ngworx.ag

Building a StableNet Ecosystem

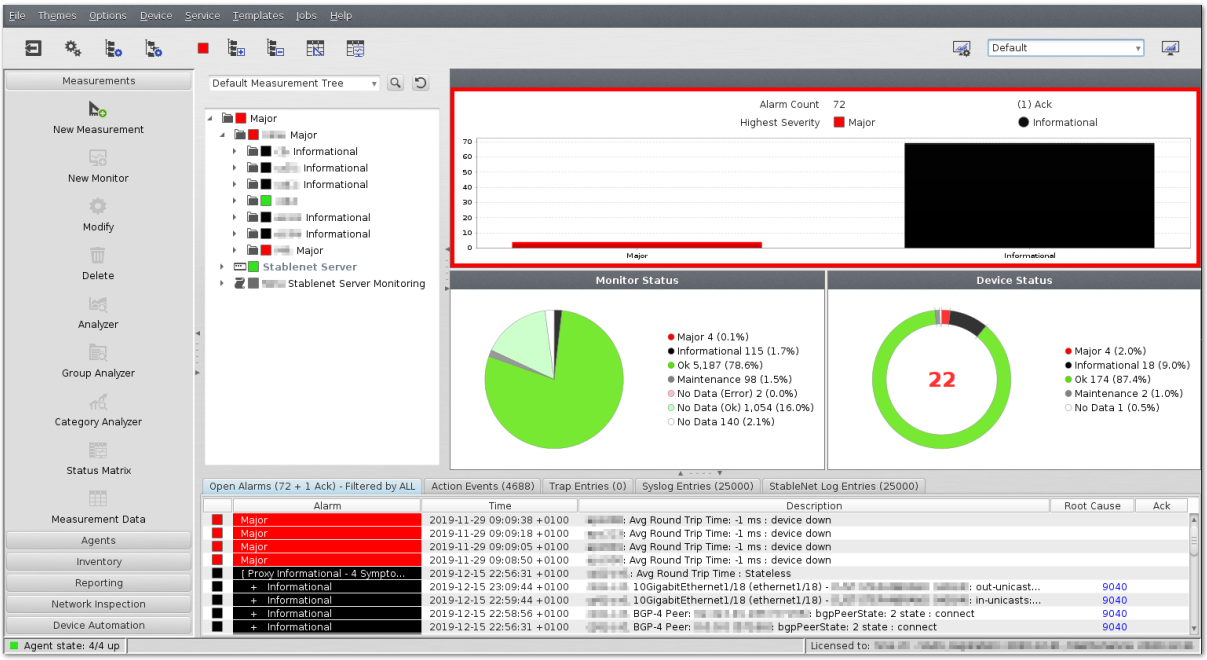
December 2019
written by Fabian Waeber
Network Engineer & Developer

In this article, I want to present some of our StableNet extensions, especially those that go further than the large choice of possibilities already shipped by default in StableNet.

As a Partner of Infosim, we work with many instances of StableNet throughout our customer base. In some cases, StableNet is used only as a monitoring and alarming solution without any need for customization. In other cases, StableNet is used to automate the provisioning of the new end customer services or the integration of non-standard, external measurements – to satisfy those needs, we had to develop several extensions and can safely call it an „ecosystem“ we grew around StableNet.

Introduction to StableNet

Infosim describes their product as follows: „StableNet provides Fault, Performance, Configuration, and Services Management on a single platform.“ – please see this page for full details about StableNet.



(StableNet Start Screen)

In the bigger context, StableNet is „just another“ Network Management System (NMS)… We have been asked several times to implement a subset of StableNet‘s features with free software and I can assure you that you really get a great deal with StableNet: although you will find free tools implementing some the functionality, you will have to integrate additional tools and/or write functionality on your own to even get close to what StableNet offers all-in-one.

Beside the key features of automatic discovery of new devices and services, monitoring, alarming, and reporting, the most used feature by our customers is configuration automation: StableNet comes with support for a XML based scripting language for automation of simple and even more complex tasks. And it is just as easy to integrate some database queries to fetch your data and use it for configuration. But enough for the introduction, let‘s see what additional extensions to StableNet we were asked by our customers to add.

External Versioning with GitLab

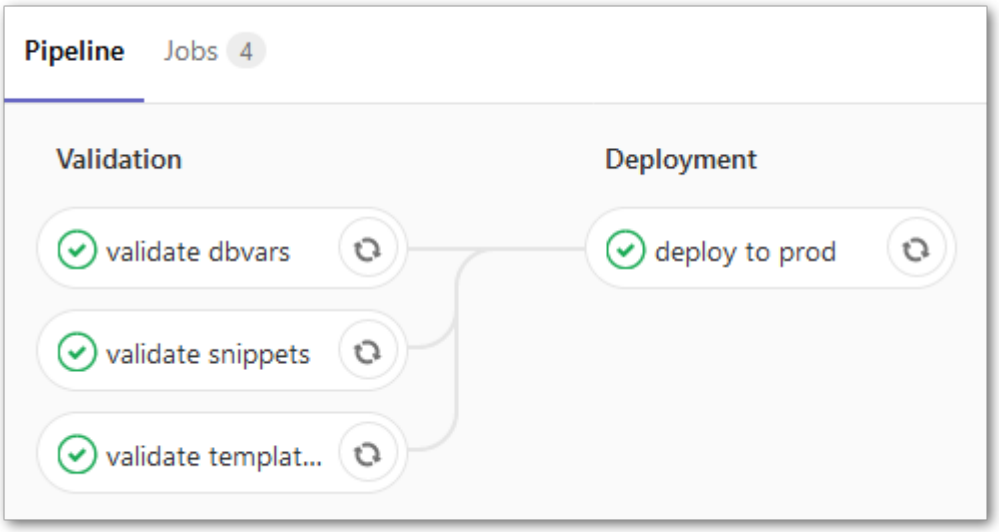
StableNet offers a lot of customization possibilities: It is possible to write reusable Database queries, scripts for configuration of templates to be re-used in device configurations. All those customizations are managed through XML text files and can be up- or downloaded from StableNet for external modification in your favorite text editor. Recently StableNet implemented versioning of those XML files internally, but as the number of developers grows, a solution using a full-fledged

version control system was required. Its availability as a self-hosted solution, CI/CD capabilities and popularity narrowed the choice down to GitLab CE.

The development of any file-based StableNet configuration, but especially for „database variables“, „configuration templates“ and „configuration snippets“ can now be done in GitLab‘s web IDE or by checking them out to a local workstation through git. All well-known features of GitLab, but especially access control for merge requests including discussions can now be used to synchronize, review and manage changes to the StableNet XML files.

CI/CD pipelines are then used on protected branches to run automatic XML validation and synchronization to StableNet:

- Any change to the master branch will trigger simple XML validation. This prevents merge requests with broken XML.
- When a merge request to the „test“ branch is accepted, CI/CD will automatically upload all changes to the configured StableNet test instance.
- Once the configurations are confirmed to work on the test instance, a merge to the „production“ branch will trigger synchronization to the according StableNet server.



(Tasks for Deployment to Production)

Besides teaching network engineers how to properly use version control, the challenge for this task was the quick evolution of the available REST API calls: Infosim extended the possibility to manage

the configuration files during our development phase, which lead to several extensions due to new possibilities.

Related Service See how we help businesses with our network automation services:

Network Automation With our automation services we help you to an efficient network operation. See more

Migration Sanity Checks

During a lifecycle project, one of our customers exchanged his PE routers with new devices from another vendor. As most of the end customers have managed CE routers, it was possible to run sanity checks before and after the migration on both CE and PE devices. The goal was to check interfaces, BGP sessions and advertised routes before and after the migration allowing the engineer in charge to check if the customer has the same service state after the change.

StableNet offers highly customizable backup jobs which basically produce text output (normally the device configuration) to be stored in StableNet. For the migration sanity checks, we developed a custom backup script generating the required output text from different commands (for example the output of „show interfaces brief“ or „show IP route“). The challenge, especially for the swapped PE devices, was to generate „normalized“ output so that the output of a Cisco device is comparable to the output of a Huawei or Juniper device. StableNet calls them „scripts“, but in reality, we developed and delivered a java executable also containing additional libraries to connect to all required devices (mainly ssh, but also telnet to check the public route servers). Why Java you may ask? The main reason is simply that StableNet itself is based on Java and therefore all StableNet Servers and Agents have a JVM at their disposal – and shipping the additional libraries bundled within the jar-archive eliminated the need to install any dependencies on the StableNet Agents.

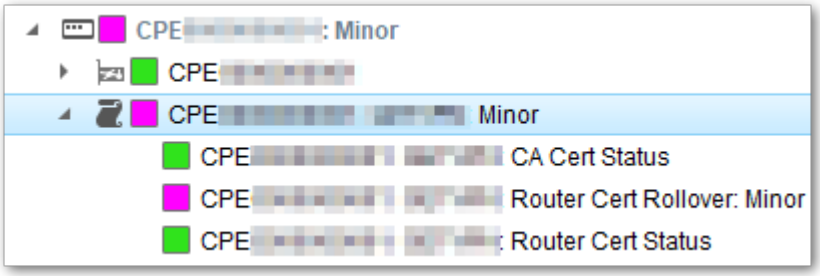
Line	Content
1	<source index='0'>
2	[default| .80/29]: AS-Path: [ ]
3	[default| .80/29]: Metric: [0]
4	[default| .80/29]: Localpref: [100]
5	[default| .80/29]: Community: [N/A]
6	</source>
7	<source index='1'>
8	[default| .80/29]: AS-Path: [ ]
9	[default| .80/29]: Metric: [0]
10	[default| .80/29]: Localpref: [100]
11	[default| .80/29]: Community: [N/A]
12	</source>

(Normalized BGP Session Example)

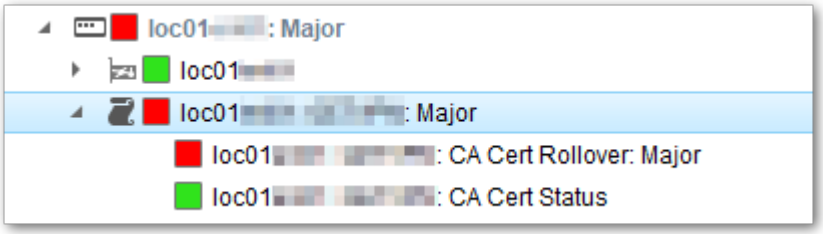
Certificate Check and Alarming

In another project, our customer needs to keep track of certificate validity for a bigger amount of VPN connections. As our customer already uses StableNet for monitoring and alarming, the task was to introduce warnings for certificates due to rollover or alarms for expired certificates. Also, the validity of the CA certificate has to be monitored.

Even though StableNet offers scripting features for measurements, we had to write Java-based extensions to connect to different platforms through ssh and to do the math for the certificate expiration. The result are two so-called „business process scripts“, one for the clients and one for the CA. Those scripts can be seamlessly integrated into StableNet‘s discovery features, the same as the scripts delivered by Infosim:



(Minor Alarm for CPE Certificate due to roll-over)



(Major Alarm because of missing CA roll-over)

Integrating External Measurement Data

As you may know, we developed ng.upp: a system composed of embedded devices measuring a variety of highly customizable things like WiFi signal strength, connection times or iPerf performance. Usually, we offer Grafana dashboards to view and analyze the data collected centrally in an InfluxDB. Again, our customer already uses StableNet for monitoring and alarming – so the task was to integrate all ng.upp measurements into StableNet.

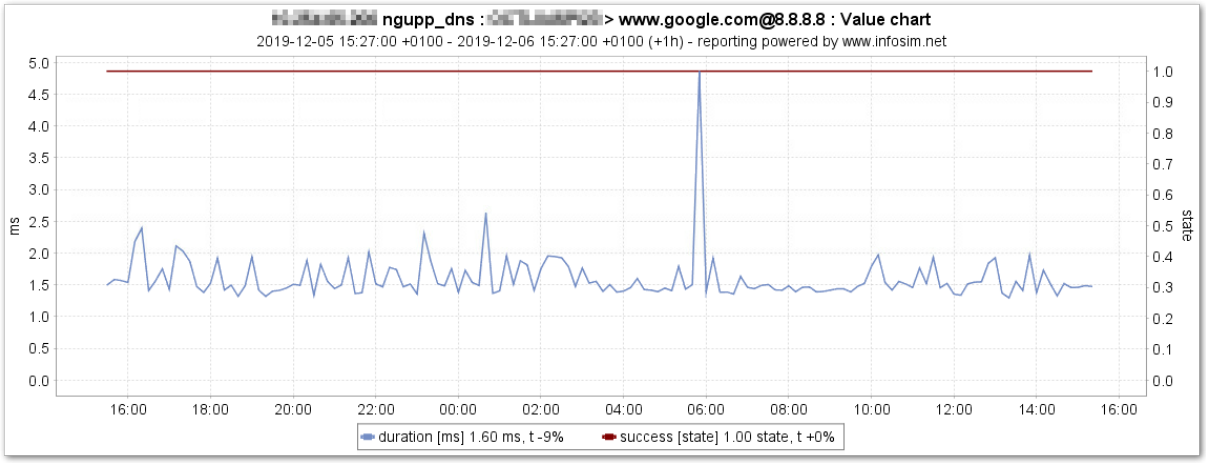
Luckily, StableNet offers a way to import multiple measurements with only one execution of an „External Measurement Data“ script. Because of the already available JRE, the choice fell again on writing Java „scripts“. The script connects to the InfluxDB through a ssh tunnel and gathers all measurement values from the requested time period.

As StableNet needs to be aware of the value types collected by the individual ng.upp measurements, we make use of the same XML configuration to tell the script which data to fetch from ng.upp:

```
<external>
    <name>ngupp_dns</name>
    <version>1</version>
    <output>
        <name>duration</name>
        <unit>ms</unit>
        <mon cat0="ng.upp" default="false"/>
        <id>1</id>
    </output>
    <output>
        <name>success</name>
        <unit>state</unit>
        <mon cat0="ng.upp" lower="1.0" upper="1.0" nodataalarm="true" sticky="false"/>
        <id>2</id>
    </output>
</external>
```

(Shared Definition for ng.upp DNS Check)

Reusing the same definition allows us to deploy new ng.upp measurement types and integrate them without any change to the script code – we just have to write and upload a new StableNet XML definition as shown above.



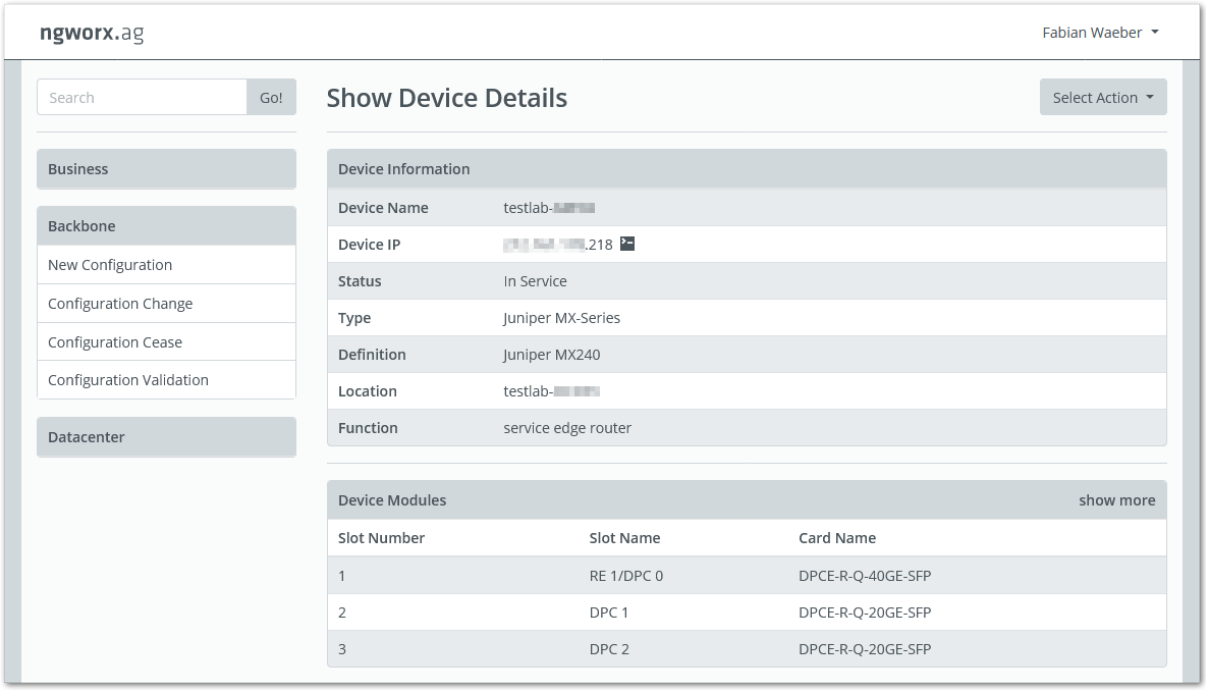
(ng.upp DNS Check Result in StableNet)

Through StableNet's discovery system, threshold values for alarms can now be assigned to any ng.upp measurement.

Provisioning Web Interface

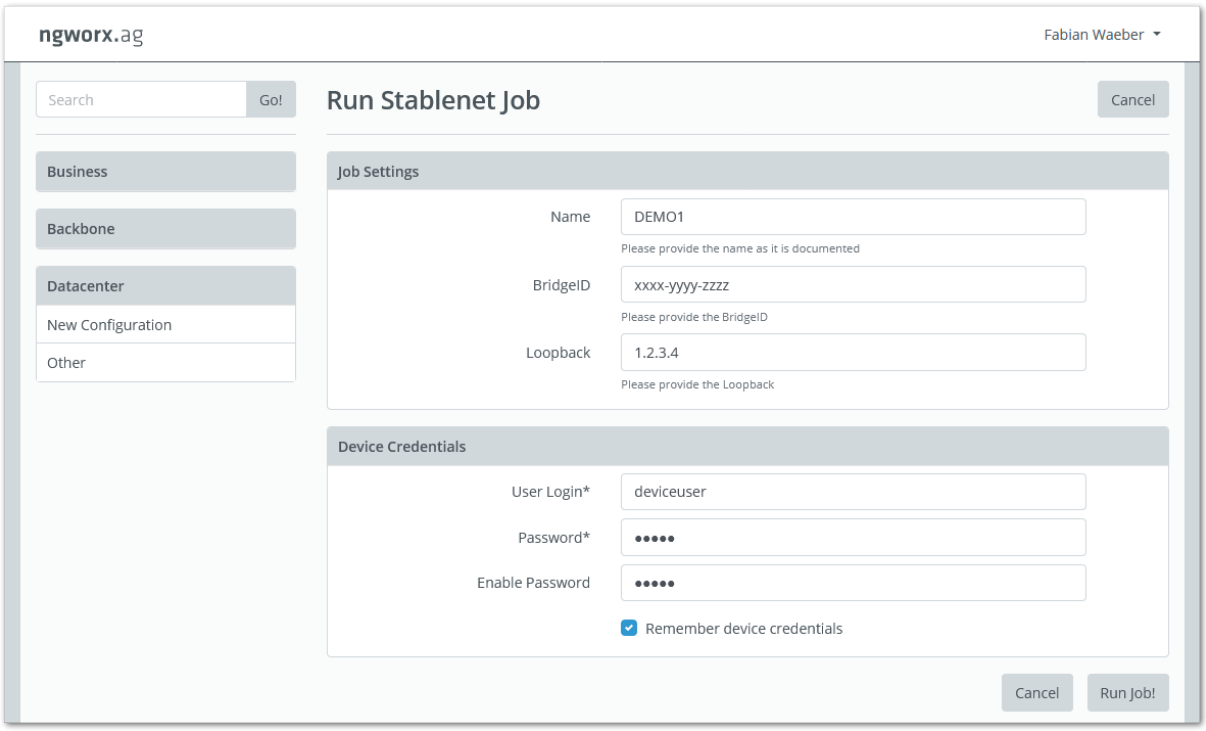
The biggest tool in our StableNet ecosystem is the „Provisioning Interface"… Unlike the other extensions, it is not directly an extension of any StableNet feature: it is a PHP/Laravel based web application, connecting to any DCIM and helping to find and run matching StableNet jobs.

The provisioning interface provides a customizable search through the DCIM data: for example, one search could be configured to only return routers of a specific model with the keyword in the device name. Also, the detail view of a result is completely customizable – but most importantly, only the matching StableNet jobs are directly selectable for execution:



(Detail View of a Device Search Result)

Alternatively, the StableNet jobs can also be categorized by department and category for easy access. Regardless of how a StableNet job is selected to be run, the definition is fetched from StableNet and the input fields are rendered into an HTML input form (and even prefilled if possible):



(Rendered HTML Form of a StableNet Job)

The final advantage of running the StableNet job through the web interface is that „personal device credentials“ are always set automatically opposed to changing the job manually before it‘s execution.

Conclusion

When working so closely with a certain Software, you will always find some points you wish where different… the same goes for StableNet! But it gives a great value out of the box and if you need anything more, StableNet really offers many interfaces to build your own extension for whatever special needs you may have.