

# SD-WAN Lab Deployment Tool for Cisco Modeling Labs

## Contents

---

[Introduction](#)

[Background Information](#)

[Key Benefits of the Lab Deployment Tools](#)

[CML Requirements for SD-WAN Use Cases](#)

[SD-WAN Lab Deployment Tool Details](#)

[Troubleshoot](#)

[Python Version Mismatch](#)

---

## Introduction

This article describes how to create a Catalyst SD-WAN Fabric in [Cisco Modeling Labs](#) (CML) in an automated fashion using the [open-source Lab Deployment Tool](#).

## Background Information

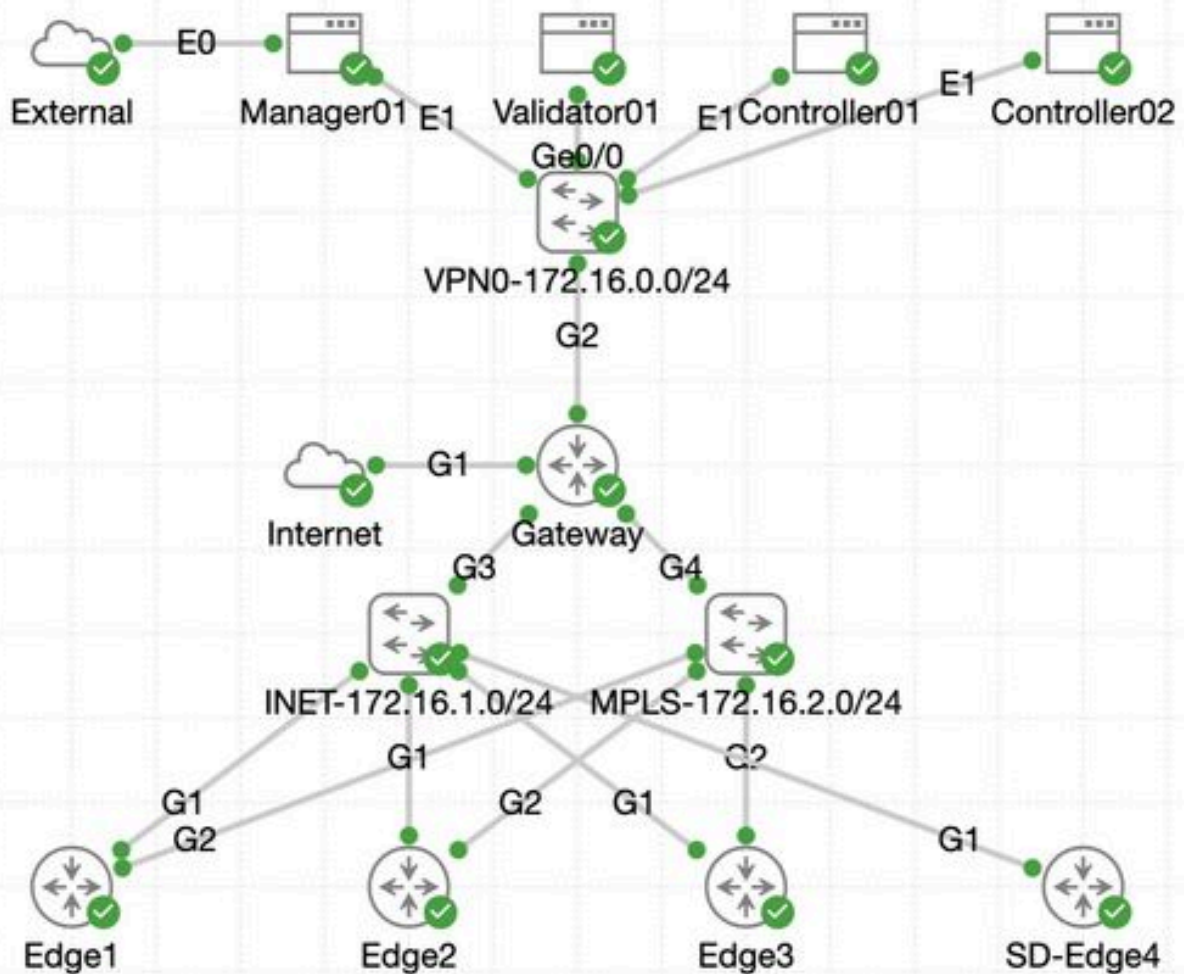
The SD-WAN Lab Deployment Tool automates SD-WAN fabric creation in CML and dramatically simplifies a lot of steps including adding SD-WAN Edge Routers to the Fabric, certificate installation and other onboarding tasks.

Before the SD-WAN Lab Tool users had to manually deploy SD-WAN controllers and edge routers in CML - see for details.

Now, the whole process is fully automated and takes minutes, not hours. See a short demo of the Lab Tool in [this YouTube video](#).

## Key Benefits of the Lab Deployment Tools

- Fully automated creation of SD-WAN Controllers
- Add multiple Branches with SD-WAN Edge routers in one automated task
- Two SD-WAN Transports: Internet and MPLS
- Ability to introduce latency as WAN Link Emulation
- Backup and Restore capabilities
- Open Source Repo on GitHub
- The [Cisco-internal version](#) of the SD-WAN Lab tool has been used **over 900 times** since its creation in different organizations including sales, CX (TAC and PS) and BU. See more details on the Cisco-internal version .



## CML Requirements for SD-WAN Use Cases

- CML can be deployed in an on-prem server (bare-metal or as VM on ESXi) or on AWS.
- Supports clustering for increasing capacity.
- For Cisco internal users: license can be obtained .
- For external customers: different license tiers available [here](#).
- Installation Guide (same for internal and external users) is [here](#).
- The recommended option for SD-WAN Simulation: on-prem CML cluster with few servers.

## SD-WAN Lab Deployment Tool Details

- The tool is currently supported on Linux/MacOS systems as a dependency package (pyats) is available only for those systems.
- To run the tool on Windows, you will need a Linux VM/container or Linux on Windows with [WSL](#). You can find all the dependencies in the `pyproject.toml` file: <https://github.com/cisco-open/sdwan-lab-deployment-tool/blob/main/pyproject.toml>
- See all installation steps, usage examples and source code on GitHub.
- Ensure that the network connection between the host running the Lab Tool and the CML server/cluster is fast enough. This helps during the SW image upload of the SD-WAN Controllers.
- Got questions about the SD-WAN Lab Deployment Tool? Send an email to [cml-sdwan-lab-](mailto:cml-sdwan-lab-@cisco.com)

## Troubleshoot

### Python Version Mismatch

As of the End of April 2024, the Lab Tools requires Python version 3.9. If you have a more recent Python version, the installation may fail. Please note, that python 3.12 support will come soon.

Example of a failed installation because of the wrong Python version:

```
(venv) [csdwan] $ pip install --upgrade catalyst-sdwan-lab
Collecting catalyst-sdwan-lab
  Downloading catalyst_sdwan_lab-2.0.9-py3-none-any.whl.metadata (20 kB)
...
  Downloading catalyst_sdwan_lab-2.0.6-py3-none-any.whl.metadata (20 kB)
ERROR: Cannot install catalyst-sdwan-lab==2.0.6, catalyst-sdwan-lab==2.0.7, catalyst-sdwan-lab==2.0.8 a
```

The conflict is caused by:

```
catalyst-sdwan-lab 2.0.9 depends on pyats<24.0 and >=23.1
catalyst-sdwan-lab 2.0.8 depends on pyats<24.0 and >=23.1
catalyst-sdwan-lab 2.0.7 depends on pyats<24.0 and >=23.1
catalyst-sdwan-lab 2.0.6 depends on pyats<24.0 and >=23.1
```

To fix this you could try to:

1. loosen the range of package versions you've specified
2. remove package versions to allow pip attempt to solve the dependency conflict

```
ERROR: ResolutionImpossible: for help visit https://pip.pypa.io/en/latest/topics/dependency-resolution/
(venv) [csdwan] $
```

To fix the issue, you can put Python 3.9 in front of 3.12 in your environment variables, use `pyenv` to switch between different versions ("pyenv global 3.9") or downgrade from 3.12 to 3.9.