

The default topology image will now be displayed on the left side of the image screen.

General
Image
Control
Device
Remote PC
Advanced
Build

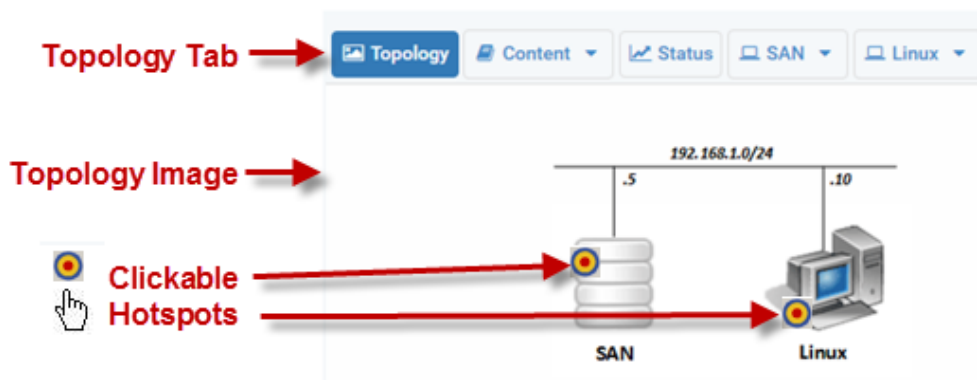
File Name	Type	Width	Height
advanced_ethical_hacking_topology.png	PNG	580	400

no image set

2.1.2 Hotspots

You may define clickable hotspots for devices and PCs shown on the default topology image. A hotspot is an invisible rectangular area placed on top of a router, switch, firewall, or PC in the image. When a user clicks on a hotspot, NETLAB+ will launch the appropriate viewer and connect to the device.

In order to set hotspots, at least one device or remote PC must be defined in the pod design. The process for adding devices and remote PCs is documented in subsequent sections.



3.5.3 Control Switch VLAN Numbers

The following table shows the control switch VLAN numbering scheme used by NETLAB+.

VLAN(s)	Purpose in NETLAB+
1	Management VLAN. Used for internal communication between the NETLAB+ server, control devices, and VMware servers.
2-99	Reserved for NETLAB+. NETLAB+ may add or delete VLANs in this range.
100-899	Used for pod dynamic VLAN maps. The maximum number of dynamic VLANs is 26 per pod and 800 per system.
900-999	Reserved for customer special needs. NETLAB+ will NOT add or delete VLANs in this range.
1000+	Reserved for Cisco. Do not use.

Control switches do not run the Cisco VLAN Trunking Protocol (VTP). NETLAB+ sets each control switch to *VTP transparent* mode during the initial switch configuration. It then uses SNMP to maintain a consistent VLAN database on each control switch. This method avoids VTP conflicts and other problems that can arise when lab switches are connected to control switches.



NETLAB+ version 22 transparently assigns and manages the pool of 800 VLANs available for pod designs. Previous versions of NETLAB+ required additional understanding and configuration of VLAN allocation.

3.5.4 VLAN Mapping

VLAN mapping is a NETLAB+ feature that can significantly increase the number of labs that can be performed on a single pod design, thereby decreasing the overall cost of lab equipment.

During normal operation, pod ports can be set to OFF or placed in a VLAN to simulate one or more Ethernet segments required by the topology of the pod.

A VLAN map tells NETLAB+ which VLAN to assign to each control switch port at the beginning of a lab reservation.

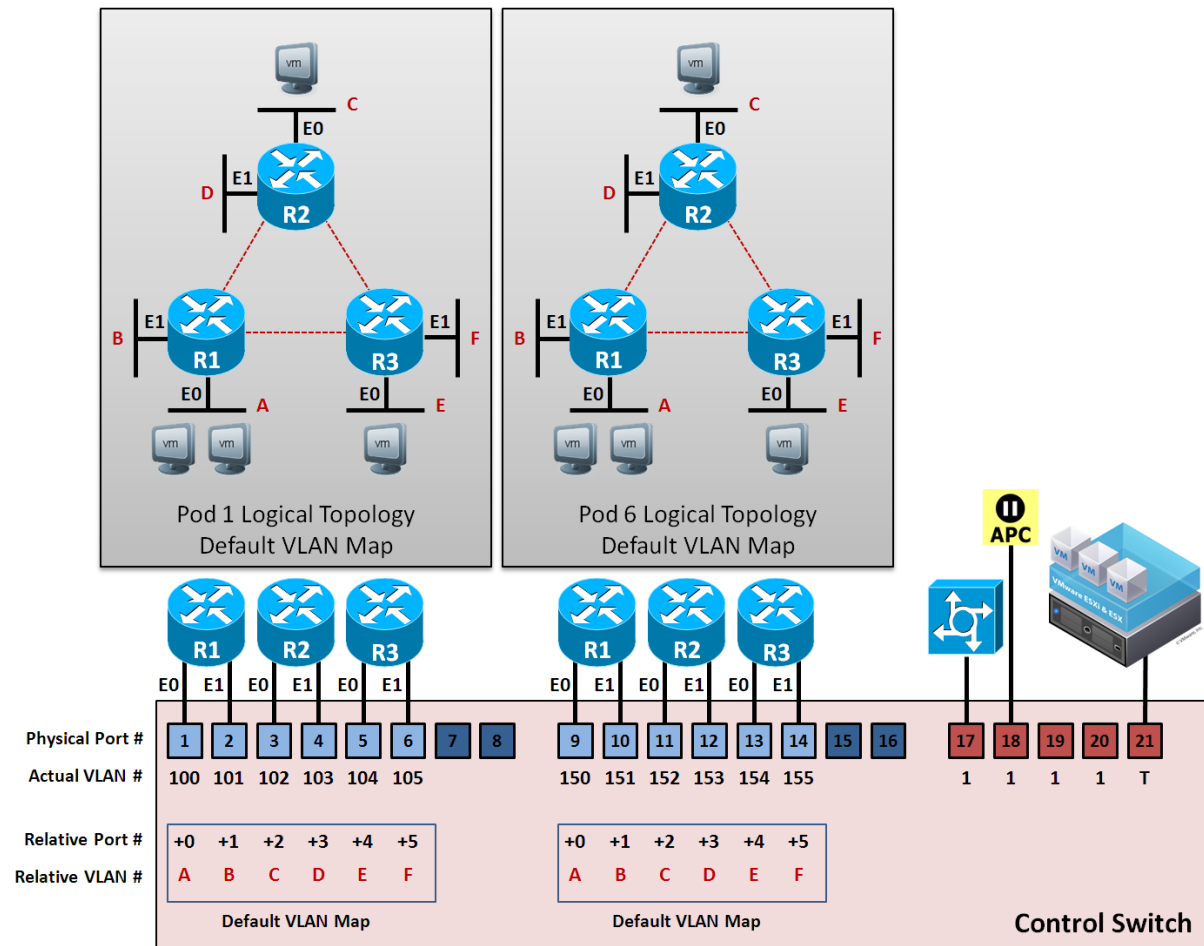
- A VLAN map can be provided for each lab exercise (these VLAN maps are specified in Lab Designer).

VLAN Map Example 1

This example below shows the default VLAN map for a NETLAB+ Basic Router Pod (version 2). The VLAN pool size for this pod design is 6 (VLAN A to VLAN F). This provides enough VLANs so that each router port can be attached to a stub network.

Because each control switch port is placed in a unique VLAN, each LAN segment behaves as a separate routed segment, as shown in the Topology diagrams.

The PCs in these pods are virtual machines on a VMware host server. Discussed in the next section.

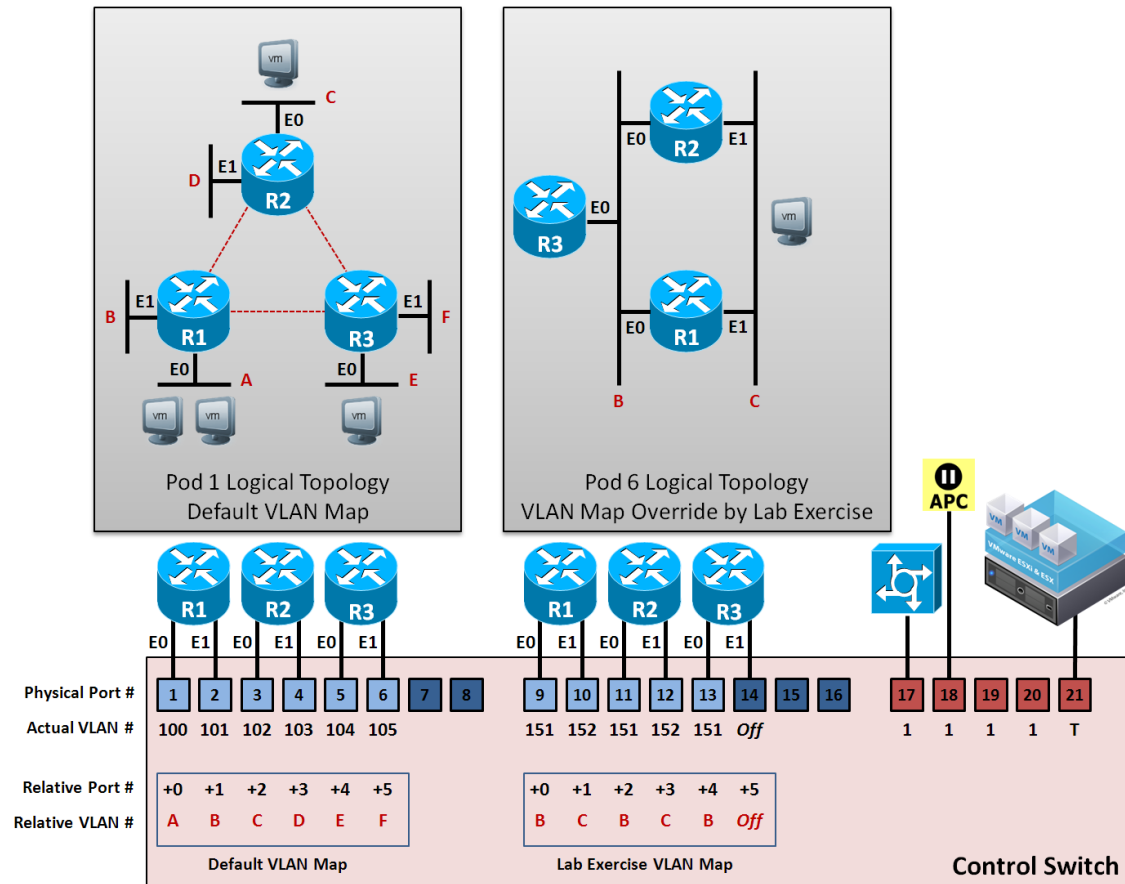


VLAN Map Example 2

The default VLAN map is normally applied to the pod at the beginning of a lab reservation. However, you may also create VLAN maps for each lab exercise (using Lab Designer). Each exercise may specify its own VLAN map or simply use the default VLAN map specified in the pod design.

By using VLAN maps in your lab exercises, you can make logical alterations to the lab topology without changing the cabling of the pod (which is always fixed). If the pod and lab designs utilize VLAN maps, it may be possible to perform many more labs than could be done with static mapping and/or direct connections between lab devices. By decreasing the number of physical lab topologies required, the potential cost of lab equipment to teach a curriculum is greatly reduced.

Consider the two pods below. Both pods are Basic Router Pod Version 2 pods and are physically wired as such. However, pod 6 has loaded an “HSRP” lab exercise that uses a dynamic VLAN map. The pod 6 topology will behave much differently than its pod 1 counterpart for the duration of the lab exercise. The lab exercise also presents the topology picture shown for pod 6, so the user actually sees the modified topology.



4 Lab Device Settings

This section applies to real equipment pods and hybrid pods.

Lab device settings do not apply to Pure Virtual Machine Pods. You may skip this section if your pod design does not include real equipment.

Lab devices can be classified into 3 categories.

- **Managed Lab Devices:** Real equipment that users can connect to and configure, automated by NETLAB+.
- **Generic Console Device:** Real equipment that users can connect to and configure but is not automated by NETLAB+ (except for user control over power).
- **Supporting Lab Devices:** Supporting real equipment that is not visible or configurable by the user.

4.1 Managed Lab Devices and Generic Console Devices

Managed lab devices are real devices that lab users will actually configure during the lab. NETLAB+ automation drivers are provided for a variety of Cisco routers, switches, and firewalls.

Each managed lab device uses a NETLAB+ device driver. Managed lab devices must be specifically supported by NDG. A list of supported lab devices can be found in the support section of the NDG website.

Generic Console Devices are real equipment that users can connect to and configure but are not automated by NETLAB+ (except for user control over power). This allows devices that are not supported (or automated) by NDG to be used without an automation driver, provided they have a serial console port and can be accessed via an access server connection.

The decision to use a managed device or generic console device is made on a per-pod basis. The pod design settings are the same for either.

4.1.1 NETLAB+ Automation Features for Managed Lab Devices

The following table shows the automation features provided for supported managed lab devices. The table does not apply if the administrator chooses a Generic Console Device setting for the given lab device.

NETLAB+ Feature	Cisco Routers	Cisco Switches	Cisco Firewalls
Connect directly to the console port	Yes	Yes	Yes
Built-in terminal application to access device console ports	Yes	Yes	Yes
Share console with several users at the same time	Yes	Yes	Yes
Turn the device ON or OFF	Yes	Yes	Yes
Automatically load or save configuration files via the console port (does not rely on networking and TFTP)	Yes	Yes	Yes
Automatically perform password recovery	Yes	Yes	Yes
Reset device to a clean configuration upon request or at the end of the lab reservation	Yes	Yes	Yes
Automatically recover an image that has been erased from flash memory	Partial	No	No

See the [NETLAB+ Supported Lab Devices](#) page for a list of supported managed equipment.

4.1.2 Lab Routers

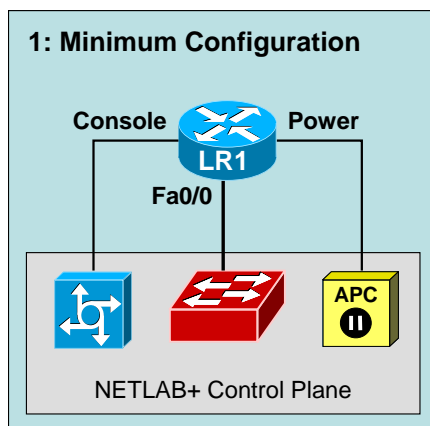


The following connections are typical for each lab router.

- One connection to an access server port.
- One connection to a switched outlet (i.e., APC device)
- One or more connections to a control switch.

Refer to the three configuration scenarios below. LR1 is a lab router. LR1 has connections to the NETLAB+ control plane: access server (left), control switch (middle), and switch outlet device (right).

1: Minimum Configuration

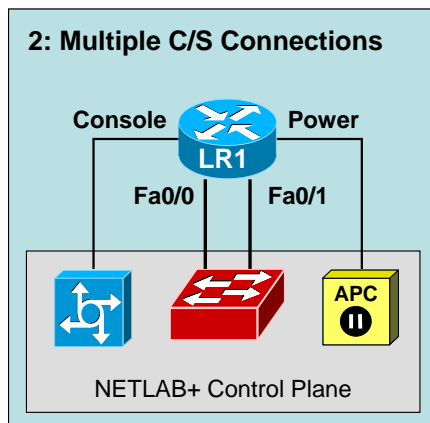


Configuration 1 depicts the minimum connection requirement for each lab router.

The lowest numbered Ethernet (typically Fa0/0) port may be used to recover the router's image should it be erased from flash on previously supported Cisco routers. Currently supported routers recover images from a read-only USB storage device, if capable.

During the lab, this port may be turned off or mapped dynamically to a VLAN on the control switch to simulate a LAN segment. Dynamic VLAN mapping is a powerful feature that may be used to alter the lab topology. This can be done for each lab exercise.

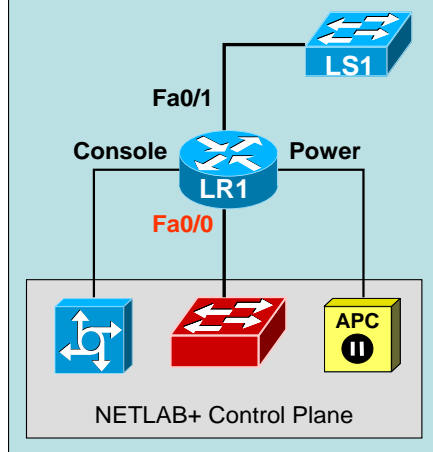
2: Multiple C/S Connections



Configuration 2 depicts both built-in Ethernet ports connected to the control switch.

Each Ethernet port connected to the control switch may participate in dynamic VLAN mapping. This technique can be used to create very flexible lab topologies from a single statically cabled pod.

3: Lab Switch Connection



Configuration 3 shows a direct connection between a lab router (RT1) and a lab switch (LS1).

Direct connections between lab devices are required to teach core switching concepts such as trunking, VTP, CDP, inter-VLAN routing, and port channeling. Direct connections preserve the nature of these protocols. A path through the control switch will not work in these scenarios.

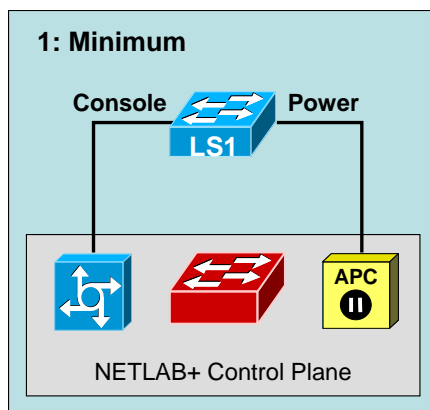
4.1.3 Lab Switches



The following connections are required for each lab switch.

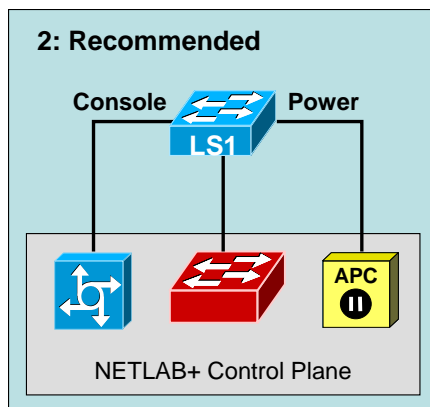
- One console connection to an access server port.
- One power connection to a switched outlet (i.e., APC device).
- One connection to a control switch for every VMware virtual machine will appear as a directly attached PC.

Refer to the four configuration scenarios below. LS1 is a lab switch. LS1 has connections to the NETLAB+ control plane: access server (left), control switch (middle, recommended), and switch outlet device (right).



Configuration 1 depicts the minimum connection requirement for each lab switch.

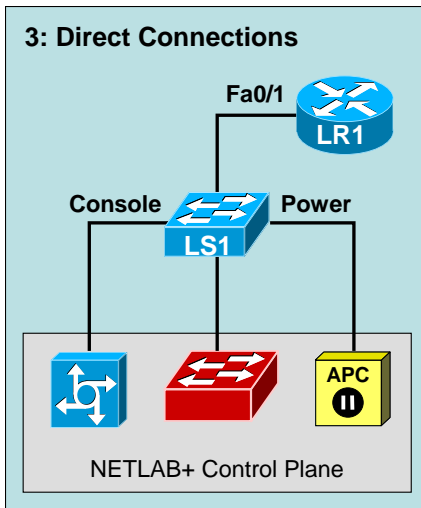
Please note this configuration will not support flash recovery, should this feature be implemented for lab switches in the future. Therefore, we recommend one of the other configuration options below.



Configuration 2 depicts the recommended configuration. If LS1 does not connect to other devices (such as VMware PCs) through the control switch, then the control switch connection can be turned off during the lab exercise. This connection also enhances NDG's ability to perform diagnostics and troubleshooting on the lab switch.

Whenever a lab switch is connected to a control switch, additional configuration commands are required on the control switch to prevent unwanted spanning tree, trunking, and VTP side effects. This is explained in the following section.

3: Direct Connections



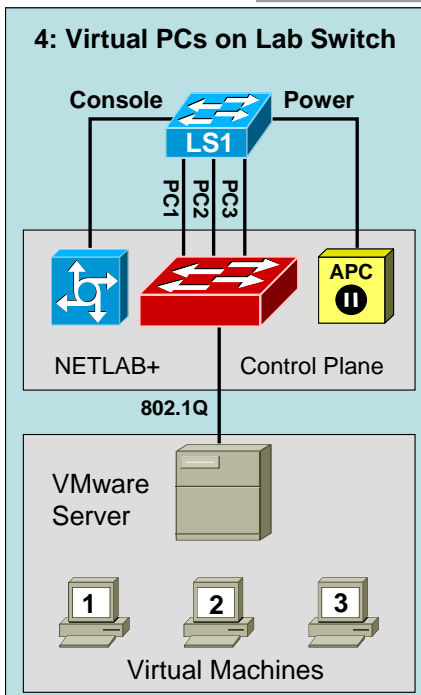
Configuration 3 shows a direct connection between a lab switch (LS1) and a lab router (RT1).

Direct connections between lab devices are required to teach core switching concepts such as trunking, VTP, CDP, inter-VLAN routing, and port channeling. Direct connections preserve the nature of these protocols. A path through the control switch will not work in these scenarios.

The connection between LS1 and the control switch is turned OFF during the lab. As in configuration 2, this is a management connection.

Whenever a lab switch is connected to a control switch, additional configuration commands are required on the control switch to prevent unwanted spanning tree, trunking, and VTP side effects. This is explained in the following section.

4: Virtual PCs on Lab Switch



Configuration 4 shows how virtual PCs can be tied to a lab switch via the control switch. To the lab user, each PC appears to be directly connected to ports on the lab switch.

Whenever a lab switch is connected to a control switch, additional configuration commands are required on the control switch to prevent unwanted spanning tree, trunking, and VTP side effects. This is explained in the following section.

4.1.3.1 Special Configuration for Pods Containing Lab Switches



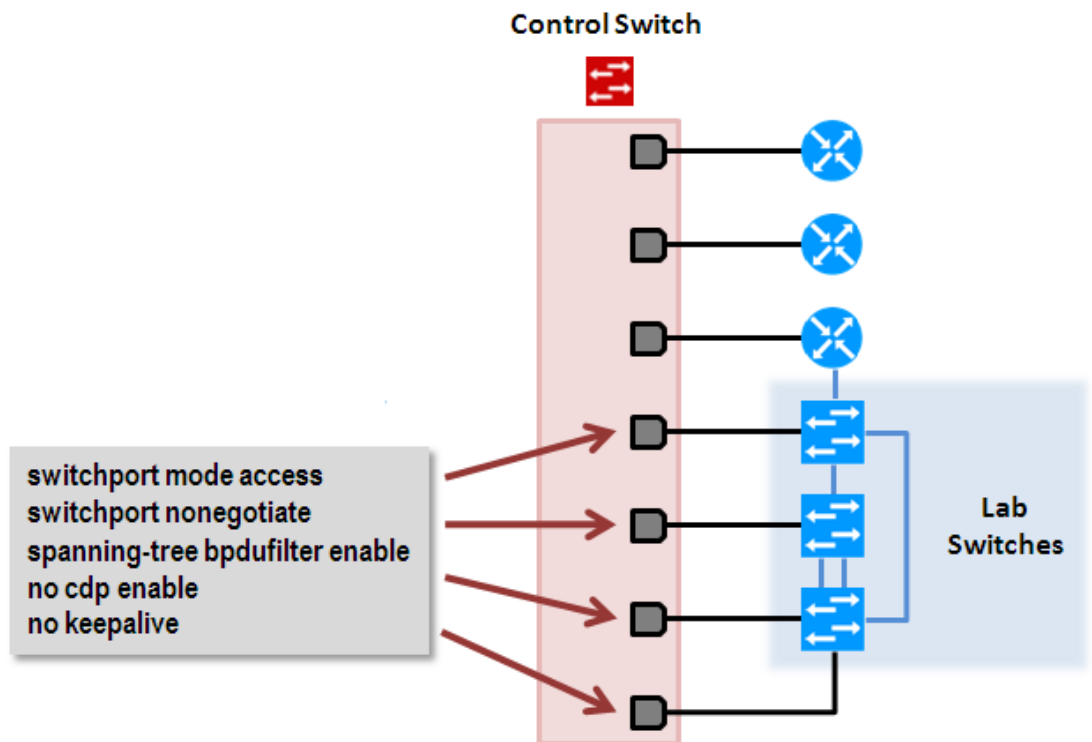
Whenever a pod contains lab switches and ports on the lab switch connect to a control switch, NETLAB+ will apply additional commands to the control switch ports to prevent adverse interactions between the lab switch and control switches.



When setting up a pod containing lab switches, be sure to add the pod to NETLAB+ and configure control switch lab ports physically attaching the equipment to the pod.

BEFORE CONNECTING LAB EQUIPMENT: the system administrator should perform the following steps in this order to prevent packet loops between control switches and lab switches.

1. Determine which access server (ATS) lines, control switch (CSW) lab ports, and power distribution outlets (PDU) will be assigned to the pod.
2. Ensure that all ATS, CSW, and PDU units are configured in NETLAB+.
3. Designate the control switch lab ports and configure them into the NETLAB+ free lab port pool. This step applies the necessary commands to the control switch lab ports to prevent packet loops. In NETLAB+ version 22, these commands are configured automatically.



4. Add the pod in NETLAB+.
5. Assign access server lines (ATS) to the pod.

6. Assign power distribution outlets (PDU) to the pod.
7. Assign control switch lab ports to the pod.
8. Now lab devices for the pod equipment can be safely connected to the designated control switch(es).



If lab equipment is connected (step 3) before configuration steps 1 or 2, serious side effects will likely occur. These include packets loops, spanning tree problems, VTP problems, and trunking problems. These issues can cause high CPU utilization, error-disabled ports, on all control switches leading to complete connectivity loss for all pods. Be sure to include this guidance in any support documentation for pod designs containing lab switches that you make available to others.

The following special commands (step 2) are configured on each control switch port that connects to a lab switch. They are provided for reference. You do not need to configure them manually on NETLAB+ version 22 systems.

- **switchport mode access**
 - Prevents the link from becoming a trunk port.
 - The labs will not work as designed if the link between the control switch and the lab switch is trunking.
 - Trunking on ports that should be access ports, combined with BPDU filtering, creates loops that are not prevented by spanning trees.
- **switchport nonegotiate**
 - Prevents the interface from sending DTP messages.
 - Disabling DTP messages is not critical but will hide the control switch's MAC address from lab switches when users perform commands to see the CAM table.
- **spanning-tree bpduguard enable**
 - Instructs control switch port not to send and receive spanning tree BPDU frames to and from the lab switch.
 - Spanning tree in the lab must not mangle with the spanning tree on the control switch. This would cause several undesirable effects in both the lab and on the control switches.
- **no cdp enable**
 - Disabling CDP is not critical but will hide the control switch from lab switch users performing CDP commands.
- **no keepalive**
 - Prevents the interface from sending L2 keepalive messages.
 - Disabling L2 keepalives messages is not critical but will hide the control switch's MAC address from lab switches when users performing commands to see the CAM table.

Here is an example configuration for a control switch control port that is connected to a lab switch:

```
interface FastEthernet0/X
  description control port to lab switch LS1 port 7
  switchport mode access
  switchport nonegotiate
  spanning-tree bpdu filter enable
  no cdp enable
  no keepalive
```



These commands are now automatically configured by NETLAB+ version 22 and later. Without them, serious side effects will likely occur.

4.1.3.2 VLAN Map Special Rule for Pods Containing Lab Switches

When connecting lab switches to control switches, special care must be taken not to create loops and other side effects. The commands and rules outlined in the previous help prevent this, but there is another rule that pod and lab design authors must follow to avoid trouble.



When specifying VLAN maps, place pod ports connecting to lab switches in unique VLANs. Pod ports that connect to lab switches should NEVER share a common VLAN, unless there is absolutely no possibility of loop formation.

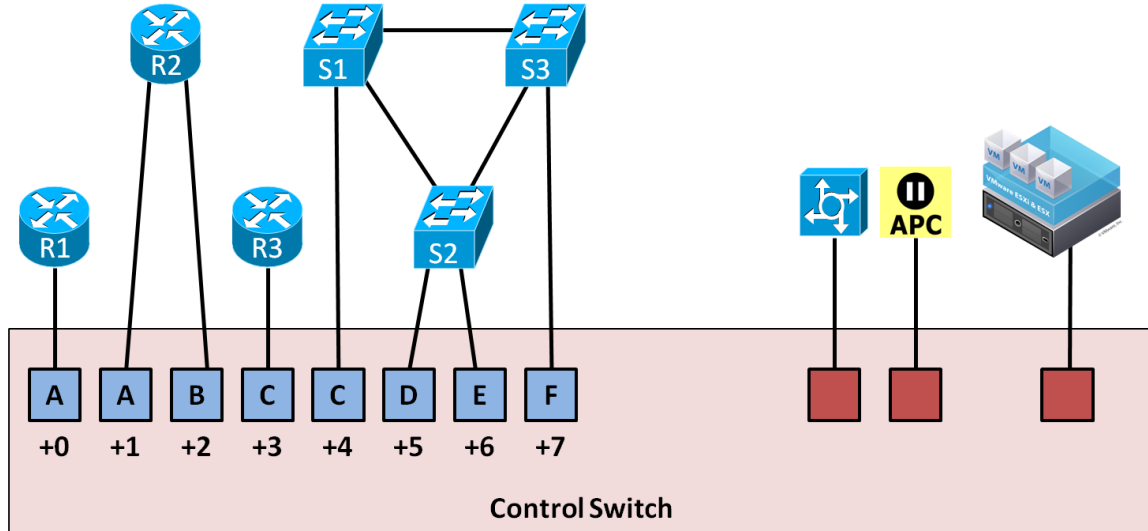
Explanation

1. Recall from the previous section that the spanning tree is intentionally disabled between the control switch (BPDU disabled) and lab switches. This creates a barrier between spanning tree domains.
 - a. Connections between lab switches and control switches are always non-trunking (switchport mode access, nonegotiate). Thus, VTP information is also isolated. That is, VLANs used on control switches are completely independent of VLANs that users may create on lab switches.
 - b. All packets between control switch and lab switches have no VLAN tagging.
2. By default, all lab switches ports will be placed in VLAN 1.
 - a. Any physical loops between lab switches will be prevented by the lab switch spanning tree domain.
 - b. However, spanning tree does not run between lab switch and control switch. Loop potential exists here.

These factors create the potential for loops to form between lab switches and control switches unless VLAN maps are properly constructed by pod and lab designers. By ensuring that control switch ports are placed on unique VLANs, you can ensure that interaction between control and lab switches will be loop-free, regardless of the configuration that takes place on the lab switches.

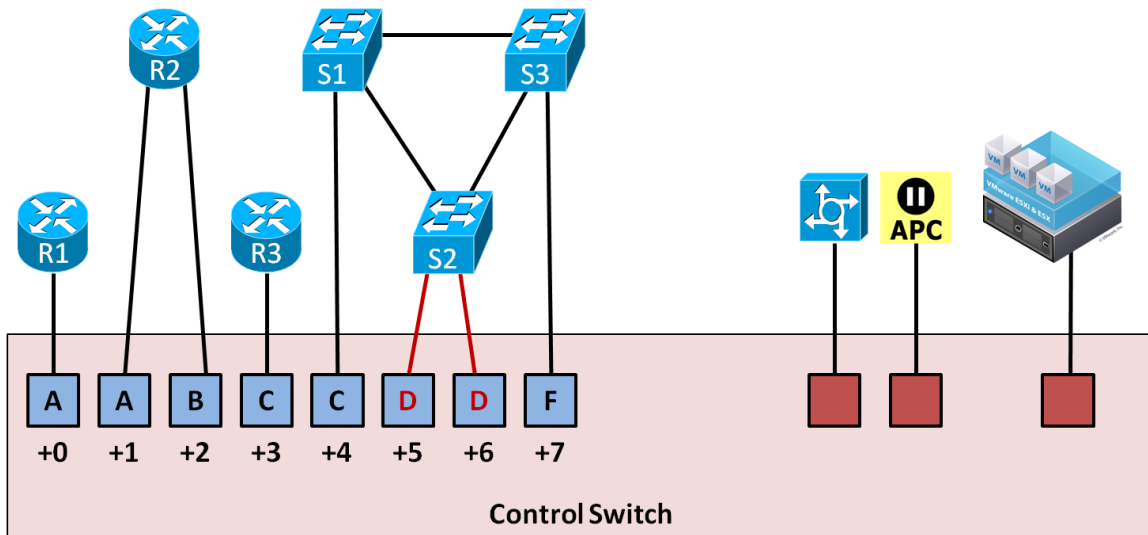
Example 1.

Our first example (below) shows a **valid** configuration. Each pod port connection on the control switch is placed in a unique VLAN (C, D, E, F). There is no possibility of looping through the control switch, regardless of how the user configures the lab switches.



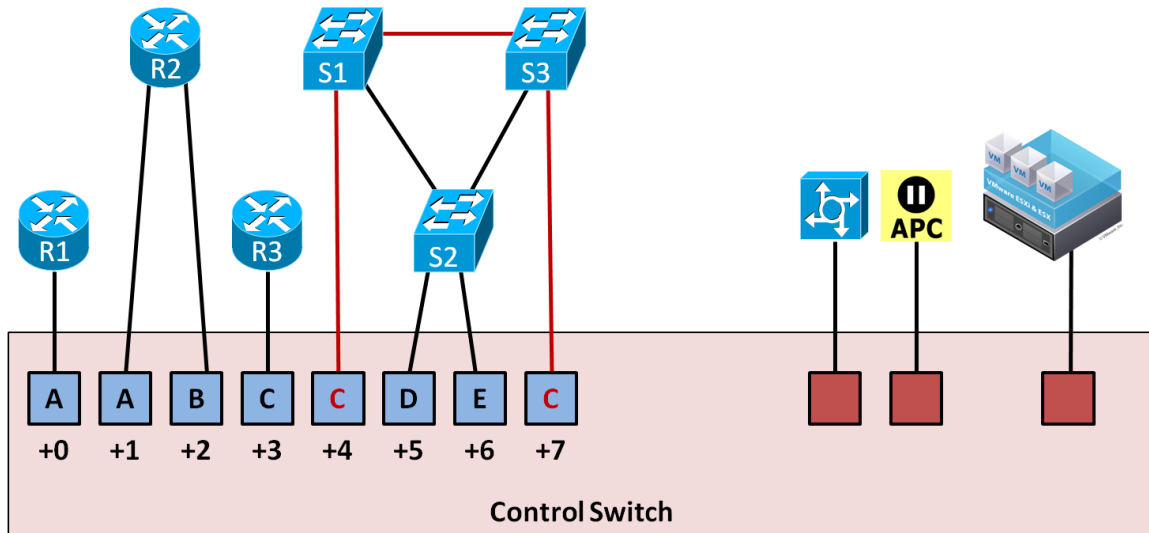
Example 2.

Our second example (below) shows an invalid VLAN map. Two ports share VLAN D. A loop will form between S2 and control switch. Broadcast traffic will quickly multiply and consume all the lab switches in this pod. More importantly, it will also consume the control switches with broadcast traffic.



Example 3.

This scenario is also invalid for the same reason as example 2.

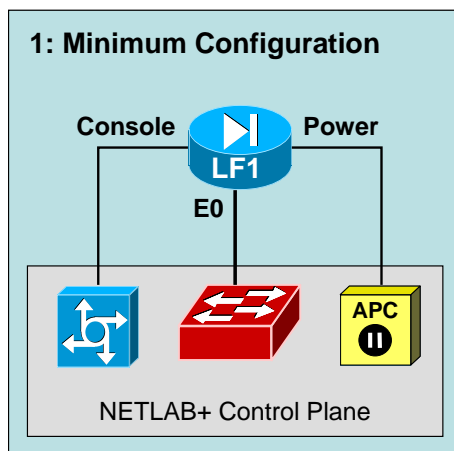


4.1.4 Lab Firewalls

The following connections are required for each lab firewall.

- One connection to an access server port.
- One connection to a switched outlet (i.e., APC device)
- One connection from the lowest numbered built-in Ethernet port to the control switch (i.e., Ethernet0).

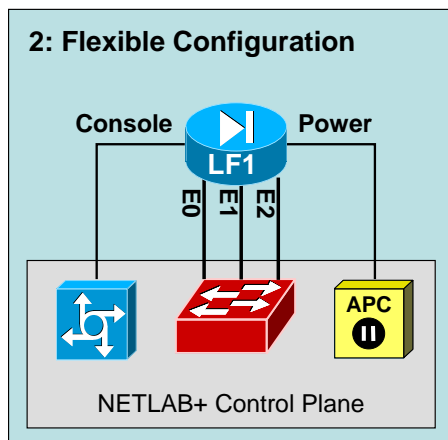
Refer to the three configuration scenarios below. LF1 is a lab firewall. LF1 has connections to the NETLAB+ control plane: access server (left), control switch (middle, recommended), and switch outlet device (right).



Configuration 1 depicts the minimum connection requirement for each lab firewall.

The lowest numbered port (typically Ethernet0) is used to manage the IOS image.

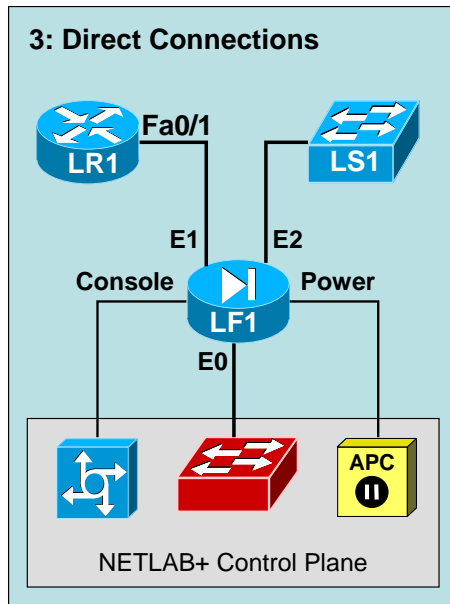
During the lab, this port may be turned off or mapped dynamically to a VLAN on the control switch to simulate a LAN segment. Dynamic VLAN mapping is a powerful feature that may be used to alter the lab topology. This can be done for each lab exercise.



Configuration 2 depicts all built-in Ethernet ports connected to the control switch.

Each Ethernet port connected to the control switch may participate in dynamic VLAN mapping. This technique can be used to create very flexible lab topologies from a single statically cabled pod.

3: Direct Connections



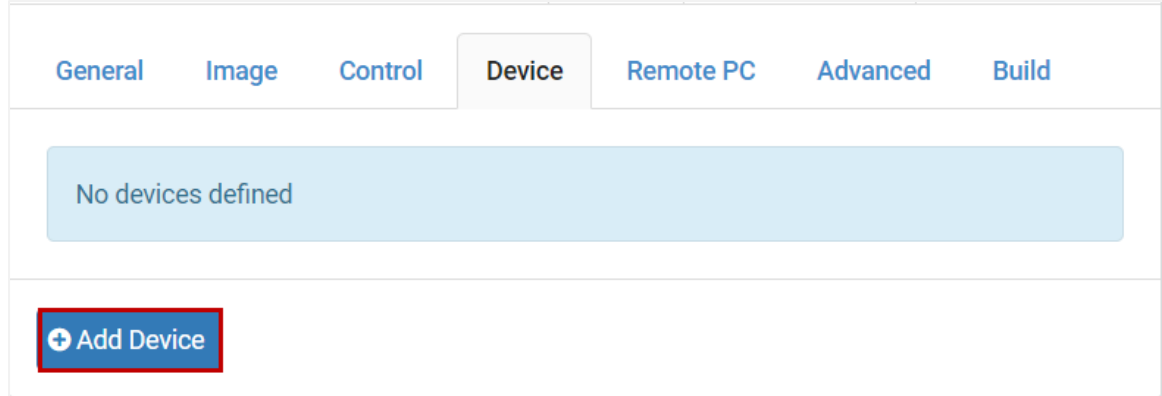
Configuration 3 shows a direct connection between a lab firewall (LF1) and two other lab devices (LR1, LS1).

Direct connections between firewalls and other devices may be required if the control switch would interfere with protocols that must be preserved to teach the core concepts (i.e., CDP). This is not usually the case with firewalls, as they typically are simple Ethernet ports.

4.1.5 Adding a Lab Device

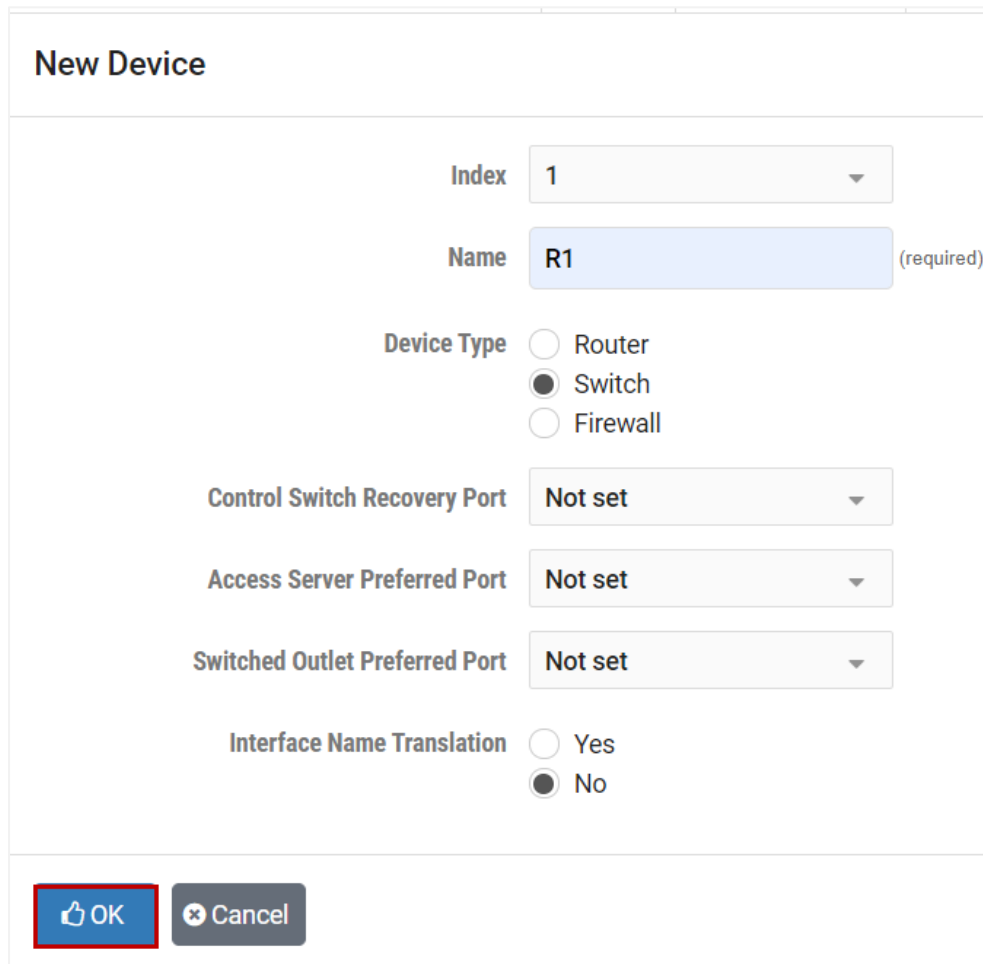
Follow the steps in this section to add a lab device to the pod design.

1. On the Device tab, click **Add Device** to add a new managed device.



The screenshot shows the 'Device' tab selected in the pod designer interface. The main area displays 'No devices defined'. A red box highlights the '+ Add Device' button at the bottom left.

2. The New Device page is displayed. On the Device tab, click **Add Device** to add a new managed device. Enter a name for the device and select the appropriate settings (refer to the field descriptions below). Click OK.



The 'New Device' configuration page is shown. The 'Index' is set to 1. The 'Name' is 'R1' (required). The 'Device Type' is 'Switch'. The 'Control Switch Recovery Port', 'Access Server Preferred Port', and 'Switched Outlet Preferred Port' are all set to 'Not set'. The 'Interface Name Translation' is set to 'No'. A red box highlights the 'OK' button at the bottom left.

Field Descriptions – New Device

- **Index:** This value determines the order in which the device will appear in the various lab access tabs.
- **Name:** The default name of the device. This may be overridden by lab exercises.
- **Device Type:** The type of managed device required: router, switch, or firewall. This will determine the icon displayed on NETLAB+ screens.
- **Control Switch Recovery Port:** The relative control switch port that will be used to recover an IOS image that has been erased flash. This port is always connected to the lowest-numbered Ethernet port on the lab device. A value of “not set” can be specified to save a control switch port and/or if the NETLAB+ device driver does not support image recovery. This feature only applies to older Cisco devices that are no longer supported. Newer models will recover from USB and/or flash if recovery is supported.
- **Access Server Preferred Port:** The suggested port on the access server the lab device should use relative to the first port. This value will be used if the administrator requests consecutive ports. However, the administrator may override this value and explicitly choose an access server and port.
- **Switched Outlet Preferred Port:** The suggested port on the switched outlet device this lab device should use relative to the first port. This value will be used if the administrator requests consecutive ports. However, the administrator may override this value and choose any switched outlet device and port.
- **Interface Name Translation:** The feature allows configuration files saved by NETLAB+ to be loaded, without errors, on a different pod with different router types. This setting is currently supported on managed routers and ignored by managed switches and firewalls. As a configuration is loaded, NETLAB+ will substitute the correct interface names for the actual router types being used (if necessary). To do this, NETLAB+ maintains a fixed table of interface names that should be present on each supported router model. Currently, this table supports two Ethernet and two serial interfaces.

If possible, you should always use the same hardware models on a similar pod type. This way, the interface names will not be a factor during automated load and save operations.

4.1.6 Adding Hotspots

You may define clickable hotspots for devices shown on the default topology image. A hotspot is an invisible rectangular area placed on top of a device in the image. When a user clicks on a device, NETLAB+ will launch a terminal application and connect to the console of the device.


Please see section [2.1.2](#) for instructions on adding hotspots.

4.1.7 Cable Guidance

Each managed device can provide cabling guidance for the administrator. This feature is currently used by NDG for NETLAB+ pods. This feature is not yet documented for general use.

4.1.8 Modifying a Managed Lab Device

To modify a managed lab device, select the device by clicking **Edit** on the Action dropdown.

Index	Type	Name	Hotspot	CS	AS	SO	INX	Action
1		pfSense	322,136,382,196				No	<div style="border: 1px solid #ccc; padding: 2px;"> ⚙️ Edit 🔌 View Cables </div>

See section 4.1.5 for a description of each field on the page.

Device [pfSense]

Index	1	
Name	pfSense	(required)
Device Type	<input type="radio"/> Router <input type="radio"/> Switch <input checked="" type="radio"/> Firewall	
Control Switch Recovery Port	Not set	
Access Server Preferred Port	Not set	
Switched Outlet Preferred Port	Not set	
Interface Name Translation	<input type="radio"/> Yes <input checked="" type="radio"/> No	

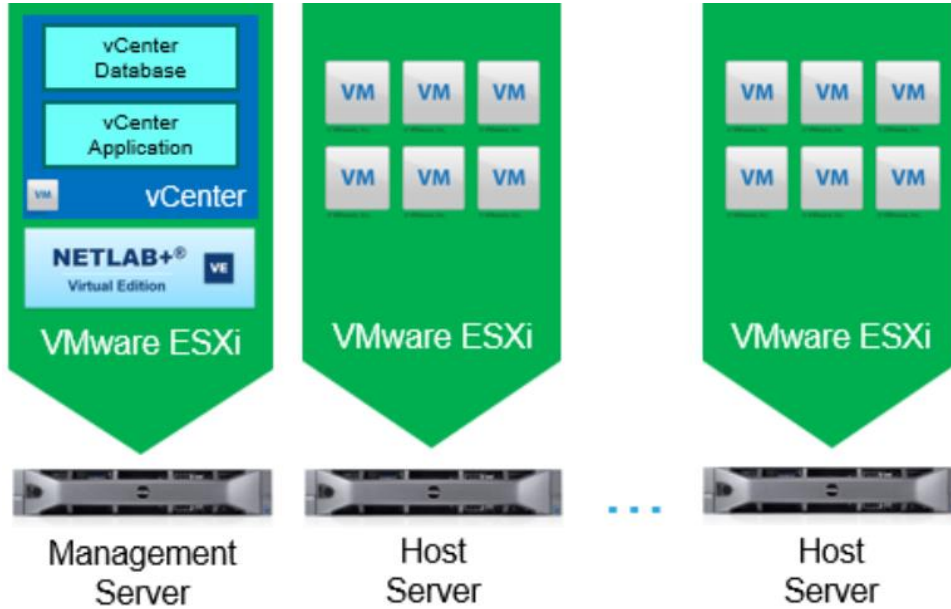
Select **OK** to save your modifications.

4.1.9 Deleting a Managed Lab Device

To delete a managed lab device, select the device from the Device tab by clicking **Edit** on the Action dropdown (see section [4.1.8](#)) and then select the **Remove Device** button.

5 Remote PC

A *remote PC* is a virtualized PC or virtualized server that is remotely accessible and configurable by lab users. Virtual machines are implemented using supported third-party virtualization products such as VMware ESXi.






The flexibility and broad selection of operating systems and configurations that may be provisioned on a virtual machine offer great potential to support IT training in a wide range of disciplines using NETLAB+.

Non-virtualized "physical" PCs and servers are no longer supported as Remote PCs.

5.1 Managing Remote PCs

Adding remote PCs using Pod Designer is fairly straightforward. Each remote PC is given a name and an icon designation of “PC” or “server”. A pod may contain very little information about remote PCs. Most of the remote PC settings are determined by the administrator when an actual pod is added to the system. Remote PCs are added and managed from the Remote PC tab.

General	Image	Control	Device	Remote PC	Advanced	Build
Index	Icon	Name	Hotspot	Actions		
1		PC1	Not set	▼		
2		PC2	Not set	▼		
3		PC3	Not set	▼		



5.1.1 Adding a Remote PC

Click on the Add PC button. NETLAB+ will display the New Remote PC dialog.

Name	Build	Committed	Author ID	Global ID
Test123	1	No	TEST123	___6501_C07C

New Remote PC

Index

Display Icon  

Name (required)

Index. Determines the order in which the PC will appear in the various lab access tabs.

Display Icon. Determines which icon (PC or server) will be associated with the PC and displayed on various pages. This setting only affects the icon that is shown. It does not affect functionality.

Name. The name of the PC.







5.1.2 Setting Hotspots

You may define clickable hotspots for PCs shown on the default topology image. A hotspot is an invisible rectangular area placed on top of a remote PC in the image. When a user clicks on a hotspot, NETLAB+ will launch the appropriate remote PC viewer and connect to the virtual machine.

Please see section [2.1.2](#) for instructions on adding hotspots.

5.1.3 Modifying a Remote PC

To modify the settings for a Remote PC, select **Edit** on the Actions dropdown.



Index	Icon	Name	Hotspot	Actions
1		PC1	Not set	
2		PC2	Not set	
3		PC3	Not set	

Please refer to the previous section for descriptions of each field.

Name	Build	Committed	Author ID	Global ID
Test123	1	No	TEST123	__6501_C07C

Remote PC [PC1]

Index

Display Icon  

Name (required)

After completing modifications, select the **OK** button.



5.1.4 Removing a Remote PC

To delete a Remote PC, select the device by clicking Edit on the Actions dropdown and then select the **Remove PC** button.

Name	Build	Committed	Author ID	Global ID
Test123	1	No	TEST123	__6501_C07C

Remote PC [PC1]

Index

Display Icon  

Name (required)

6 Advanced Settings

NETLAB+'s Advanced configuration settings are based on simple "key=value" strings. Advanced settings allow pod designers to take advantage of new features and capabilities that are still evolving and have not been backed by a final user interface in NETLAB+.

Advanced settings for a pod design can be accessed and viewed from the Advanced tab.

Name	Build	Committed	Author ID	Global ID
Test123	1	No	JD224	4581_B0FF_9639_65A8_26BB

General
Image
Control
Device
Remote PC
Advanced
Build

Advanced Settings

```

pod.requires.vm_same_host=1
pod.requires.vm_types=AVMI,ABSENT

pod.auto.net.supported=1
pod.auto.net.enabled.default=1
pod.auto.net.host.setup.supported=1
pod.auto.net.host.setup.enabled.default=1
pod.auto.net.host.teardown.supported=1
pod.auto.net.host.teardown.enabled.default=1

# vmware virtual switches created on host
vswitch.0.name.include.pod_id=1
vswitch.0.name.suffix=VS1

```

6.1 Modifying Advanced Settings

To modify Advanced settings:

1. Click the Advanced tab.
2. Click the Modify Design button.
3. Enter the pod modification password, if any.
4. Click the Modify Advanced Settings Button.
5. Enter configuration directives, one per line in the text area. Each line may contain one of the following elements:
 - a. A `key=value` configuration directive. Supported directives are discussed in the next section.
 - b. A blank line.
 - c. A comment beginning with the pound symbol (`#`) in the first column.

Name	Build	Committed	Author ID	Global ID
Test123	1	No	JD224	4581_B0FF_9639_65A8_26BB

Advanced Settings (NDG)

Advanced Settings

```

pod.requires.vm_same_host=1
pod.requires.vm_types=AVMI,ABSENT

pod.auto.net.supported=1
pod.auto.net.enabled.default=1
pod.auto.net.host.setup.supported=1
pod.auto.net.host.setup.enabled.default=1
pod.auto.net.host.teardown.supported=1
pod.auto.net.host.teardown.enabled.default=1

# vmware virtual switches created on host
vswitch.0.name.include.pod_id=1
vswitch.0.name.suffix=VS1
vswitch.0.ports=6

vswitch.0.portgroup.0.id=common
vswitch.0.portgroup.0.name.include.pod_id=1
vswitch.0.portgroup.0.name.suffix=PG1
vswitch.0.portgroup.0.vlan.mode=UNTAGGED
vswitch.0.portgroup.0.vlan.id=1
          
```

OK
Cancel

Avoid trailing whitespace at the end of a line.

6.2 Advanced Configuration Directive Reference

The following configuration directives are currently supported. These may be complemented or replaced by user interface elements in future versions. Please see subsequent sections for examples; it may be easier to learn from examples than to return to the reference.

6.2.1 Pod Requirement Configuration Directives

Pod requirement configuration directives allow the pod designer to ensure certain requirements are met by the NETLAB+ administrator.

6.2.1.1 `pod.requires.vm_same_host`

When enabled, it will force the administrator to place all VMs for this pod on the same virtual host system. VMs on the same host can communicate with each other using virtual networking on the host and do not require external networking. This should generally be enabled on any pod that supports automatic networking.

Synopsis

```
pod.requires.vm_same_host=1
```

Values

- 1** enabled
- 0** disabled (default)

Deprecated. This setting should always be set to enabled (1). Virtual machines of a single pod should be run on the same host system. Future versions of NETLAB+ may ignore this setting.

6.2.1.2 `pod.requires.vm_types`

This directive enforces the types of virtual machines that can be used with this pod. The primary use of this directive is to indicate that the pod relies on the advanced features of the NETLAB+ Virtual Machine Inventory, ESXi 4.X+, and VMware vCenter.

You would typically use this setting for Pure Virtual Machine Pods, which would be difficult to replicate and stand up without pod cloning and automatic networking support. For example, NDG uses this setting on the VMware ICM pod to ensure that the proper virtual machine host environment will support pod cloning and automatic networking; it would be difficult to configure and support an ICM pod on a standalone VMware host without the help of VMware vCenter. NDG omits this setting on its

standard hybrid pods to allow all older versions and/or standalone versions of VMware to be used.

Synopsis

```
pod.requires.vm_types=AVMI, ABSENT
```

Values

These are the intended values to be used by this directive.

AVMI pod may contain virtual machine inventory / vCenter VMs

ABSENT pod may contain missing VMs

Example

```
pod.requires.vm_types=AVMI, ABSENT
```

This directive requires that each VM in the pod must be registered in the NETLAB+ Virtual Machine Inventory, which also implies that it is supported by vCenter and the advanced automation capabilities of NETLAB+. Alternatively, the administrator may choose ABSENT on any particular VM. Any other virtual machine host platforms will be disallowed for pods of this type. For example, VMware Server 2 or standalone ESXi virtual machines would not be a configurable option for pods of this type.



Deprecated. This setting is ignored in NETLAB+ VE version 22.X.X and higher.

6.3 Automatic Networking Configuration Directives

The following scenarios are currently supported by automatic networking:

- Pure Virtual Machines Pods with no external networking
- Hybrid pods with no external networking

The following scenarios are not currently supported by automatic networking but may be in future software versions:

- Pods connecting to external networks or the internet.

6.3.1 pod.auto.net (category)

The pod.auto.net configuration directives specify which automatic networking features are supported by the pod. The administrator may choose to disable these features if manual networking is desired.

6.3.1.1 pod.auto.net.supported

Enables or disabled automatic networking features for the pod design. Requires other directives if enabled. The administrator may disable automatic networking on a per-pod basis if manual networking is desired.

Synopsis

```
pod.auto.net.supported=1
```

Values

- 1** enable automatic networking support for pod design
- 0** disable automatic networking support for pod design (default)

6.3.1.2 pod.auto.net.host.setup.supported

If enabled, it allows virtual switches and/or portgroups to be automatically created on the virtual machine host per the pod design specifications. Switches and portgroups are created when a pod reservation starts. The administrator may disable this feature on a per-pod basis if manual setup is preferred.

Synopsis

```
pod.auto.net.host.setup.supported=1
```

Values

- 1** enable automatic vswitch/portgroup setup for pod design
- 0** disable automatic vswitch/portgroup setup for pod design (default)

6.3.1.3 pod.auto.net.host.teardown.supported

If enabled, it allows virtual switches and/or portgroups to be automatically removed from the virtual machine host when the pod reservation ends. This frees networking resources from the host when they are not being used by the pod. The administrator may disable this feature on a per-pod basis if manual teardown (or persistent networking) is preferred.

Synopsis

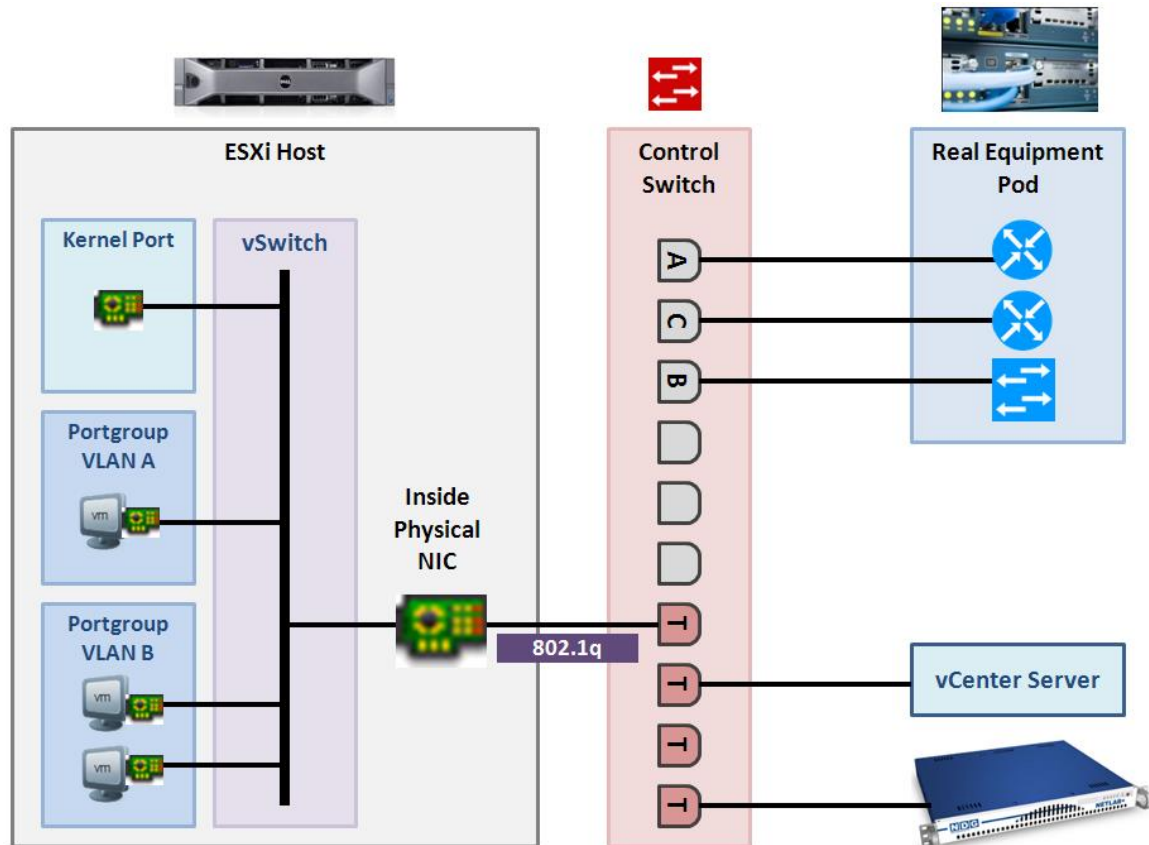
```
pod.auto.net.host.teardown.supported=1
```

Values

- 1** enable automatic vswitch/portgroup teardown for pod design
- 0** disable automatic vswitch/portgroup teardown for pod design (default)

6.3.2 Automatic Networking Support for Hybrid Pods

Hybrid pods contain both real equipment and virtual machines. To transport network traffic between virtual machines and real equipment, each virtual machine host server is connected to a control switch and configured for 802.1q (trunking mode).



The network interface that connects to the control switch on the VM host server is associated with the NETLAB+ inside virtual switch (typically vswitch1). The designated inside vswitch is specified by the NETLAB+ administrator in the Virtual Machine Host Server settings and can be specified on a per-host basis. Each VLAN within the inside vswitch is backed by a virtual portgroup within the virtual switch.

- Using NETLAB+ versions 2011.R2 (and later) with vCenter implementation, VLAN portgroups for hybrid pods can be automatically created and removed from the inside vswitch using the following configuration directives in the pod design:
 - `pod.auto.net.control_switch.vlans`
 - `pc.net.vnic.vlan`
- When using NETLAB+ versions prior to 2011.R2 and/or standalone VMware host server implementations, the NETLAB+ administrator must create these portgroups manually.

The `pod.auto.net.control_switch.vlans` setting is used in conjunction with the `pc.net.vnic.vlan` directive, which bind virtual network adapters (vNIC) to a particular

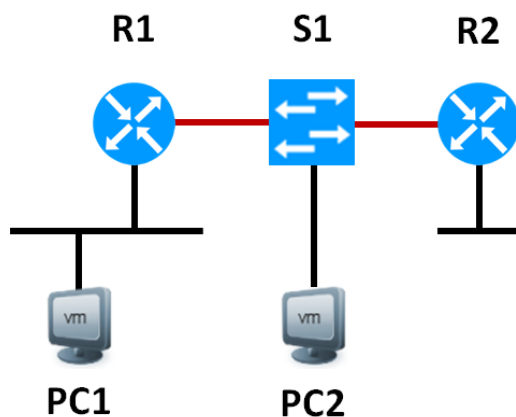
VLAN. VLAN letters are used instead of numbers, because the actual VLAN numbers cannot be determined at design time (as explained in section 3.5.4).

VLAN maps cannot be currently specified for virtual machine networking interfaces. Virtual machine VLANs remain fixed regardless of lab exercise. This is to maintain backward compatibility with a large installed base of standalone VMware host implementations that cannot support this feature. To overcome this limitation, real network device interfaces must move into the same VLANs with virtual machines (which remain fixed regardless of lab exercise).

Deprecated. Virtual machines in a pod always use the same VLAN, regardless of lab exercise. Real lab equipment may change VLANs on a per lab basis.

Example 1

We wish to provide automatic networking for virtual machines in the following hybrid topology.



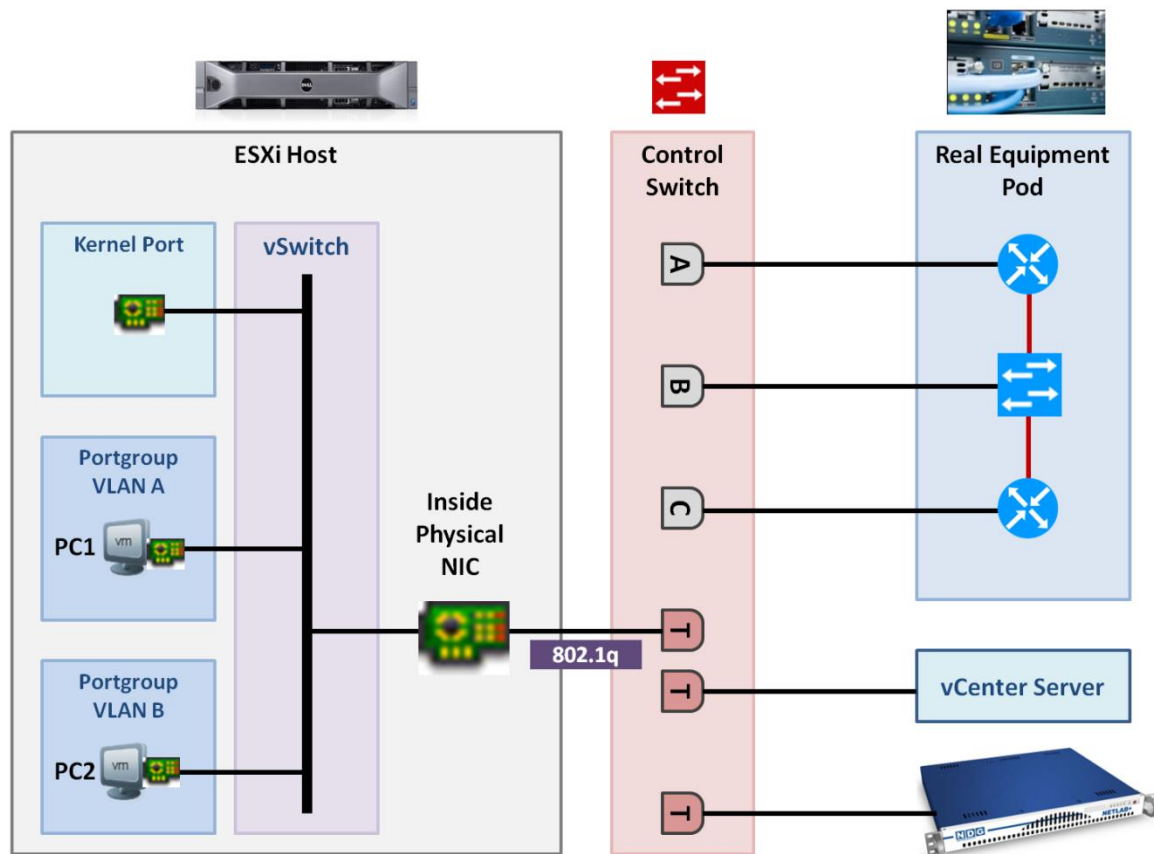
The black Ethernet segments:

- Are implemented through the control switch
- Can participate in VLAN maps
- Can connect to virtual machines
- Cannot use 802.1q

The red Ethernet segments:

- Are direct hard-wired connections between lab routers and lab switch
- Can support 802.1q if configured

The topology is implemented as follows with respect to the VM host server, control switch, and default VLAN map.



The following advanced configuration directives are implemented in the pod design to support automatic networking of the virtual machines.

```
pod.auto.net.supported=1
pod.auto.net.host.setup.supported=1
pod.auto.net.host.teardown.supported=1

pod.auto.net.control_switch.vlans=A,B

pc.1.net.vnic.1.vlan=A
pc.2.net.vnic.1.vlan=B
```

Note that VLAN C is not required in the `pod.auto.net.control_switch.vlans` directive, as there are no virtual machines in VLAN C. This avoids unnecessary setup and teardown of an empty portgroup.

VLAN maps for virtual machines are not currently supported. However, it is still possible to move virtual machines in this topology. For example, if a lab exercise wanted to relocate PC1 from the R1's stub network to R2's stub network, you could change the VLAN map from A-B-C to C-B-A; the router stub network moves to the PC, rather than the PC moving to the router stub network.

When a pod of this type starts up, a port group for VLAN A and VLAN B is created on the inside vswitch. Given a pod ID of 1 and a NETLAB+ global system ID of 1, the portgroups would be named NETLAB001_POD1_VLAN100_A and NETLAB001_POD1_VLAN101_B, respectively.

When the pod reservation is complete, NETLAB+ will remove the two portgroups from the inside vswitch.

6.3.3 pod.auto.net.control_switch.VLANs

This configuration directive specifies which VLANs are used by virtual machines in hybrid pod designs. It is used in conjunction with the pc.net.vnic.vlan directive, which binds virtual network adapters (vNIC) to a particular VLAN.

Synopsis

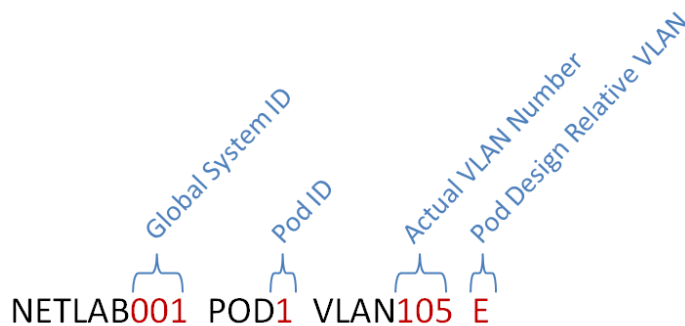
```
pod.auto.net.control_switch.vlans=A, B, C, ..., Z
```

Upon pod startup, a portgroup for each specified VLAN is created on the inside vswitch. These portgroups are removed when the pod reservation ends.

VLAN letters are used instead of numbers since the actual VLAN numbers cannot be determined at design time (as explained in section 3.5.4).

Only VLANs that contain virtual machines should be listed. Omit VLAN letters that do not contain virtual machines to avoid setup and teardown of empty portgroups.

The portgroups that are created by NETLAB on the inside vswitch are named as follows.



- **Global System ID:** A unique number assigned to the NETLAB+ system. Normally 001, but can be altered by the administrator when multiple NETLAB+ systems share a virtual host environment.
- **Pod ID:** The numeric ID assigned to the pod by the administrator.

- **Actual VLAN Number:** This is the 802.1q VLAN ID bound to the portgroup. It is automatically calculated from pod ID and relative VLAN letter.
- **Pod Design Relative VLAN:** This is the relative VLAN letter as specified in the pod design. It is shown in the portgroup name for easy association with the pod design VLANs.

Synopsis

```
pod.auto.net.control_switch.vlans=A,B,C,...,Z
```

Values

A comma-separated list of relative VLAN letters that will be used by virtual machines in the pod.

Example

```
pod.auto.net.supported=1
pod.auto.net.host.setup.supported=1
pod.auto.net.host.teardown.supported=1

pod.auto.net.control_switch.vlans=A,D,E

pc.1.net.vnic.1.vlan=A
pc.2.net.vnic.1.vlan=A
pc.3.net.vnic.1.vlan=D
pc.3.net.vnic.2.vlan=E
```

In this example, three portgroups are created for virtual machines residing in VLAN A, VLAN D, and VLAN E. The pod also uses VLAN B and VLAN C, but they are omitted from the configuration directive because they do not contain virtual machines.

Actual network bindings are performed by the pc.net.vnic.vlan directives.

- PC1 and PC2 have one network adapters that are bound to VLAN A.
- PC3 is dual-homed. Network adapter 1 is bound to VLAN D. Network adapter 2 is bound to VLAN E.

6.3.4 vswitch (category)

Synopsis

```
who=dat
```

Values

1 enabled
0 disabled

Example

```
who=date
```

6.4 Automatic Networking Configuration Directives

The following scenarios are currently supported by automatic networking:

- Pure Virtual Machines Pods with no external networking
- Hybrid pods with no external networking

The following scenarios are not currently supported by automatic networking but may be in future software versions:

- Pods connecting to external networks or the internet

6.4.1 Automatic Networking for Pure Virtual Machine Pods

```
pod.auto.net.supported=1  
pod.auto.net.host.setup.supported=1  
pod.auto.net.host.teardown.supported=1
```

```
pod.auto.net.control_switch.vlans=A,C,E
```

```
pc.1.net.vnic.1.vlan=A  
pc.2.net.vnic.1.vlan=A  
pc.3.net.vnic.1.vlan=C  
pc.4.net.vnic.1.vlan=E
```

6.4.2 PC-Only Pod Designs

When a pod design contains only PCs and no managed lab devices, it is not necessary to allocate lab device ports and VLANs for control switches in the pod design. This is generally true if there are no special connectivity requirements and the PCs in the pod just communicate amongst themselves. Simply connect them to one or more virtual switches (VMnets).

If your pod has one PC, or PCs that do not need to communicate, then no virtual switches are required. The NETLAB+ viewers work with the VMware host to display the mouse, keyboard, and screen (MKS); the virtual machine does not require a network adapter or network connection for MKS.

6.4.3 Hybrid Pod Designs

In a hybrid pod design, remote PCs and lab devices communicate with each other. Virtual machines (which are the remote PCs) connect to virtual switches on a virtual machine host (i.e., VMware ESXi). They are bound to real VLAN numbers by NETLAB+ and trunked to a pod's designated lab device ports. In NETLAB+ version 22, this happens automatically.

In NETLAB+ version 22, this is an automated process.

6.5 Virtual Machine Deferred Booting

Sometimes, it is necessary to defer the booting of a virtual machine to allow other virtual machines or resources to become available first. NETLAB+ provides three advanced configuration settings that can control deferred booting.

```
pc.<i>.boot.delay  
pc.<i>.after_pc_index=<j>  
pc.<i>.after_pc_extra=<j>
```

These settings are enforced:

- At the beginning of a lab reservation.
- When a user switches lab exercises during a reservation, if the user chooses to reset virtual machines at that time.

6.5.1 `pc.<x>.boot.delay=<seconds>`

The directive delays the boot of a virtual machine by a specified number of seconds. The index `<i>` is the index of the remote PC, as shown in the Pod Designer. The default is 0 seconds (no delay).

Synopsis

```
pc.<i>.boot.delay=<seconds>
```

Example

```
pc.2.boot.delay=30
```

This example will cause the remote PC with index 2, if present, to delay booting of the corresponding virtual machine 30 seconds after a lab reservation is started or lab exercise is switched (with VM resets).

Caveats

- This directive was added in NETLAB+ version 22.7.3 and was ignored in previous versions.
- The specified delay time will be accurate to within 5 seconds.
- Other settings may cause this value to be ignored or adjusted.

6.5.2 pc.<x>.boot.after_pc_index=<j>

This directive causes the remote PC with index <i> to boot after the PC with index <j> boots. Both <i> and <j> are the remote PC indices of the virtual machines, as shown in the Pod Designer. NETLAB+ considers a virtual machine to have booted when the VMware Tools process is running on the virtual machine.

Synopsis

```
pc.<x>.boot.after_index=<j>
```

Example

```
pc.2.boot.after_index=5
```

This example will cause the remote PC with index 2 to delay booting until the remote PC with index 5, if present, has booted.

Caveats

- This directive was added in NETLAB+ version 22.7.3 and was ignored in previous versions.
- This setting is ignored for pc.<i> if pc.<j> is absent, or pc.<j> is not used by a lab exercise.
- If pc.<i>.boot.delay is not specified, a minimum value of 120 seconds is set when pc.<x>.boot.after_index=<j> is specified. This ensures that pc.<i> will eventually

boot, even if VMware Tools on pc.<j> is never detected. This delay may be too long, so it is always best to set boot.delay to a reasonable value when using boot.after_index.

6.5.3 pc.<x>.boot.after_pc_extra=<seconds>

Used in conjunction with boot.after_index, this directive causes the remote PC with index <i> to wait an extra amount of time (in seconds) to boot, after VMware Tools is detected on pc.<j>.

Synopsis

```
pc.<x>.boot.after_extra=<seconds>
```

Example 1

```
pc.2.boot.after_index=5  
pc.2.boot.after_extra=15  
pc.2.boot.delay=60
```

This example will cause the remote PC with index 2 to defer booting until one of the following conditions occurs (whichever comes first):

- VMware Tools is running on the PC with index 5, AND 15 extra seconds have elapsed.
- 60 seconds have elapsed since the pod reservation was started.

Example 2

```
pc.2.boot.after_index=5  
pc.2.boot.after_extra=10
```

This example will cause the remote PC with index 2 to defer booting until one of the following conditions occurs (whichever comes first):

- VMware Tools is running on the PC with index 5, AND 10 extra seconds have elapsed.
- 120 seconds have elapsed since the pod reservation was started. This is the default setting for pc.2.boot.delay if not specified AND pc.2.boot.after_index is set.

Caveats







- This directive was added in NETLAB+ version 22.7.3 and was ignored in previous versions.
- This setting is ignored for pc.<i> if pc.<j> is absent, or pc.<j> is not used by a lab exercise.

- This setting is also ignored if the setting `pc.<i>.boot.after_index=<j>` is omitted.
- If `pc.<i>.boot.delay` is not specified, a minimum value of 120 seconds PLUS the `boot.after_extra` setting. This ensures that `pc.<i>` will eventually boot, even if VMware Tools on `pc.<j>` is never detected. This delay may be too long, so it is always best to set `boot.delay` to a reasonable value when using `boot.after_index`.

6.6 Remote PCs

A pod design contains very little information about remote PCs. Most of the remote PC settings are determined by the administrator when an actual pod is added to the system.

Remote PCs are optional.

General	Image	Control	Device	Remote PC	Advanced	Build
Index	Icon	Name	Hotspot	Actions		
1		PC1	Not set			
2		PC2	Not set			
3		PC3	Not set			



6.6.1 Adding a Remote PC

Click on the Add PC button. NETLAB+ will display the New Remote PC dialog.

Name	Build	Committed	Author ID	Global ID
Test123	1	No	TEST123	__6501_C07C

New Remote PC

Index







Display Icon  

Name (required)

- **Index:** Determines the order in which the PC will appear in the various lab access tabs.
- **Display Icon:** Determines which icon (PC or server) will be associated with the PC and displayed on various pages. This setting only affects the icon that is shown. It does not affect functionality.
- **Name:** The name of the PC.

6.6.2 Modifying a Remote PC

To modify the settings for a Remote PC, select **Edit** on the Actions pull-down.

Index	Icon	Name	Hotspot	Actions
1		PC1	Not set	
2		PC2	Not set	
3		PC3	Not set	

Please refer to the previous section for descriptions of each field.

Remote PC [PC1]


Index

Display Icon  

Name (required)

After completing modifications, select the **OK** button.

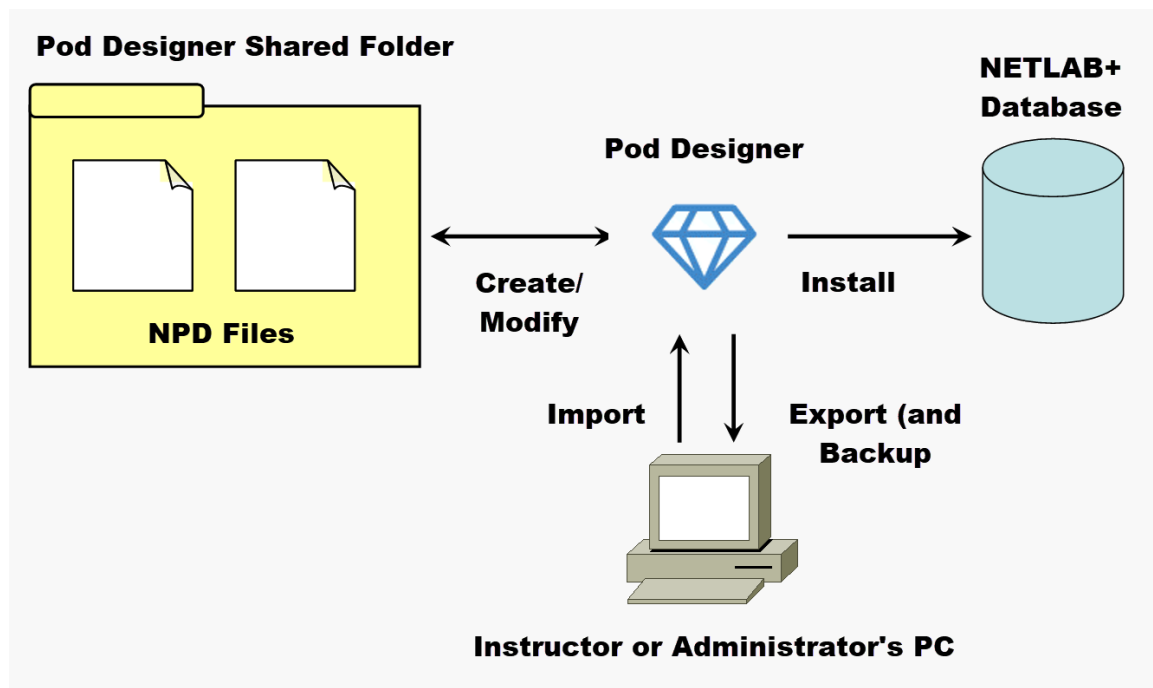
6.6.3 Removing a Remote PC

To delete a Remote PC, select the device by clicking the appropriate  button on the Remote PC tab (See section 6.6.2). The **Remove PC** button is available on the Remote PC page (See previous section).

7 Working With Pod Design Files and Builds

Pod Designer does not create actual pods. Rather, it produces an NPD file, which is a pod template. The administrator and each instructor share a common folder where NPD files are stored and manipulated by Pod Designer. NPD files can be exported to your PC, shared with others, and transferred to other NETLAB+ systems.

The NPD file format has built-in version control. Each version is called a *build*. Builds are managed from the build tab. When all desired changes have been made to the NPD file, the build is *committed*. Only a committed build can be *installed* into the NETLAB+ database, which allows the administrator to add actual pods based on the pod design.



A new build must be created to make further changes to the pod design. All changes must be made to the NPD file.

Certain settings cannot be changed once the first build is committed unless the pod design is retooled. Please see section [7.12](#) for discussion.

You cannot modify pod designs directly in the NETLAB+ database. Therefore, it is important to export, backup, and protect your NPD files.

Always keep backup copies of your pod designs. Pod design files are not automatically backed up by the system. The export function is used to make backups and is described in section [7.7](#).

When you are done working with a pod design, always click on the **Close Pod Design** button at the bottom of the tabbed interface. This will ensure that all changes are saved and the file is unlocked.

7.1 The Build Tab

Many of the tasks described in this section are managed from the Build tab. A set of buttons for different actions will appear near the end of the Build tab page. The buttons shown on the build tab will vary depending on the state of the build and if the NPD file is being modified.

Name	Build	Committed	Author ID	Global ID
Test123	2	Yes	TEST123	___6501_C07C

General	Image	Control	Device	Remote PC	Advanced	Build
Installed	Build #	Description				Retool
	2	Committed 2024-01-17 15:58:58 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5				No
	1	Committed 2024-01-16 21:53:58 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5				No

Install
New Build
Export
Delete

Close Pod Design
Modify Design

7.2 Modifying a Build

To modify a build, click on the Modify Design Button at the bottom of the tabbed interface. Once you are modifying a build, you can make changes to almost all of the settings contained within the design file.

If you are already modifying, the Modify Design button will not appear. If the current build is committed, you must create a new build as described in section [7.11](#) before changes can be made.

Name	Build	Committed	Author ID	Global ID
Test123	2	Yes	TEST123	__6501_C07C

General Image Control Device Remote PC Advanced Build

Installed	Build #	Description	Retool
	2	Committed 2024-01-17 15:58:58 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5	No
	1	Committed 2024-01-16 21:53:58 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5	No

Install New Build Export Delete

Close Pod Design Modify Design

Button does not appear if the design file is already being modified.

If a modification password is assigned to the design file, you will be asked to provide the correct password before modification is allowed.

🔒 Password Required

This file is password protected. Before changes are allowed, you must supply the **current** modification password.

Enter the current modification password: (required)

OK
Cancel
Help



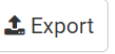

7.3 Committing a Build

The NPD file format has built-in version control. Each version is called a *build*. Builds are managed from the build tab. When all desired changes have been made to the NPD file, the build is *committed*. Only a committed build can be *installed* into the NETLAB+ database. To commit the current build, click on the Commit button in the Build tab.


You must modify the design file in order for the Commit button to appear.

Name	Build	Committed	Author ID	Global ID
Test123	3	No	TEST123	__6501_C07C

Installed	Build #	Description	Retool
	3	Modified 2024-01-17 17:03:21 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5	No
	2	Committed 2024-01-17 15:58:58 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5	No
	1	Committed 2024-01-16 21:53:58 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5	No


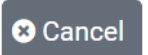
			
---	---	---	---

Confirm that you wish to proceed with the commit action by selecting **Commit** again.

 **Commit Build**

Commit build?

- Once committed, this design cannot be changed without creating a new build.

After confirming the commit, further changes to the build will be locked out. You can now install the build in the NETLAB+ database.

All parameters in the pod design can be changed during the initial (first) build. After you commit the initial build, certain settings (see below) cannot be changed unless you *retool* the pod design. Retooling will require you to reinstall any actual pods based on the design. The retooling procedure is explained in section [7.12](#).

Settings that can always be changed:	Settings that cannot be changed without retooling:
Images and Hotspots VLAN Pool Size (partial) Default VLAN Map Cable Guidance	Number of Access Server Ports Number of Switched Outlets Number of Consecutive Control Switch Ports Lab Devices Remote PCs

7.4 Installing a Build




Before a pod design can be used, it must be installed into the NETLAB+ database. If the current build is committed, an install button will appear on the build tab.


To install the current build, click on the install button located on the build tab.


Name	Build	Committed	Author ID	Global ID
Test123	3	Yes	TEST123	__6501_C07C

General
Image
Control
Device
Remote PC
Advanced
Build


Installed	Build #	Description	Retool
	3	Committed 2024-01-17 17:10:02 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5	No
	2	Committed 2024-01-17 15:58:58 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5	No
	1	Committed 2024-01-16 21:53:58 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5	No


 Install
+ New Build
 Export
 Delete

 **Install Build**



Install this build?

 **Install Build**

 Cancel

NETLAB+ will not install the current build if it is already installed.

- NETLAB+ will display a warning message if you are about to install a build that is older than the currently installed build.

If the pod design requires an installation password, you will be required to enter the correct password at this time.

Once you have installed your pod design, it becomes an asset that your classes can use and rely upon. As always, it is important to keep a backup copy of the pod design. You will need the pod design file if you want to make changes. Please see section [7.2](#).

7.5 Adding Pods

Pods are added to the NETLAB+ system using the Pods administrator function.

1. Select the **Create New Pod** button.

Pod List Search

Pod ID	Type	Pod Name	Category	Activity	State	Action
100	NDG Security Ethical Hacking v2	NDG_EH_v2_M100	Master	IDLE	ONLINE	<input type="checkbox"/>
200	NDG Security Forensics v2	NDG_Forensics_v2_M200	Master	IDLE	ONLINE	<input type="checkbox"/>

Show All entries Showing 1 to 2 of 2 items < 1 >

Create New Pod Selected Items

2. The list shows pod types provided by NDG, along with any installed custom pod designs among the selections. Select a pod type by clicking on any row.

Installed Pod Types Search

Type	Description / Notes	Author / Organization
CISCO ITE v6	AE CITE v6 - English Cisco IT Essentials v6 For use only within the Cisco Net Academies. 2016 http://netdevgroup.com/content/	
Academy CIS Pod	EMC CIS 1 EMC Cloud Infrastructure and Se 2015 http://www.netdevgroup.com/sup	
Academy ISM v2	EMC ISM v2 EMC ISM v2 2014 http://www.netdevgroup.com/sup	
NDG Security Ethical Hacking	NDG Ethical Hacking NDG Ethical Hacking 2015 http://www.netdevgroup.com/sup	
Networking A+ v2	NISGTC A+ v2 NISGTC A+ v2 2015 http://support.netdevgroup.com	

Show 5 entries Showing 1 to 5 of 15 items < 1 2 3 >

Cancel


- After selecting the pod type, the New Pod Wizard will prompt for information on control device settings. Please refer to the *Adding New Pods* section of the *NETLAB+ Administrator Guide* for details on continuing this process.

7.6 Creating Lab Exercises for Pods

In order for a pod to be available for use by a class, the pod type must be referenced in at least one lab exercise from a lab design selected for the class. Therefore, it is often desirable to create a lab exercise that allows a class to reserve a particular type of pod. This exercise will typically have no lab activity and will use the settings shown here:

Reserve MAP Pod (no lab exercise specified)

Index	3
Global Lab ID	CCNARS51_0050_56BF_5E52_5A03_37B1_0006
Lab Name	Reserve MAP Pod (no lab exercise specified)
Pod Type	AE Multi-purpose Academy Pod
Time Limit	up to class maximum
Lab Document	none
Lab Image	none
Preset Configuration	none
Assessment Options	<ul style="list-style-type: none"> • disable the action tab (AE/PE only)
VLAN Map	use default VLAN map for pod
Device Customizations	none

 Dismiss

At least one lab exercise must be created using Lab Designer in order to make the pod available to classes. Please refer to the *Relationship Between Classes, Lab Exercises, Reservations and Pods* section of the *NETLAB+ Lab Designer Guide*.

7.7 Exporting and Backing Up Pod Design Files

The export function downloads a copy of a pod design from the pod design folder on the NETLAB+ server to your local PC. You can use the export function to make a backup copy of your pod designs. Similarly, you can use the import function (see 0) to restore a pod design. Import and export can be used to share pod designs with other users and/or other NETLAB+ systems. However, only one instance of a pod design can be installed per system (see 7.4).

Name Test123	Build 3	Committed Yes	Author ID TEST123	Global ID __6501_C07C	
<div style="display: flex; justify-content: space-between; border-bottom: 1px solid #ccc;"> General Image Control Device Remote PC Advanced Build </div>					
Installed	Build #	Description			Retool
	3	Committed 2024-01-17 17:10:02 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5			No
	2	Committed 2024-01-17 15:58:58 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5			No
	1	Committed 2024-01-16 21:53:58 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5			No
<div style="display: flex; justify-content: center; gap: 10px;"> Install + New Build Export Delete </div>					

7.8 Importing and Restoring Pod Design Files

The import function uploads a copy of a pod design on your local PC to the pod design folder on the NETLAB+ server. Import is the opposite of export. You can use the import function to restore a backup copy of your pod designs. Import and export can be used to share pod designs with other users and/or other NETLAB+ systems. However, only one instance of a pod design can be installed per system (see 7.4).

1. Select the **Import Design** button located at the bottom of the main Pod Designer page.

Name	Build	Commit	Author ID	GID	Author	Size	Modified
Test123	3	Yes	TEST123	...C07C		4K	2024-01-17

Showing 1 to 1 of 1 items

Dismiss
Create New Pod Design
Show Installed Pod Designs
Import Pod Design

Select the upload button to browse to the location of the file, which must end with the **NPD** extension. Select the **OK** button to upload the specified pod design.

Import Pod Design File

To import a pod design, specify the path of an NPD file on your PC or workstation, or click the browse/choose button.

Path to Pod Design File (.npd) 📁

OK Cancel

7.9 Viewing a List of Installed Pod Designs

To see a list of the pod designs that are installed on your NETLAB+ system, select the **Show Installed Pod Designs** button located at the bottom of the main Pod Designer page.

1. The **Pod Design List** is displayed when the Pod Designer option is selected on the administrator interface.
2. To see the pod designs that have been installed in the local NETLAB+ pod type database on this system, click **Show Installed Pod Designs**.

Pod Designer									
Name	Build	Commit	Author ID	GID	Author	Size	Modified		
NISGTC Network+	12	Yes	DOLNETP	...15D6	NDGTH	124K	2014-02-10		
NISGTC Security+	9	Yes	DOLSECPV2	...5A23	NDGJJ	35K	2015-08-21		
RHSA7	3	Yes	RHSA7	...A22E	NDGJJ	37K	2016-01-13		
VMware VCA DCV 6	4	Yes	VMVCADC6	...F6DE	NDGJJ	32K	2016-01-15		
VMware vSphere ICM 6	4	Yes	VCPICM6F1	...78BE	NDGTK	72K	2016-01-14		

Buttons: Dismiss, Create New Pod Design, **Show Installed Pod Designs**, Import Pod Design

3. The currently installed pod designs will be displayed. The **Used** column indicates the number of instantiated pods of each type. Click on a pod for additional details and management options. Select the **View** option on the Action dropdown to see additional details for any of the pods displayed.

Manage Installed Pod Designs							
Name	Build	Author/Org	Used	Icon/Type	Action		
NISGTC Network+	12	NDGTH NISGTC	4	n/a	▼		
NISGTC Security+	9	NDGJJ NISGTC	3		View Uninstall		
RHSA7	3	NDGJJ NDG	0	n/a	▼		
VMware VCA DCV 6	4	NDGJJ NDG	0	n/a	▼		
VMware vSphere ICM 6	4	NDGTK NDG	0	n/a	▼		

Buttons: Dismiss


7.10 Uninstalling a Pod Design From the NETLAB+ Database

The option to uninstall a pod is available only to the NETLAB+ administrator. Select the option to **Show Installed Pod Designs** (See section 7.9) and select the file name of the pod design to uninstall. The uninstall button will display if uninstalling is permitted for the pod design. The uninstall button is not displayed for NETLAB+ pods provided by NDG.



Caution: If a pod design has been installed on your system and used to add pods to your NETLAB+ system (using the Equipment Pods administrator function; see the *NETLAB+ Administrator Guide* for details), the pods must first be deleted from the system before uninstalling the pod design. This is necessary to prevent access conflicts to control resources.

The option to uninstall a pod is available only to the NETLAB+ administrator. Select the option to Show Installed Pod Designs (See section 7.9) and select the file name of the pod design to uninstall. The uninstall button will display if uninstalling is permitted for the pod design. The uninstall button is not displayed for NETLAB+ pods provided by NDG.





General	Image	Control	Device	Remote PC	Advanced	Build
Name	Test123					
Description	Created for documentation					
Author						
Organization						
Copyright						
Support URL	http://					
Note / Comment						
Modification Password	<i>not required</i>					
Installation Password	<i>not required</i>					
Cloning Permitted	No					
Cloning Password	<i>not applicable</i>					
						

7.11 Creating a New Build

The NPD file format has built-in version control. Each version is called a *build*. A new build must be created to make further changes to a pod design. All changes must be made to the NPD file. Once a build is committed, no further modifications may be made. In order to make additional changes to the pod design, a new build must be created.

Name	Build	Committed	Author ID	Global ID
Test123	3	Yes	TEST123	___6501_C07C

Installed	Build #	Description	Retool
	3	Committed 2024-01-17 17:10:02 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5	No
	2	Committed 2024-01-17 15:58:58 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5	No
	1	Committed 2024-01-16 21:53:58 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5	No

			
---	---	---	---

The **New Build** button will be available on the build tab only if the current build has been committed. Select the checkbox to **remove pod design files from previous builds** if you wish to have the previous builds for this pod design deleted from the pod design folder on the NETLAB+ server.

You may wish to keep your previous build should any circumstance cause you to decide not to proceed with the installation of the changes you plan to make to the pod design.

+ New Build

Create new build?

remove pod design files from previous builds

OK
Cancel

Select **OK** to proceed with the build.

7.12 Retooling a Build





All parameters in the pod design can be changed during the initial (first) build. After you commit the initial build, certain settings (see below) cannot be changed unless you retool the pod design. Retooling will require you to reinstall any actual pods based on the design.

Setting that can always be changed:	Settings that cannot be changed after the first build without retooling:
Images and Hotspots VLAN Pool Size (partial) Default VLAN Map Cable Guidance	Number of Access Server Ports Number of Switched Outlets Number of Consecutive Control Switch Ports Lab Devices Remote PCs

If the settings you must change to the pod design fall into the retooling category in the table above, proceed with the **Retool** option, available on the Build tab.


1. Select the **Retool Design** button to proceed.

General	Image	Control	Device	Remote PC	Advanced	Build
Installed	Build #	Description				Retool
	4	Created 2024-01-17 18:43:40 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5				No
	3	Committed 2024-01-17 17:10:02 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5				No
	2	Committed 2024-01-17 15:58:58 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5				No
	1	Committed 2024-01-16 21:53:58 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5				No

 Commit
 Retool
 Export
 Delete

Keep in mind that it will be necessary to reinstall any pod based on the design. The extent of the modifications that may be performed using the Retool option allow modifications to the pod design which can significantly change it from the previous build. Reinstalling any pods using the design is necessary in order to prevent access conflicts to control resources.

2. Select the **Retool Design** button.

 **Retool Build**

Retooling permits unrestricted modification of the design, but will break compatibility with installed pods.




WARNING: to install a retooled build, you must delete and reinstall any pod based on this design.

After retooling, unrestricted modification of the pod design settings will be permitted.

7.13 Cloning a Build


Use the Clone feature to create derivative works based on a pod design. The cloning feature must be enabled in the general settings.

General	Image	Control	Device	Remote PC	Advanced	Build
Installed	Build #	Description				Retool
	4	Modified 2024-01-17 18:55:18 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5				Yes
	3	Committed 2024-01-17 17:10:02 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5				No
	2	Committed 2024-01-17 15:58:58 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5				No
	1	Committed 2024-01-16 21:53:58 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5				No



 Clone
 Export
 Delete

The Clone button does not appear if cloning permitted is not checked in the design.

If a Cloning password requirement has been set, the password must be entered in order to proceed.

 **Clone build**

Clone build?

 Clone
 Cancel

You will be required to enter a new Pod Design ID and modify the pod design name in order to create the clone.

A cloned pod design is a derivative work, not an exact copy. NETLAB+ automatically assigns a new globally unique ID number to the clone.

🔍 Delete Design

Delete this design file?

This will only delete the design file. Any installation of this design or pod instances will not be deleted.

🗑 Delete File
✖ Cancel

7.14 Deleting a Pod Design

To delete a pod design file from the folder, select the Delete button located on the build tab.

General	Image	Control	Device	Remote PC	Advanced	Build
Installed	Build #	Description				Retool
	1	Cloned 2024-01-17 19:12:59 UTC by administrator on NDG-VE-2ABB-CEA5-6322-F6D5				No
<div style="display: flex; justify-content: space-between; align-items: center;"> 📄 Clone 📄 Export 🗑 Delete </div>						

Please note that deleting the design file does not automatically delete the pod design from your NETLAB+ system database. To proceed with the deletion, select the **Delete File** button.

🔍 Delete Design

Delete this design file?

This will only delete the design file. Any installation of this design or pod instances will not be deleted.

🗑 Delete File
✖ Cancel