eNMS

1.0

2018 07 25

Contents

CHAPTER 1

eNMS ?

eNMS designed to help automate networks. While network automation traditionally requires scripting skills, eNMS provides a way to automate networks **graphically**. It encompasses the following aspects of network automation:

- **Configuration management**: commit/rollback of a configuration via NAPALM.
- **Netmiko scripting**: using netmiko to push a configuration, or display the result of a set of commands.
- **Ansible support**: sending and managing ansible playbooks.
- **Workflows**: all scripts can be organized in workflows (conditional graph of scripts executed in a specific order).
- **Scheduling**: any script/workflow can be scheduled to run at a specific time, periodically or not.

1.1 Design philosophy

eNMS provides a way to automate networks **graphically**, in a few simple steps:

1. Creation of the network (e.g by importing a spreadsheet describing the network topology).
2. Creation of the scripts and workflows.
3. Visualization of the network on a world map, or via a force-based algorithm.
4. Selection of the target devices graphically, and scheduling of the script/workflow.

1.2 Application stack

eNMS is built on the *Flask* Python framework and utilizes either a *SQLite* database, or a *PostgreSQL* database. It runs as a WSGI service behind your choice of HTTP server.

Function	Component
HTTP Service	nginx or Apache
WSGI Service	gunicorn or uWSGI
Application	Flask/Python
Database	SQLite or PostgreSQL

CHAPTER 2

Installation

2.1 First steps

```
# download the code from github:
git clone https://github.com/afourmy/eNMS.git
cd eNMS

# install the requirements:
pip install -r requirements.txt
```

2.2 Start eNMS in debugging mode

```
# set the FLASK_APP environment variable
(Windows) set FLASK_APP=enms.py
(Unix) export FLASK_APP=enms.py

# set the FLASK_DEBUG environment variable
(Windows) set FLASK_DEBUG=1
(Unix) export FLASK_DEBUG=1

# run the application
flask run --host=0.0.0.0
```

2.3 Start eNMS with gunicorn (better)

```
# start gunicorn
gunicorn --config gunicorn.py enms:app
```

2.4 Start eNMS as a docker container (even better)

```
# download & run the container
docker run -d -p 5000:5000 --name enms --restart always afourmy/enms
```

2.5 Once eNMS is running, go to http://127.0.0.1:5000

CHAPTER 3

Objects

3.1 Objects

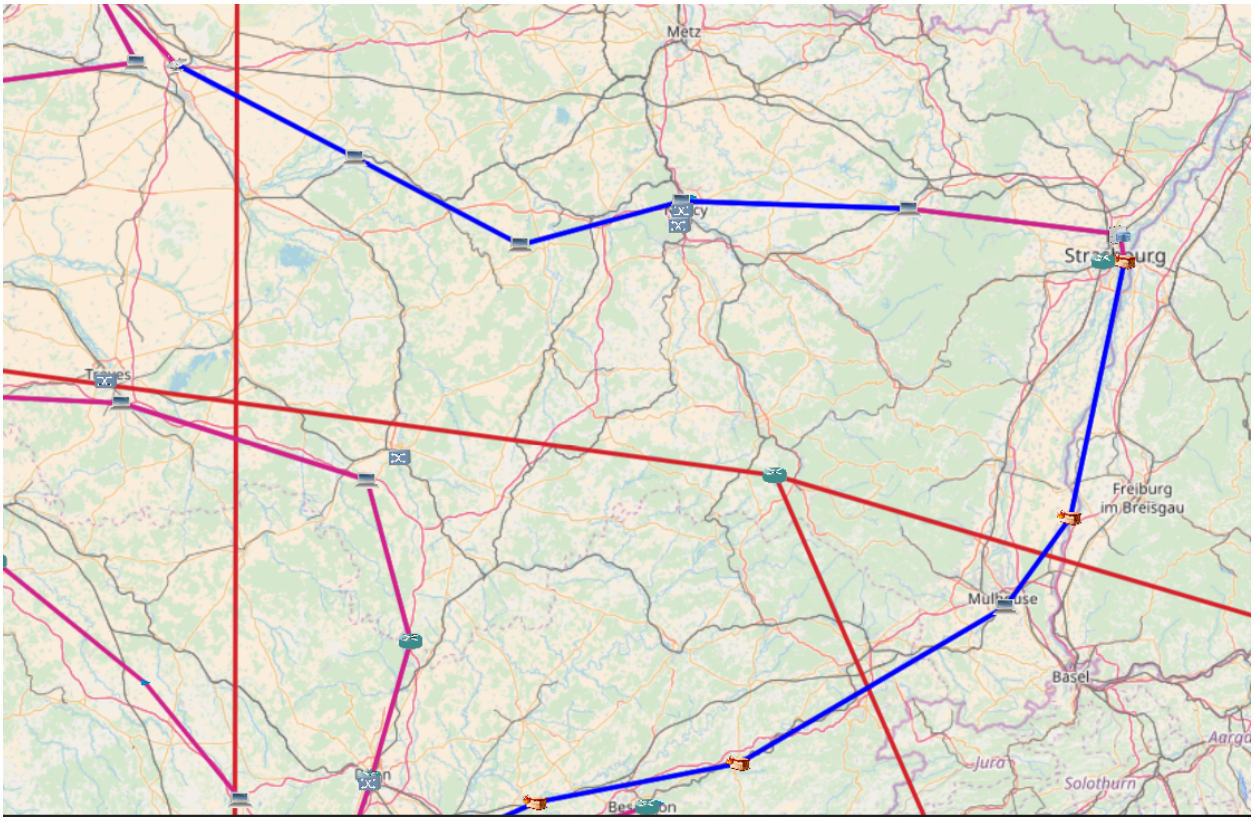
3.1.1 Object creation

Type of objects

There are different types of nodes and links available in eNMS.

- **Node**: Router, Switch, Optical switch, Server, Host, Antenna, Regenerator, Firewall.
- **Link**: Ethernet link, Optical link, Etherchannel (LAG), Optical channel, Pseudowire, BGP peering.

Each type of node (resp. link) has a specific icon (resp. color) when displayed graphically:



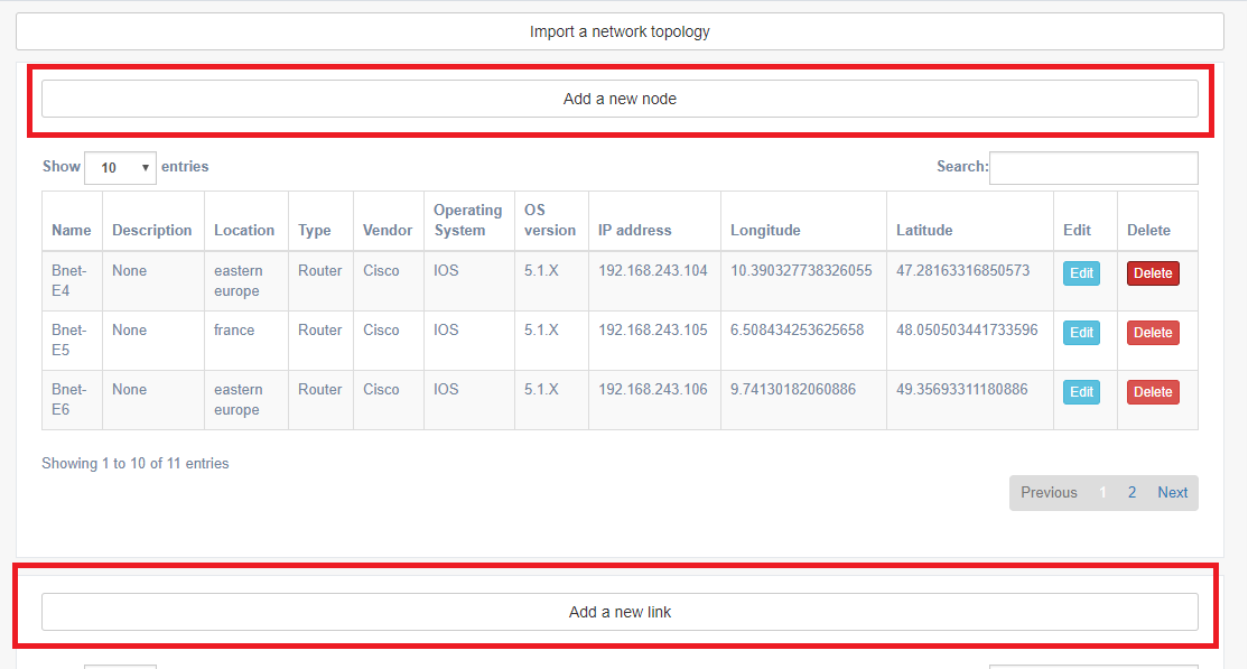
Creation

Objects can be created from the *objects/object_management* page, in two different ways:

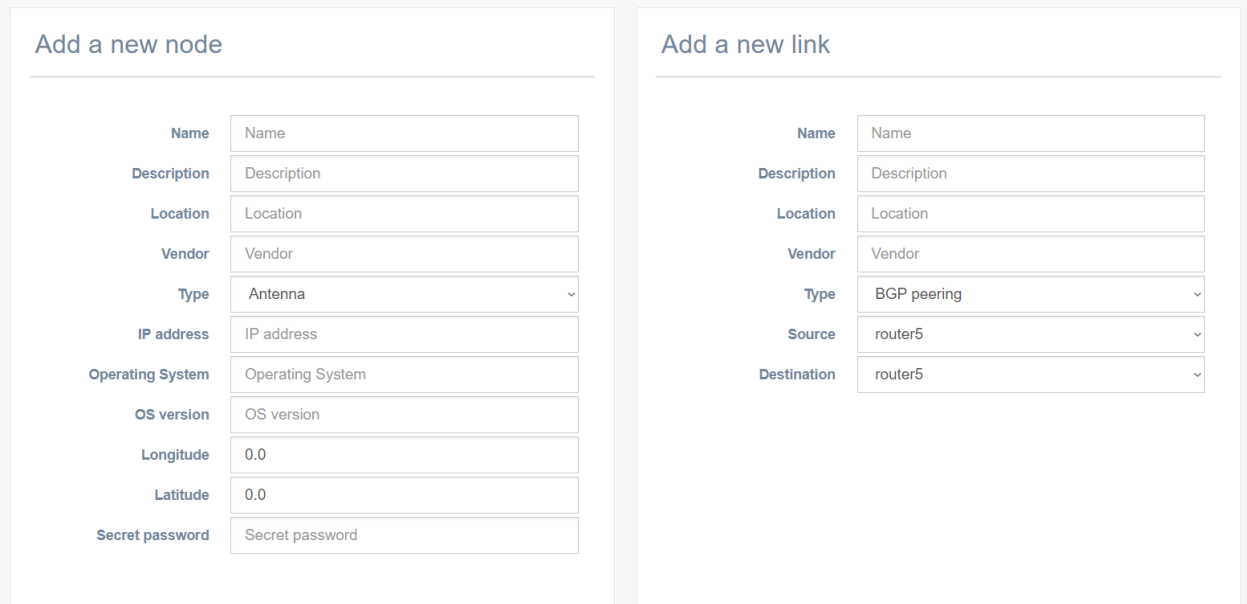
- Manually, by entering the value of each property in a form. With this method, objects have to be created one by one.
- By importing an Excel file (.xls, .xlsx).

Manual creation

Clicking on the `Add a new node` or `Add a new link` buttons will open a form with the list of all properties of the object.



Fill the form and click on the `Submit` button.



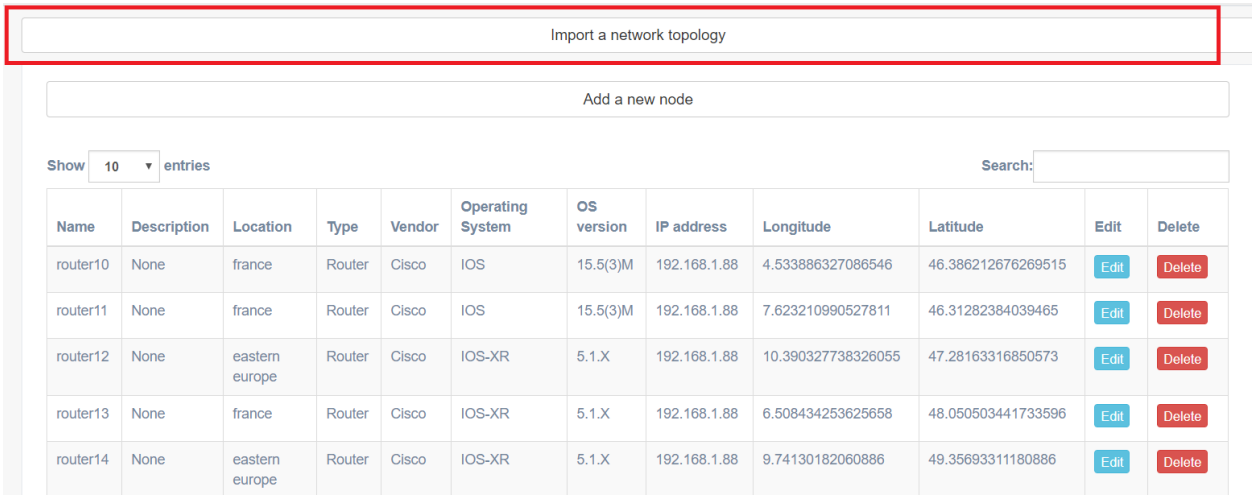
Creation via import

All objects can be created at once by importing an Excel file. Each spreadsheet corresponds to a type of object. The first line of a spreadsheet contains the properties, the following lines define the objects, as demonstrated in the example below.

	A	B	C	D	E	F	G	H
1	name	longitude	latitude	ip_address				
2	PSS00222	3,2516295	48,1906625826	127.0.0.1				
3	PSS00282	5,1038716	43,650057334	127.0.0.1				
4	PSS00309	5,4257466	43,3835984573	127.0.0.1				
5	PSS00318	5,8454402	43,0530959941	127.0.0.1				
6	PSS00114	-1,002317	46,9013737948	127.0.0.1				
7	PSS00112	-0,74675	47,1477630559	127.0.0.1				
8	PSS00295	-1,959262	48,2880336584	127.0.0.1				
9	PSS00113	-0,903417	47,0966509344	127.0.0.1				
10	PSS00029	2,427925	47,112136027	127.0.0.1				
11	PSS00032	2,0926905	47,2416520502	127.0.0.1				
12	PSS00155	3,0886	50,6221843839	127.0.0.1				
13	PSS00179	2,2273361	48,76928239	127.0.0.1				
14	PSS00339	2,2192851	48,7878597134	127.0.0.1				
15	PSS00340	2,2192851	48,7878597134	127.0.0.1				
16	PSSM00001	2,2192851	48,7878597134	127.0.0.1				
17	PSSM00002	2,2192851	48,7878597134	127.0.0.1				
18	PSSM00003	2,2192851	48,7878597134	127.0.0.1				
19	PSSM00004	2,2192851	48,7878597134	127.0.0.1				
20	PSSM00005	2,2192851	48,7878597134	127.0.0.1				
21	PSSM00006	2,2192851	48,7878597134	127.0.0.1				
22	PSSM00007	2,2192851	48,7878597134	127.0.0.1				
23	PSSM00008	2,2192851	48,7878597134	127.0.0.1				
24	PSSM00009	2,2192851	48,7878597134	127.0.0.1				
25	PSS00177	2,0357938	48,7944240057	127.0.0.1				
26	PSS00327	2,0357938	48,7944240057	127.0.0.1				

Antenna | Firewall | Host | Optical switch | Regenerator | **Router** | Switch | Server | BGP peering | Etherchannel | Ethernet link

For eNMS to let you choose an Excel spreadsheet to import, click on the `Import a network topology` button in the `object_management` page.



: You can find examples of such spreadsheets in *eNMS/projects*.

: If an imported object already exists, its properties will be updated.

Properties

Some properties are mandatory:

- Name: objects are uniquely defined by their name.
- Source and destination: a link needs a source and a destination to be created.

: In order to visualize the network topology on a map, nodes must have geographical coordinates (longitude and latitude).

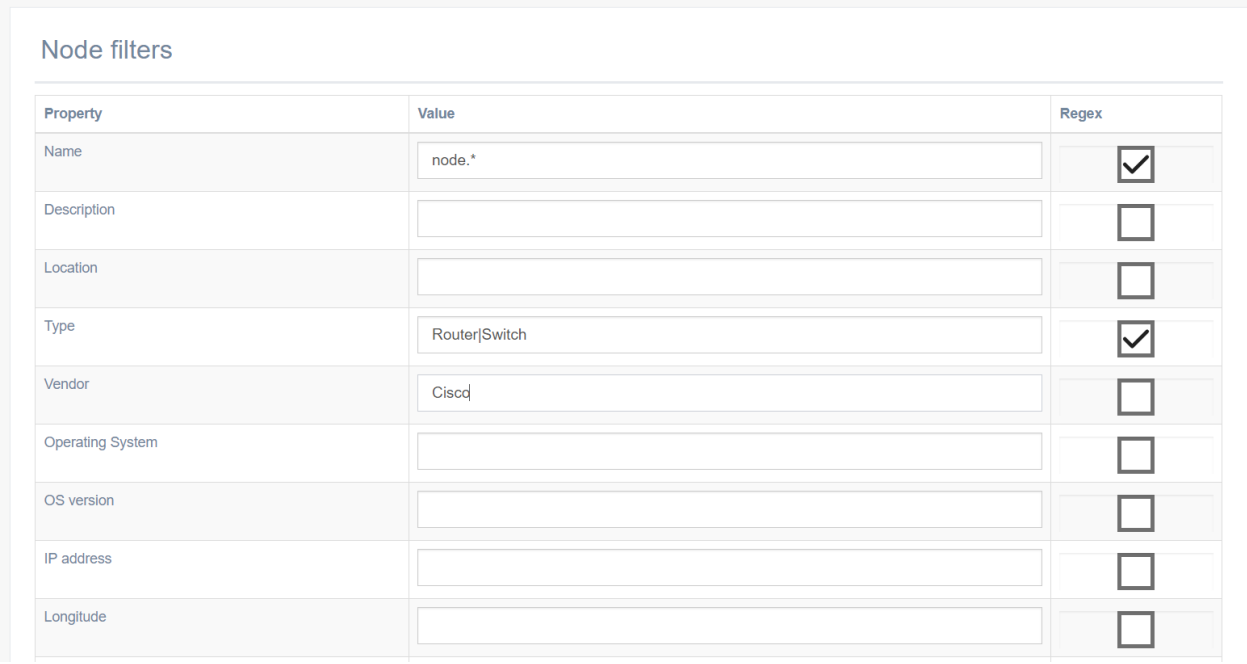
3.1.2 Object filtering

The filtering system allows to display only a subset of the network in the graphical view. A filter is a combination of values (or regular expressions) for the properties of an object.

If the properties of an object does not match **all** properties of the filter, the object will be undisplayed when the filter is selected.

Filters can be created in *objects/object_filtering*.

Node filtering



This filter enforces the following conditions:

- name: `node.*` — this regular expression matches all nodes which name starts with `node`.
- type: `Router|Switch` — matches routers and switches (nodes which type is either `Router`, or `Switch`.
- vendor: `Cisco` — for this property, the regular expression box is not ticked. This means the value must be exactly `Cisco`.

In summary, all Cisco routers or switches which name starts with `node` will match those conditions. All others will be filtered (that is, undisplayed from the graphical view).

: All properties which field is left empty are simply ignored.

Link filtering

This filter enforces the following conditions:

- type: `Ethernet link` — matches all Ethernet links.
- source: `bnet6` — matches all links which source is the node `bnet6`.

In summary, all Ethernet links starting from the node `bnet6` will be considered, all others ignored.

Apply a filter

Filters are applied from the geographical or logical view. You can switch between filters with the drop-down list in the top-right corner of the screen (framed in red below).

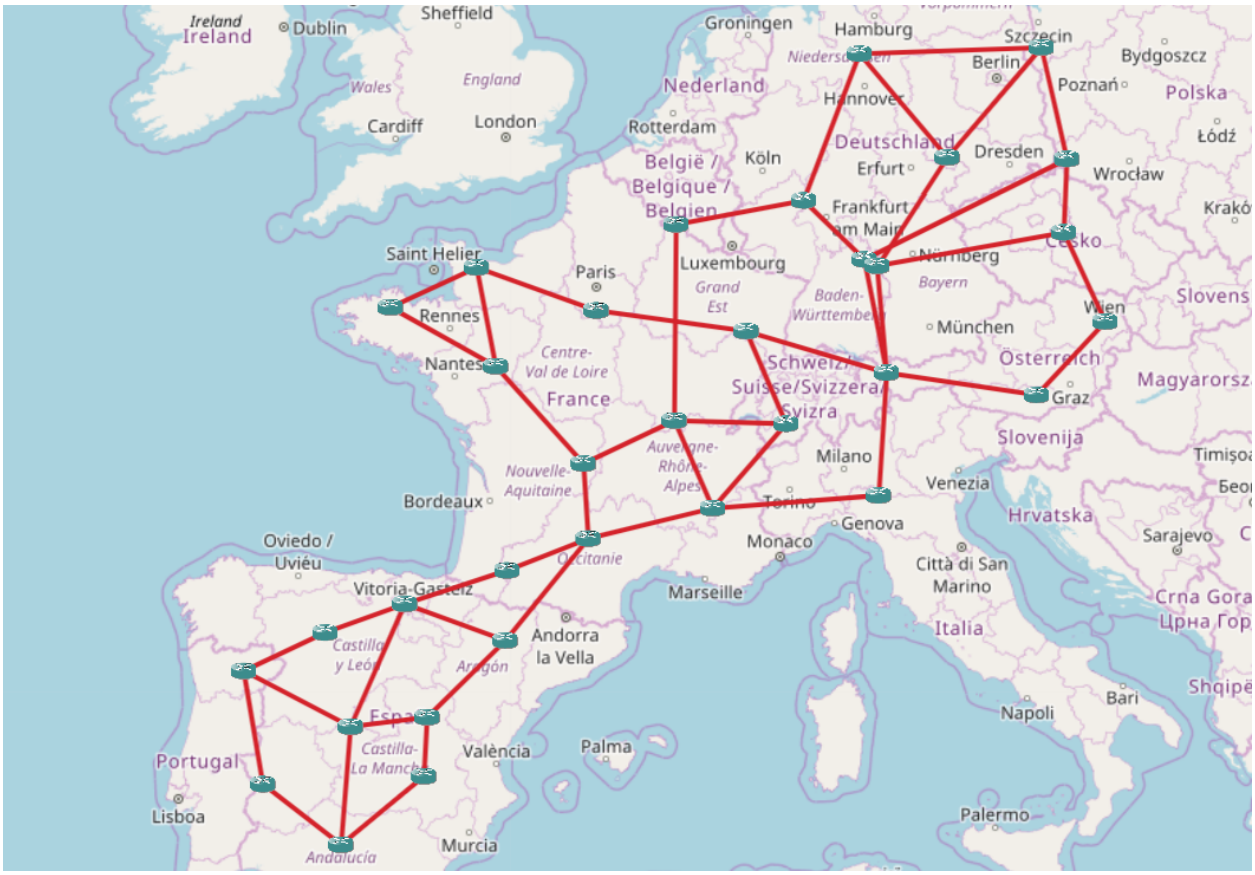
Example

Initial network

In this first example, we consider the following network:

	A	B	C	D	E	F	G	H
1	name	location	longitude	latitude	ip_address	operating_system	vendor	os_version
2	router5	france	-0.4080065865215727	47.403792182567656	192.168.1.88	IOS	Cisco	15.5(3)M
3	router6	france	2.366904644610879	48.421248383492745	192.168.1.88	IOS	Cisco	15.5(3)M
4	router7	france	4.553994745638234	49.98884227261501	192.168.1.88	IOS	Cisco	15.5(3)M
5	router8	eastern europe	8.084503313789972	50.41433737503949	192.168.1.88	IOS	Cisco	15.5(3)M
6	router9	france	2.0137607972959803	45.559733221855026	192.168.1.88	IOS	Cisco	15.5(3)M
7	router10	france	4.533886327086546	46.386212676269515	192.168.1.88	IOS	Cisco	15.5(3)M
8	router11	france	7.623210990527811	46.31282384039465	192.168.1.88	IOS	Cisco	15.5(3)M
9	router12	eastern europe	10.390327738326055	47.28163316850573	192.168.1.88	IOS-XR	Cisco	5.1.X
10	router13	france	6.508434253625658	48.050503441733596	192.168.1.88	IOS-XR	Cisco	5.1.X
11	router14	eastern europe	9.74130182060886	49.35693311180886	192.168.1.88	IOS-XR	Cisco	5.1.X
12	router15	eastern europe	10.109935828061502	49.22507496520804	192.168.1.88	IOS-XR	Cisco	5.1.X
13	router16	france	5.595254381372841	44.69860318923541	192.168.1.88	IOS-XR	Cisco	5.1.X
14	router17	france	2.1511313297338606	44.098423304433176	192.168.1.88	IOS-XR	Cisco	5.1.X
15	router18	eastern europe	12.041475611784332	51.16509496536982	192.168.1.88	IOS-XR	Cisco	5.1.X
16	router19	eastern europe	14.637732184922653	53.002511516276 22	192.168.1.88	IOS-XR	Cisco	5.1.X
17	router20	eastern europe	9.620913979151592	52.898908251546565	192.168.1.88	Junos	Juniper	17.2R1.13
18	router21	eastern europe	15.322252610306014	51.13069724514281	192.168.1.88	Junos	Juniper	17.2R1.13
19	router22	eastern europe	14.509961201590082	46.86709038736285	192.168.1.88	Junos	Juniper	17.2R1.13
20	router23	eastern europe	16.394243292075355	48.2265031612354	192.168.1.88	Junos	Juniper	17.2R1.13
21	router24	eastern europe	15.23113134650369	49.84702977845239	192.168.1.88	Junos	Juniper	17.2R1.13
22	router25	france	-0.9363500525806724	49.20978690554773	192.168.1.88	Junos	Juniper	17.2R1.13
23	router26	france	-3.304732896119501	48.49233507313489	192.168.1.88	Junos	Juniper	17.2R1.13
24	router27	france	-0.07442133974525358	43.45727942401689	192.168.1.88	Junos	Juniper	17.2R1.13
25	router28	spain	-7.341918530493528	41.418677137969205	192.168.1.88	Junos	Juniper	17.2R1.13
26	router29	spain	-5.1153688818914205	42.187262151915185	192.168.1.88	Junos	Juniper	17.2R1.13

Unfiltered, this network results in the following view:



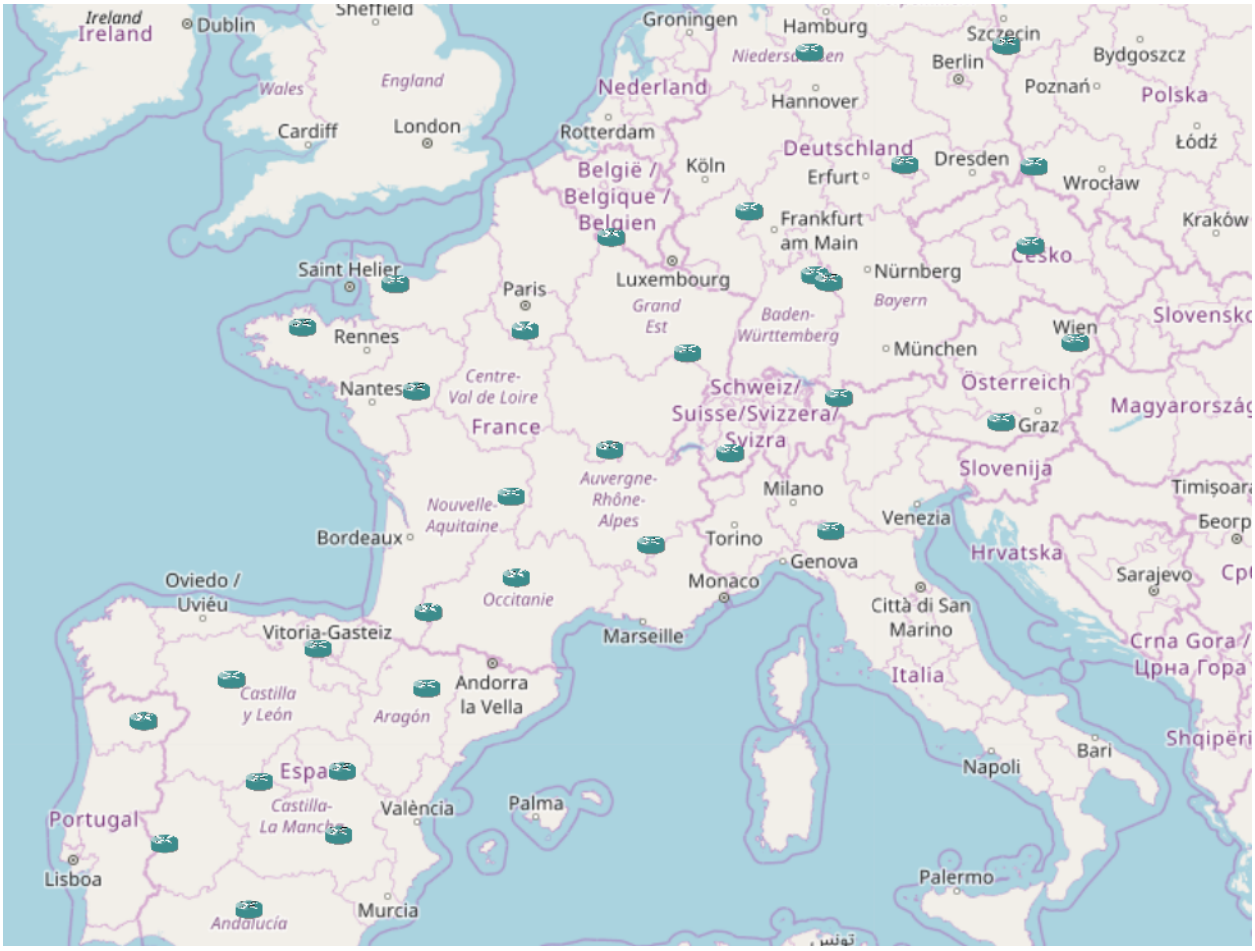
Filter all links

We create a filter with a condition on the `Name` of a link:

Property	Value	Regex
Name	a	☐

There isn't a single link which name is `a`: all links will be filtered.

This result in the following view:

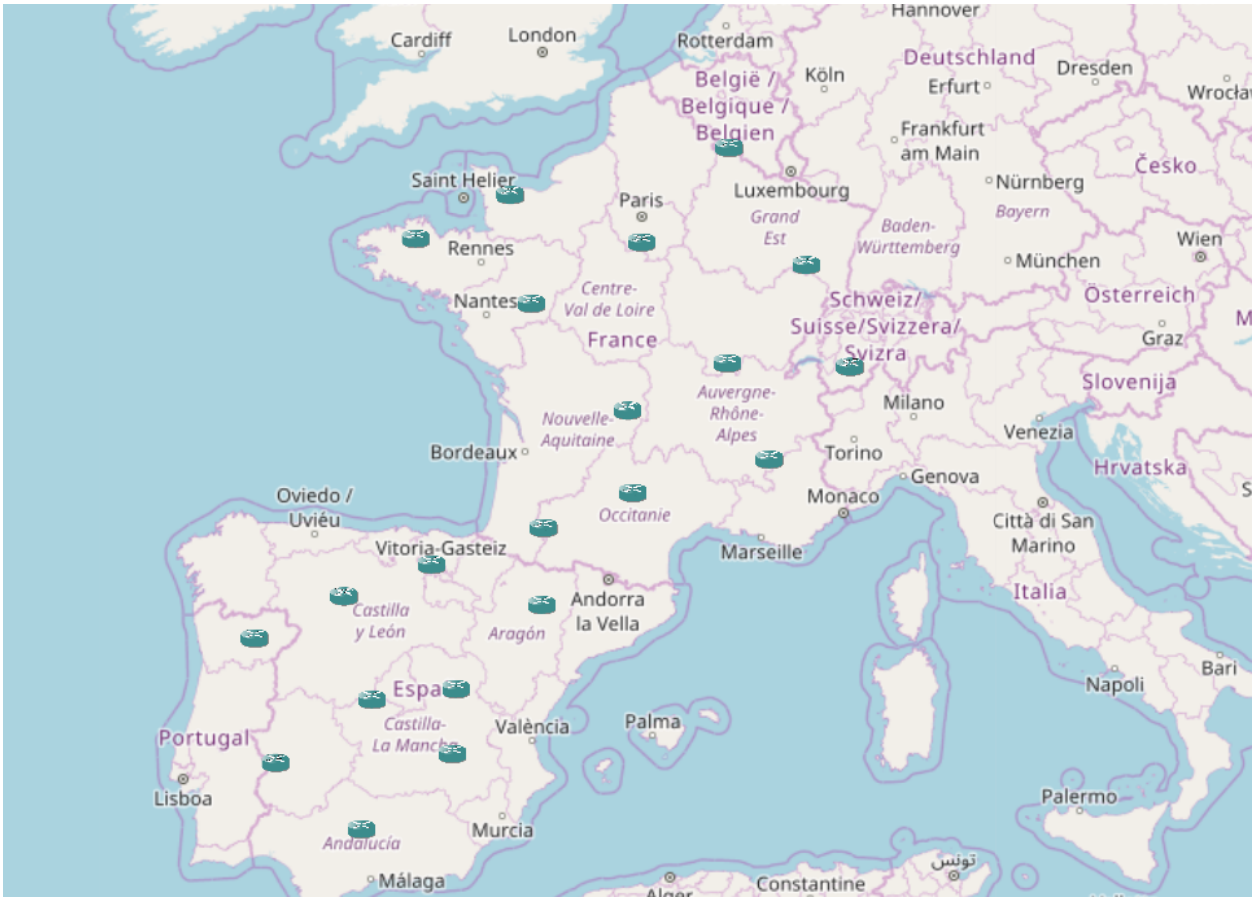


Filter all nodes outside of France or Spain

We add a new condition on the `Location` of a node to exclude all nodes that are located outside of France or spain:

Location	france|spain	☑

This result in the following view:

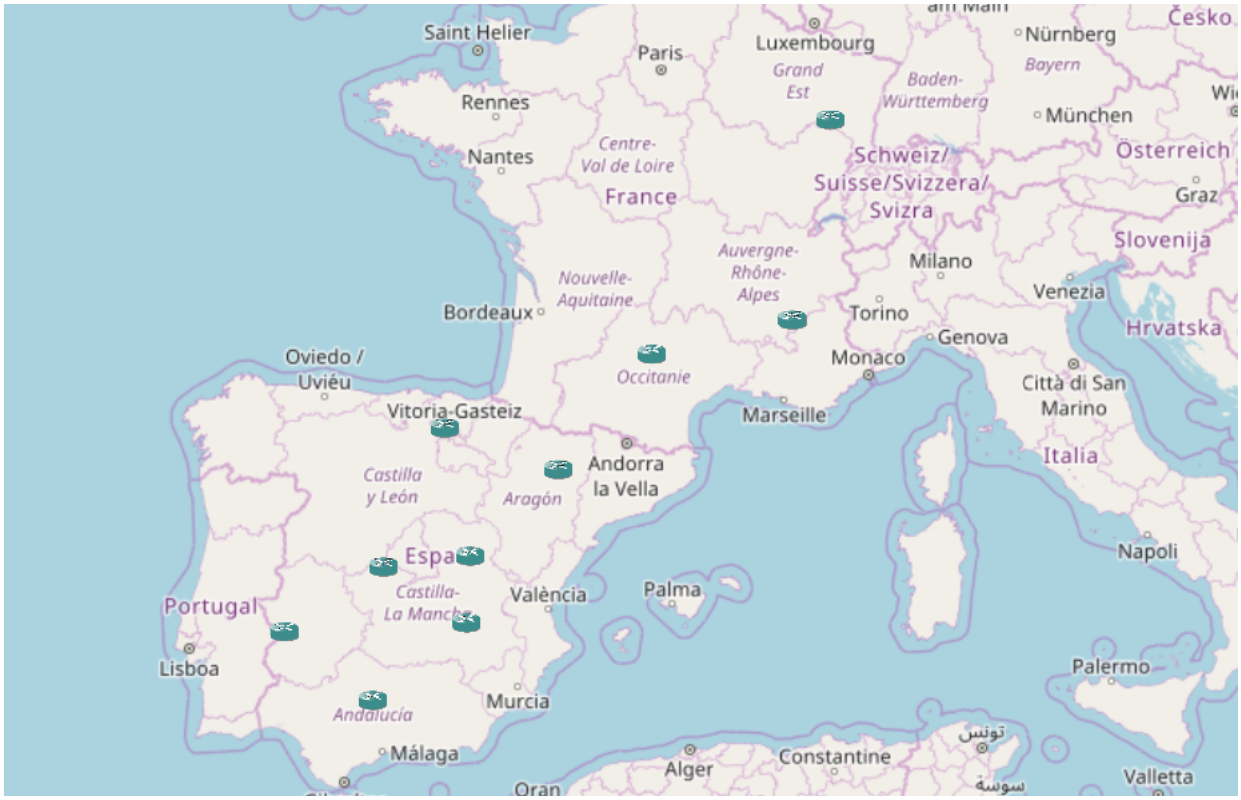


Restrict to nodes with IOS-XE or IOS-XR

Finally, out of the remaining nodes, we exclude all nodes which operating system is not IOS-XE or IOS-XR:

Operating System	IOS-(XE|XR)	☑

This result in the following view:



: Using the filtering system is important because network automation in eNMS is done graphically, by selecting nodes in the graphical view. See the *automation* documentation for more information.

CHAPTER 4

Views

4.1 Views

4.1.1 Geographical view

Once the network has been created, it can be displayed on a geographical map.

All nodes are displayed at their exact location (they must have been created with a longitude and latitude) on the map. The icon of a node and the color of a link depend on their type.

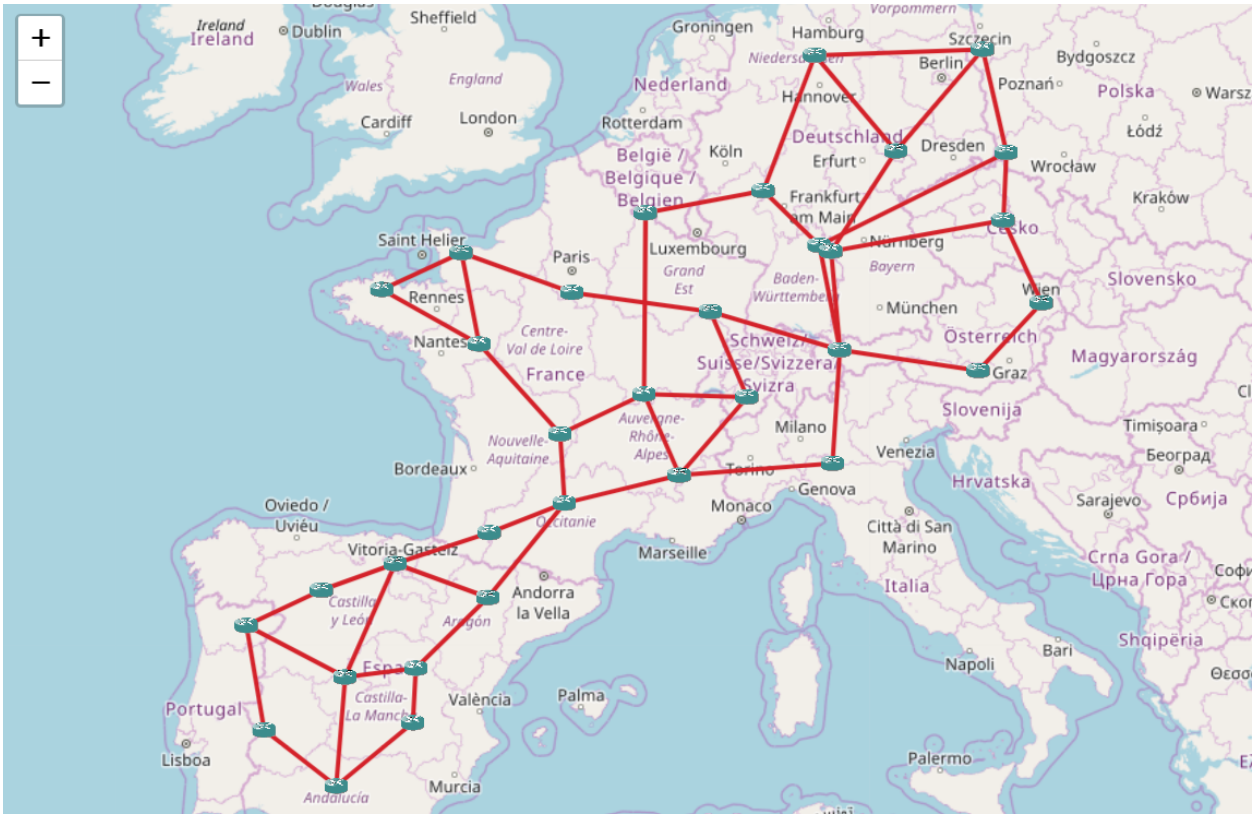
Geographical display

There are three types of geographical displayed available.

2D map

The classic 2D map is based on the *Leaflet* JavaScript library. All nodes and links are displayed on a 2D representation of the Earth, based on the `Google Mercator (EPSG:3857)` projection.

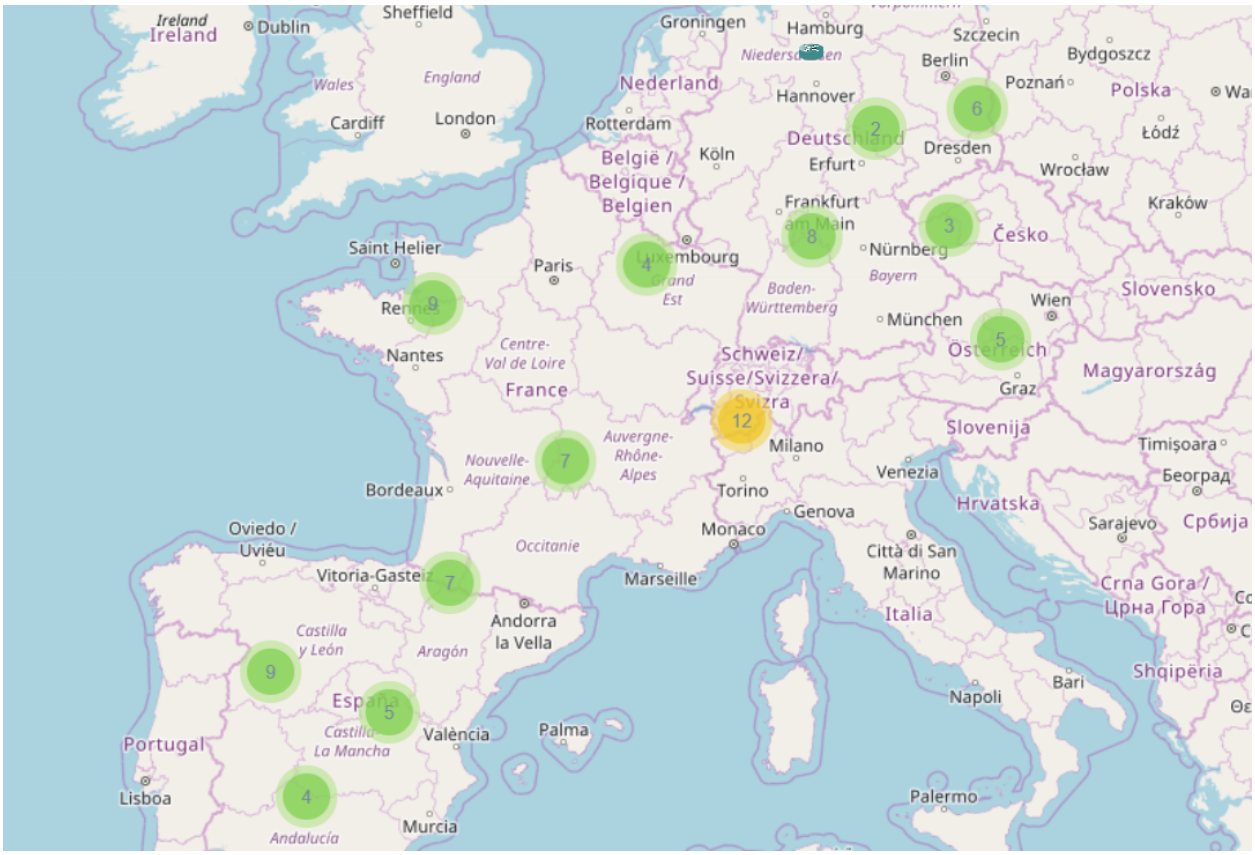
The classic 2D map works well for small networks (less than 5000 nodes), when there are no colocated nodes (colocated nodes cannot be distinguished from one another).



Clusterized 2D map

The clusterized 2D map is based on the *Leaflet MarkerCluster* JavaScript library. Nodes and links are displayed as clusters, which size depends on the zoom level.

The clusterized map works well for large networks (until 50000 nodes), and it supports colocated nodes. Clicking on a group of colocated nodes will expand the group.



3D map

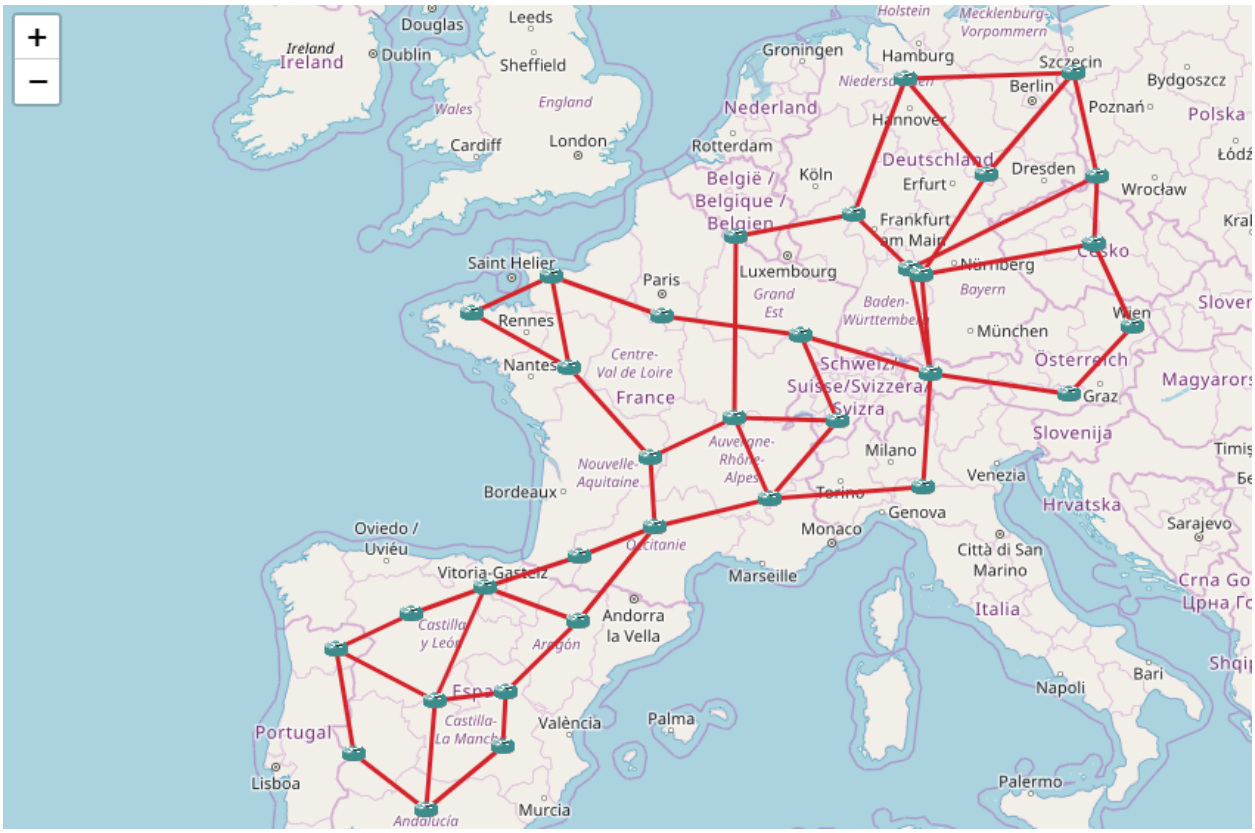
The 3D map is based on the *WebGL-Earth* JavaScript library (which itself uses *Cesium*) All nodes and links are displayed on a 3D representation of the Earth.

The 3D map works well for small networks (less than 500 nodes) with no colocated nodes.

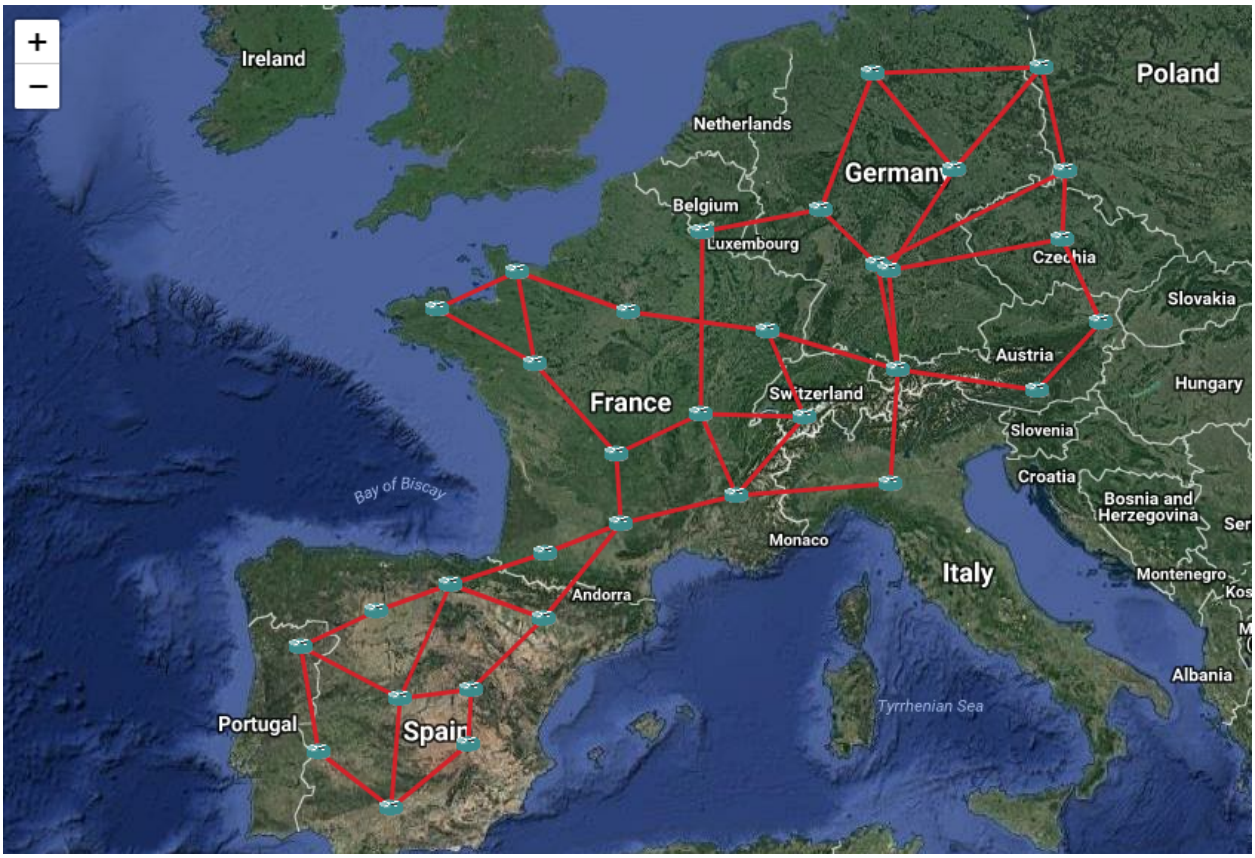
Tile layers

There are three types of tile layer available for the geographical display.

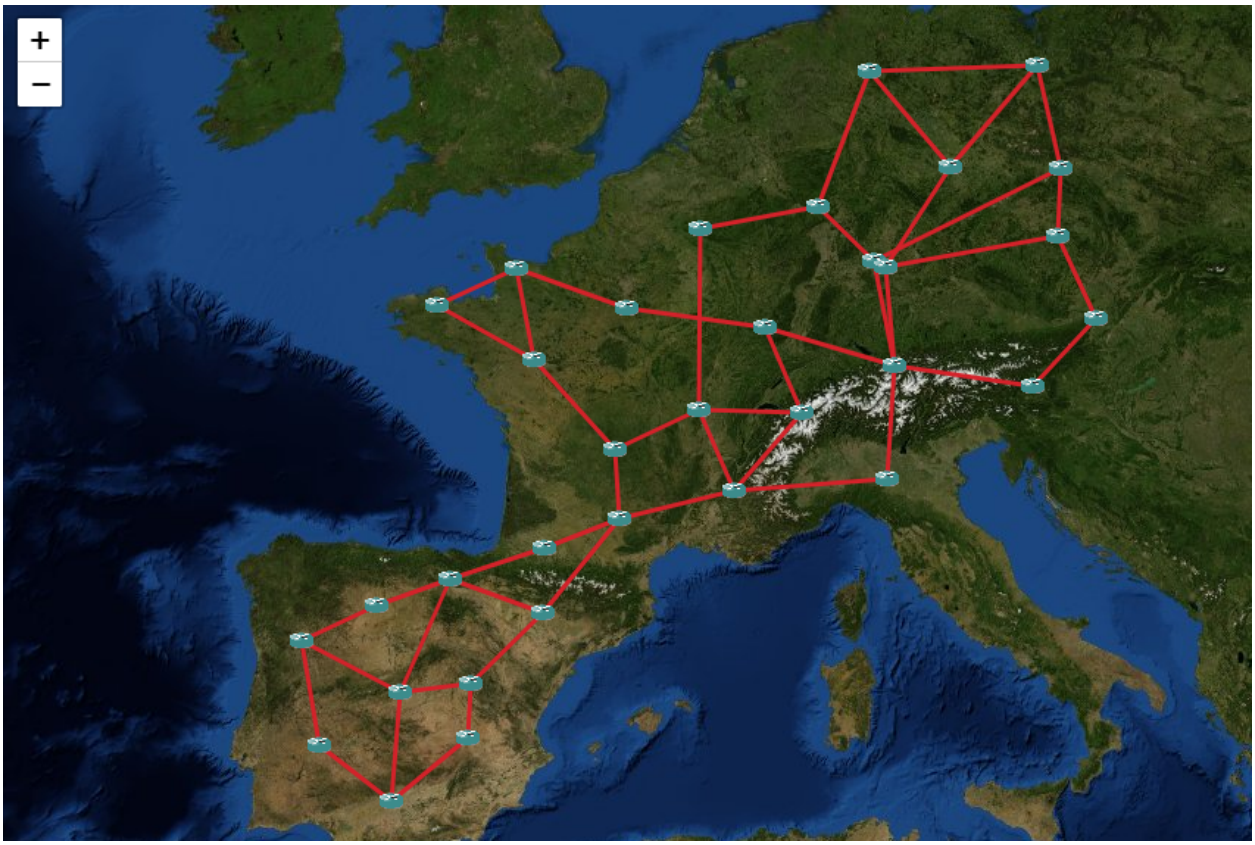
OpenStreetMap tiles



Google Map tiles



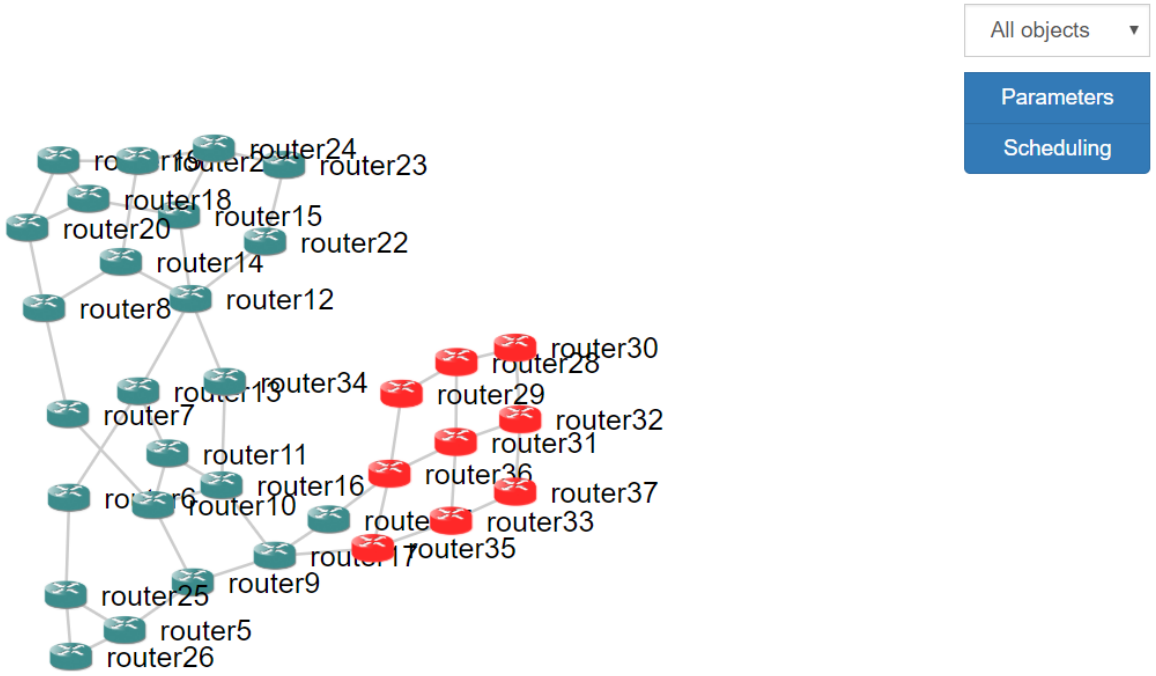
NASA tiles



4.1.2 Logical view

The geographical visualization only works if we have all GPS coordinates: it isn't always the case. Another way to visualize a network is to use a graph drawing algorithm to display the network.

The logical view relies on the *d3.js* JavaScript library to display the network in a visually pleasing fashion.

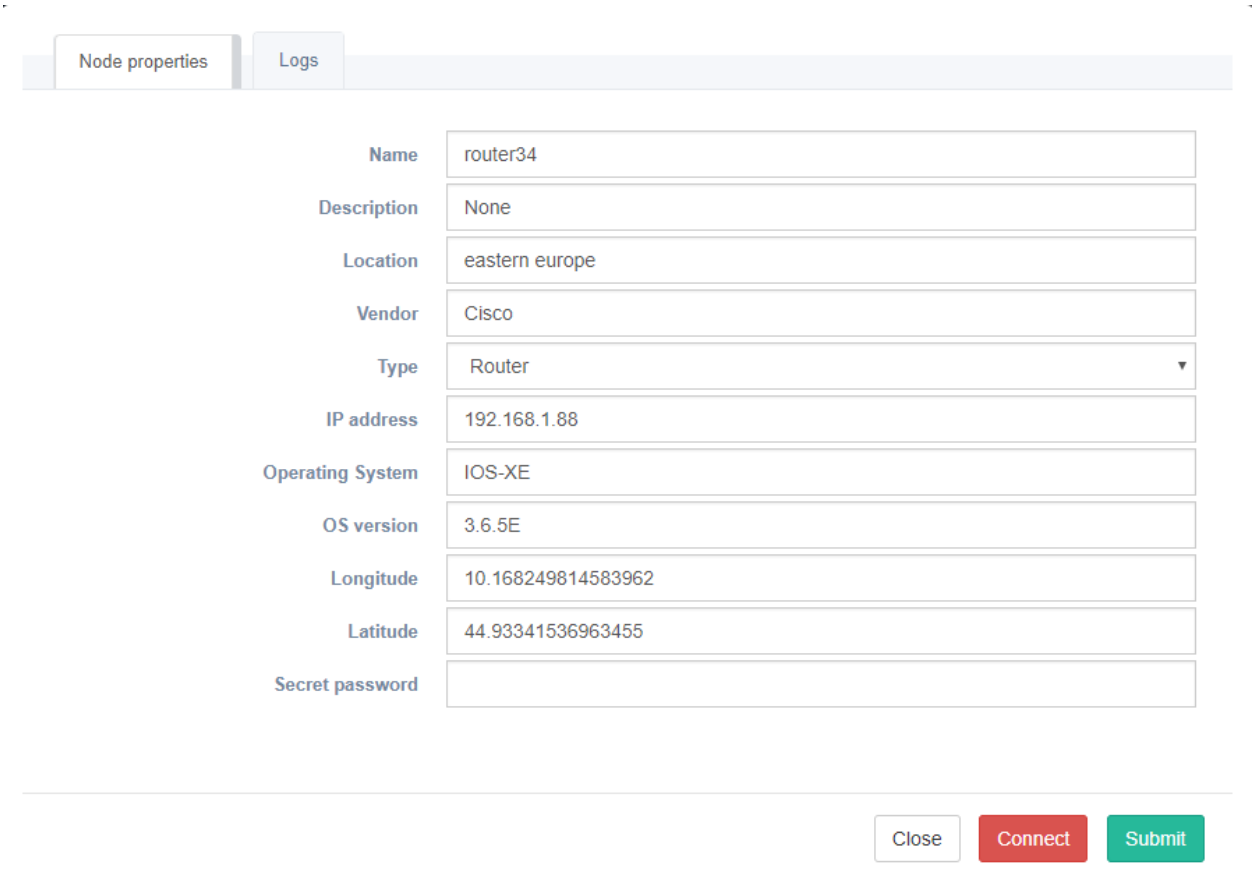


4.1.3 Bindings

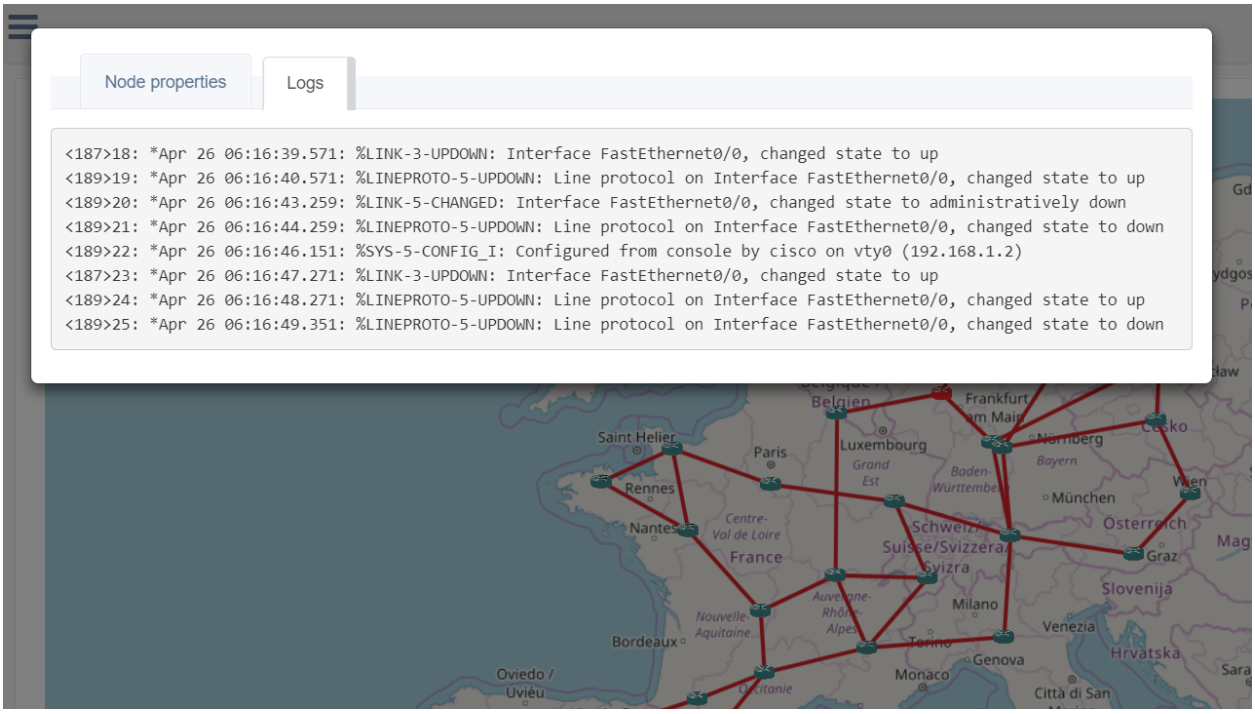
For both the geographical and the logical views, bindings are the following :

- *Left-click on a node*: selection of the node.
- *Double left-click on an object*: open the property panel.

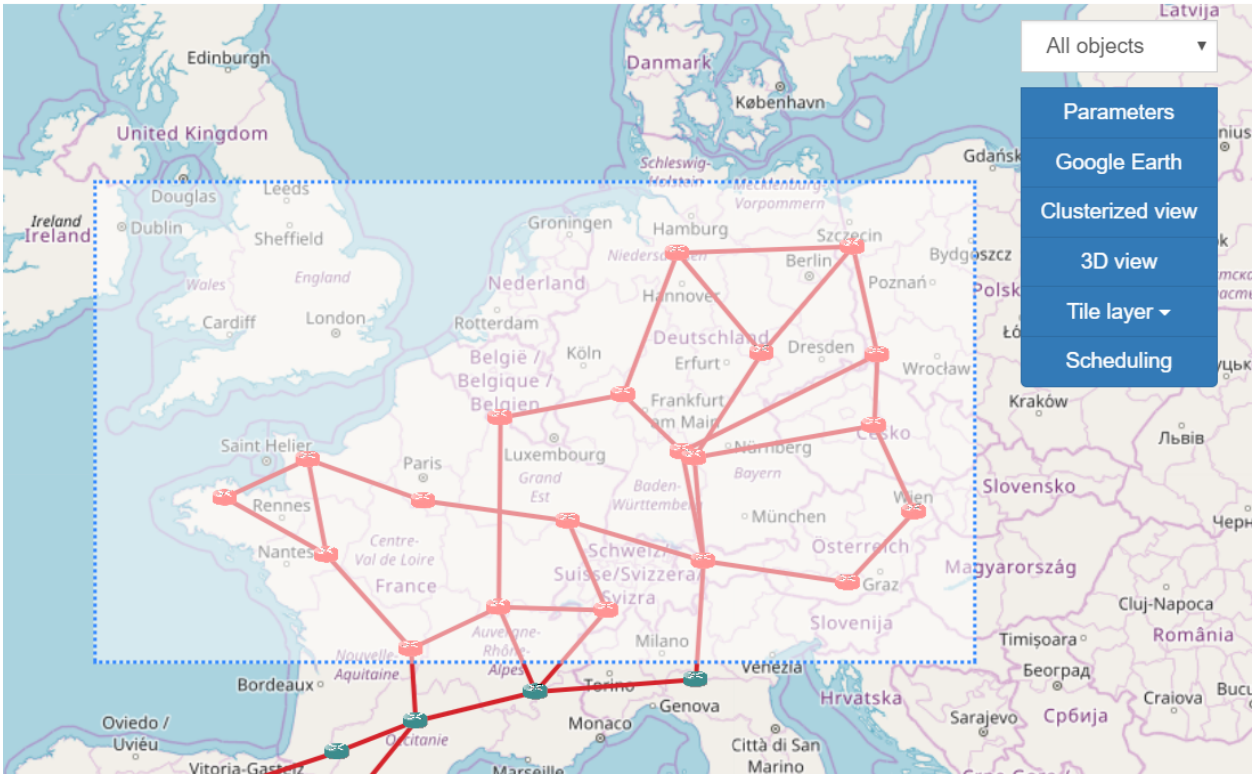
The property panel displays all properties of the object, and any property can be modified. If the object is a node, it also contains a `Connect` button to automatically start an SSH to the device.



If the SYSLOG server has been activated, eNMS stores all logs it receives. The property panel also contains a `Logs` tabs that display all logs sent by this specific node.



- *Shift + left-click*: draw a selection rectangle to select multiple nodes at once.



- *Right-click*: Unselect all nodes.

CHAPTER 5

Automation

5.1 Automation

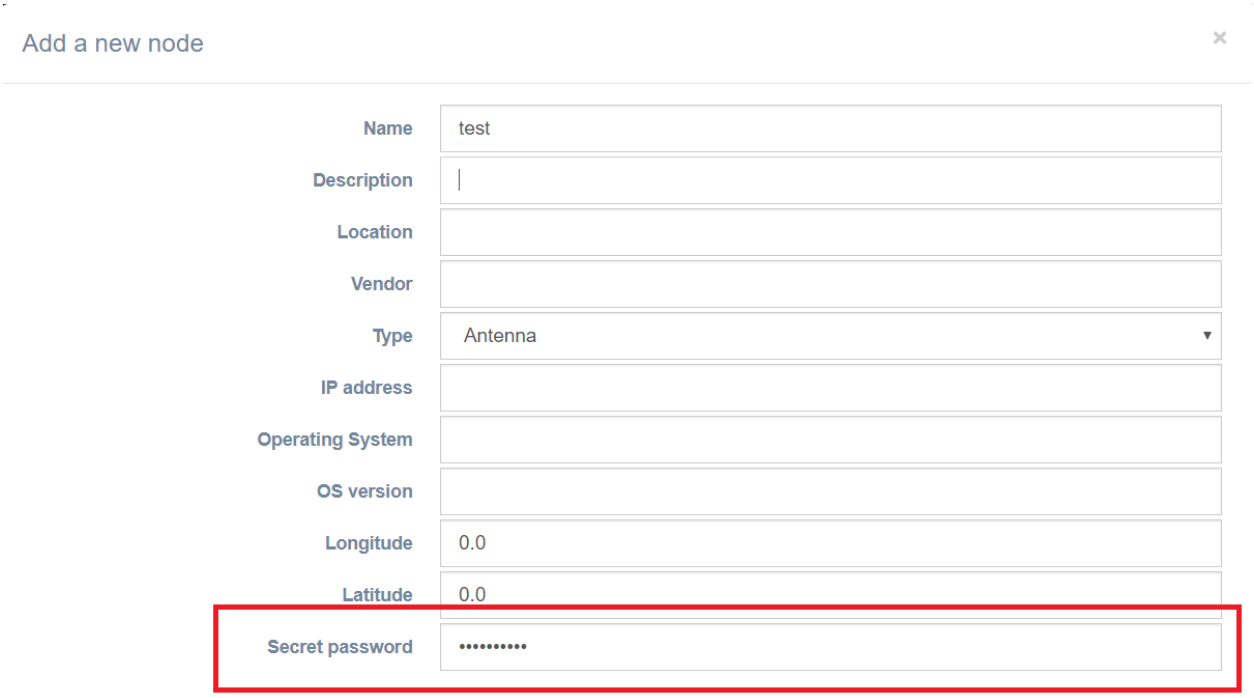
5.1.1 Credentials

Login & username

The login and username used to connect to a device are the ones you use for eNMS. If you create an eNMS account with login `cisco` and password `cisco`, eNMS will use the same credentials to try and connect to the devices upon running a script.

Secret password

Unlike the username and password, the secret password is a property of the device itself. The secret password is set when the node is created (whether manually or from excel import):



5.1.2 Scripts

Scripts are created from the *Scripts* menu. The following types of script are available in eNMS.

Netmiko configuration script

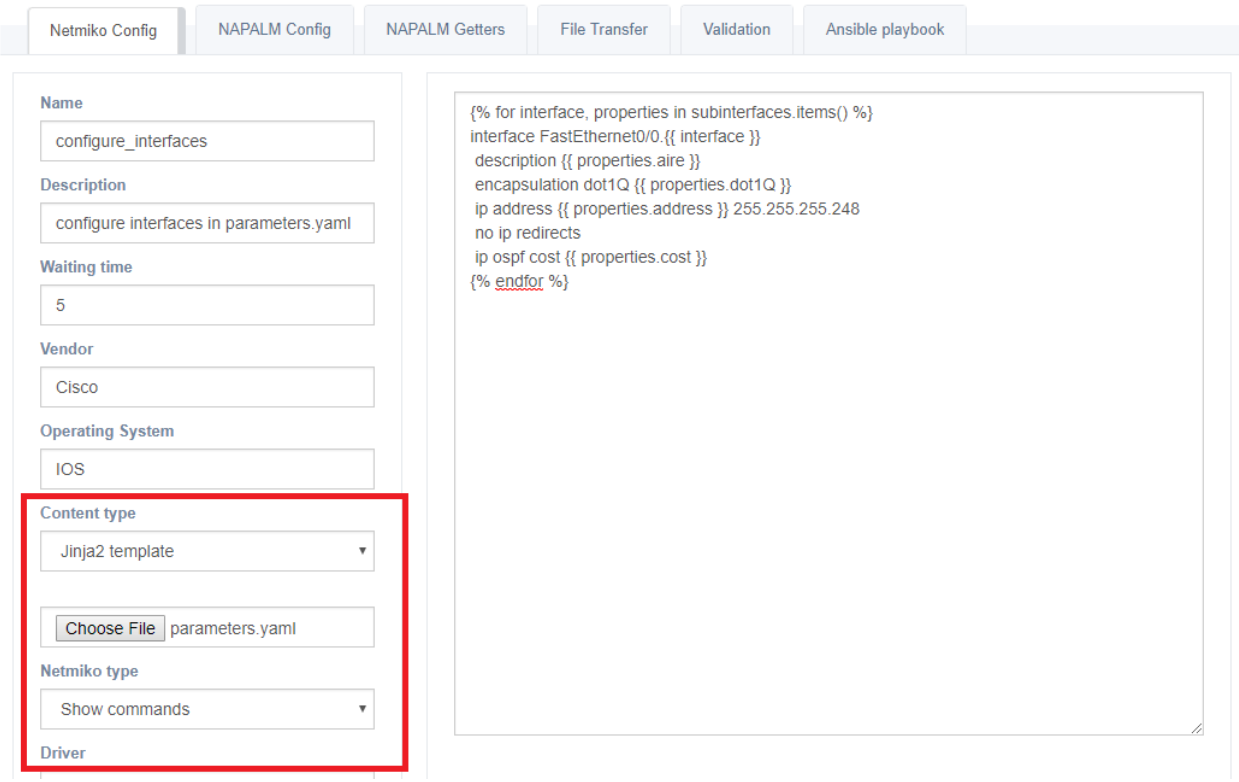
There are two types of Netmiko configuration scripts:

- **show commands**: a list of "show commands" which output will be displayed in the logs.
- **configuration**: a list of commands to be configured on the devices.

For each type, the content of the script can be:

- `text-based`: a list of configuration commands to be sent to the device.
- `template-based`: the script is a Jinja2 template. A .YAML file containing all parameters must be provided.

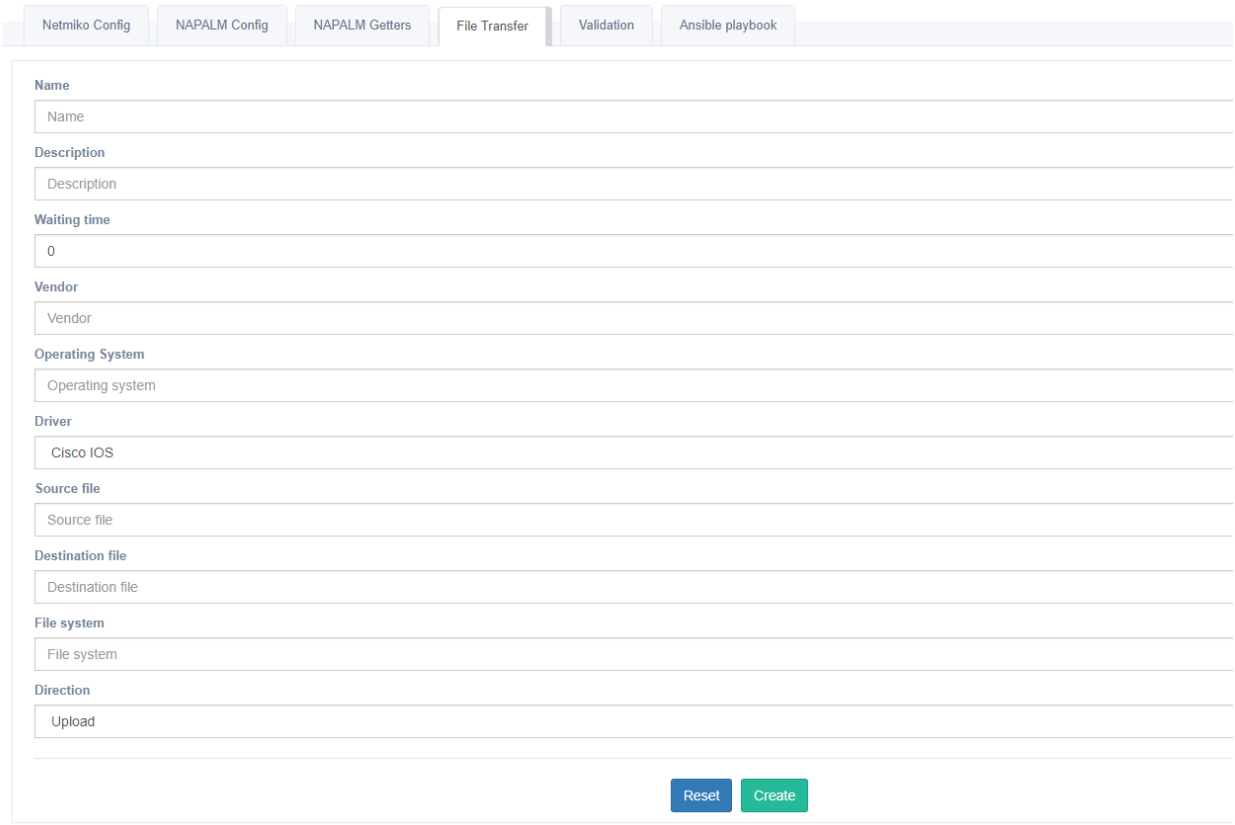
Finally, a **driver** must be selected among all available netmiko drivers.



Netmiko File transfer script

A file transfer script sends a file to a device, or retrieve a file from a device. It relies on Netmiko file transfer functions.

If you want to send a file to a device, you must place the file in the `eNMS/file_transfer` folder.

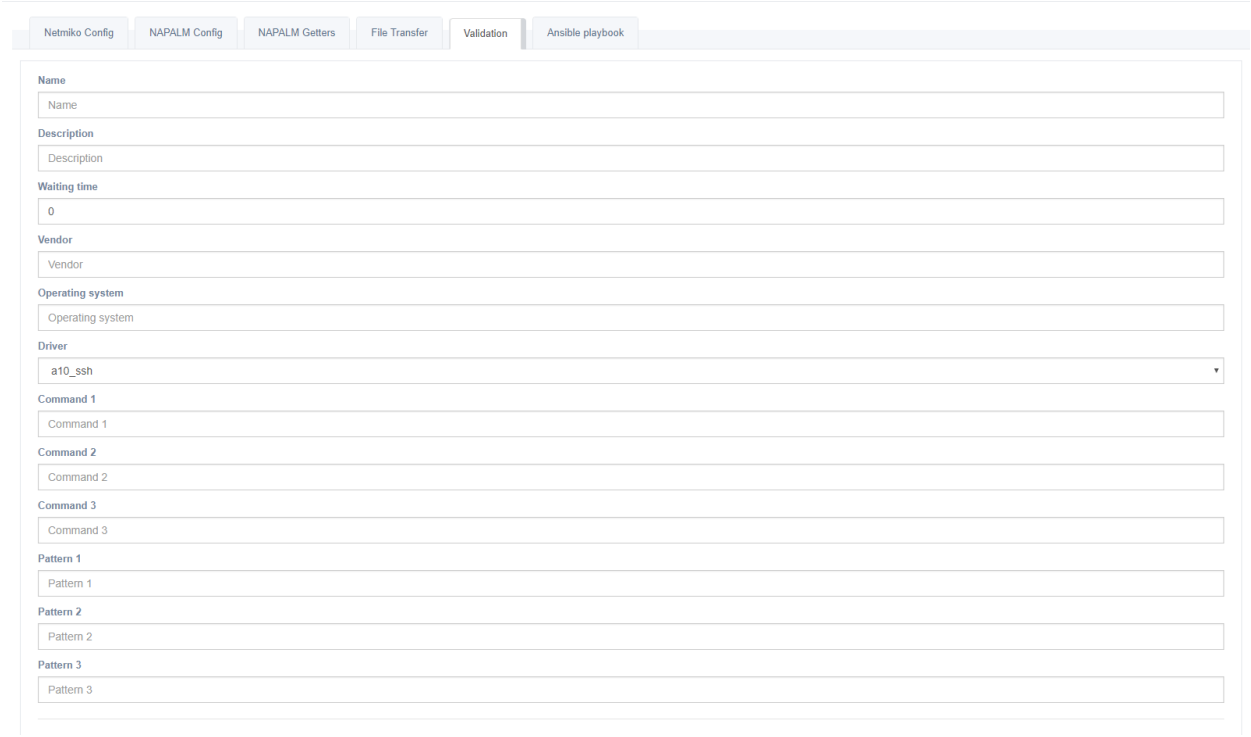


: File-transfer scripts only works for IOS, IOS-XE, IOS-XR, NX-OS and Junos.

Netmiko validation script

A `Netmiko validation` script is used to check the state of a device, in a workflow (see the `Workflow` section for examples about how it is used).

There are 3 `command` field and 3 `pattern` field. For each couple of command/pattern field, eNMS will check if the expected pattern can be found in the output of the command. If the result is positive for all 3 couples, the script will return `True` (allowing the workflow to go forward, following the `success` edges), else it will return `False`.



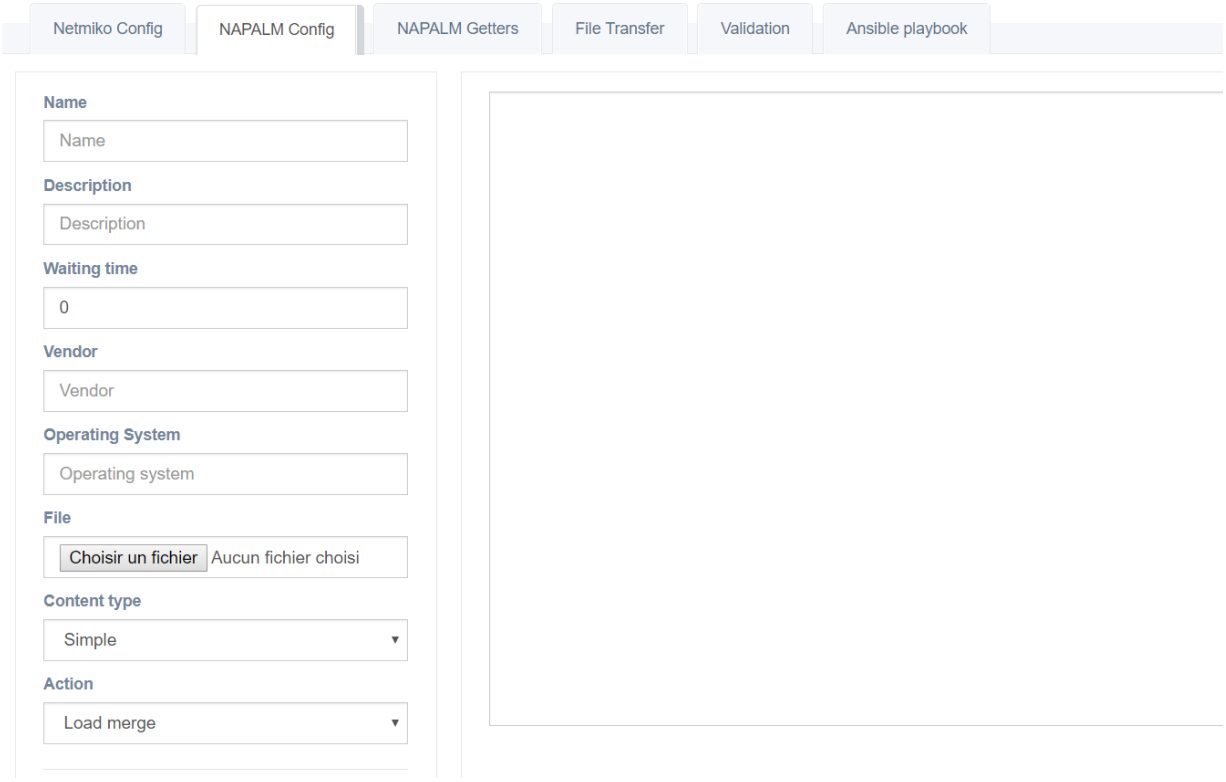
NAPALM configuration script

This type of script uses NAPALM to update the configuration of a device.

There are two types of operations:

- `load merge`: add the script configuration to the existing configuration of the target.
- `load replace`: replace the configuration of the target with the script configuration.

Just like with the `Netmiko configuration` script, a configuration can be either `text-based`, or `template-based`.



: The NAPALM driver used by eNMS is the one you configure in the Operating System property of a node.

The NAPALM drivers name must be respected: `ios, iosxr, nxos, junos, eos`.

: This script does not by itself commit the configuration. To do so, a `NAPALM action` script must be used (see below).

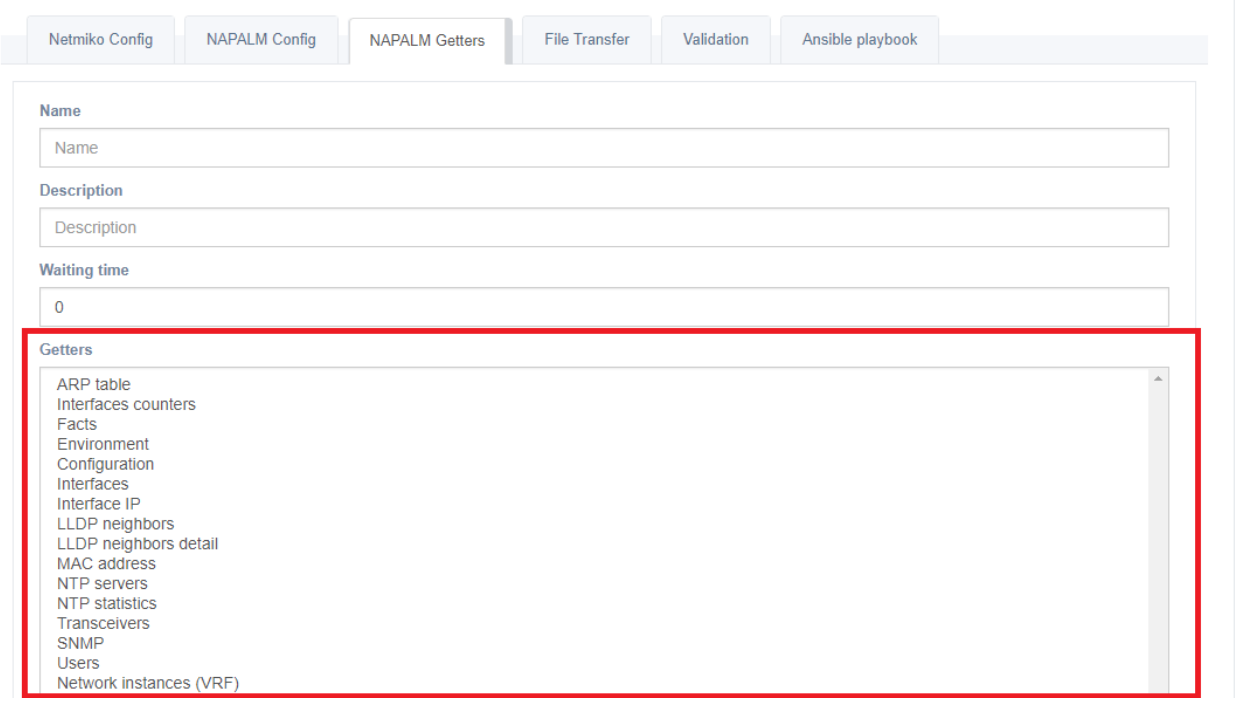
NAPALM action script

`NAPALM action` scripts do not have to be created: they are created by default when eNMS runs for the first time. There are three actions:

- `commit`: commits the changes pushed with `load replace` or `load merge`.
- `discard`: discards the changes before they were committed.
- `rollback`: rollbacks the changes after they have been committed.

NAPALM getters script

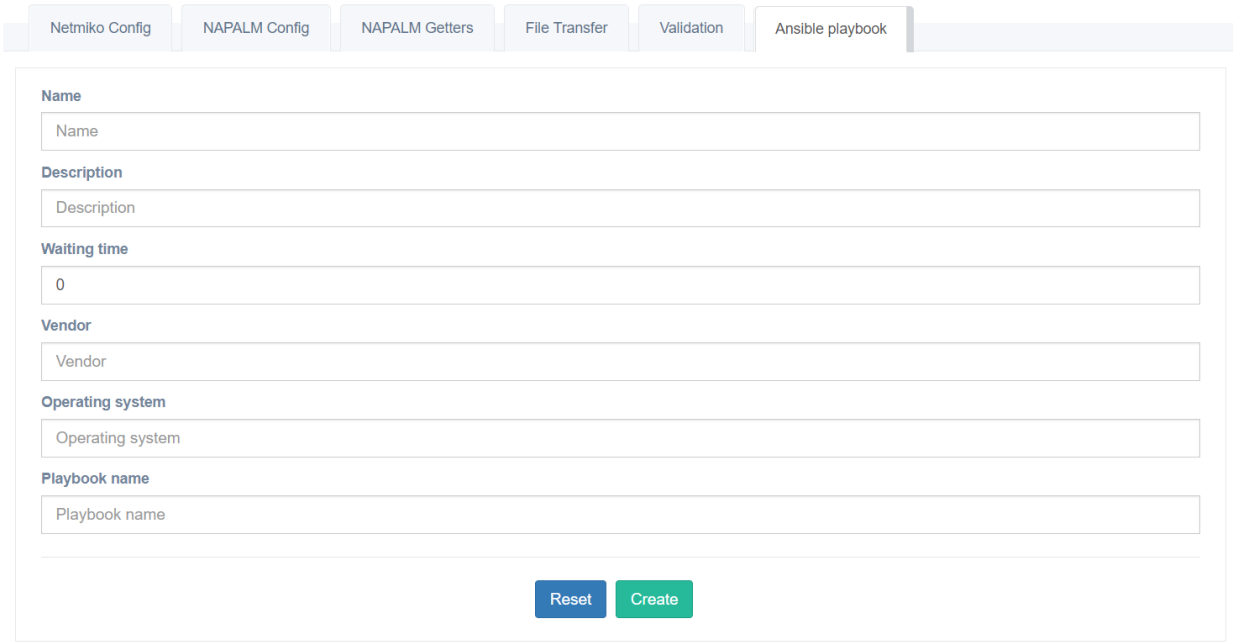
A `NAPALM getters` script is a list of getters which output is displayed in the logs.



: just like with the `NAPALM configuration` scripts, the NAPALM driver used by eNMS is the one configured in the Operating System property of a node. The NAPALM drivers name must be respected: `ios, iosxr, nxos, junos, eos`.

Ansible playbook script

An `Ansible playbook` script sends an ansible playbook to the devices. The playbook file must be placed in the `eNMS/playbooks` folder, along with the Ansible configuration file (`ansible.cfg`). To create an `Ansible playbook` script, simply enter the name of the playbook (example: `the_playbook.yml`) in the `Playbook name` field of the form.



Custom script

eNMS also gives you the option to create your own script. Once created, a custom script is automatically added to the web interface and can be used like any other script. To create a custom script, open the file `eNMS/source/scripts/custom_scripts.py` and use the following template:

- a function that contains the code of the script
- a dictionnary that contains the parameters of your new script, and an key `job_name` which value is the name of the job function.

```
def job_example(args):
    task, node, results = args
    # add your own logic here
    # results is a dictionnary that contains the logs of the script
    results[node.name] = 'what will be displayed in the logs'
    # a script returns a boolean value used in workflows (see the workflow section)
    return True if 'a condition for success' else False

example_parameters = {
    'name': 'script that does nothing',
    'waiting_time': 0,
    'description': 'does nothing',
    'vendor': 'none',
    'operating_system': 'all',
    'job_name': 'job_example'
}
```

You must also the update the `create_custom_scripts` function at the bottom of the file:

```
def create_custom_scripts():
    for parameters in (
```

(continues on next page)

()

```
        example_parameters,
        the_parameters_of_the_script_you_created
    ):
        try:
            custom_script = CustomScript(**parameters)
            db.session.add(custom_script)
            db.session.commit()
        except exc.IntegrityError:
            db.session.rollback()
```

Finally, restart the application.

You can take a look at the other scripts for inspiration (in `eNMS/source/scripts/models.py`). `custom_scripts.py` also contains a script called `NornirPingScript` that shows how to use the Nornir automation framework to ping a device on ports 23 and 443.

5.1.3 Workflows

A workflow is a directed graph which nodes are scripts.

Each script in eNMS returns a boolean value: - `True` if it ran successfully. - `False` otherwise.

There are two types of edge in a workflow: `success edge` and `failure edge`. The `success edge` indicates where to move in the graph if the source script was executed with success, while the `failure edge` does the same thing in case of failure.

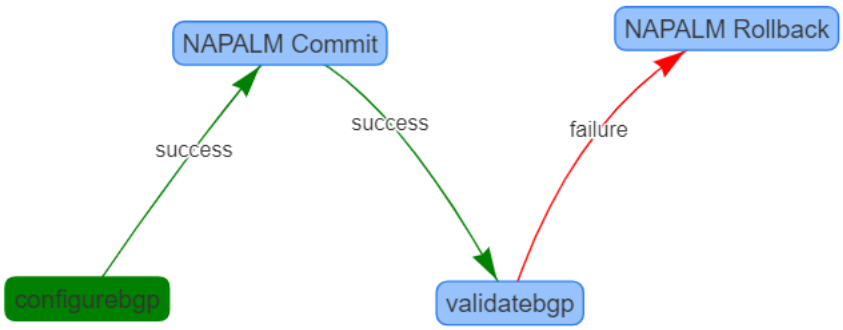
Workflows are created and managed from the *workflows/workflow_management* page.

A first example

Let's consider a workflow made of four scripts:

- `configure-bgp`, a `NAPALM configuration` script that configures a BGP neighbor.
- `NAPALM Commit`, that commits the changes with NAPALM.
- `validate-bgp`, a `Netmiko validation` script that checks that the neighbor appears in the `show ip bgp neighbors` command.
- `NAPALM Rollback`, that rollbacks the changes with NAPALM.

The workflow below uses these four scripts together to configure a new BGP neighbor and rollbacks in case of problem.



The green color of `configure-bgp` indicates that this is the beginning of the workflow (the first script to be executed).

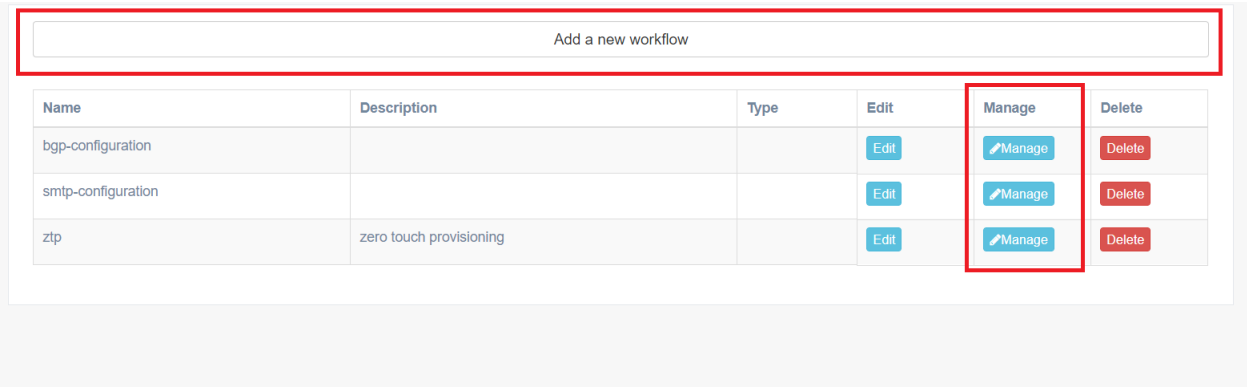
If `configure-bgp` is a success (it returns the boolean value `True`, the `success edge` will be used, and the `NAPALM Commit` script will be executed.

If `NAPALM Commit` runs successfully, `validate-bgp` will run and check that the neighbor was properly configured.

If `validate-bgp` is a success, the workflow will stop here as there is no `success edge` starting from `validate-bgp`. On the other hand, if it fails, the workflow will go on using the `failure edge` and the configuration will be rolled back with NAPALM.

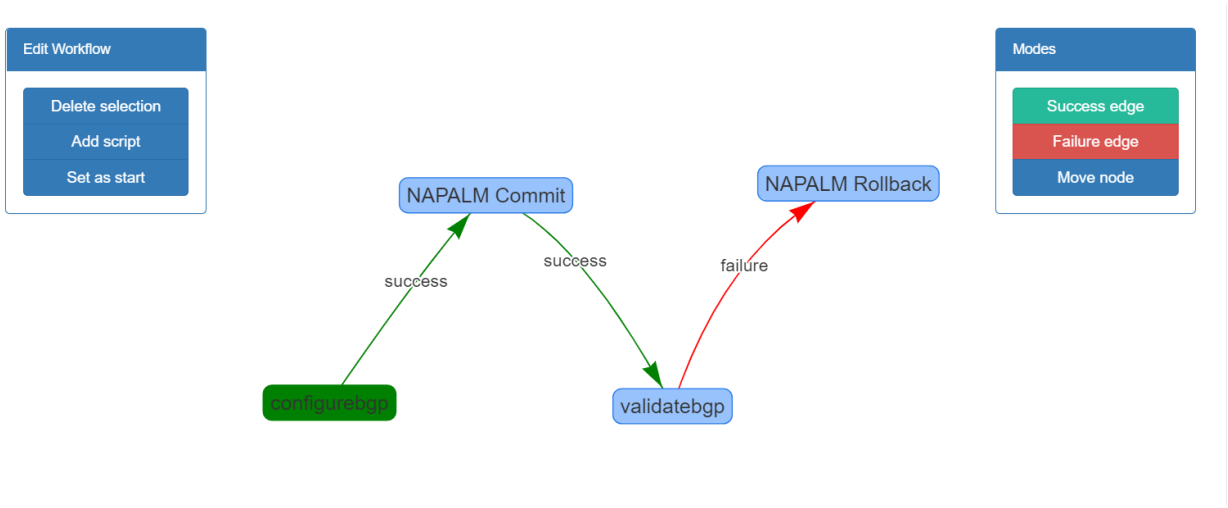
Creation of a workflow

In the *workflows/workflow_management* page, click on the button `Add a new workflow` and fill the workflow creation form. The new workflow will be automatically added to the table of worflows.



Clicking on the `Manage` button in the table of workflows to open the `Workflow builder`.

Workflow builder



- *Add script*: open a window to select which script you want to add to the workflow.
- *Delete selection*: delete the selected script or edge.
- *Set as start*: the selected script is set as the beginning of the workflow. It will be highlighted in green.
- *Success edge*: switch to the `success edge` creation mode, allowing you to draw `success edge` between scripts.
- *Failure edge*: same as `success edge`.
- *Move node*: switch to the motion node, allowing you to drag the scripts on the canvas to better visualize the workflow.

: You can double-click on a script to update its properties.

Create a workflow step by step

Let's create the BGP workflow discussed in the first paragraph.

Creation of the validatebgp script

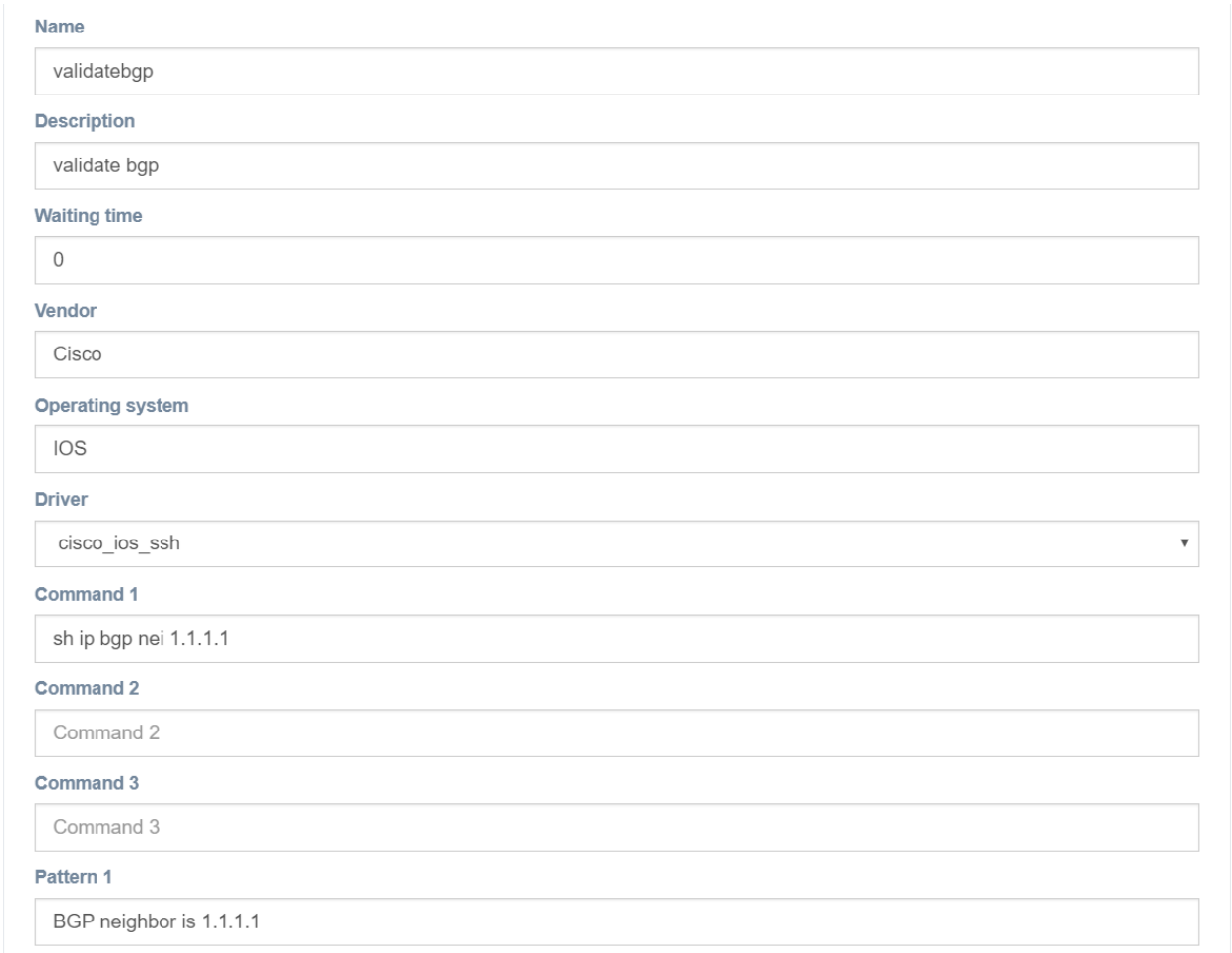
In the *scripts/script_creation* page, we create a `NAPALM configuration` script to configure the BGP neighbor on the device.

Configuration:

Creation of the validatebgp script

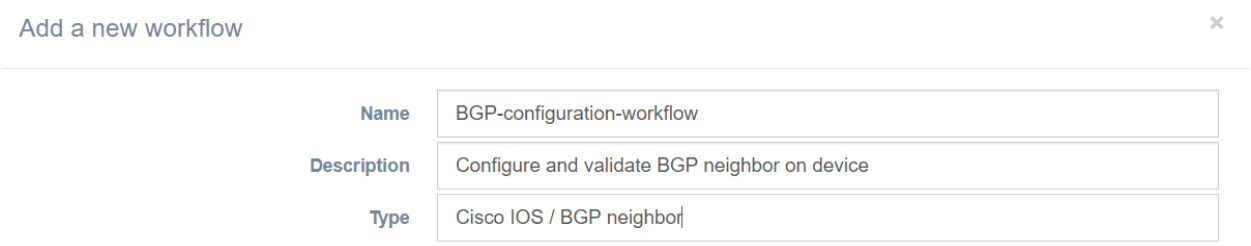
In the *scripts/script_creation* page, we create a `Netmiko validation` script to check that 1.1.1.1 is indeed considered a BGP neighbor on the device.

Specifically, we are checking that the output of `show ip bgp neighbors 1.1.1.1` contains the line `BGP neighbor is 1.1.1.1`.



Creation of the workflow

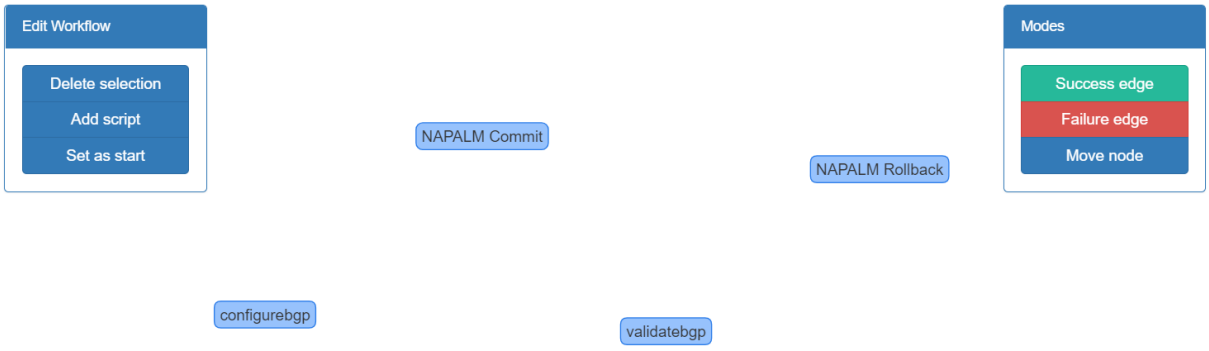
In the *workflows/workflow_management* page, click on the button `Add a new workflow` and fill the workflow creation form.



Building the workflow

In the *workflows/workflow_management* page, click on the button `Manage` of the newly created workflow. This opens the `Workflow builder`.

Click on the `Add script` button, and add all 4 scripts: `configurebgp`, `validatebgp`, `NAPALM Commit` and `NAPALM Rollback`.

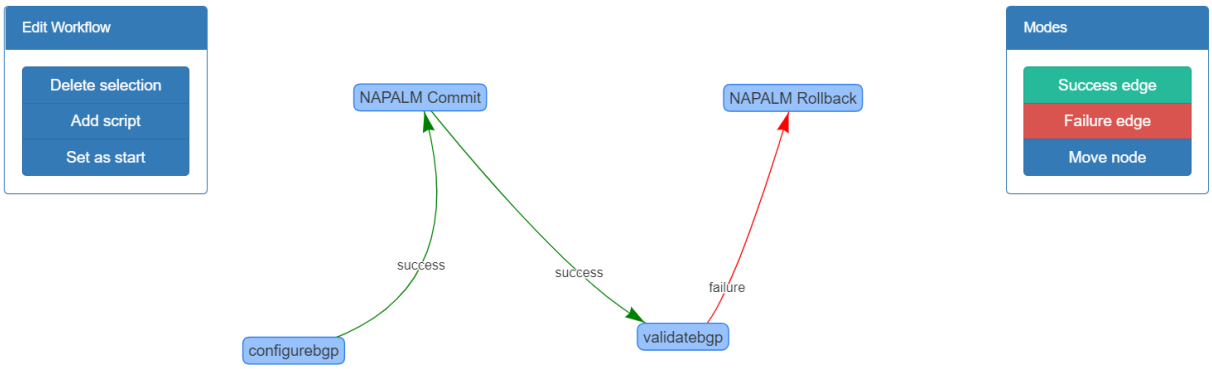


Finally, create:

- a `success edge` from `configurebgp` to `NAPALM Commit`.
- a `success edge` from `NAPALM Commit` to `validatebgp`.
- a `failure edge` from `validatebgp` to `NAPALM Rollback`.

Select `configurebgp` and click on the `Set as start` button to tell eNMS that this is where the workflow begins.

The workflow is done and ready to be executed:



5.1.4 OS upgrade

We consider a Cisco router with IOS `12.4(13r)T` and the IOS image `c1841-spservicesk9-mz.124-8.bin`. Let's create a workflow to upgrade to `c1841-adventerprisek9-mz.124-8a.bin`.

Creation of the scripts

The workflow will be composed of the following scripts:

1. Version check

We use a `Netmiko validation` script to make sure that the current IOS image used by the current is the one we want to update. In other words, we check that:

- The output of `show version` contains `System image file is "flash:c1841-spservicesk9-mz.124-8.bin"`.
- The output of `dir` contains `c1841-spservicesk9-mz.124-8.bin`.

If either of this condition fails, the `Netmiko validation` script will fail, and the workflow will stop.

We create the following script `version-check-before-reload` from the *script/script_creation* page.

2. Transferring the new IOS image

In order to transfer the new IOS image, we will use a `Netmiko File Transfer` script. We place the `c1841-adventerprisek9-mz.124-8a.bin` in the *eNMS/file_transfer* folder, and we create the file transfer script:

Name

transfer-new-image

Description

Transfer the new IOS image on the router

Waiting time

0

Vendor

Cisco

Operating System

IOS

Driver

Cisco IOS

Source file

c1841-adventerprisek9-mz.124-8a.bin

Destination file

c1841-adventerprisek9-mz.124-8a.bin

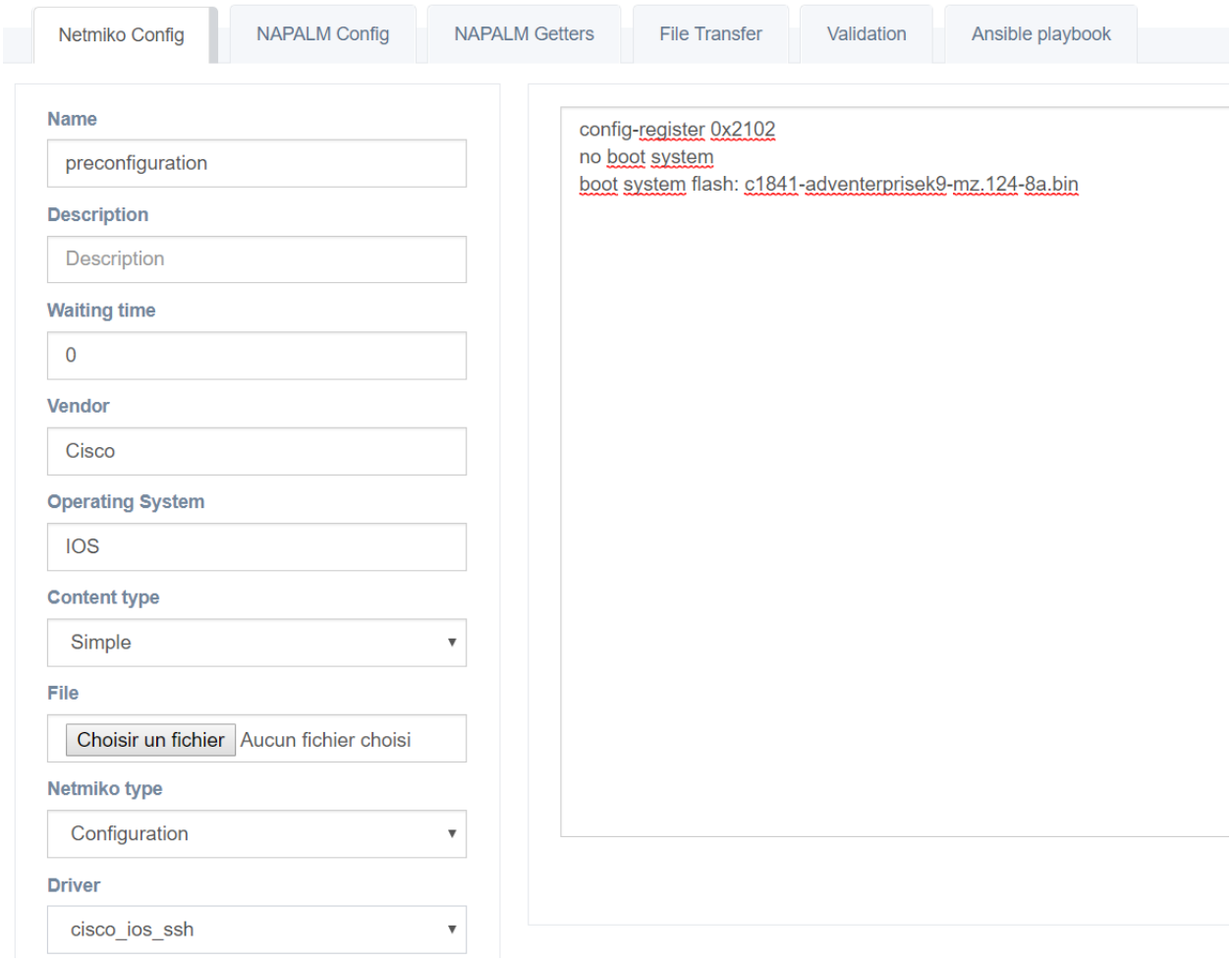
File system

flash:

3. Preconfigure the router for the upgrade

We need to upgrade the configuration register to `0x2102`, and tell the router to boot from the IOS image that we've uploaded in the last step.

We create a `Netmiko configuration` script of type `configuration` with the following commands:

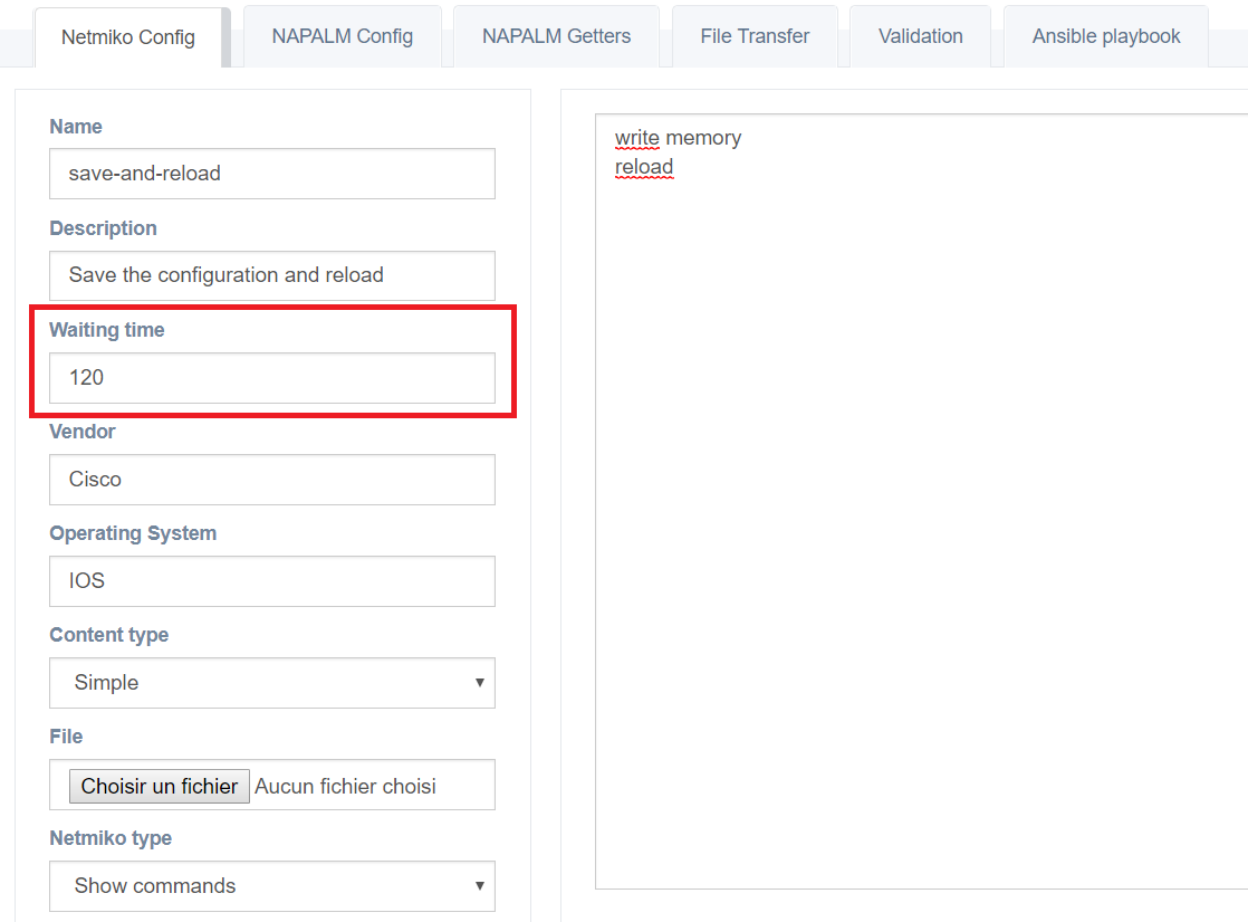


4. Save and reload

We use a `Netmiko configuration` of type `"show" commands` to save the latest changes and reload the device.

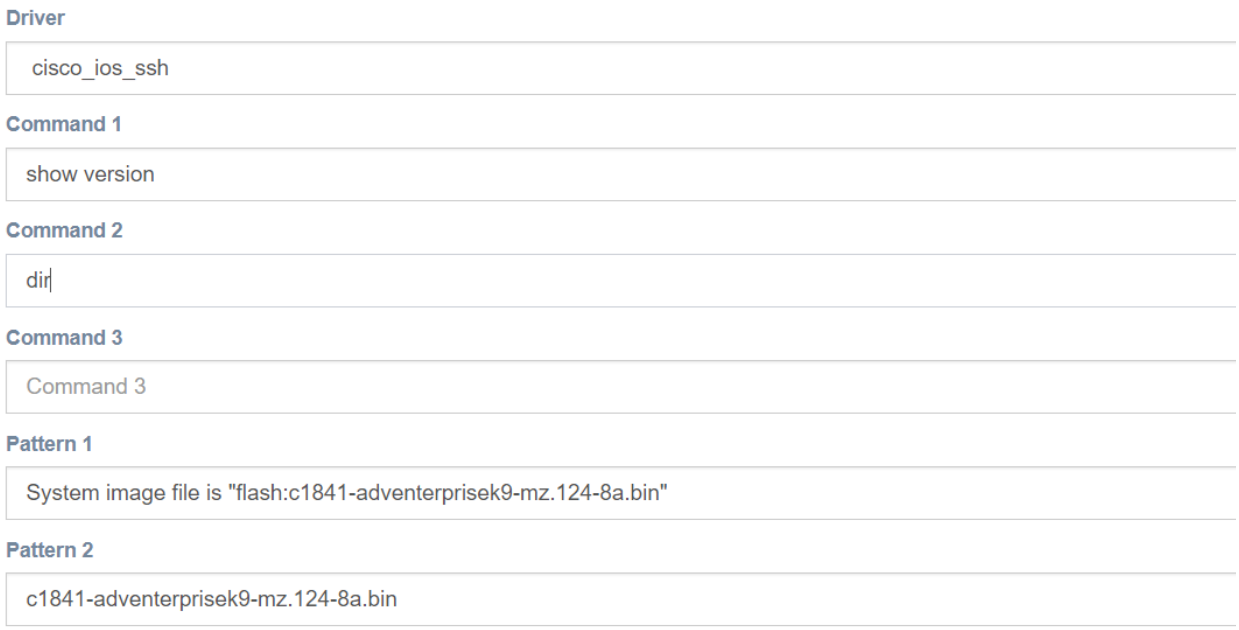
Each script has a `Waiting time` parameter (seconds) that tells eNMS how much time it must wait before proceeding to the next script in the workflow.

After sending the script, we have to wait a bit for the device to reload and be available again: we set the `waiting time` of the script to `120` (2 minutes).



5. Post-reload version check

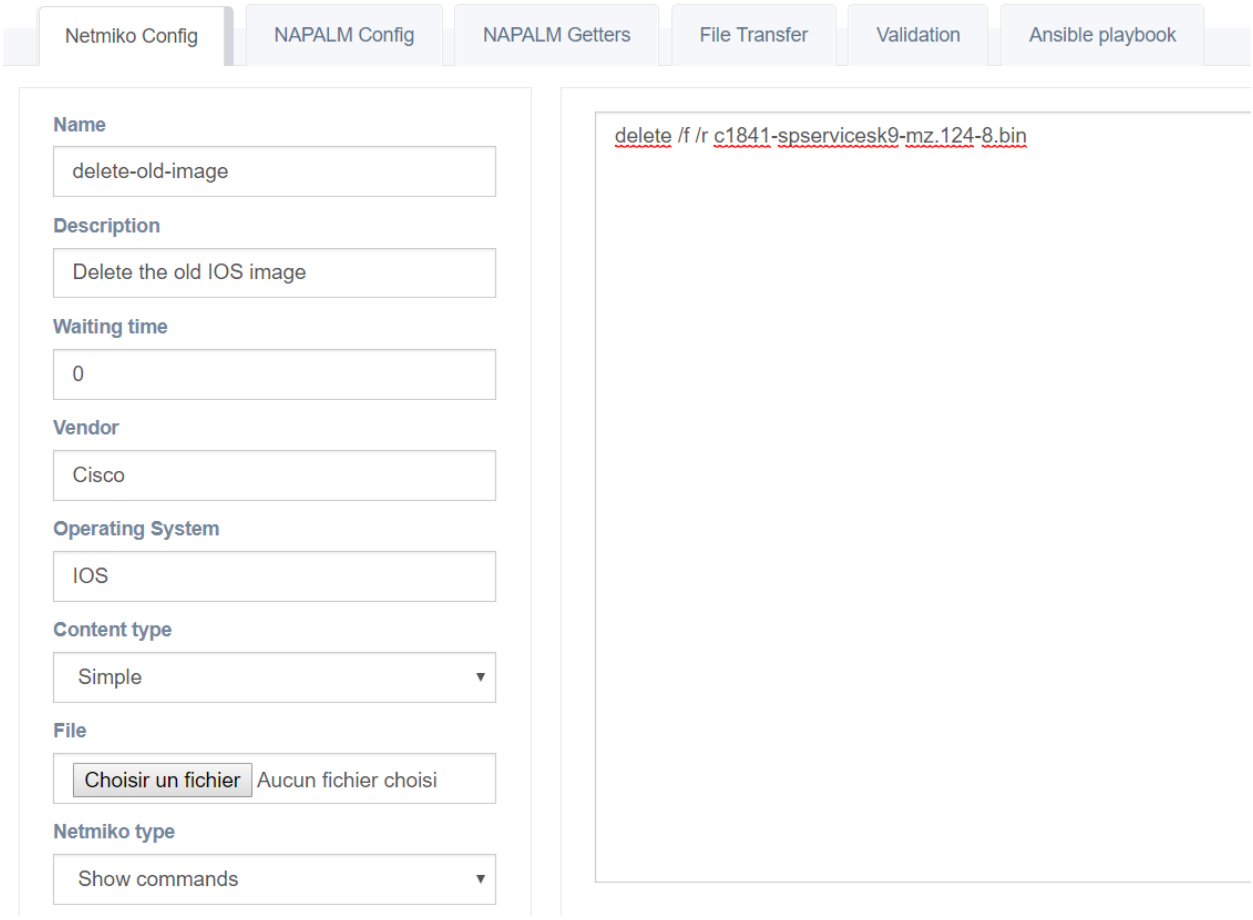
We create a `Netmiko validation` script similar to the one used in the first step to check that the IOS image used by the router is indeed the new one.



6. Delete the old IOS image

Since we have uploaded a new IOS image and moved it to the flash memory, we need some space. We erase the old IOS image from the memory: `delete /f /r c1841-spservicesk9-mz.124-8.bin`.

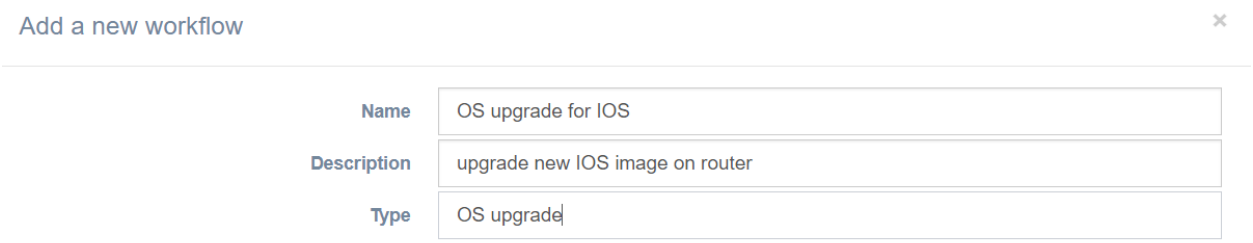
We do not need to enter the configuration mode to type this command and delete the file, so we create a `Netmiko configuration` script of type `"show" commands` (and not `configuration`).



Creation of the workflow

1. Creation form

In the *workflows/workflow_management* page, click on the button `Add a new workflow` and fill the workflow creation form.



2. Building the workflow

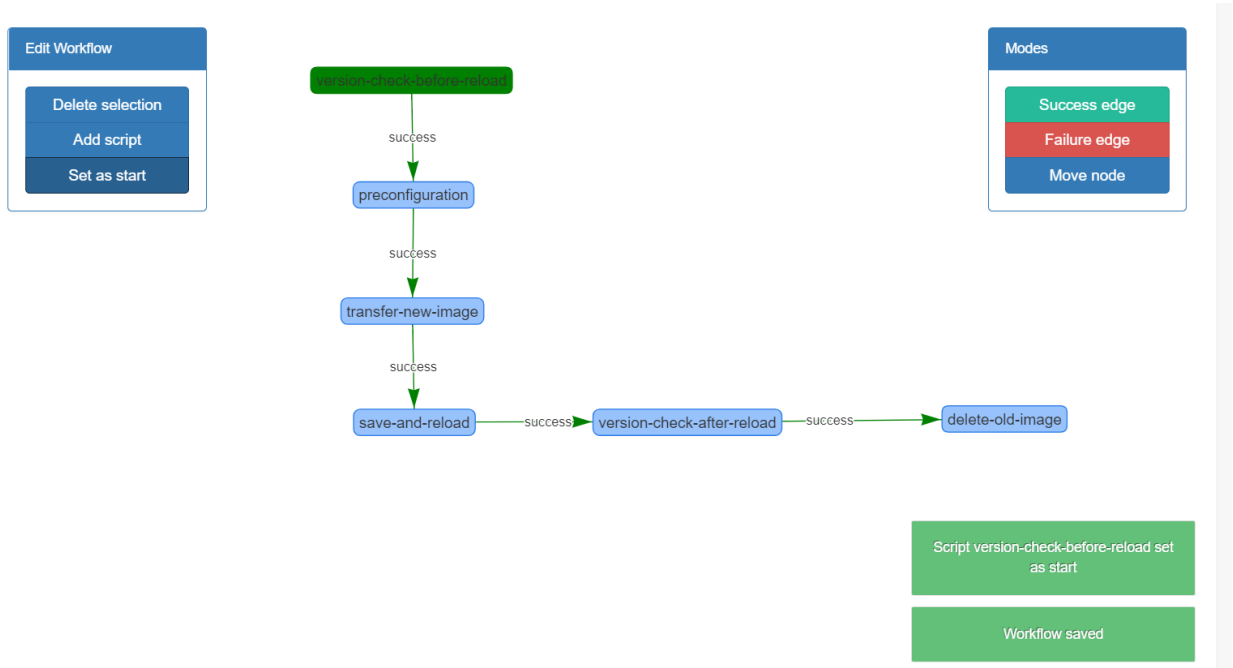
In the *workflows/workflow_management* page, click on the button `Manage` of the newly created workflow. This opens the `Workflow builder`.

Click on the `Add script` button, and add all 6 scripts:

- `version-check-before-reload`
- `preconfiguration`
- `transfer-new-image`
- `save-and-reload`
- `version-check-after-reload`
- `delete-old-image`

Between each consecutive pair of scripts, we create a success edge, and we set `version-check-before-reload` as the beginning of the workflow.

The workflow is done and ready to be executed:



5.1.5 ReST API

eNMS has a ReST API allowing to:

- retrieve an object.
- execute a task.

Retrieve an object

```
# via a GET call to the following URL
http://IP_address/rest/get/object_type/object_name
```

`object_type` can be any of the following: `node`, `link`, `user`, `script`, `workflow`, `task`.

Execute a task

```
# via a GET call to the following URL
http://IP_address/rest/execute_task/task_name
```

The task will start immediately (and its properties are displayed).


127.0.0.1:5000/rest/execute_task/task-napalm-rollback
{
"result": {
"id": "1",
"name": "task-napalm-rollback",
"result": "None",
"status": "active",
"start_date": "2018-06-21 18:05:17",
"end_date": "None",
"frequency": "",
"creator": "cisco",
"scripts": [
{
"id": "1",
"name": "NAPALM Rollback",
"type": "napalm_action",
"description": "",
"vendor": "None",
"operating_system": "None"
}
],
"nodes": [
{
"id": "4",
"name": "router8",
"description": "eastern europe",
"model": "None",
"location": "eastern europe",
"type": "Router",
"vendor": "Cisco",
"operating_system": "IOS",

CHAPTER 6

Tasks

6.1 Tasks

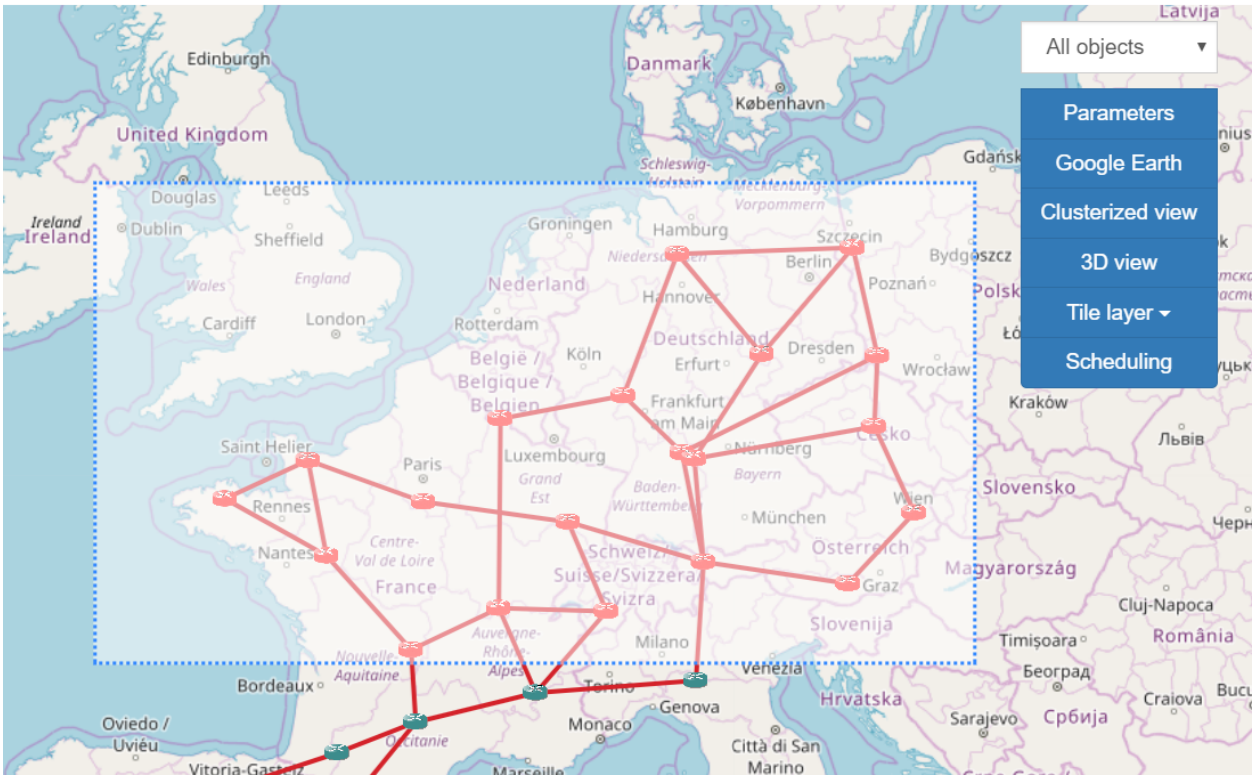
6.1.1 Scheduling

Scheduling a script or a workflow is done from the graphical view of the network, in two steps:

1. Selection of the target devices
2. Scheduling the script

Target selection

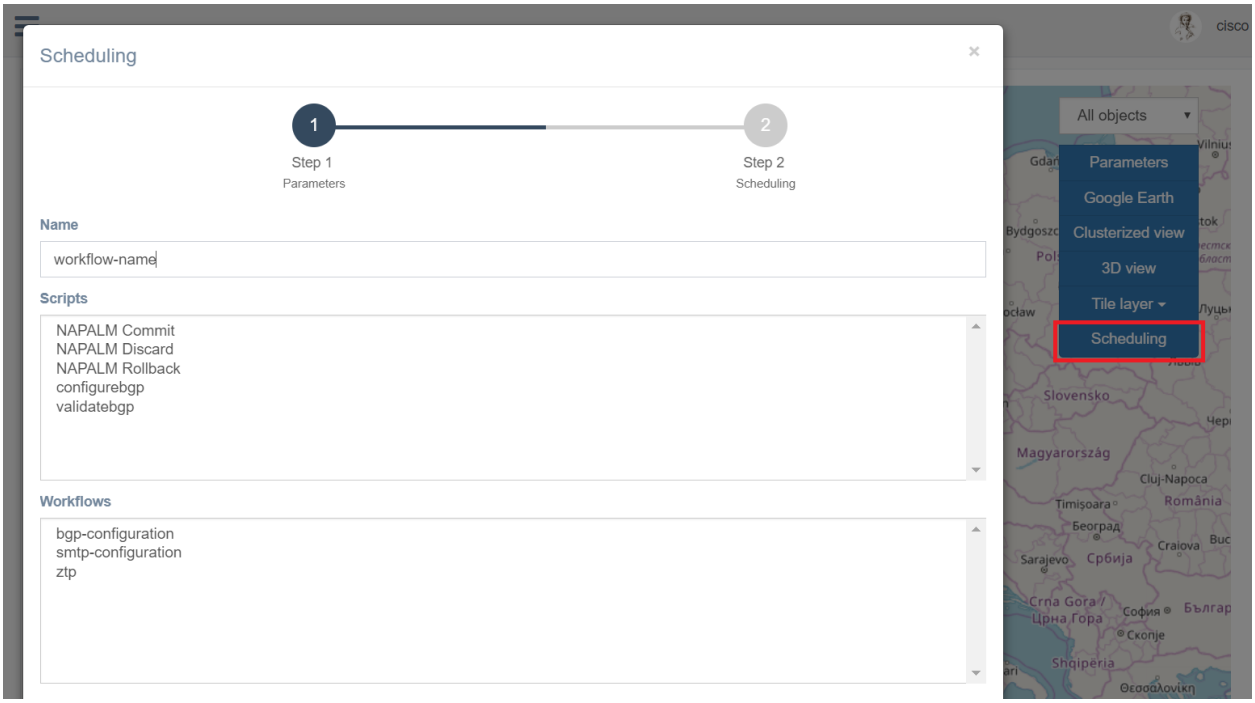
You can `left-click` on a device to select it, or use `shift + left-click` to draw a selection rectangle and select multiple devices at once. All selected devices are highlighted in red. A `right-click` will automatically unselect all devices.



Refer to the *views/bindings* section of the docs for more information.

Script & Workflow selection

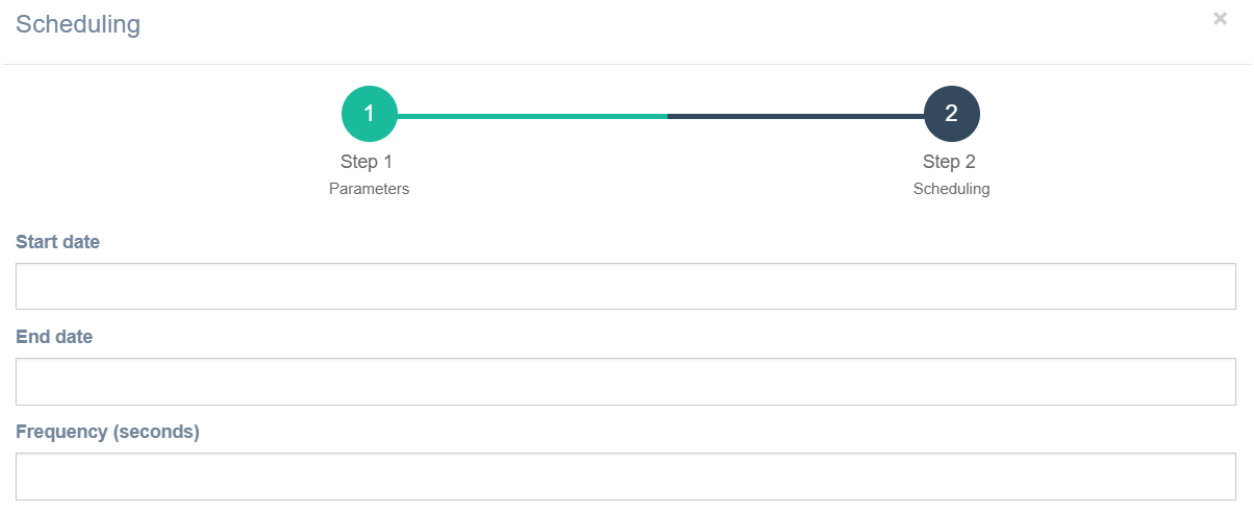
The geographical and logical views have an `Scheduling` button. After the target devices have been selected, click on this button to open the scheduling panel. Enter the name of the task, and select all the scripts and workflows to run.



Scheduling

A task can be scheduled to run at a specific time, once or periodically.

For a periodic task, set the frequency in seconds in the `Frequency` field. The task will run indefinitely, until it is stopped or deleted from the task management page (*tasks/task_management*). Optionally, an `End date` can be scheduled for the script to stop running automatically.

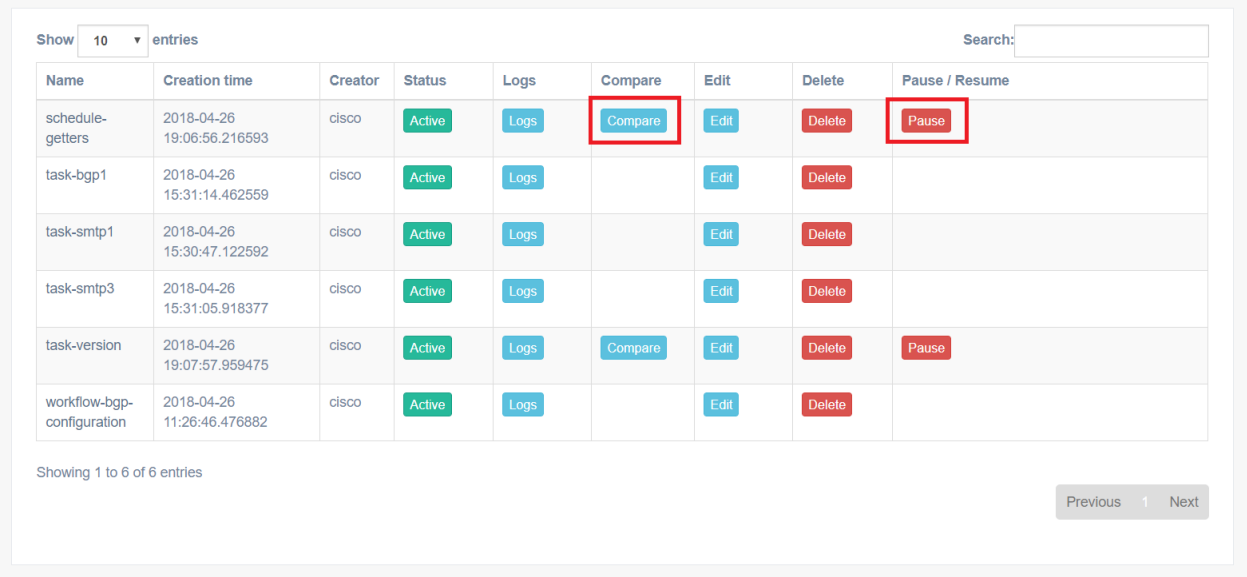


: If the `Start date` field is left empty, the script will run immediately.

6.1.2 Management

Task overview

In the *tasks/task_management* page, you can find a summary of all existing tasks.

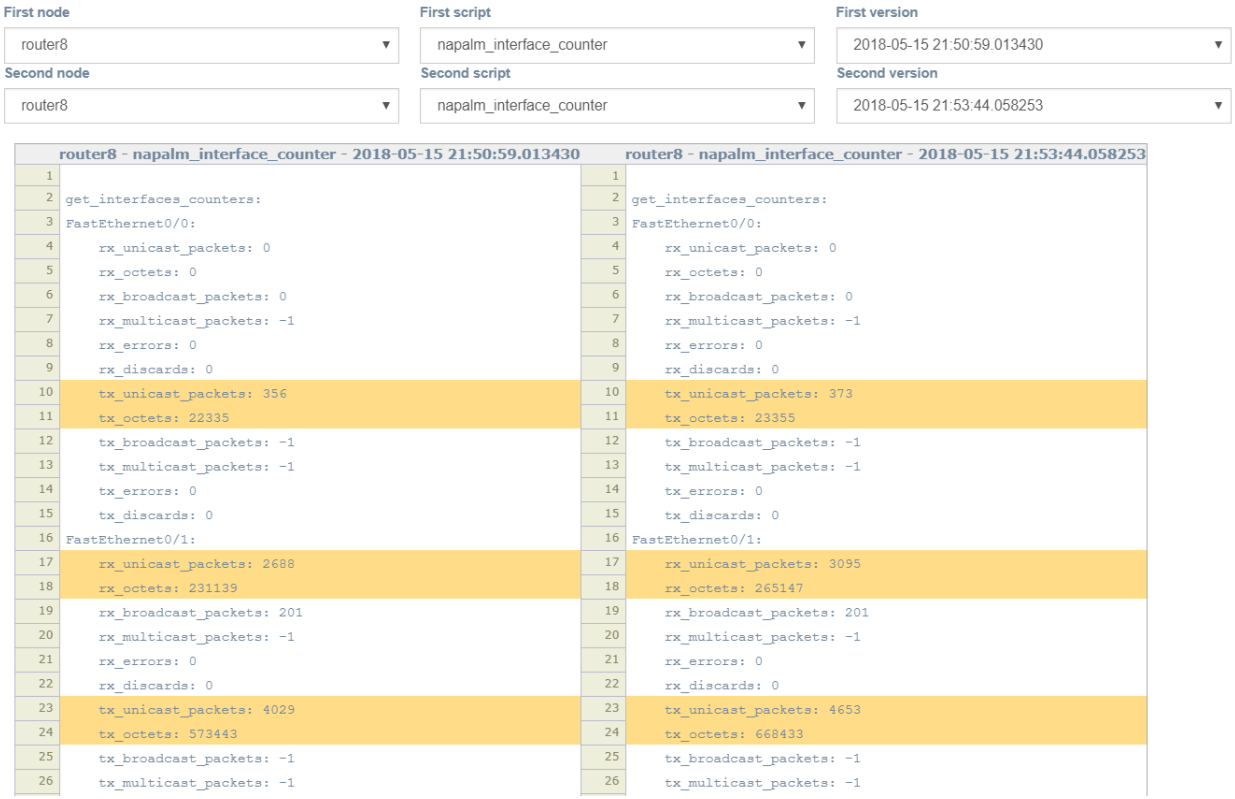


From this table, you can:

- view the logs of the task.
- edit the task's properties, including the scheduling properties (dates and frequency).
- delete the task.

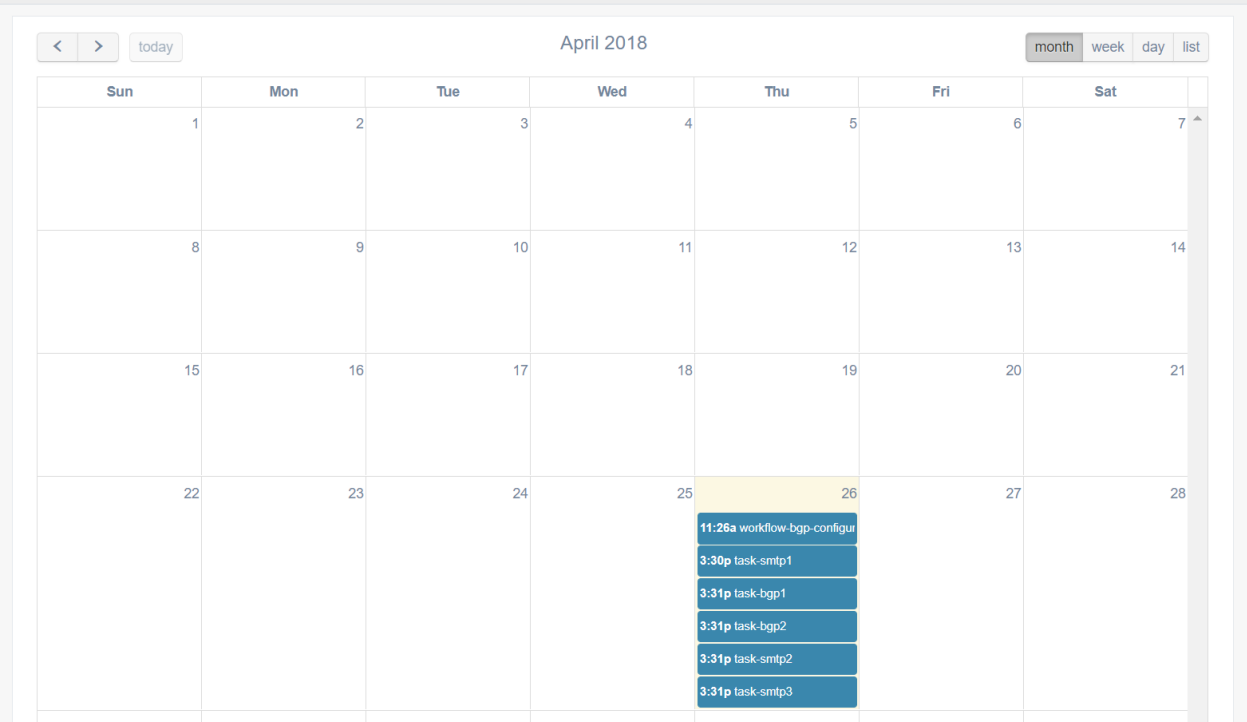
There are two additional options for periodic tasks (tasks that run periodically at a user-defined frequency):

- pause or resume the task
- compare the logs, between:
  - two different devices.
  - two different scripts or workflows.
  - two different times.



Calendar

The *tasks/calendar* page provides an overview of all tasks, organized in calendar. Clicking on a task opens a panel to edit its properties.


today
April 2018
month
week
day
list
Sun
Mon
Tue
Wed
Thu
Fri
Sat
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
11:26a workflow-bgp-configur
3:30p task-smtp1
3:31p task-bgp1
3:31p task-bgp2
3:31p task-smtp2
3:31p task-smtp3

CHAPTER 7

License

7.1 License

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 3, 29 June 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The GNU General Public License is a free, copyleft license for
software and other kinds of works.

  The licenses for most software and other practical works are designed
to take away your freedom to share and change the works.  By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users.  We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors.  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

  To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights.  Therefore, you have
certain responsibilities if you distribute copies of the software, or if
```

(continues on next page)

()

```
you modify it: responsibilities to respect the freedom of others.

  For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received.  You must make sure that they, too, receive
or can get the source code.  And you must show them these terms so they
know their rights.

  Developers that use the GNU GPL protect your rights with two steps:
(1) assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.

  For the developers' and authors' protection, the GPL clearly explains
that there is no warranty for this free software.  For both users' and
authors' sake, the GPL requires that modified versions be marked as
changed, so that their problems will not be attributed erroneously to
authors of previous versions.

  Some devices are designed to deny users access to install or run
modified versions of the software inside them, although the manufacturer
can do so.  This is fundamentally incompatible with the aim of
protecting users' freedom to change the software.  The systematic
pattern of such abuse occurs in the area of products for individuals to
use, which is precisely where it is most unacceptable.  Therefore, we
have designed this version of the GPL to prohibit the practice for those
products.  If such problems arise substantially in other domains, we
stand ready to extend this provision to those domains in future versions
of the GPL, as needed to protect the freedom of users.

  Finally, every program is threatened constantly by software patents.
States should not allow patents to restrict development and use of
software on general-purpose computers, but in those that do, we wish to
avoid the special danger that patents applied to a free program could
make it effectively proprietary.  To prevent this, the GPL assures that
patents cannot be used to render the program non-free.

  The precise terms and conditions for copying, distribution and
modification follow.

                       TERMS AND CONDITIONS

  0. Definitions.

  "This License" refers to version 3 of the GNU General Public License.

  "Copyright" also means copyright-like laws that apply to other kinds of
works, such as semiconductor masks.

  "The Program" refers to any copyrightable work licensed under this
License.  Each licensee is addressed as "you".  "Licensees" and
"recipients" may be individuals or organizations.

  To "modify" a work means to copy from or adapt all or part of the work
in a fashion requiring copyright permission, other than the making of an
exact copy.  The resulting work is called a "modified version" of the
earlier work or a work "based on" the earlier work.
```

(continues on next page)

()

```
  A "covered work" means either the unmodified Program or a work based
on the Program.

  To "propagate" a work means to do anything with it that, without
permission, would make you directly or secondarily liable for
infringement under applicable copyright law, except executing it on a
computer or modifying a private copy.  Propagation includes copying,
distribution (with or without modification), making available to the
public, and in some countries other activities as well.

  To "convey" a work means any kind of propagation that enables other
parties to make or receive copies.  Mere interaction with a user through
a computer network, with no transfer of a copy, is not conveying.

  An interactive user interface displays "Appropriate Legal Notices"
to the extent that it includes a convenient and prominently visible
feature that (1) displays an appropriate copyright notice, and (2)
tells the user that there is no warranty for the work (except to the
extent that warranties are provided), that licensees may convey the
work under this License, and how to view a copy of this License.  If
the interface presents a list of user commands or options, such as a
menu, a prominent item in the list meets this criterion.

  1. Source Code.

  The "source code" for a work means the preferred form of the work
for making modifications to it.  "Object code" means any non-source
form of a work.

  A "Standard Interface" means an interface that either is an official
standard defined by a recognized standards body, or, in the case of
interfaces specified for a particular programming language, one that
is widely used among developers working in that language.

  The "System Libraries" of an executable work include anything, other
than the work as a whole, that (a) is included in the normal form of
packaging a Major Component, but which is not part of that Major
Component, and (b) serves only to enable use of the work with that
Major Component, or to implement a Standard Interface for which an
implementation is available to the public in source code form.  A
"Major Component", in this context, means a major essential component
(kernel, window system, and so on) of the specific operating system
(if any) on which the executable work runs, or a compiler used to
produce the work, or an object code interpreter used to run it.

  The "Corresponding Source" for a work in object code form means all
the source code needed to generate, install, and (for an executable
work) run the object code and to modify the work, including scripts to
control those activities.  However, it does not include the work's
System Libraries, or general-purpose tools or generally available free
programs which are used unmodified in performing those activities but
which are not part of the work.  For example, Corresponding Source
includes interface definition files associated with source files for
the work, and the source code for shared libraries and dynamically
linked subprograms that the work is specifically designed to require,
such as by intimate data communication or control flow between those
subprograms and other parts of the work.
```

(continues on next page)

()

```
  The Corresponding Source need not include anything that users
can regenerate automatically from other parts of the Corresponding
Source.

  The Corresponding Source for a work in source code form is that
same work.

  2. Basic Permissions.

  All rights granted under this License are granted for the term of
copyright on the Program, and are irrevocable provided the stated
conditions are met.  This License explicitly affirms your unlimited
permission to run the unmodified Program.  The output from running a
covered work is covered by this License only if the output, given its
content, constitutes a covered work.  This License acknowledges your
rights of fair use or other equivalent, as provided by copyright law.

  You may make, run and propagate covered works that you do not
convey, without conditions so long as your license otherwise remains
in force.  You may convey covered works to others for the sole purpose
of having them make modifications exclusively for you, or provide you
with facilities for running those works, provided that you comply with
the terms of this License in conveying all material for which you do
not control copyright.  Those thus making or running the covered works
for you must do so exclusively on your behalf, under your direction
and control, on terms that prohibit them from making any copies of
your copyrighted material outside their relationship with you.

  Conveying under any other circumstances is permitted solely under
the conditions stated below.  Sublicensing is not allowed; section 10
makes it unnecessary.

  3. Protecting Users' Legal Rights From Anti-Circumvention Law.

  No covered work shall be deemed part of an effective technological
measure under any applicable law fulfilling obligations under article
11 of the WIPO copyright treaty adopted on 20 December 1996, or
similar laws prohibiting or restricting circumvention of such
measures.

  When you convey a covered work, you waive any legal power to forbid
circumvention of technological measures to the extent such circumvention
is effected by exercising rights under this License with respect to
the covered work, and you disclaim any intention to limit operation or
modification of the work as a means of enforcing, against the work's
users, your or third parties' legal rights to forbid circumvention of
technological measures.

  4. Conveying Verbatim Copies.

  You may convey verbatim copies of the Program's source code as you
receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
```

(continues on next page)

()

```
recipients a copy of this License along with the Program.

  You may charge any price or no price for each copy that you convey,
and you may offer support or warranty protection for a fee.

  5. Conveying Modified Source Versions.

  You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:

    a) The work must carry prominent notices stating that you modified
    it, and giving a relevant date.

    b) The work must carry prominent notices stating that it is
    released under this License and any conditions added under section
    7.  This requirement modifies the requirement in section 4 to
    "keep intact all notices".

    c) You must license the entire work, as a whole, under this
    License to anyone who comes into possession of a copy.  This
    License will therefore apply, along with any applicable section 7
    additional terms, to the whole of the work, and all its parts,
    regardless of how they are packaged.  This License gives no
    permission to license the work in any other way, but it does not
    invalidate such permission if you have separately received it.

    d) If the work has interactive user interfaces, each must display
    Appropriate Legal Notices; however, if the Program has interactive
    interfaces that do not display Appropriate Legal Notices, your
    work need not make them do so.

  A compilation of a covered work with other separate and independent
works, which are not by their nature extensions of the covered work,
and which are not combined with it such as to form a larger program,
in or on a volume of a storage or distribution medium, is called an
"aggregate" if the compilation and its resulting copyright are not
used to limit the access or legal rights of the compilation's users
beyond what the individual works permit.  Inclusion of a covered work
in an aggregate does not cause this License to apply to the other
parts of the aggregate.

  6. Conveying Non-Source Forms.

  You may convey a covered work in object code form under the terms
of sections 4 and 5, provided that you also convey the
machine-readable Corresponding Source under the terms of this License,
in one of these ways:

    a) Convey the object code in, or embodied in, a physical product
    (including a physical distribution medium), accompanied by the
    Corresponding Source fixed on a durable physical medium
    customarily used for software interchange.

    b) Convey the object code in, or embodied in, a physical product
    (including a physical distribution medium), accompanied by a
    written offer, valid for at least three years and valid for as
```

(continues on next page)

()

```
    long as you offer spare parts or customer support for that product
    model, to give anyone who possesses the object code either (1) a
    copy of the Corresponding Source for all the software in the
    product that is covered by this License, on a durable physical
    medium customarily used for software interchange, for a price no
    more than your reasonable cost of physically performing this
    conveying of source, or (2) access to copy the
    Corresponding Source from a network server at no charge.

    c) Convey individual copies of the object code with a copy of the
    written offer to provide the Corresponding Source.  This
    alternative is allowed only occasionally and noncommercially, and
    only if you received the object code with such an offer, in accord
    with subsection 6b.

    d) Convey the object code by offering access from a designated
    place (gratis or for a charge), and offer equivalent access to the
    Corresponding Source in the same way through the same place at no
    further charge.  You need not require recipients to copy the
    Corresponding Source along with the object code.  If the place to
    copy the object code is a network server, the Corresponding Source
    may be on a different server (operated by you or a third party)
    that supports equivalent copying facilities, provided you maintain
    clear directions next to the object code saying where to find the
    Corresponding Source.  Regardless of what server hosts the
    Corresponding Source, you remain obligated to ensure that it is
    available for as long as needed to satisfy these requirements.

    e) Convey the object code using peer-to-peer transmission, provided
    you inform other peers where the object code and Corresponding
    Source of the work are being offered to the general public at no
    charge under subsection 6d.

  A separable portion of the object code, whose source code is excluded
from the Corresponding Source as a System Library, need not be
included in conveying the object code work.

  A "User Product" is either (1) a "consumer product", which means any
tangible personal property which is normally used for personal, family,
or household purposes, or (2) anything designed or sold for incorporation
into a dwelling.  In determining whether a product is a consumer product,
doubtful cases shall be resolved in favor of coverage.  For a particular
product received by a particular user, "normally used" refers to a
typical or common use of that class of product, regardless of the status
of the particular user or of the way in which the particular user
actually uses, or expects or is expected to use, the product.  A product
is a consumer product regardless of whether the product has substantial
commercial, industrial or non-consumer uses, unless such uses represent
the only significant mode of use of the product.

  "Installation Information" for a User Product means any methods,
procedures, authorization keys, or other information required to install
and execute modified versions of a covered work in that User Product from
a modified version of its Corresponding Source.  The information must
suffice to ensure that the continued functioning of the modified object
code is in no case prevented or interfered with solely because
modification has been made.
```

(continues on next page)

()

```
  If you convey an object code work under this section in, or with, or
specifically for use in, a User Product, and the conveying occurs as
part of a transaction in which the right of possession and use of the
User Product is transferred to the recipient in perpetuity or for a
fixed term (regardless of how the transaction is characterized), the
Corresponding Source conveyed under this section must be accompanied
by the Installation Information.  But this requirement does not apply
if neither you nor any third party retains the ability to install
modified object code on the User Product (for example, the work has
been installed in ROM).

  The requirement to provide Installation Information does not include a
requirement to continue to provide support service, warranty, or updates
for a work that has been modified or installed by the recipient, or for
the User Product in which it has been modified or installed.  Access to a
network may be denied when the modification itself materially and
adversely affects the operation of the network or violates the rules and
protocols for communication across the network.

  Corresponding Source conveyed, and Installation Information provided,
in accord with this section must be in a format that is publicly
documented (and with an implementation available to the public in
source code form), and must require no special password or key for
unpacking, reading or copying.

  7. Additional Terms.

  "Additional permissions" are terms that supplement the terms of this
License by making exceptions from one or more of its conditions.
Additional permissions that are applicable to the entire Program shall
be treated as though they were included in this License, to the extent
that they are valid under applicable law.  If additional permissions
apply only to part of the Program, that part may be used separately
under those permissions, but the entire Program remains governed by
this License without regard to the additional permissions.

  When you convey a copy of a covered work, you may at your option
remove any additional permissions from that copy, or from any part of
it.  (Additional permissions may be written to require their own
removal in certain cases when you modify the work.)  You may place
additional permissions on material, added by you to a covered work,
for which you have or can give appropriate copyright permission.

  Notwithstanding any other provision of this License, for material you
add to a covered work, you may (if authorized by the copyright holders of
that material) supplement the terms of this License with terms:

    a) Disclaiming warranty or limiting liability differently from the
    terms of sections 15 and 16 of this License; or

    b) Requiring preservation of specified reasonable legal notices or
    author attributions in that material or in the Appropriate Legal
    Notices displayed by works containing it; or

    c) Prohibiting misrepresentation of the origin of that material, or
    requiring that modified versions of such material be marked in
```

(continues on next page)

()

```
    reasonable ways as different from the original version; or

    d) Limiting the use for publicity purposes of names of licensors or
    authors of the material; or

    e) Declining to grant rights under trademark law for use of some
    trade names, trademarks, or service marks; or

    f) Requiring indemnification of licensors and authors of that
    material by anyone who conveys the material (or modified versions of
    it) with contractual assumptions of liability to the recipient, for
    any liability that these contractual assumptions directly impose on
    those licensors and authors.

  All other non-permissive additional terms are considered "further
restrictions" within the meaning of section 10.  If the Program as you
received it, or any part of it, contains a notice stating that it is
governed by this License along with a term that is a further
restriction, you may remove that term.  If a license document contains
a further restriction but permits relicensing or conveying under this
License, you may add to a covered work material governed by the terms
of that license document, provided that the further restriction does
not survive such relicensing or conveying.

  If you add terms to a covered work in accord with this section, you
must place, in the relevant source files, a statement of the
additional terms that apply to those files, or a notice indicating
where to find the applicable terms.

  Additional terms, permissive or non-permissive, may be stated in the
form of a separately written license, or stated as exceptions;
the above requirements apply either way.

  8. Termination.

  You may not propagate or modify a covered work except as expressly
provided under this License.  Any attempt otherwise to propagate or
modify it is void, and will automatically terminate your rights under
this License (including any patent licenses granted under the third
paragraph of section 11).

  However, if you cease all violation of this License, then your
license from a particular copyright holder is reinstated (a)
provisionally, unless and until the copyright holder explicitly and
finally terminates your license, and (b) permanently, if the copyright
holder fails to notify you of the violation by some reasonable means
prior to 60 days after the cessation.

  Moreover, your license from a particular copyright holder is
reinstated permanently if the copyright holder notifies you of the
violation by some reasonable means, this is the first time you have
received notice of violation of this License (for any work) from that
copyright holder, and you cure the violation prior to 30 days after
your receipt of the notice.

  Termination of your rights under this section does not terminate the
licenses of parties who have received copies or rights from you under
```

(continues on next page)

()

```
this License.  If your rights have been terminated and not permanently
reinstated, you do not qualify to receive new licenses for the same
material under section 10.

  9. Acceptance Not Required for Having Copies.

  You are not required to accept this License in order to receive or
run a copy of the Program.  Ancillary propagation of a covered work
occurring solely as a consequence of using peer-to-peer transmission
to receive a copy likewise does not require acceptance.  However,
nothing other than this License grants you permission to propagate or
modify any covered work.  These actions infringe copyright if you do
not accept this License.  Therefore, by modifying or propagating a
covered work, you indicate your acceptance of this License to do so.

  10. Automatic Licensing of Downstream Recipients.

  Each time you convey a covered work, the recipient automatically
receives a license from the original licensors, to run, modify and
propagate that work, subject to this License.  You are not responsible
for enforcing compliance by third parties with this License.

  An "entity transaction" is a transaction transferring control of an
organization, or substantially all assets of one, or subdividing an
organization, or merging organizations.  If propagation of a covered
work results from an entity transaction, each party to that
transaction who receives a copy of the work also receives whatever
licenses to the work the party's predecessor in interest had or could
give under the previous paragraph, plus a right to possession of the
Corresponding Source of the work from the predecessor in interest, if
the predecessor has it or can get it with reasonable efforts.

  You may not impose any further restrictions on the exercise of the
rights granted or affirmed under this License.  For example, you may
not impose a license fee, royalty, or other charge for exercise of
rights granted under this License, and you may not initiate litigation
(including a cross-claim or counterclaim in a lawsuit) alleging that
any patent claim is infringed by making, using, selling, offering for
sale, or importing the Program or any portion of it.

  11. Patents.

  A "contributor" is a copyright holder who authorizes use under this
License of the Program or a work on which the Program is based.  The
work thus licensed is called the contributor's "contributor version".

  A contributor's "essential patent claims" are all patent claims
owned or controlled by the contributor, whether already acquired or
hereafter acquired, that would be infringed by some manner, permitted
by this License, of making, using, or selling its contributor version,
but do not include claims that would be infringed only as a
consequence of further modification of the contributor version.  For
purposes of this definition, "control" includes the right to grant
patent sublicenses in a manner consistent with the requirements of
this License.

  Each contributor grants you a non-exclusive, worldwide, royalty-free
```

(continues on next page)

()

```
patent license under the contributor's essential patent claims, to
make, use, sell, offer for sale, import and otherwise run, modify and
propagate the contents of its contributor version.

  In the following three paragraphs, a "patent license" is any express
agreement or commitment, however denominated, not to enforce a patent
(such as an express permission to practice a patent or covenant not to
sue for patent infringement).  To "grant" such a patent license to a
party means to make such an agreement or commitment not to enforce a
patent against the party.

  If you convey a covered work, knowingly relying on a patent license,
and the Corresponding Source of the work is not available for anyone
to copy, free of charge and under the terms of this License, through a
publicly available network server or other readily accessible means,
then you must either (1) cause the Corresponding Source to be so
available, or (2) arrange to deprive yourself of the benefit of the
patent license for this particular work, or (3) arrange, in a manner
consistent with the requirements of this License, to extend the patent
license to downstream recipients.  "Knowingly relying" means you have
actual knowledge that, but for the patent license, your conveying the
covered work in a country, or your recipient's use of the covered work
in a country, would infringe one or more identifiable patents in that
country that you have reason to believe are valid.

  If, pursuant to or in connection with a single transaction or
arrangement, you convey, or propagate by procuring conveyance of, a
covered work, and grant a patent license to some of the parties
receiving the covered work authorizing them to use, propagate, modify
or convey a specific copy of the covered work, then the patent license
you grant is automatically extended to all recipients of the covered
work and works based on it.

  A patent license is "discriminatory" if it does not include within
the scope of its coverage, prohibits the exercise of, or is
conditioned on the non-exercise of one or more of the rights that are
specifically granted under this License.  You may not convey a covered
work if you are a party to an arrangement with a third party that is
in the business of distributing software, under which you make payment
to the third party based on the extent of your activity of conveying
the work, and under which the third party grants, to any of the
parties who would receive the covered work from you, a discriminatory
patent license (a) in connection with copies of the covered work
conveyed by you (or copies made from those copies), or (b) primarily
for and in connection with specific products or compilations that
contain the covered work, unless you entered into that arrangement,
or that patent license was granted, prior to 28 March 2007.

  Nothing in this License shall be construed as excluding or limiting
any implied license or other defenses to infringement that may
otherwise be available to you under applicable patent law.

  12. No Surrender of Others' Freedom.

  If conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License.  If you cannot convey a
```

(continues on next page)

()

```
covered work so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you may
not convey it at all.  For example, if you agree to terms that obligate you
to collect a royalty for further conveying from those to whom you convey
the Program, the only way you could satisfy both those terms and this
License would be to refrain entirely from conveying the Program.

  13. Use with the GNU Affero General Public License.

  Notwithstanding any other provision of this License, you have
permission to link or combine any covered work with a work licensed
under version 3 of the GNU Affero General Public License into a single
combined work, and to convey the resulting work.  The terms of this
License will continue to apply to the part which is the covered work,
but the special requirements of the GNU Affero General Public License,
section 13, concerning interaction through a network will apply to the
combination as such.

  14. Revised Versions of this License.

  The Free Software Foundation may publish revised and/or new versions of
the GNU General Public License from time to time.  Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

  Each version is given a distinguishing version number.  If the
Program specifies that a certain numbered version of the GNU General
Public License "or any later version" applies to it, you have the
option of following the terms and conditions either of that numbered
version or of any later version published by the Free Software
Foundation.  If the Program does not specify a version number of the
GNU General Public License, you may choose any version ever published
by the Free Software Foundation.

  If the Program specifies that a proxy can decide which future
versions of the GNU General Public License can be used, that proxy's
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

  Later license versions may give you additional or different
permissions.  However, no additional obligations are imposed on any
author or copyright holder as a result of your choosing to follow a
later version.

  15. Disclaimer of Warranty.

  THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW.  EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.  THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
IS WITH YOU.  SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

  16. Limitation of Liability.
```

(continues on next page)

()

```
  IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

  17. Interpretation of Sections 15 and 16.

  If the disclaimer of warranty and limitation of liability provided
above cannot be given local legal effect according to their terms,
reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
copy of the Program in return for a fee.

                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program.  It is safest
to attach them to the start of each source file to most effectively
state the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

  If the program does terminal interaction, make it output a short
notice like this when it starts in an interactive mode:

    <program>  Copyright (C) <year>  <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.
```

(continues on next page)

()

```
The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License.  Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

  You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see
<http://www.gnu.org/licenses/>.

  The GNU General Public License does not permit incorporating your program
into proprietary programs.  If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library.  If this is what you want to do, use the GNU Lesser General
Public License instead of this License.  But first, please read
<http://www.gnu.org/philosophy/why-not-lgpl.html>.
```

CHAPTER 8

Contributing

8.1 Contributing

Contributions are welcome. If you want to contribute, you should join the #enms channel in the networktocode slack (http://networktocode.herokuapp.com/).

8.1.1 For developers

eNMS uses flake8 to make sure that the code is PEP8-compliant, and pytest for the test suite. There is a dedicated `requirements_dev.txt` file to install these libraries:

```
pip install -r requirements_dev.txt
```

Before opening a pull request with your changes, you should make sure that:

```
# your code is PEP8 (flake8) compliant
flake8

# all the tests are passing
cd /tests
pytest
```

If one of these checks fails, so will Travis CI after opening the pull request.

If you are updating the documentation, you can build a local version of the docs:

```
# build a local version of the docs
cd /docs
make html
```

CHAPTER 9

Indices and Tables

- genindex
- glossary