opConfig Virtual Operator

Introduction

Opmantek's Virtual Operator can be used to help create jobs comprised of command sets to be run on various nodes, reporting to see job results and troubleshooting to diagnoses nodes which raise conditions through opConfig's plugin system.

Quick Actions are Virtual Operator templates which can save you time from creating commonly run jobs. Or give operators easy access to run commands on remote systems.

New Virtual Operator Job

To create a new Virtual Operator job go to Virtual Operator and click New Virtual Operator Job.

You will need to select which nodes you want to run commands on, these are auto-completed from the list of currently activated nodes in opConfig.

Next you can select which command sets should be run on the nodes, these are auto-completed from all command sets which opConfig has loaded.

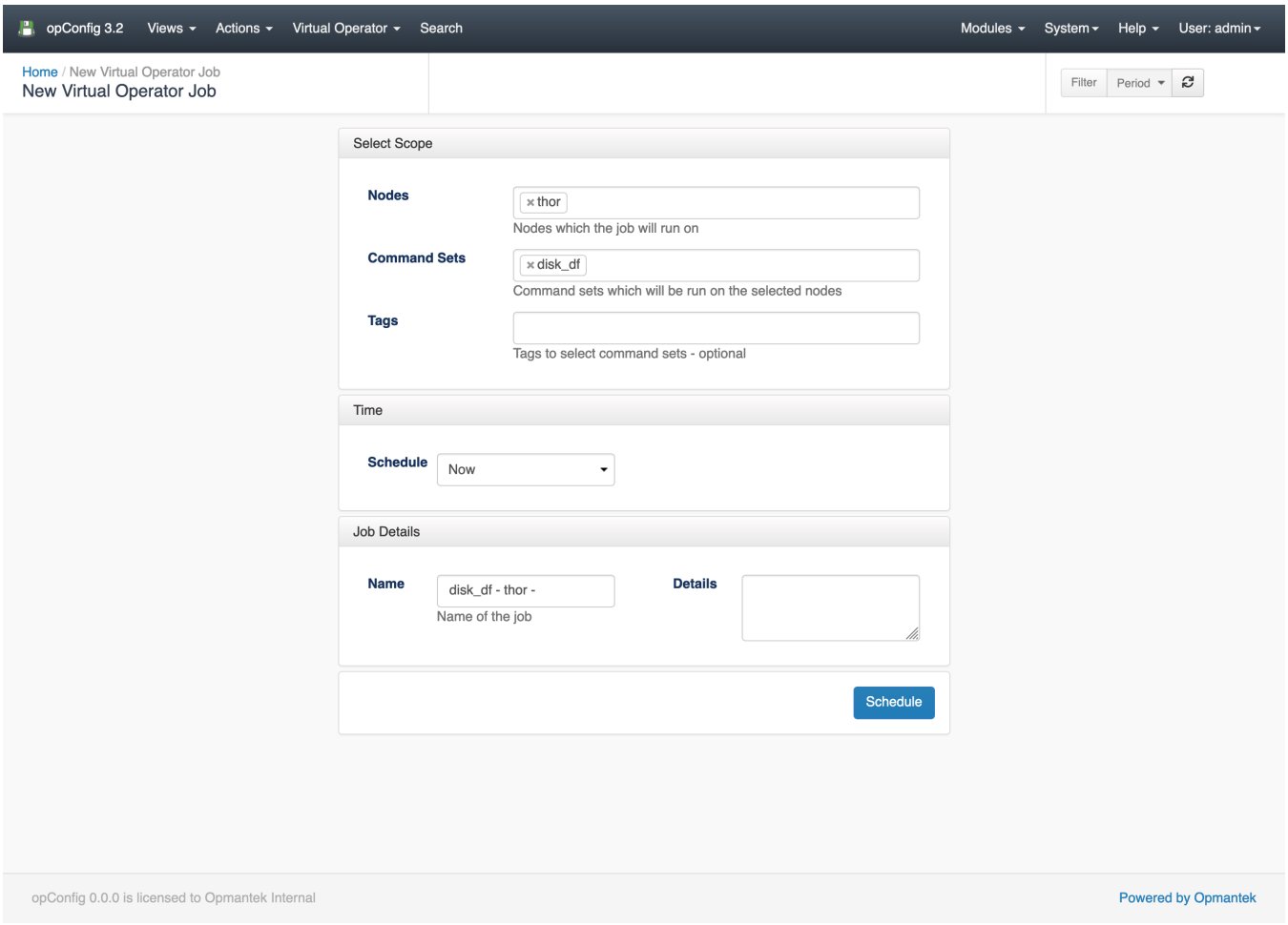
You can also use tags to select which command sets should be run.

Schedule can be run Now or Later and selecting later will bring a time-picker to schedule when this job shall be run.

The name is auto generated from data you have input, but can be anything you decide.

Details is a free text field for keeping notes about this job.

Clicking schedule will add the Job to opConfig's queue and take you to the report schedule.



Creating a new Virtual Operator job

Virtual Operator Report

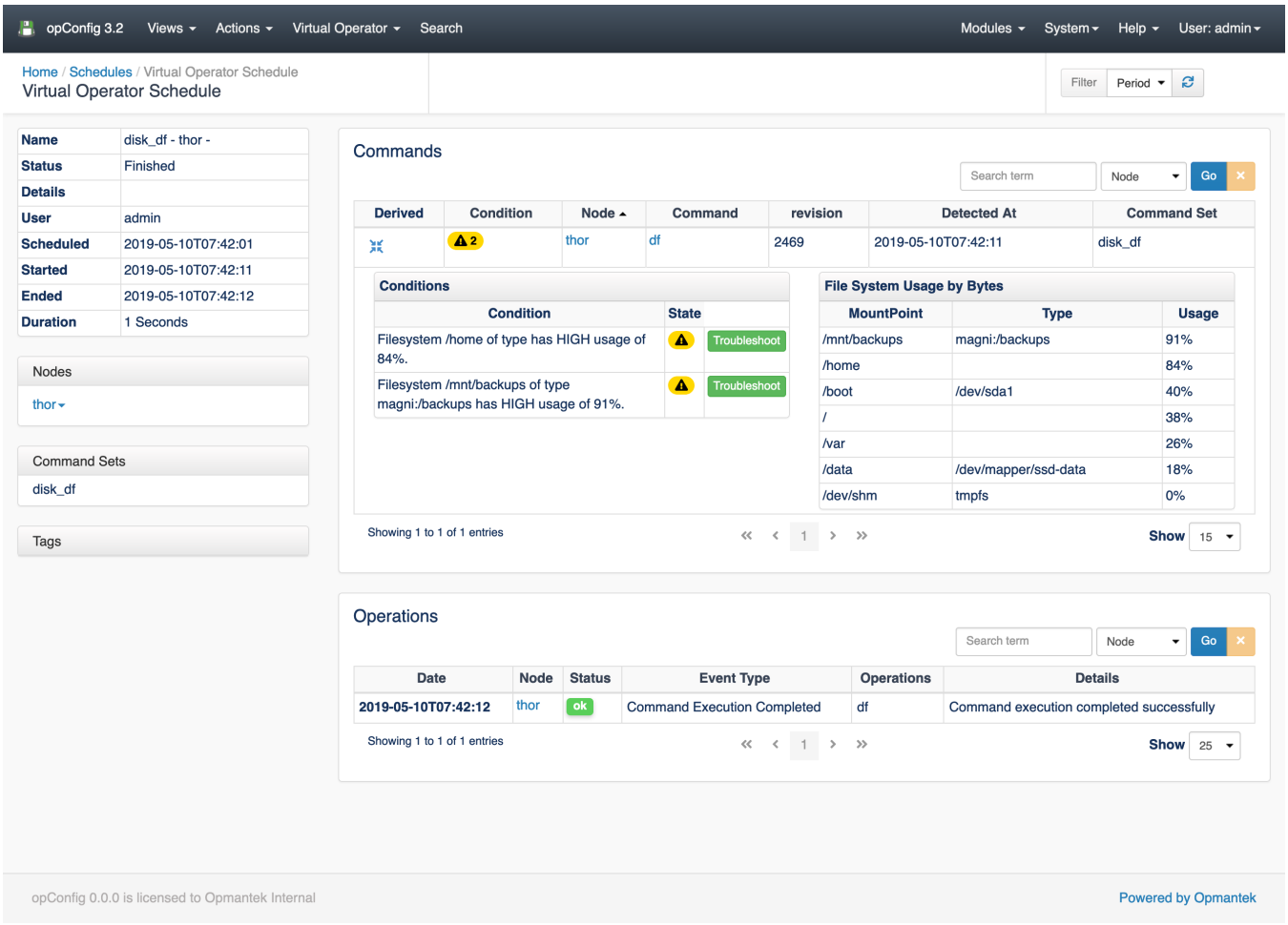
A Virtual Operator Report is an aggregation of all data collected from your virtual operator job.

On the left panel you have meta data about the job, how it was created who whom and when its going to be or when it was run.

Commands panel is a paginated table of the successful commands which were run for the current job. If the command set is using a plugin to show derived data or report conditions these results are shown inline by clicking the expand icon in the derived column.

If the condition has a tag this can be used to help filter down command sets for creating linked virtual operator jobs off these conditions.

All operations for the current job are shown to help diagnose connection or command issue which have been run.



Virtual Operator Job Report

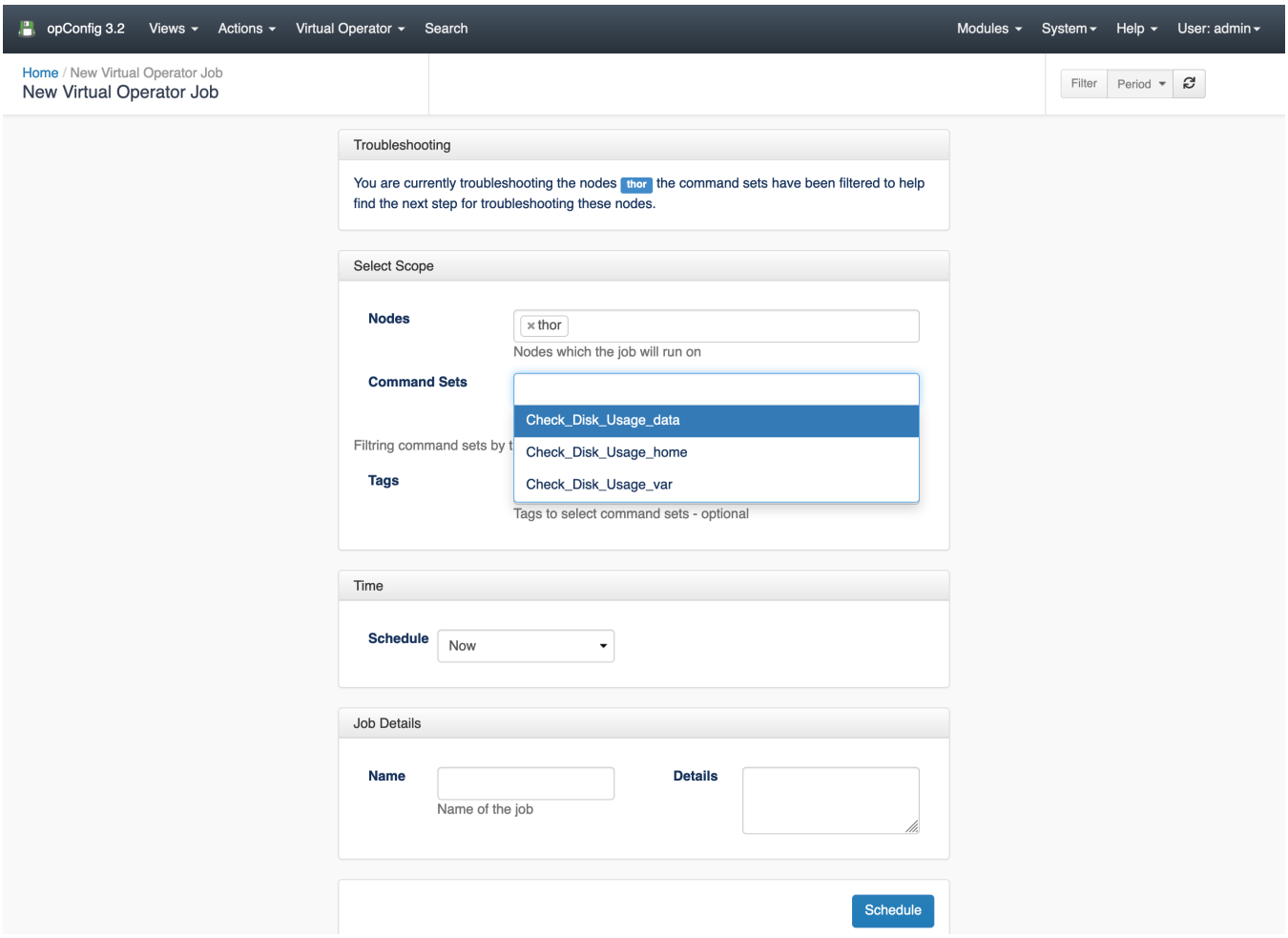
Virtual Operator Troubleshooting

If you have clicked troubleshoot from a report condition you are taken to the New Virtual Operator Job screen but there are a couple of key differences.

The node has already been filled out and the command sets have been filtered down using a tag, in this example we have three command sets with the tag disk.

This can help to create workflows where conditions are tagged to limit what the operator can select for the next steps.

When this job is created the parents job ID is also recorded and the parents job name is is shown in the newly created report.



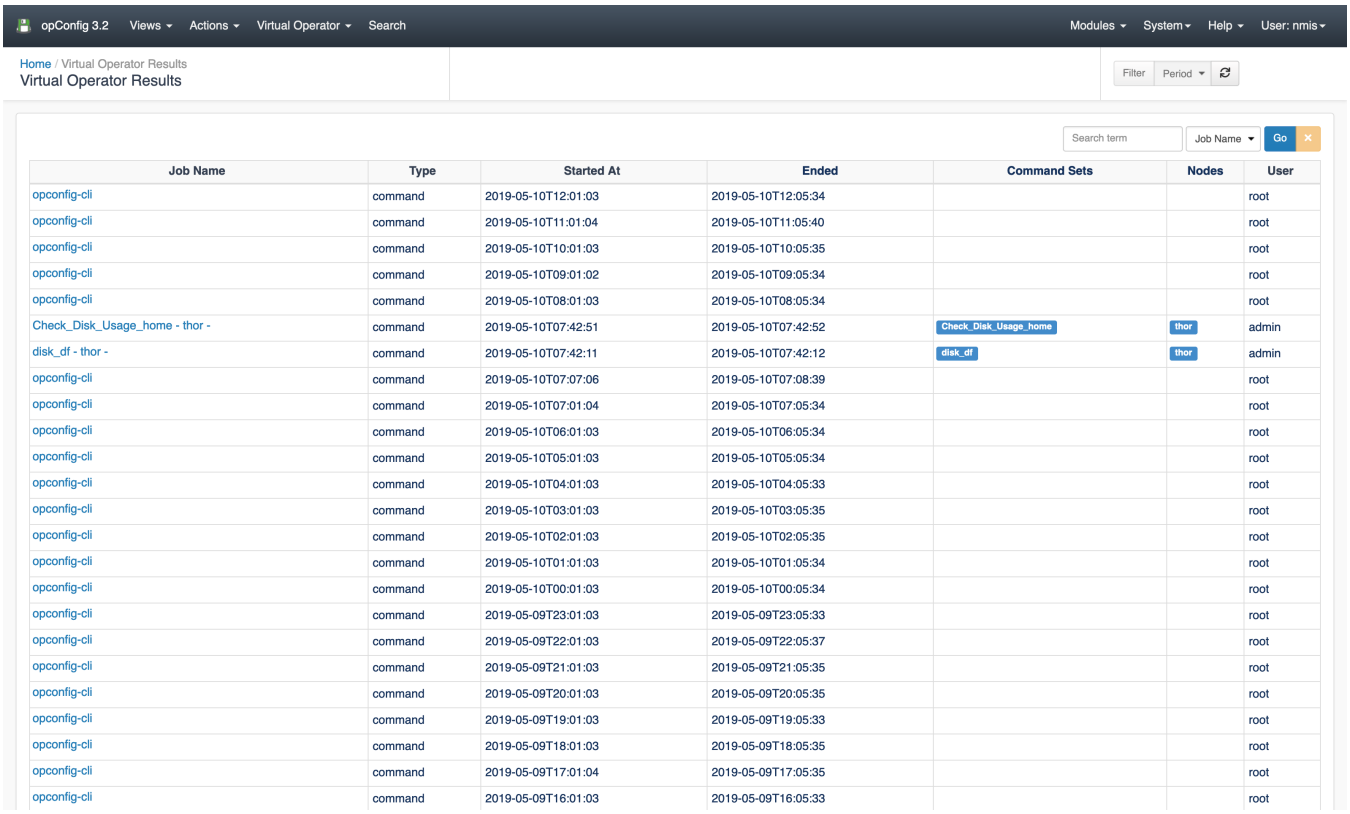
Creating a new linked job from a report which has conditions.

Virtual Operator Results & Schedules

Under troubleshooting there are two pages which have scheduled virtual operator jobs and completed virtual operator jobs.

Scheduled shows user created running jobs and ones which are scheduled in the future.

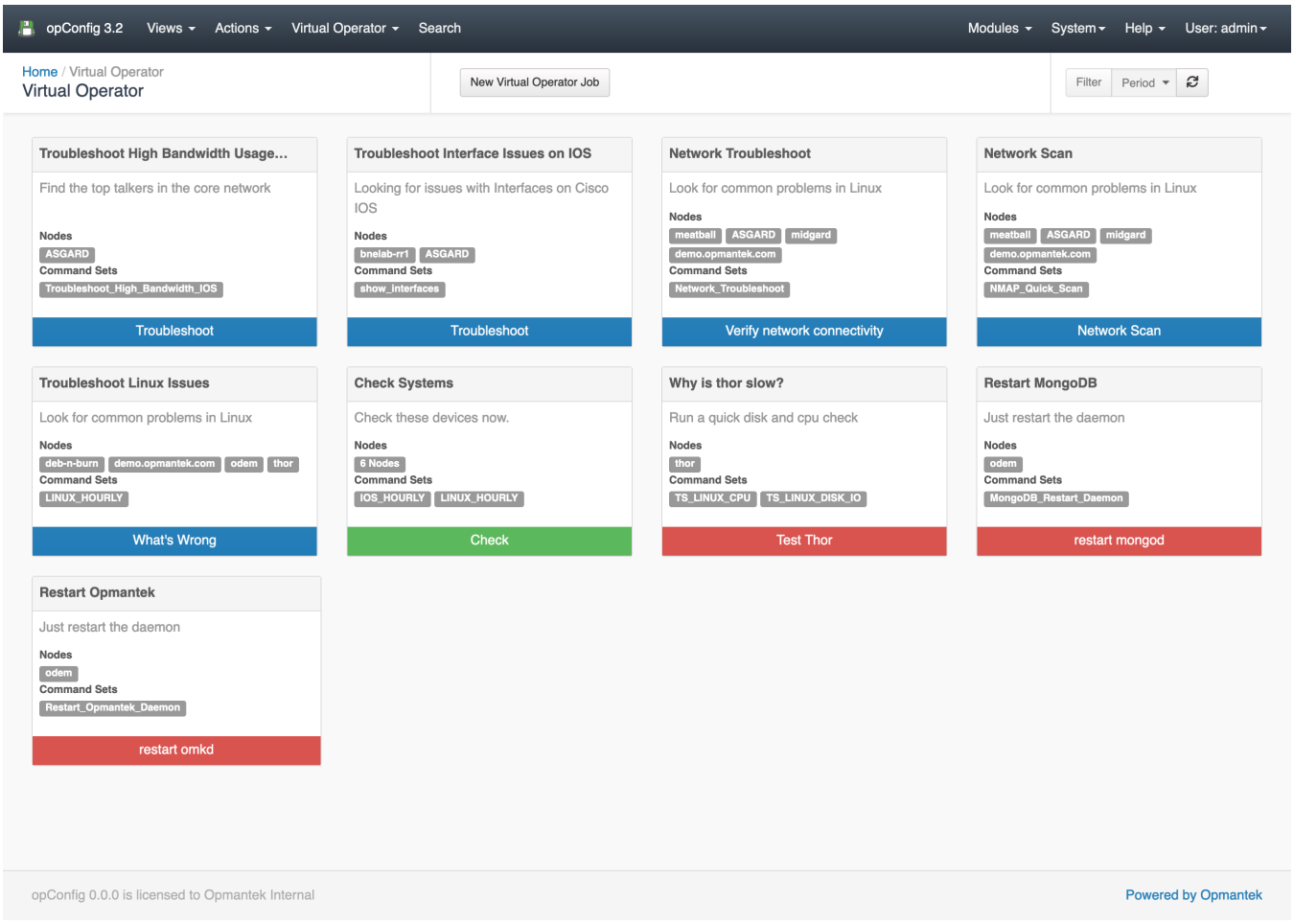
Results shows all the completed jobs which were user created and cli run.



Job Name	Type	Started At	Ended	Command Sets	Nodes	User
opconfig-cli	command	2019-05-10T12:01:03	2019-05-10T12:05:34			root
opconfig-cli	command	2019-05-10T11:01:04	2019-05-10T11:05:40			root
opconfig-cli	command	2019-05-10T10:01:03	2019-05-10T10:05:35			root
opconfig-cli	command	2019-05-10T09:01:02	2019-05-10T09:05:34			root
opconfig-cli	command	2019-05-10T08:01:03	2019-05-10T08:05:34			root
Check_Disk_Usage_home - thor -	command	2019-05-10T07:42:51	2019-05-10T07:42:52	Check_Disk_Usage_home	thor	admin
disk_df - thor -	command	2019-05-10T07:42:11	2019-05-10T07:42:12	disk_df	thor	admin
opconfig-cli	command	2019-05-10T07:07:06	2019-05-10T07:08:39			root
opconfig-cli	command	2019-05-10T07:01:04	2019-05-10T07:05:34			root
opconfig-cli	command	2019-05-10T06:01:03	2019-05-10T06:05:34			root
opconfig-cli	command	2019-05-10T05:01:03	2019-05-10T05:05:34			root
opconfig-cli	command	2019-05-10T04:01:03	2019-05-10T04:05:33			root
opconfig-cli	command	2019-05-10T03:01:03	2019-05-10T03:05:35			root
opconfig-cli	command	2019-05-10T02:01:03	2019-05-10T02:05:35			root
opconfig-cli	command	2019-05-10T01:01:03	2019-05-10T01:05:34			root
opconfig-cli	command	2019-05-10T00:01:03	2019-05-10T00:05:34			root
opconfig-cli	command	2019-05-09T23:01:03	2019-05-09T23:05:33			root
opconfig-cli	command	2019-05-09T22:01:03	2019-05-09T22:05:37			root
opconfig-cli	command	2019-05-09T21:01:03	2019-05-09T21:05:35			root
opconfig-cli	command	2019-05-09T20:01:03	2019-05-09T20:05:35			root
opconfig-cli	command	2019-05-09T19:01:03	2019-05-09T19:05:33			root
opconfig-cli	command	2019-05-09T18:01:03	2019-05-09T18:05:35			root
opconfig-cli	command	2019-05-09T17:01:04	2019-05-09T17:05:35			root
opconfig-cli	command	2019-05-09T16:01:03	2019-05-09T16:05:33			root

Results from user created virtual operator jobs and opConfig jobs created from the command line.

Quick Actions



Quick Actions are templates for new Virtual Operator jobs, we have shipped four sample jobs but you can create your own. The default four jobs are defined in the file: /usr/local/omk/lib/json/opConfig/table_schemas/opConfig_action-elements.json

Clicking the quick action button on the Virtual Operate screen will take you to the New Virtual Operator Job screen and fill out the specified fields.

You can create your own Quick Actions by copying the /usr/local/omk/lib/json/opConfig/table_schemas/opConfig_action-elements.json file to /usr/local/omk /conf/table_schemas/ (you may need to create the table_schemas folder if it does not exist) and then editing it. Available field options are described below.

```
{
   "name": "IOS Hourly Collection",
   "description": "Hourly baseline collection for Cisco IOS.",
   "command_sets": ["IOS_DAILY"],
   "buttonLabel": "Collect Now",
   "buttonClass": "btn-primary"
 }
```

Key	Datatype	About
name	string	Name which is shown at the top of the quick action element
description	string	Text shown under the quick action name, useful to describe what the action does
command_sets	array of strings	Command set keys which you wish to be run
nodes	array of strings	Names of nodes which you wish the command sets to be run against
buttonLabel	string	Text of the run button

buttonClass	string	CSS class applied to the button to colour it. • btn-default • btn-primary (default, blue) • btn-success (green) • btn-warning (orange) • btn-danger (red)