Cisco Modeling Labs 2.0 Release Notes

First Published: 2020-04-10

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com go trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

CONTENTS

CHAPTER 1

Cisco Modeling Labs 2.0

Overview of CML 2.0

Cisco Modeling Labs 2.0 is a major update of the entire Cisco Modeling Labs (CML) network simulation platform. While the platform still uses KVM as the hypervisor to run the same network OS virtual machine (VM) images, we have completely rewritten the rest of the platform. For example, we replaced the desktop GUI application with a new HTML5 browser-based user interface (UI). The software that orchestrates and runs the simulation is brand new and has a much smaller memory footprint. We greatly simplified the installation and initial simulation creation to improve the user experience. The virtual machines in the network simulations are connected via a custom-designed fabric. These changes provide for a more secure, easier-to-use network simulation platform and enable new core concepts in the product.

Starting with CML 2.0, you can think of each of your network topologies as a *lab*. You create and modify your labs on the CML server. With some limitations, you can modify the topology while the lab simulation is running. For example, you can change the connections between nodes, and you can add new nodes and connect them to the topology without stopping the simulation. Labs are also persistent by default now, unlike in the 1.x versions of the product. That is, when you stop a simulation, the disk images for the VMs in the lab are not discarded. This persistence preserves the state of each node, including crypto keys, license keys, and newly-installed packages.

CML 2.0 is built on top of REST-based web service APIs designed with both security and automation in mind. You can use these APIs to create labs and drive the entire simulation lifecycle programmatically. The new release was designed "API first" to ensure that fine-grained operations are exposed via the APIs in a consistent way. The product uses these APIs in its own user-facing interfaces:

- the HTML5 UI
- companion utilities, such as the Breakout Tool
- the Python client library

CML enables you to create and run virtual networks. You can use these labs for personal study for certification, for teaching networking classes, and for testing out new protocols or configuration changes. With the changes in the 2.0 release, CML also becomes part of a larger NetDevOps ecosystem, enabling you to test and validate network changes in an automated workflow. CML 2.0 is a complete rewrite of the product and introduces fundamental changes. If you use CML 1.x or Cisco VIRL Personal Edition 1.x, then we recommend that you read the entire CML 2.0 release notes before you get started.

Summary of CML 2.0 Changes

Feature	Description	Details
New User Interface	The new HTML5 web application provides an intuitive and feature-rich user interface.	See *Using and Configuring Cisco Modeling Labs 2.0* for instructions on using the HTML5 UI.
New Simulation Engine	This component orchestrates and controls the network simulations. The new simulation engine enables new functionality, such as more intelligent simulation launch sequencing and the ability to modify a running lab.	See *Administering Cisco Modeling Labs 2.0* for more details on the new engine.
Improved Local Resource Management	An idle CML server now consumes less memory, providing approximately 4 GB more RAM for simulations to use.	See *Administering Cisco Modeling Labs 2.0* for more details.
Console Multiplexing	Connections to the consoles of simulated nodes pass through a multiplexer, permitting multiple simultaneous connections to the same console. The first interface of each node is no longer reserved for management access.	See *Administering Cisco Modeling Labs 2.0* for more details.
External Connector	Leveraging a single virtual interface, external connectivity is now managed via a single node type, configurable to use either NAT or Bridge mode.	See *Using and Configuring Cisco Modeling Labs 2.0* for instructions on configuring the external connector.
Breakout Tool	You can run the Breakout Tool on your local machine to expose the consoles of your lab's nodes as local ports. The Breakout Tool lets you use your favorite terminal emulator application to connect to the nodes over an authenticated, encrypted connection.	See *Using and Configuring Cisco Modeling Labs 2.0* for information on installing and setting up the Breakout Tool.
SCP-enabled	SCP is now enabled out-of-the-box, enabling a simplified custom image upload process. After you upload an image via `scp`, it will appear on the **Node and Image definitions** page of the UI.	See *Using and Configuring Cisco Modeling Labs 2.0* for instructions on installing custom images.
APIs and Programmability	CML 2.0 is ready for integration into your NetDevOps automated tests with a redesigned set of REST-based web service APIs. This release also includes a Python client library that simplifies automating CML.	The API documentation is included with the product itself. For more information about the client library, visit the client library's PyPi page.

Feature	Description	Details
Dedicated System Administration	This release introduces the System Administration Cockpit, a dedicated web console for administering the CML server itself.	See *Administering Cisco Modeling Labs 2.0* for more information on performing specific changes using the System Administration Cockpit.
Smart Licensing	CML-Enterprise edition no longer uses PAK licenses. CML-Personal edition no longer uses the Cisco Salt servers for licensing. If you have an active license for 1.x, you may convert it to a smart license for the 2.0 release.	See *Administering Cisco Modeling Labs 2.0* for instructions on applying your license.

New Features in CML 2.0

HTML5 User Interface

CML 2.0 includes a brand new HTML5 web-based user interface. CML 2.0 no longer uses the VM Maestro / CML GUI application from the 1.x releases, and users will no longer need to install a GUI application on their local machines. The HTML5 UI is much simpler to use and no longer enforces a separation between design and simulation activities. The HTML5 UI provides a graphical canvas where you can create network topologies. It uses the same Canvas Network Topology component as the Cisco DNA-C product, providing high-performance rendering in a clean interface in your web browser.

You also use the same UI to interact with the running simulation for your lab. You can access the consoles of the nodes in the simulation directly in the web interface. If the node type provides a VNC server for a graphical desktop, the UI will also provide access to the VNC connection. You can use the UI to start a packet capture in your lab and even view the captured packets in a simple view without leaving the browser. The UI also leverages capabilities provided by the new simulation engine to enable you to modify a lab while the simulation is running. You can, with some restrictions, add and remove nodes in a lab or rewire their connections without restarting the simulation.

New Simulation Engine

The simulation engine in CML 2.0 is a purpose-built middleware component and no longer uses OpenStack. While the new simulation engine still uses KVM and the same virtual machine images for the network devices, the rest of the code is brand new. The new simulation engine enables many new features in CML 2.0, including the ability to modify the topology of a running simulation. This capability enables dynamic scripting for Network DevOps tests. It can also save you time when you need to modify a lab that would take a long time to restart, such as labs that consist of hundreds of nodes, use the larger VMs, or are running on a resource-constrained system.

In CML 2.0, all labs are persistent by default. That is, when you stop a node in the network simulation, the disk image for its VM is preserved for the next boot. Therefore, the startup configuration, generated cryptography keys, and licensing information that you apply to your lab will be preserved and available the next time you restart the node or lab simulation. When you stopped a simulation in CML 1.x, all of the disk state was discarded, and each time a simulation started, nodes were booting a pristine, unconfigured VM image and applying the node's bootstrap config. In CML 2.0, the persistent state for the nodes in the lab is not discarded unless you explicitly **wipe** the state. The simulation engine only applies the bootstrap configuration when a node has no associated state, which only happens the first time the node is booted or after you have wiped its persistent state.

The new simulation engine implements intelligent launch sequencing, which controls how node VMs are booted based on your hardware capacity and load. This feature should eliminate the need to stagger the launch of individual nodes manually. When you click the **Start** button to launch a lab simulation, the simulation engine will automatically sequence the launch of the nodes' VMs, resulting in a more stable system. Virtual machines should no longer come up in a failed state due to resource contention during boot-up.

Improved Resource Utilization

With the new simulation engine and the elimination of the Open Stack dependencies, CML 2.0 provides new capabilities while reducing the overall resource requirements. The memory footprint of an idle CML server, when not running a simulation, has been reduced from nearly 5 GB in the 1.x releases to approximately 600 MB. The Kernel Samepage Merging (KSM) module is enabled on the CML server by default, reducing the amount of memory used by simulations that have many nodes that use the same node type and VM image. The reduced memory footprint gives you additional capacity to run larger simulations when CML is installed on a smaller system, such as a laptop.

The elimination of the Open Stack dependencies reduces complexity in the installation, upgrade, and runtime operation of the CML server. The size of the OVA has been reduced to approximately 700 MB. Virtual machine images for the bundled node types are now delivered in a separate ISO file. CML 2.0 also requires only a single network interface, reduced from the 5 interfaces required by the 1.x releases. All of these changes create a lighter and more stable system, which is important for installation, day-to-day use, and integration into a Network DevOps pipeline.

Console Multiplexing

Starting with CML 2.0, all connections to the consoles of simulated nodes pass through a multiplexer, which permits multiple simultaneous connections to the same console. Abandoned connections won't block the console, and multiple users can view the console simultaneously. Viewing the same console simultaneously can be useful in a training, teaching, or troubleshooting environment, permitting you to view the console session of another user, assuming that you have permission to see that user's lab. Because console multiplexing prevents console connections from being blocked, CML 2.0 no longer reserves the first network interface of each virtual machine for out-of-band management access.

There are multiple ways to access console connections, and they all use the same multiplexer. You can access the console in the HTML5 UI in your web browser, via SSH to the terminal server running on the CML server, or via a connection to the **Breakout Tool**. Some node types, such as Linux desktop VMs, provide a VNC server for access to the graphical desktop. The VNC connections are multiplexed in the same manner as console connections, providing concurrent access over secure, authenticated connections via the HTML5 UI or via the **Breakout Tool** and a VNC client application on the local machine.

External Connector

CML 2.0 simplifies external connectivity, removing the concepts of FLAT and SNAT. The 2.0 release provides a single **External Connector** node type instead. You can toggle the configuration of the external connector to bridge mode or NAT mode. In bridge mode, the connection of the CML server itself is shared with the simulation, whereas in NAT mode the CML server will perform network address translation from a private IP range to the network to which the CML server connects.

Breakout Tool

With the Breakout Tool, you can use your favorite terminal emulator app to connect to your nodes' consoles on configurable local ports. In the 1.x release, each node's console was normally exposed as another open port on the CML server itself. In CML 2.0, all console connections pass through the console multiplexer. If you don't want to access consoles in the web interface or via an SSH terminal server, you can run the Breakout Tool on your local machine to authenticate to the CML server and create an encrypted connection to the console multiplexer. You can configure the Breakout Tool to expose some or all of your nodes' consoles locally. You can also configure specific ports for each lab and node, providing a consistent set of ports for a particular lab's nodes that does not interfere with the settings used by other users of the same CML server. The Breakout Tool provides access to console, AUX, and even VNC connections.

APIs and Programmability

We have completely redesigned the product's REST-based web service APIs for CML 2.0. The APIs are more consistent, accepting and returning JSON data structures and using a uniform approach to error handling. The entire product is built on top of the same APIs that are available to you. For example, the HTML5 UI uses the APIs to create and modify labs and to start and stop simulations. The Breakout Tool uses the APIs to list your lab simulations and their nodes and to open connections to the nodes.

You can use these APIs to create labs and drive the entire simulation lifecycle programmatically. The new APIs provide access to fine-grained operations, such as adding a node or link. You can also use the APIs to start and stop nodes or to break connections in the running simulation, which facilitates writing Network DevOps tests of failure scenarios. The APIs are documented using Swagger UI, and the documentation is included with the product itself. Login to the UI, and on the **Lab Manager** page, select the menu item **Tools** > **API Documentation.**

We have also released a Python client library for automating CML. The client library uses the APIs but handles the details of the web service requests so that your code doesn't have to. It provides a higher-level Python API for you to use in automating CML. For more information about the client library, visit the client library's PyPi page.

System Administration Cockpit

CML 2.0 introduces the System Administration Cockpit, a robust web console dedicated to system administration. The System Administration Cockpit console is based on Cockpit, https://cockpit-project.org/ . It allows the administrator to manage and troubleshoot the CML server in an easy-to-use and understandable web interface. The System Administration Cockpit is *not* part of CML's HTML5 UI or simulation engine: it is a separate web application that runs on the CML server. This separation allows for easier integration and more flexibility. System administrators must use this admin console to make all system changes, such as CML server IP address management, hard disk capacity, and the NTP server configuration.

By default, the System Administration Cockpit runs on port `9090` on the CML server. For example, if you access the CML UI at https://nnn.nnn.nnn.nnn/, then you would find the System Administration Cockpit at https://nnn.nnn.nnn.nnn:9090.

Smart Licensing

Starting with the 2.0 release, both CML-Enterprise and CML-Personal use Cisco Smart licensing. CML-Enterprise no longer uses PAK licenses. CML-Personal no longer uses the Cisco Salt servers for licensing. If you are a CML-Personal customer, then the Cisco Learning Network (CLN) Store allocates the Smart license on your behalf and simply provides you with auth tokens for you to use when licensing your installation. If you have a Smart license management account, you will *not* see the CML-Personal entitlement in your Smart Account. If you are a CML-Enterprise customer, Smart licensing works the same as it does for other Cisco products. Your CML-Enterprise entitlements will show up in your Smart Account in Cisco Smart Software Manager (*CSSM*), and your organization's Smart Account administrator may subdivide the node capacity entitlements to different virtual accounts, if desired.

If you have an active license for 1.x, you may convert it to a smart license for the 2.0 release. For Cisco VIRL Personal Edition customers, visit your account page on the CLN Store: https://learningnetworkstore.cisco.com/myaccount. If you have a 1.x license that is still active, you should also have a license token for licensing a CML-Personal 2.x installation.

For CML 1.x customers, you will need to convert your PAK licenses to Smart entitlements for CML-Enterprise 2.x. Visit the Cisco Software Central page, https://software.cisco.com/, and follow the link to the Smart Software Licensing page. If you are unfamiliar with Smart licensing, follow the link on the main Cisco Software Central page to Learn about Smart Accounts.

New and Changed Information

There are some product features that will not be supported until a future 2.x release. There are also some new features and changes that may impact your workflow in the product.

- **Configuration generation**. The 1.x releases used AutoNetkit (ANK) to generate detailed configurations, including IP Addresses and routing protocols, for the nodes in your topology. CML 2.0 only uses ANK to generate a basic configuration for most node types.
- **Node types**. The following node types have been replaced or removed:
    - **Ostinato** is replaced by Cisco's TRex. TRex requires two interfaces and can be managed using a GUI application that you can download separately. The TRex node's console and the node type description provide additional details.
    - **RouteM** and **iPerf** are now installed on the *Alpine* Linux node type.
    - Docker containers now run inside CoreOS, which is included as a node type and VM image.
    - Ubuntu and CoreOS nodes require a proper Cloud-Init configuration. Cloud-init takes longer when there is no IP connectivity. Cloud-init on Ubuntu takes about 30 seconds to be applied beyond the point when the "login" prompt is shown.
    - WAN emulation is achieved via a new **WAN Emulator** node type. In the 1.x releases, you could add packet loss and latency by setting properties on a connection in a running simulation. In the 2.0 release, you will instead add a Wan Emulator to your lab on the connection that you want to affect. The WAN Emulator node is basically a two-port bridge or "bump in the wire" that can be controlled via its console.
- **Clustering**. CML 2.0 does not support clustering. Adding support for clustering is on the CML-Enterprise roadmap and will be added in a future release. The architecture supports clustering, and the current CML server is effectively a "cluster of one compute."
- **System management**. The old User Workspace Management (UWM) web interface is no longer part of the product. The system administrator can manage the CML server via the System Administration Cockpit, which runs on port 9090 by default.

Note You may still import CML 1.x topologies that contain one or more of the node types that have been replaced or removed, but the imported lab may not function as expected.

Reference Platforms and Images

Disk Image Version: `refplat-20200409-fcs.iso`

Ref. Platform/Image	Description	Version
ASAv	Cisco ASA firewall image	9.12.2
CSR 1000v	IOS-XE Cloud Services Router	16.11.01b
IOS XRv	IOS XR classic image (32-bit, deprecated)	6.3.1
IOS XRv 9000	IOS XR 64-bit image	6.6.2
Nexus 7000v	NX-OS layer 3 image (deprecated)	7.3.0.d1.1
Nexus 9000v	NX-OS layer 2/3 image	9.2.3
IOSv	IOS classic layer 3 image	15.8(3)
IOSv L2	IOS classic layer 2/3 switch image	15.2
Linux Images		
TRex	Linux-based image with Cisco's packet generator	2.6.5
WAN Emulator	Linux-based image that provides WAN-like delay, jitter, and loss effects to links	3.10
Alpine Linux	Desktop Alpine Linux image that provides a graphical, Xfce interface	3.10
Tiny Core Linux	Tiny Core Linux server image	8.2.1
Ubuntu 18.04	Full-featured Ubuntu server image using cloud-init YAML configuration	18.04.3 LTS
CoreOS	Linux container-focused OS using cloud-init YAML configuration	2135.4.0

Installing CML 2.0

The CML 2.0 release only supports deployment as a virtual machine (VM). Bare metal installations may be supported in future CMLreleases. CML VM deployments are only tested and supported on specific releases of VMware products. Before you deploy the CML VM, ensure that you have installed and are running a supported release of VMware Player, Workstation, Fusion, or ESXi.

The CML 2.0 release consists of two files: a `cml2_controller` OVA file and a `refplat` ISO file. You must download **both** the OVA and ISO file before deploying the product. The OVA file is the CML server VM with the CML UI, controller, and other application software. The ISO file contains the VM images for the different node types that you can run in your CML labs.

Note Please note that Cisco images provided as part of the Cisco Modeling Labs reference platform .iso file are licensed only for use within Cisco Modeling Labs. Downloading the images and using them outside Cisco Modeling Labs without a proper license is prohibited.

For detailed installation instructions, see *Administering Cisco Modeling Labs 2.0* or *Cisco Modeling Labs 2.0 Quick Start Guide*.

Upgrade Limitations

Upgrading from Previous Releases

Cisco Modeling Labs 2.0 is a brand-new platform. In-place upgrades from the 1.6 release to the 2.0 release are not supported. If you have an existing 1.x installation of CML or Cisco VIRL Personal Edition, you will have to migrate to a new installation of CML-Enterprise or CML-Personal 2.0.

System Settings

There is no automated migration of system settings or users from 1.x to 2.0. Because of the scope of changes between the 1.6 and 2.0 releases, many of the system settings from the 1.6 release are no longer applicable. The initial set-up wizard will prompt you for required settings during the CML 2.0 installation. During the installation process, you will also create a system administrator account and an initial application user account with application administrator privileges. You can apply additional system settings in the new System Administration Cockpit after you complete the installation process using the system administrator account. You can create and manage application user accounts using the application administrator account. From the **Lab Manager** page of the UI, select **Tools** > **System Administration**.

Topology Files

CML 2.0 supports backward compatibility for .virl topologies from the 1.6 release with some limitations. You can import your legacy .virl files from your local machine to the CML server from the **Lab Manager** page in the new UI. In the 1.6 release, topology files generally reside on the local machine under the GUI's workspace folder. To find the local path to a particular topology file, right-click on the file in the **Project** view of VM Maestro or the CML UI and select **Properties** from the pop-up menu. The **Resource** > **Location** property provides the file location.

Upgrades to Future 2.x Releases

While upgrades from 1.x releases are not supported, in-place upgrades within the 2.x train will be supported for both CML-Enterprise and CML-Personal.

Related Topics

Administering Cisco Modeling Labs 2.0

Using and Configuring Cisco Modeling Labs 2.0