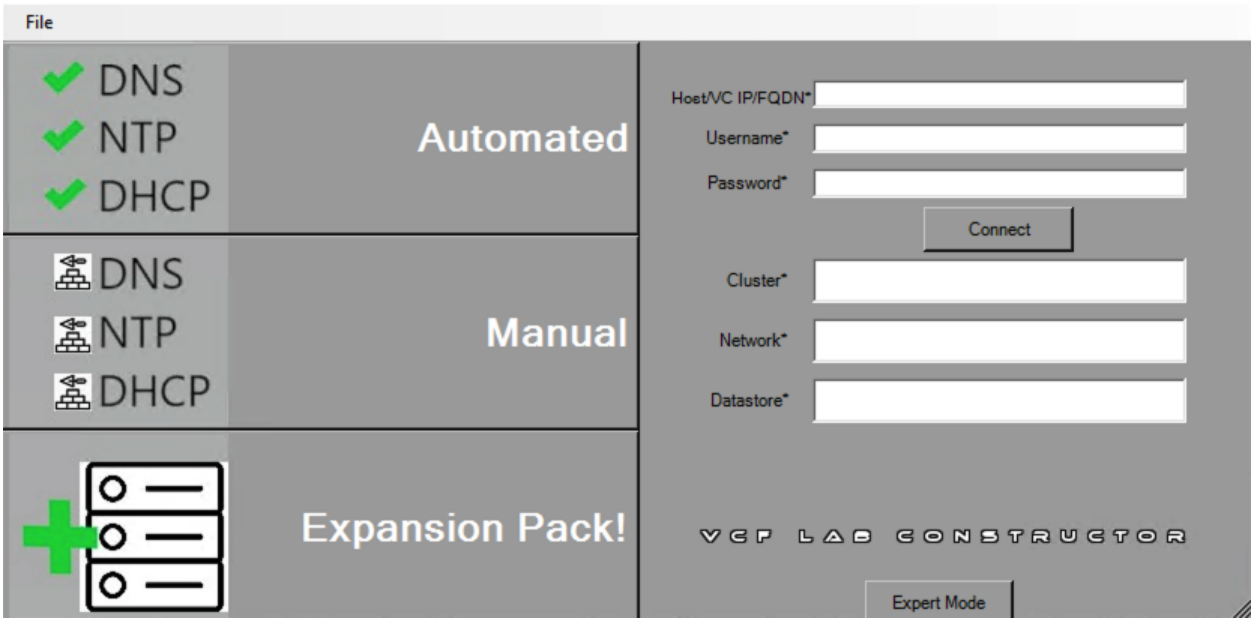


Figure 5: Initial VLC Launch UI

Click on the field titled 'VCF EMS JSON' and select the JSON file that you just entered the license keys for.
Click on the CB OVA Location field to select the location of the CB OVA.
(Optional) Enter the External GW and/or External DNS for the Cloud Builder Appliance to use. This allows you to point to a gateway that will allow internet access or a specific external DNS server.
On the right side of the UI, enter the information for where you wish to deploy to. This this can be a vCenter Login or a direct ESXi login, depending on your lab.

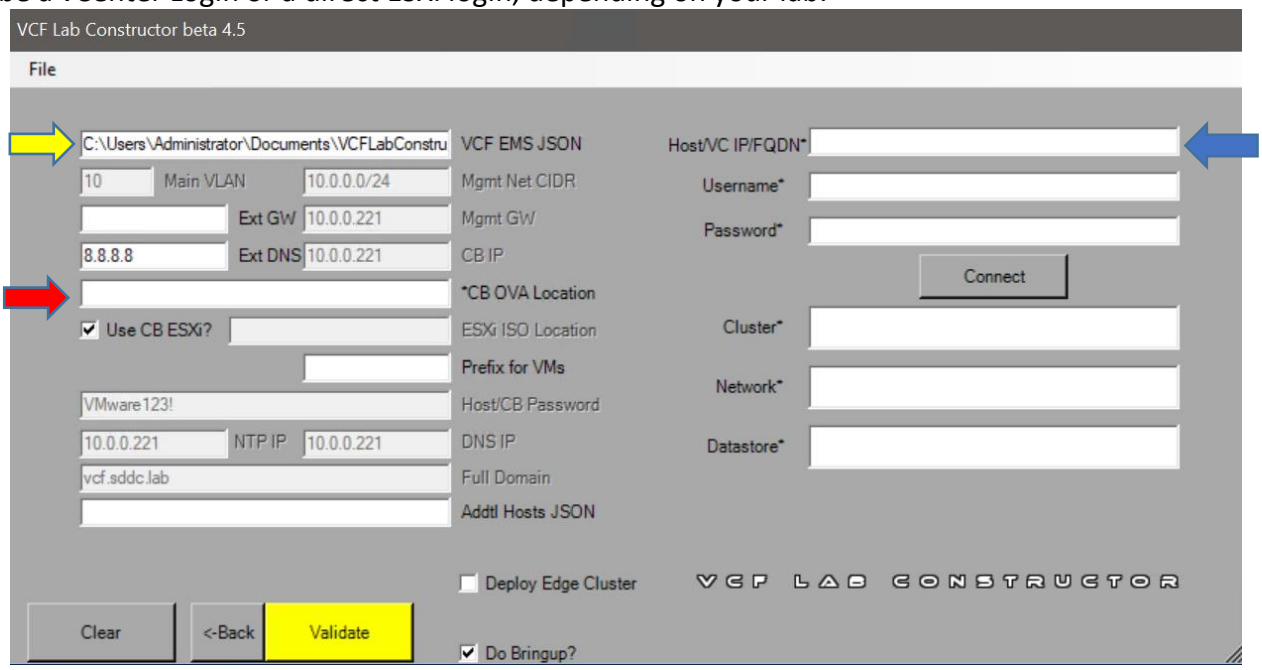


Figure 6: Automated VLC configuration initial UI

Click the Connect Button

VLC will connect to the host or vCenter you specified and will validate all necessary settings. It will then populate the Cluster, Network, and Datastore fields with information gathered from your environment.

Select the cluster, network (port group) and datastore that you desire VLC to install the nested lab to. The Cluster field will not display any value if you are deploying directly to a single ESXi host.

** If your port group does not show up, you may check to see if the following has been set explicitly on the port group and **not just the switch**:

- Promiscuous Mode = Accept
- Allow Forged Transmits = Accept
- Allow MAC Address Changes = Accept

Click the yellow Validate button
As VLC validates the information, it will mark the fields in green. When everything has been validated, the Validate button will change to a green button that says 'Construct'. If you are ready to do a basic deployment
Click Construct to begin or continue reading a few more options for an automated deployment of VMware Cloud Foundation.

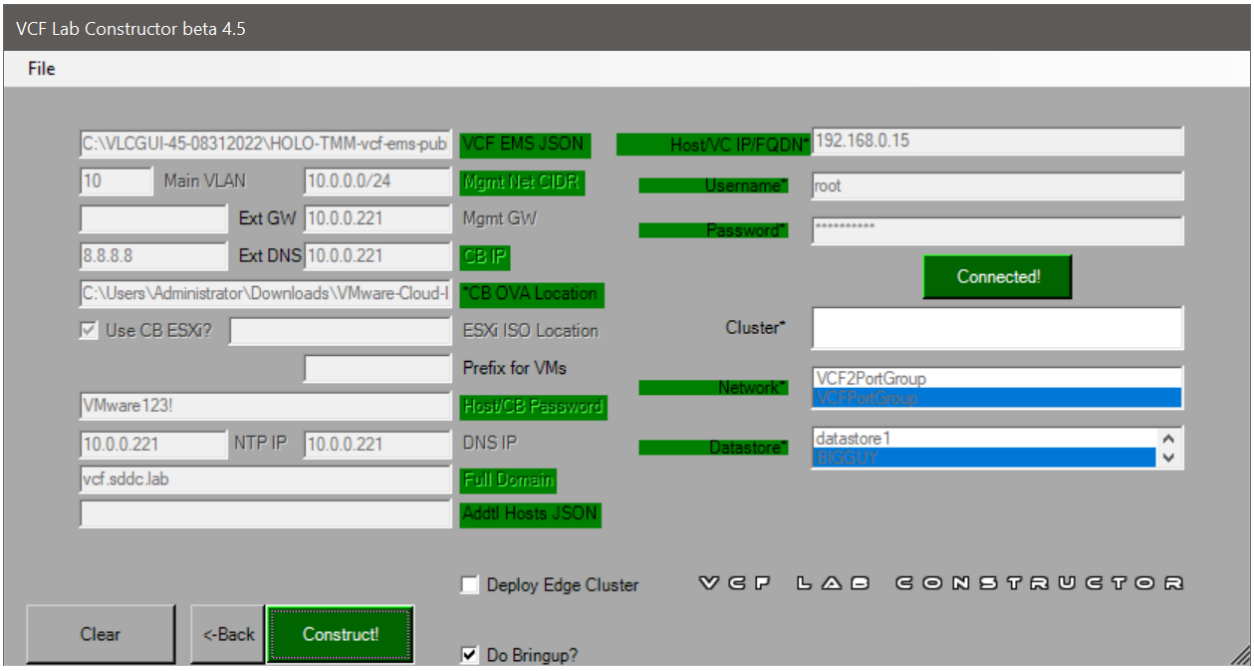


Figure 7: Populated and Verified Basic VLC Deployment

(Optional) Choice of Edges and AVN when Deploying VLC

What is AVN? See this blog

In 4.5 we have included a variety of AVN centric deployment options:

- No Edge deployment
- Edge deployment, but no AVN's
- Edge deployment and AVN's created

To facilitate these new use cases and aligning to AVNs as a Day N operation VLC now includes a few more configuration files. If you are using all the default values in Automated mode and only update the bringup JSON file by adding your license keys there is no need to modify any of the additional files. The files are located in the C:\VLC\automated_api_jsons:

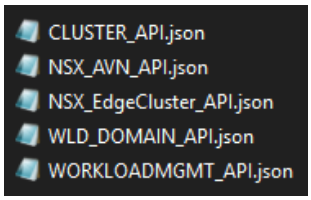


Figure 8: List of API/JSON files

1. NSX_EdgeCluster_API: This is the configuration file you would edit if you choose to make changes to the BGP configuration, gateway, or VLAN configuration.
2. NSX_AVN_API: This is the configuration file you would edit to make changes to the AVN's that VLC will create.

For VLC to use these json files to create edges and AVN's for you, before hitting Construct select the option to 'Deploy the Edge Cluster' highlighted with yellow arrow.

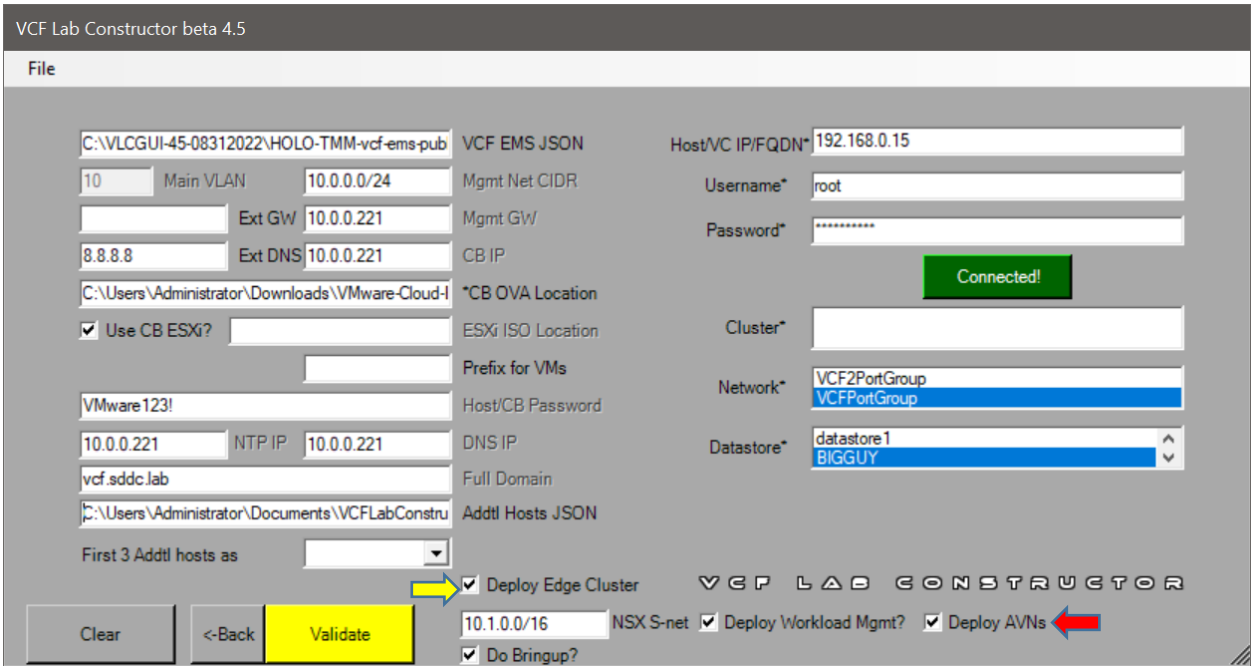


Figure 9: Deploy Edge Cluster VLC Option

If this option is selected the C:\VLC\automated_api_jsons\NSX_EdgeCluster_API.json will be used and a pair of edges will be deployed and peered via BGP to the services running on CloudBuilder. The CloudBuilder will also have a route to the edges injected for the NSX S-Net subnet listed.

Additionally, if you select the Deploy AVNs option highlighted with the red arrow the C:\VLC\automated_api_jsons\NSX_AVN_API.json file will be used and the AVNs will be created

**Note if you opt out of these during deployment you would need to utilize SDDC Manager to deploy the edge cluster and AVNs. Product documentation links for these actions:

Edge Cluster Creation
https://docs.vmware.com/en/VMware-Cloud-Foundation/4.5/vcf-admin/GUID-8FA66DA3-3166-426B-84A8-C45FA7651658.html

Create AVNs
https://docs.vmware.com/en/VMware-Cloud-Foundation/4.5/vcf-admin/GUID-59E5BEE3-B157-426D-A40C-F21171586863.html

(Optional) Adding Hosts, Workload Domains, and Clusters

As you may have seen above there were a few more files you can configure in the C:\VLC\automated_api_jsons\ folder

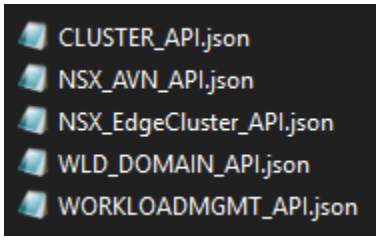


Figure 10: Additional JSON's to configure

1. Add_3_hosts.json: while technically not in this directory this file is located in C:\VLC will be our first file that may require some configuration. This will create 3 additional hosts and automatically commission them to SDDC Manager during the VCF Deployment. That would mean Management Domain created plus 3 hosts available to be commissioned.
2. CLUSTER_API: This will configure a 3 host cluster (first 3 hosts in the add_3_hosts.json file) will be used to create an additional cluster under the Management Domain.
3. WLD_DOMAIN_API: This will create a Workload domain with the values in this configuration file for vCenter and NSX using the first 3 hosts in the add_3_hosts.json file.
4. WORKLOADMGMT_API: This will enable Workload Management on the Management cluster and create a single namespace where you can instantiate Kubernetes deployments.

*Note: More information on what you can change in the above files is found later in this document

This is a very powerful option that allows for the skipping of several subsequent tasks such as utilizing the expansion pack in addition to commissioning the hosts to SDDC Manager and either creating a cluster or a WLD. Note: you cannot create a cluster and a WLD while deploy with VLC, you will need to pick one or the other.

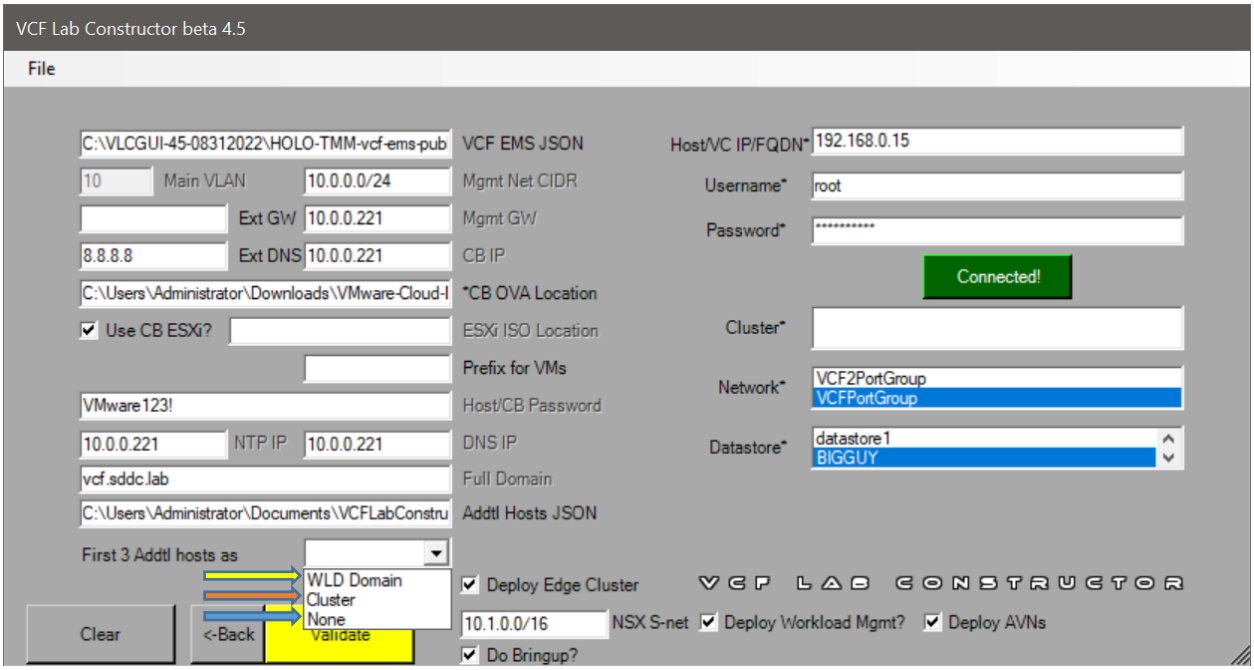


Figure 11: Creating Additional hosts, cluster, or WLD

Providing the Addtl Hosts JSON field and pointing at the add_3_hosts.json field will automatically commission them to the system if the ⟹ option None is selected.

Using the ⟹ option of Cluster will create an additional cluster in the MGMT Workload Domain of 3 hosts based upon the json.

Using the ⟹ option of WLD Domain will create an additional WLD (vCenter, NSX, and Cluster) using the first 3 hosts based upon the json.

How much Time?

The deployment and configuration process will take some time to complete. On average, a basic deploy could take more than 3.5 hours depending upon speed of underlying hardware (most deployments complete in 2.5 hours). Adding an Edge cluster and AVN's could add more than 1 hour. Adding hosts could add more than 15 minutes in deployment time. Adding a cluster could add more than 30 minutes. Adding a Workload Domain could add more than 1 hour.

Lifecycle Management – Tips

If you plan to test lifecycle management in this nested environment there will be a few fixes that you will want to ensure are put in place to give the greatest success:

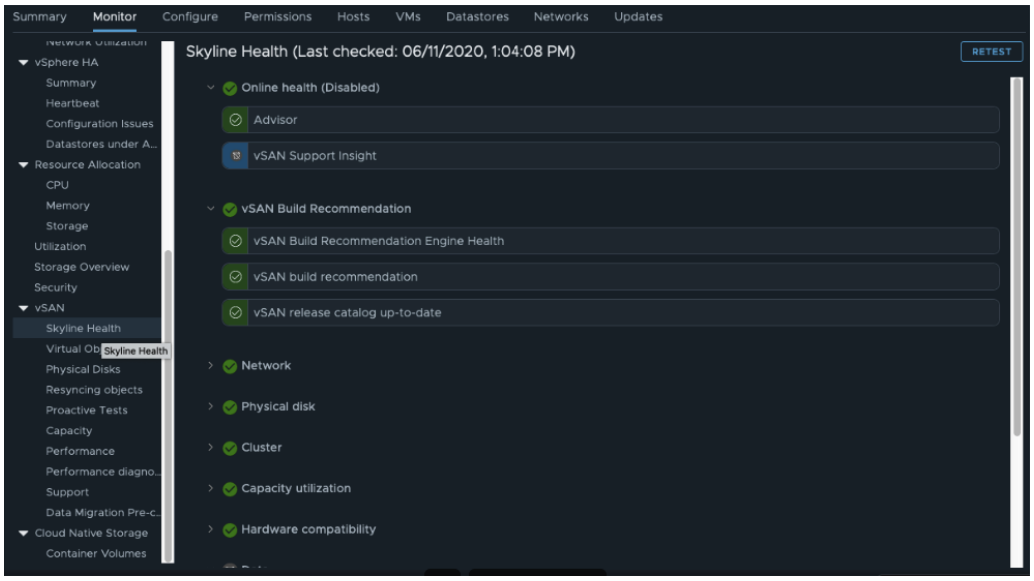
1. edit the following file /opt/vmware/vcf/lcm/lcm-app/conf/application-prod.properties file:

```
################### VSAN CONFIGURATION ###################
vsan.healthcheck.enabled=false
vsan.hcl.update.enabled=false
vsan.precheck.enabled=false
```

   After editing this file restart lcm services:

```
systemctl restart lcm
```

2. vSAN/vCenter will throw errors either during pre-check, or during upgrade.In addition go to HA settings of every cluster and disable admission control and change DRS to partially automated on the nested VCF Deployment.
3. Couple of things to note for nested upgrades though! Go into vSphere Client>Cluster>Monitor>vSAN>Skyline Health and silence all the alerts and run retests there to ensure vSAN reports 'green' Click on view reply to see more details.



Advanced Expert Mode Notes

Physical Install Requirements

Advanced Install

Use Case - Complete basic install and additionally be able to create multiple Workload Domains or install additional optional components (vROPs, vRA, vRNI)

4+ Sockets – 12+ Cores Each

Total RAM Requirements – 768GB – 2TB RAM

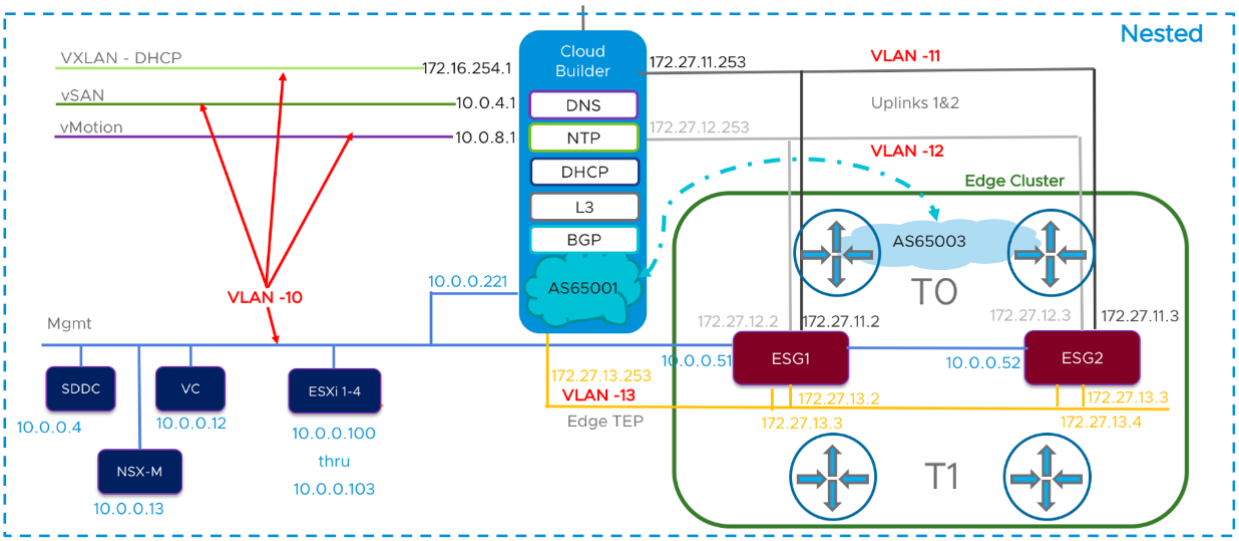
Note: your mileage may vary based upon how many of these optional components you choose to install. The more RAM and CPU you can provide the better.

- **Physical Hardware Running vSphere 7.0 or Higher (It may work on prior versions of vSphere but has not been tested)**
    - 2TB+ GB of Disk (preferably high-speed flash for best results)
        - If you try to deploy this on spinning disk, you may spend forever troubleshooting. **You have been warned!**
- **Note:** If you modify the included JSON for Bring-up or use the XLSX, ensure that the MTU of the vSAN and vMotion Network Pools is set to 8940. This is required because the nested packet being passed up to the Physical Layer DVS needs overhead packet capacity
- Disable all HA and DRS and VMotion on the Physical Host(s)
- Build your own L3 BGP Router. Vyos and PFsense are most common. (Required for VCF 3.9.1 and optional for VCF 4.0)
    - Instructions here include the use of a VLC built-in router injected using the *Automated* button with VLC using Cloud Builder. You can also use your own existing physical or virtual router. Regardless of the configuration you chose to use, the end requirement is to have a working BGP router correctly configured within the environment.
- **Optional – Other Scenarios**
    - If you are running vSAN on physical hosts, run this command on all hosts
        - esxcli system settings advanced set -o /VSAN/FakeSCSIReservations -i 1
    - If you are using an ESXi host that is controlled by a vCenter and you want to target that single host **instead** of the vCenter you will need to do the following:
        - Stop communication between the host and the vCenter Server by stopping these services with the commands:
            - /etc/init.d/vpxa stop
            - /etc/init.d/hostd restart
        - After these commands have executed, the ESXi host will have stopped communicating with the vCenter Server. At this point, execute the VLC script to build the lab. After the deployment completes, start the vpxa service to add the ESXi host back to vCenter by running the command:
            - /etc/init.d/vpxa start

Expert Mode – Manual

If you select the Manual deployment model, you are required to provide all the required VCF services. This includes DNS, DHCP, NTP, and BGP routing.

When using the Automated deployment model, VLC configures the Cloud Builder appliance to provide the required services. It will also automatically install software packages on the Cloud Builder appliances to support these services. This is represented in the diagram below. Use this diagram as a basis to build a similar network configuration for your environment, utilizing whatever you desire for these core services.



In additional to providing the network services, performing the Manual Mode of deployment requires you to provide suitable JSON files for the various options selected. Sample files are included with VLC (There is more info on these files in the sections above):

- NOLIC-45-TMM-vcf-ems-public.json
  - A sample configuration for completing bring up
- add_3_hosts.json
  - A sample file creating additional nested hosts when using VLC
- conf/default_mgmt_hosthw.json
  - Nested hardware specs for CPU,Memory and disk for the management cluster hosts

Optional during initial run in Automated mode. (These filenames are hard coded in the script and must remain the same if used). You are able to change settings detailed below, however you'll need to ensure that they are the same across files if needed and the same across your infrastructure and services (like Ips, VLANs, DNS FQDNS, etc..)

- automated_api_jsons/WLD_DOMAIN_API.json
    - Sample file for creating a workload domain when this option is selected in VLC UI
    - You can change:
        - "domainName"
        - In vcenterSpec
            - "name"
            - "rootPassword"
            - "datacenterName"
            - "vmSize"
            - "storageSize"
            - in networkDetailsSpec
            - "ipAddress","dnsName","gateway" and "subnetMask"
        - "name" in clusterSpecs
        - "datastoreName" in vsanDatastoreSpec
        - "failuresToTolerate" in vsanDatastoreSpec
        - "name" in vdsSpecs -Ensure host vdsName(s) match
        - "name" in portGroupSpecs
        - In nsxTSpec and nsxManagerSpecs
            - "name","ipAddress","dnsName","gateway","subnetMask","vip",vipFqdn","nsxManagerAdminPassword" and "formfactor"
        -
    - Host ID's, licenseKey, geneveVlanID and domainId will be obtained by VLC from SDDC Manager

- automated_api_jsons/CLUSTER_API.json
    - Sample file for creating a cluster when this option is selected in VLC UI
    - You can change:
        - "name" in clusterSpecs
        - "datastoreName" in vsanDatastoreSpec
        - "failuresToTolerate" in vsanDatastoreSpec
        - "name" in vdsSpecs -Ensure host vdsName(s) match
        - "name" in portGroupSpecs
    - Host ID's, licenseKey, geneveVlanID and domainId will be obtained by VLC from SDDC Manager

- automated_api_jsons/NSX_EdgeCluster_API.json
    - Sample file for creating an Edge cluster when this option is selected in VLC UI
    - You can change:
        - Everything (Ensure MTU is not > 8940 in nested environment)
    - clusterID will be obtained by VLC from SDDC Manager

- automated_api_jsons/NSX_AVN_API.json

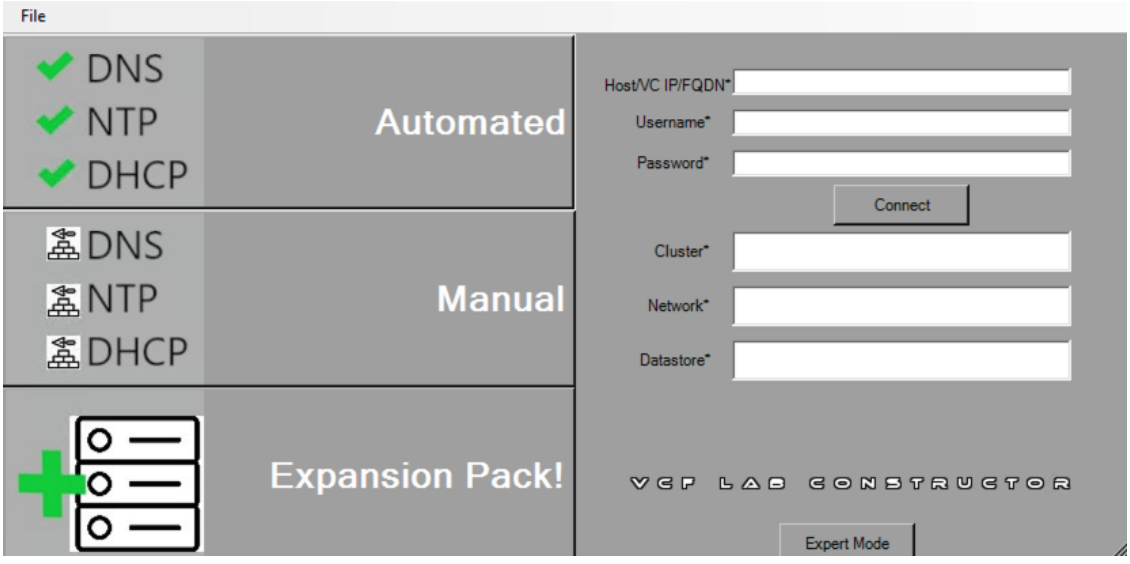
    - Sample file for creating AVNs when this option is selected in VLC UI
    - You can change:
        - Everything (Ensure MTU is not > 8940 in nested environment)
        - “routerName” *must* match "tier1Name" from NSX_EdgeCluster_API.json
- automated_api_jsons/WORKLOADMGMT_API.json
    - Sample file for enabling Workload Management when selected in VLC UI
    - Note: It is fixed that (5) IP addresses are used for the Supervisor cluster in your management subnet. Starting with .150, with the defaults this means 10.0.0.150-10.0.0.154 will be used by the Supervisor cluster.
    - You can change:
        - Egress and Ingress CIDR subnets and prefixes

When performing a Manual Mode deployment, you may wish to modify it to match your desired environment, ensure you have a backup copy available. Common changes include changing the naming, VLAN IDs, or the DNS/NTP IPs.

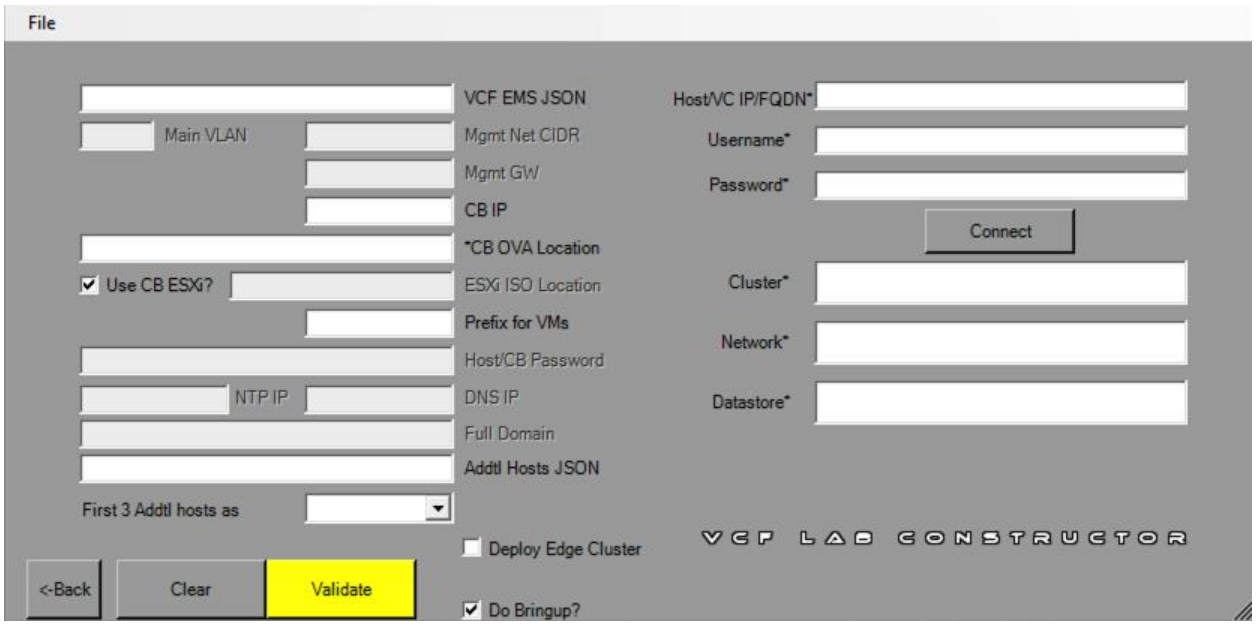
Note you can also use the Cloud Builder appliance to generate a json file based on a completed Bring-up Excel worksheet. See the VMware Cloud Foundation documentation for details.

Building in Expert Mode

1. From the jump host, execute the VLC PowerShell script: VCLGui.ps1
2. Powershell will open and run a few checks.
3. Wait for the following window to appear:



4. Select the ***Manual*** Method.



5. Enter the path to your ***Custom VCF.json***
6. Enter the IP to use for the Cloud Builder appliance
7. Enter the path for the Cloud Builder OVA
8. (Optional) Enter the path for the Additional Hosts JSON file. This will automatically create additional nested hosts that you can use to expand your VCF deployment. You can always use the Expansion Pack option to add additional hosts after a deployment.
9. (Optional) Uncheck the 'Use CB ESXi' and enter the path for the ESXi ISO
10. Enter a naming prefix for the first four hosts in the Management Domain.
11. Enter the Host or vCenter IP information of the physical environment that you would like to use to deploy to.
    a. Enter the Username and password to connect to this infrastructure, be advised that the password will be captured in clear text in the log and ini files. Secure as needed.
12. Click Connect
    a. The Constructor will now connect to the host or vCenter you specified and will gather information on suitable targets. Once completed, it will populate the information in the Cluster, Network and Datastore fields for selection.
13. Click on File, to save your data entry configuration to an .ini file before you start the deployment. This will save you time when you start over. Note: You can also load previously saved configuration information from this menu.
14. Click Validate and wait until all the components turn green. Address any errors as needed.
15. Click Construct to begin the deployment process.

Once you click construct, VCF Lab Constructor will build the nested ESXi hosts you specified in the JSON and configures the hosts to work with the Cloud Builder appliance.

Next it will automatically deploy the Cloud Builder appliance

Once the Cloud Builder is deployed and running you can connect to it at this URL:
https://<Cloud_Builder_VM_IP>

If you selected the option for VLC to perform bring-up, no action is required form you. VLC will monitor the bring-up progress for you in the PowerShell window. However, if you still want to see the UI, you can access by completing the following steps:

- Log in to the Cloud Builder UI. Use the username (admin) and password (VMware123!) that you set in the VLC GUI.
- Click the checkboxes and then continue to click Next until you get to the file upload.
- Select one of the sample json configuration files file and click Next
- IMMEDIATELY click "Cancel" in the blue bar to the left
- Change the end of the URL to /bringup-result and press enter

This will allow you to see all the bring-up UI tasks and the progress of the deployment. If there are any errors or failures this will display the relevant error. Depending on what failed, you may be able to use the retry feature to restart a failed task after you have corrected the issue.

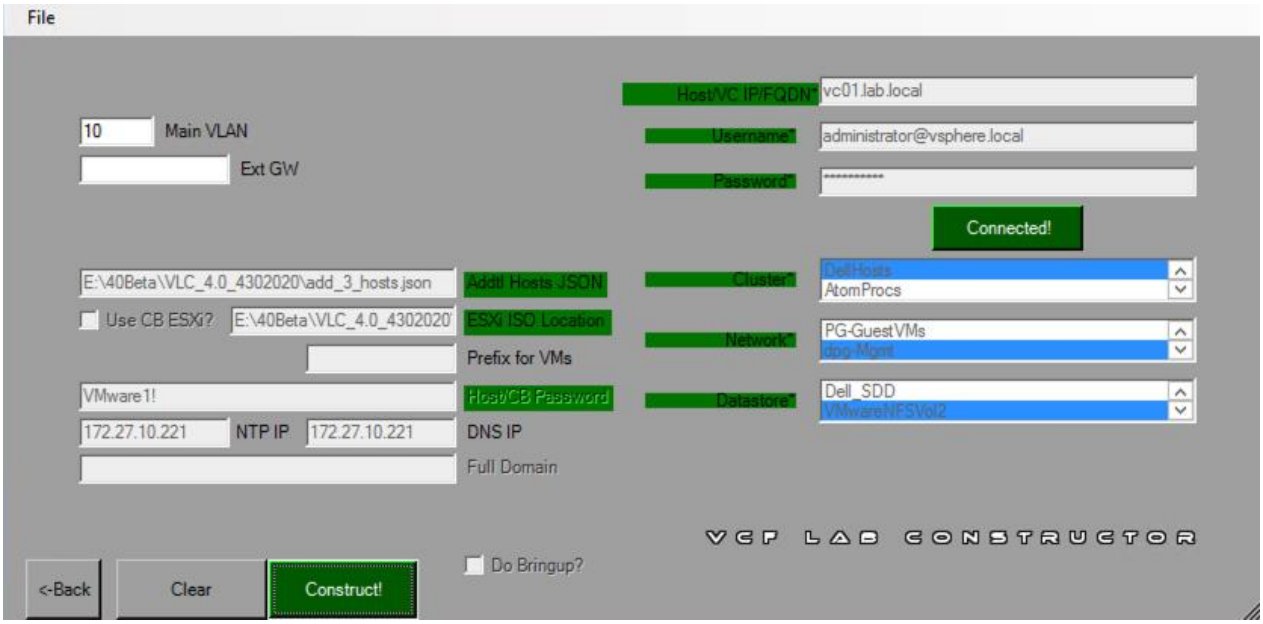
If you did not opt to have VLC perform bring-up automatically, you will need to complete the bring-up process manually as per the VMware Documentation.

Using Expansion Pack

Congratulations! If you have traveled this far that means you have successfully deployed VCF in a nested lab and you are ready to scale out your installation.

Located in the C:\VLC\add_3_hosts.json file you will be able to copy and / or edit this file to modify their hardware configuration and add more hosts..

**Note, 4 CPUs and 32gb of memory is the minimum supported configuration for ESXi hosts. Lesser memory will work but has been seen to cause issues during NSX VIB installation in our nested configs.



There are two use case scenarios for the expansion pack.

Manual > Expansion Pack

Using this method means you have created DNS entries with the host names your adding to the environment on your DNS server. VLC will continue to use your DNS server.

Automated > Expansion Pack

This method means that you have used the Automated method to deploy your environment. As such, VLC has configured the Cloud Builder appliance to provide essential infrastructure services. Before adding additional hosts, you will need to add the appropriate DNS entries to the Cloud Builder configuration.

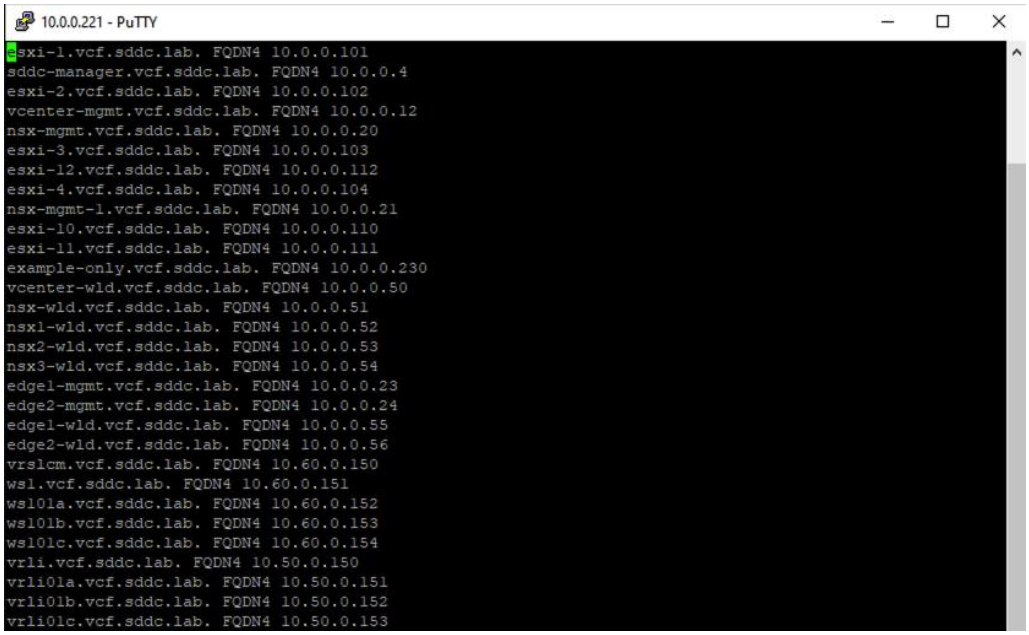
Adding DNS entries

Use SSH to connect to your Cloud Builder VM and log in using the username (admin) and the password that you specified in the VLC GUI when you deployed the environment.

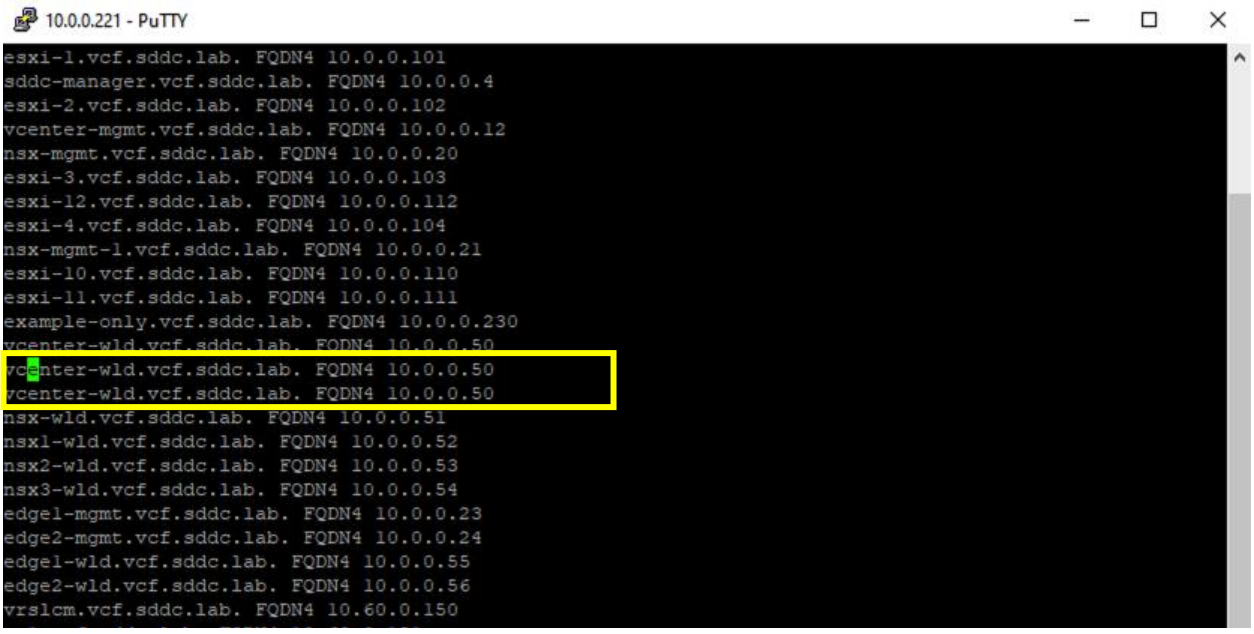
You will need to edit the DNS “db” file for the zone specified. As an example, assume that the domain ‘vcf.sddc.lab’ was used during the creation of the nested environment. This would mean the zone file would be located here: /etc/maradns/db.vcf.sddc.lab
Edit the file using vi:

```
root@LAB1-CB-01a [ ~ ]# vi /etc/maradns/db.vcf.sddc.lab
```

The result of this command opens the db zone file in the VI editor:



Duplicate the line of the currently defined vcenter by moving the cursor to that line using the arrow down key and then pressing YYPP in that order. Once that is done, you should see something like this:



After this you will change the newly copied lines name and IP addresses, ensuring that the IP addresses are not already in use! For those not familiar with VI commands here is a cheat sheet: https://devhints.io/vim

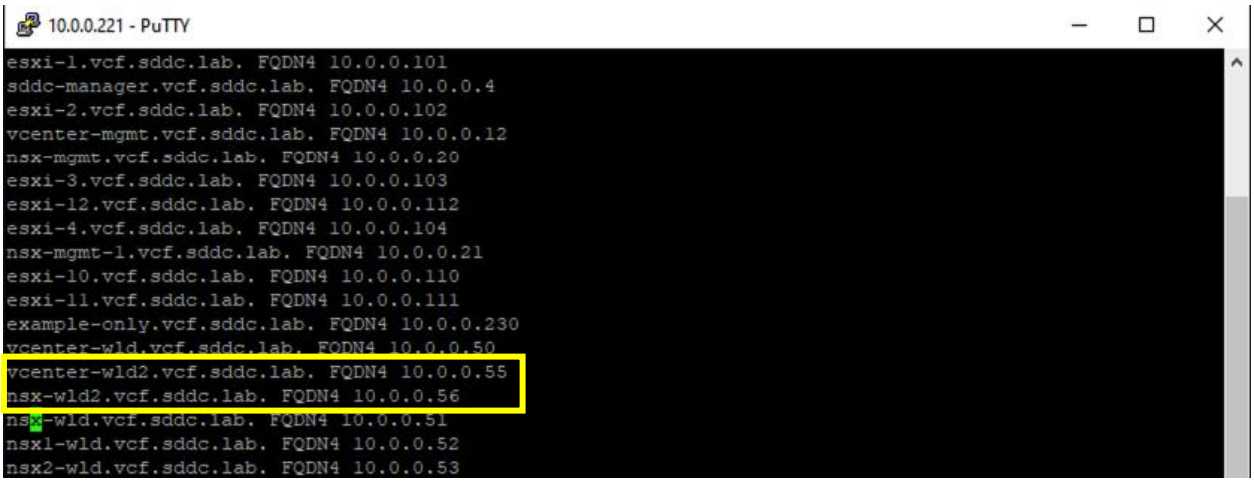


Figure 12: This may not be a complete list of DNS changes required to complete your task

After making your changes and saving the file you will need to reload maradns and the maradns.deadwood services. MaraDNS takes care of forward lookups and Deadwood takes care of Reverse DNS.

```
root@LAB1-CB-01a [ ~ ]# vi /etc/maradns/db.vcf.sddc.lab
root@LAB1-CB-01a [ ~ ]# systemctl restart maradns
root@LAB1-CB-01a [ ~ ]# systemctl restart maradns.deadwood
root@LAB1-CB-01a [ ~ ]#
```

You would follow this same procedure for adding DNS entries for vRSLCM, vROps, vRA, Horzion, or any other component. Note: Certain software (like Horizon) are not automated in VCF 4.x via SDDC Manager. You may need to follow the manual guidance presented in the VCF documentation to deploy these software packages.

Examples of the DNS files

esxi-1.vcf.sddc.lab. FQDN4 10.0.0.101
sddc-manager.vcf.sddc.lab. FQDN4 10.0.0.4
esxi-2.vcf.sddc.lab. FQDN4 10.0.0.102
vcenter-mgmt.vcf.sddc.lab. FQDN4 10.0.0.12
nsx-mgmt.vcf.sddc.lab. FQDN4 10.0.0.20
esxi-3.vcf.sddc.lab. FQDN4 10.0.0.103
esxi-12.vcf.sddc.lab. FQDN4 10.0.0.112
esxi-4.vcf.sddc.lab. FQDN4 10.0.0.104
nsx-mgmt-1.vcf.sddc.lab. FQDN4 10.0.0.21
esxi-10.vcf.sddc.lab. FQDN4 10.0.0.110
esxi-11.vcf.sddc.lab. FQDN4 10.0.0.111
example-only.vcf.sddc.lab. FQDN4 10.0.0.230
vcenter-wld.vcf.sddc.lab. FQDN4 10.0.0.50

vcenter-wld2.vcf.sddc.lab. FQDN4 10.0.0.55
nsx-wld2.vcf.sddc.lab. FQDN4 10.0.0.56
nsx-wld.vcf.sddc.lab. FQDN4 10.0.0.51
nsx1-wld.vcf.sddc.lab. FQDN4 10.0.0.52
nsx2-wld.vcf.sddc.lab. FQDN4 10.0.0.53
nsx3-wld.vcf.sddc.lab. FQDN4 10.0.0.54
edge1-mgmt.vcf.sddc.lab. FQDN4 10.0.0.23
edge2-mgmt.vcf.sddc.lab. FQDN4 10.0.0.24
edge1-wld.vcf.sddc.lab. FQDN4 10.0.0.55
edge2-wld.vcf.sddc.lab. FQDN4 10.0.0.56
vrslcm.vcf.sddc.lab. FQDN4 10.60.0.150
ws1.vcf.sddc.lab. FQDN4 10.60.0.151
ws101a.vcf.sddc.lab. FQDN4 10.60.0.152
ws101b.vcf.sddc.lab. FQDN4 10.60.0.153
ws101c.vcf.sddc.lab. FQDN4 10.60.0.154
vrli.vcf.sddc.lab. FQDN4 10.50.0.150
vrli01a.vcf.sddc.lab. FQDN4 10.50.0.151
vrli01b.vcf.sddc.lab. FQDN4 10.50.0.152
vrli01c.vcf.sddc.lab. FQDN4 10.50.0.153
vrops.vcf.sddc.lab. FQDN4 10.60.0.160
vrops01a.vcf.sddc.lab. FQDN4 10.60.0.161
vrops01b.vcf.sddc.lab. FQDN4 10.60.0.162
vra.vcf.sddc.lab. FQDN4 10.60.0.170
vra01a.vcf.sddc.lab. FQDN4 10.60.0.171
vra01b.vcf.sddc.lab. FQDN4 10.60.0.172
vra01c.vcf.sddc.lab. FQDN4 10.60.0.173
kubeapi.vcf.sddc.lab. FQDN4 10.0.0.150

Examples of rDNS zones

If you add hosts on a different network, you'll need to update the /etc/dwood3rc file and add in the in-addr.arpa for that network.

```
bind_address = "10.0.0.221"
chroot_dir = "/etc/maradns"
upstream_servers = {}
upstream_servers["."]="8.8.8.8, 8.8.4.4"
upstream_servers["0.0.10.in-addr.arpa."] = "127.0.0.1"
upstream_servers["0.60.10.in-addr.arpa."] = "127.0.0.1"
upstream_servers["0.50.10.in-addr.arpa."] = "127.0.0.1"
upstream_servers["vcf.sddc.lab."] = "127.0.0.1"
recursive_acl = "10.0.0.0/24,10.60.0.0/24,10.50.0.0/24"
filter_rfc1918 = 0
```

Adding Neighbors and routes to gobgpd

You can find the gobpg.conf file on Cloudbuilder after an automated install here: /usr/bin/gobgpd.conf This file contains all the BGP neighbors and their information and also the

local AS #. You can add neighbors to the file and restart the gobgpd service when you create a new edge cluster or, add nodes to an existing edge cluster. Below is an example stanza to add to the file, note there is no "separator" like a comma between entries:

```
[[neighbors]]
  [neighbors.config]
    neighbor-address = "172.27.12.6"
    peer-as = 65003
    auth-password = "VMware1!"
```

After adding your neighbors and saving the file you need to restart the service:

```
systemctl restart gobgpd
```

Then you can check for your new neighbors to be in the list, they should not be connected at this point:

```
gobgp neighbor
```

Clean up, Rinse and Repeat

If at any point you would like to start from a fresh deployment. You need to complete the following.

1. Login to the host or vCenter at Layer 1.
2. Power off all nested ESXi hosts and Cloud Builder
3. Delete them all from disk.
4. Delete the temp-<date> directory from the Lab Constructor directory.
5. Delete VLC_vSphere.iso from the ISO directory on the datastore.
6. Optionally remove any old log files from the Lab Constructor directory.

Troubleshooting

Some general troubleshooting hints:

- If your ESXi Host or vCenter in Physical Environment has a self-signed or untrusted SSL cert
    - Set your PowerShell environment to ignore the certificate.
        - set-powercliconfiguration -invalidcertificateaction ignore
- Examine the Bring-up logs
    - https://docs.vmware.com/en/VMware-Cloud-Foundation/4.5/vcf-deploy/GUID-BD989650-BB59-4352-A278-CDF1B9AB9B47.html
- Most times a FAILED task can be restarted (Best practice is to retry three times before investigating deeper)
- If you think a task is stuck and it's been hours, try rebooting the SDDC Manager and retrying the bring-up task.
- Depending on what stage you're having any issues at, you can login to any of the layer 2 hosts, and vCenters to troubleshoot issues.

- Post questions on the #vlc-support, join slack by signing up here: https://tiny.cc/getVLCSlack

Known Issues

- Because this software is originally made to run on physical hardware with SSD's and 10GB networks, performance issues may break your deployment.
- Sometimes VCF will fail on the prepare ESXi hosts for VSAN. ESXi 1 will show that it does not have an available SSD for cache. Cleanup and restart the process
- 32gb of RAM and 4 CPUs is minimum nested hardware config for each expansion pack host if you intend to deploy an NSX Edge
- Do not Mount the ISO for ESXi from an NFS Datastore

Logging

The VCF Lab Constructor creates a log to verify successful deployment. This log can be found in the VCF Directory where you launched the VCF Lab Constructor.

VLC Log Files are located here:
C:\VLC\Logs

During bring-up logs can be found in the Cloudbuilder appliance in the /var/log/vmware/vcf/bringup directory. The log I like the most is the vcf-bring-up-debug.log in that directory.
For problems deploying VC on bring-up look in /var/log/vmware/vcf/bring-up/ci-installer-xxxx/workflow_xxxx/vcsa-cli-installer.log

After bring-up you will look to SDDC Manager for logs. They are all rooted in the /var/log/vmware/vcf folder. Depending on what operation you are performing you can dive into one of the other folders. These are where I do most of my digging:

- Domain Manager - Used when creating/deleting/expanding/shrinking new workload domains as well as some API calls:
  /var/log/vmware/vcf/domainmanager/domainmanager.log

- Operations Manager - Used when commissioning/decommissioning hosts and resource utilization collection as well as some API calls:
  /var/log/vmware/vcf/operations/operationsmanager.log

- LCM - Used for Life cycle management activities like downloading bundles, applying updates:
  /var/log/vmware/vcf/lcm/lcm.log