Contents

[1. Power management driver 2](#_Toc101260924)

[1.1. General information 2](#_Toc101260925)

[1.2. Functionality 2](#_Toc101260926)

[1.3. Velocity updated properties 4](#_Toc101260927)

[2. Power management visualization 6](#_Toc101260928)

[2.1. General information 6](#_Toc101260929)

[2.2. Functionality 7](#_Toc101260930)

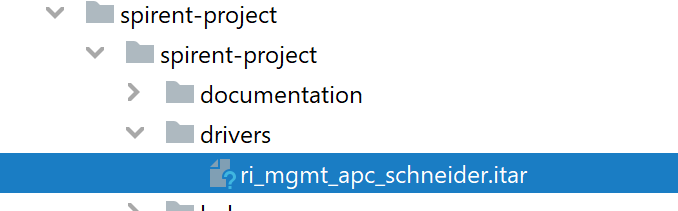
[2.3. OpenSearch Dashboards 8](#_Toc101260931)

# Power management driver

## General information

Driver functionality has been developed and tested on Schneider AP8853 model. It is possible that driver works on other Schneider PDU models. It all depends on what commands are used to extract power information and how the output of the commands looks like.

Driver file is committed on Git *drivers* directory:



## Functionality

Driver script is connecting via SSH to the PDU device and extracts the following information:

* PDU Model
* PDU Name
* PDU Firmware version
* PDU Reported Power
* Banks current (which are assimilated to ports in the JSON below).

SSH needs to be enabled on the device and the correct username / password / ipAdrress or Hostname properties need to be set on the PDU resource page in Velocity.

The driver returns the following JSON to Velocity. Velocity is parsing it and it is updating the device properties and ports properties.

{

  "properties": {

    "Model": "AP8853",

    "Name": "apc44E8C1",

    "Firmware Version": "6.8.2",

    "PDU Reported Power": "220"

  },

  "ports": [

    {

      "name": "1",

      "status": "online",

      "properties": {

        "Current": "1.0"

      }

    },

    {

      "name": "2",

      "status": "online",

      "properties": {

        "Current": "0.0"

      }

    }

  ]

}

The following device commands are used to extract device data:

root@172:~# ssh -c aes256-cbc apc@172.16.251.206

apc@172.16.251.206's password:

Schneider Electric Network Management Card AOS v6.8.2

(c) Copyright 2019 All Rights Reserved RPDU 2g APP v6.8.0

-------------------------------------------------------------------------------

Name : apc44E8C1 Date : 10/21/2021

Contact : Unknown Time : 18:25:36

Location : Unknown User : Super User

Up Time : 1 Day 23 Hours 51 Minutes Stat : P+ N4+ N6+ A+

-------------------------------------------------------------------------------

IPv4 : Enabled IPv6 : Enabled

Ping Response : Enabled

-------------------------------------------------------------------------------

HTTP : Disabled HTTPS : Enabled

FTP : Disabled Telnet : Disabled

SSH/SCP : Enabled SNMPv1 : Disabled

SNMPv3 : Disabled

-------------------------------------------------------------------------------

Super User : Enabled RADIUS : Disabled

Administrator : Disabled Device User : Disabled

Read-Only User : Disabled Network-Only User : Disabled

apc>bkReading all

E000: Success

1: 1.0 A

2: 0.0 A

apc>prodinfo

E000: Success

RPDU ID: 1\*

AOS 6.8.2

Metered Rack PDU 6.8.0

Model: AP8853

Name: apc44E8C1

Location: Unknown

Contact: Unknown

Present Outlets: 42

Switched Outlets: 0

Metered Outlets: 0

Max Current: 32 A

Present Phases: 1

Metered Phases: 1

Metered Banks: 2

Uptime: 2 Days 1 Hour 34 Minutes

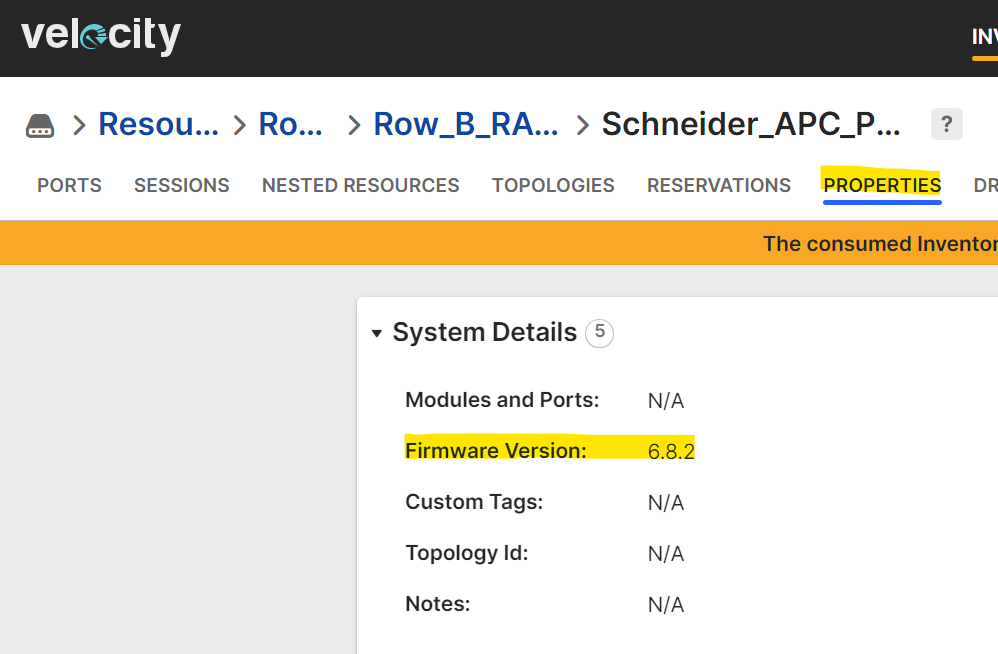
Network Link: Link Active

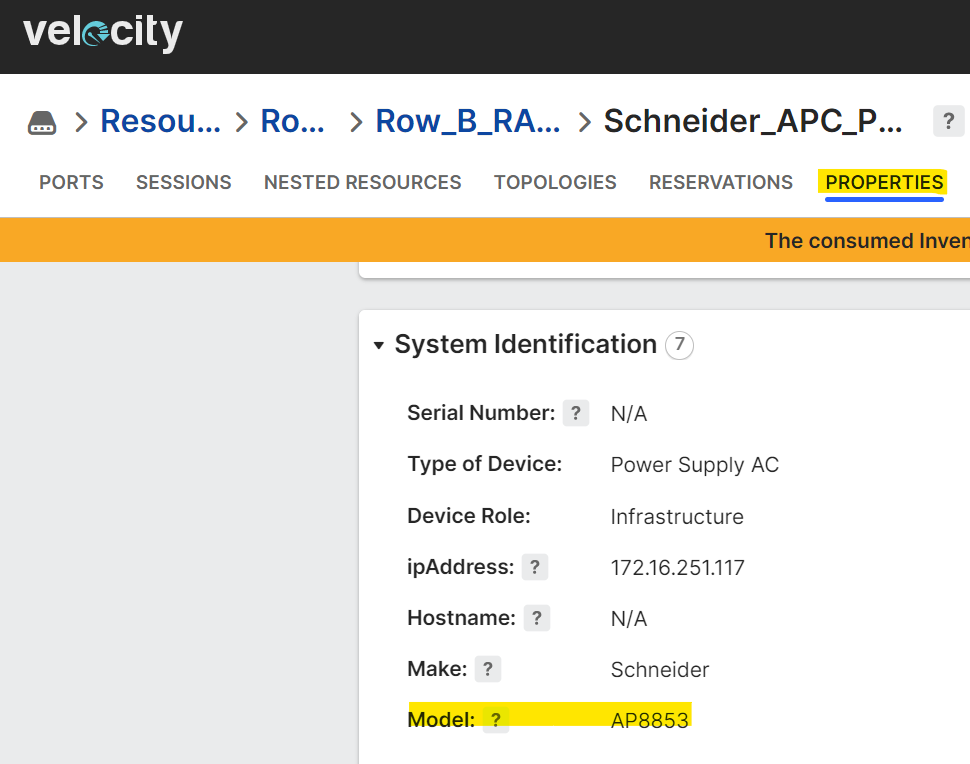
apc>

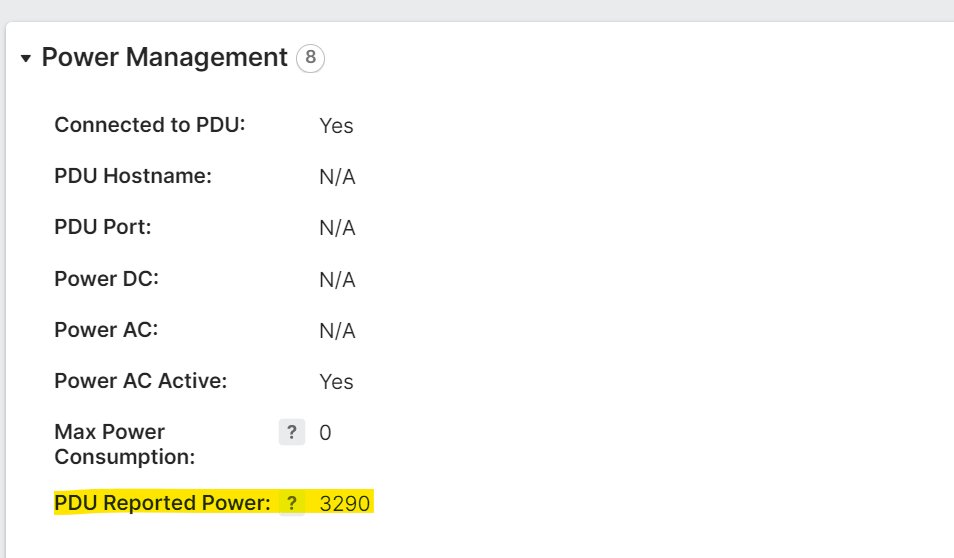
apc>phReading all power  
E000: Success  
1: 0.22 kW

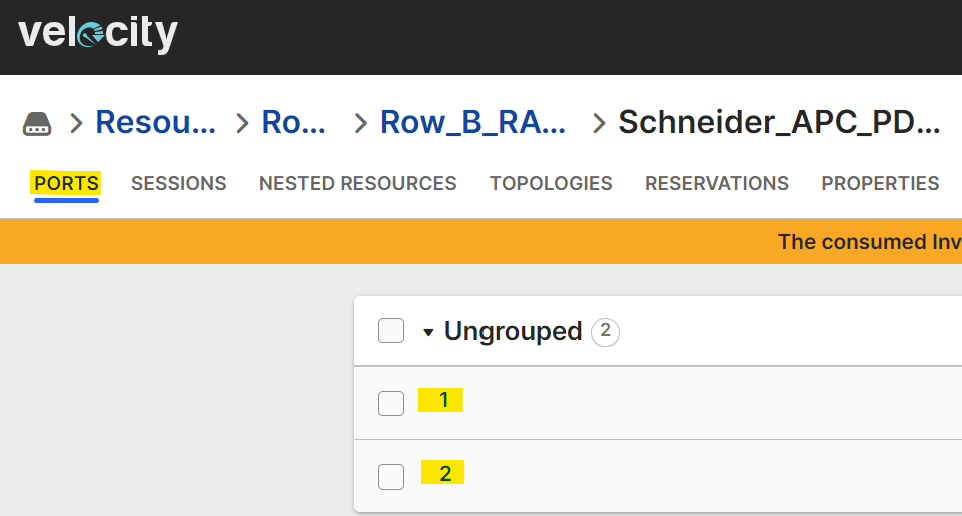
## Velocity updated properties

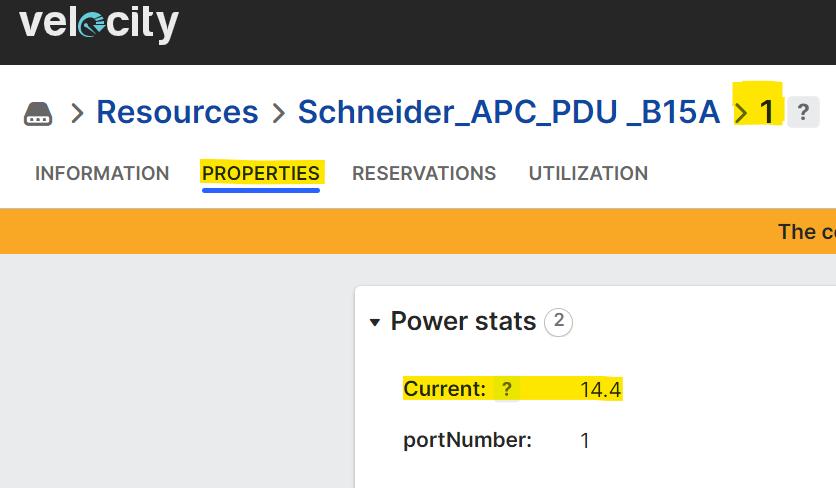
Once Velocity received the JSON from the driver, it will update the Ports and Properties page of the resource with the values received:







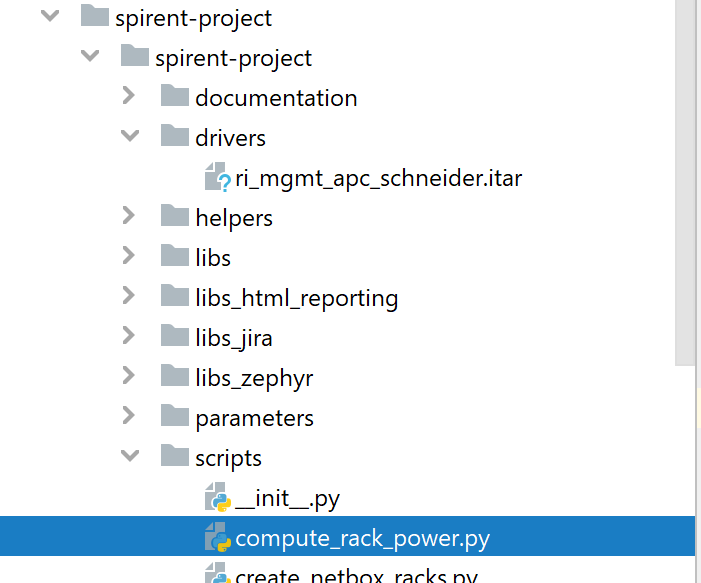




# Power management visualization

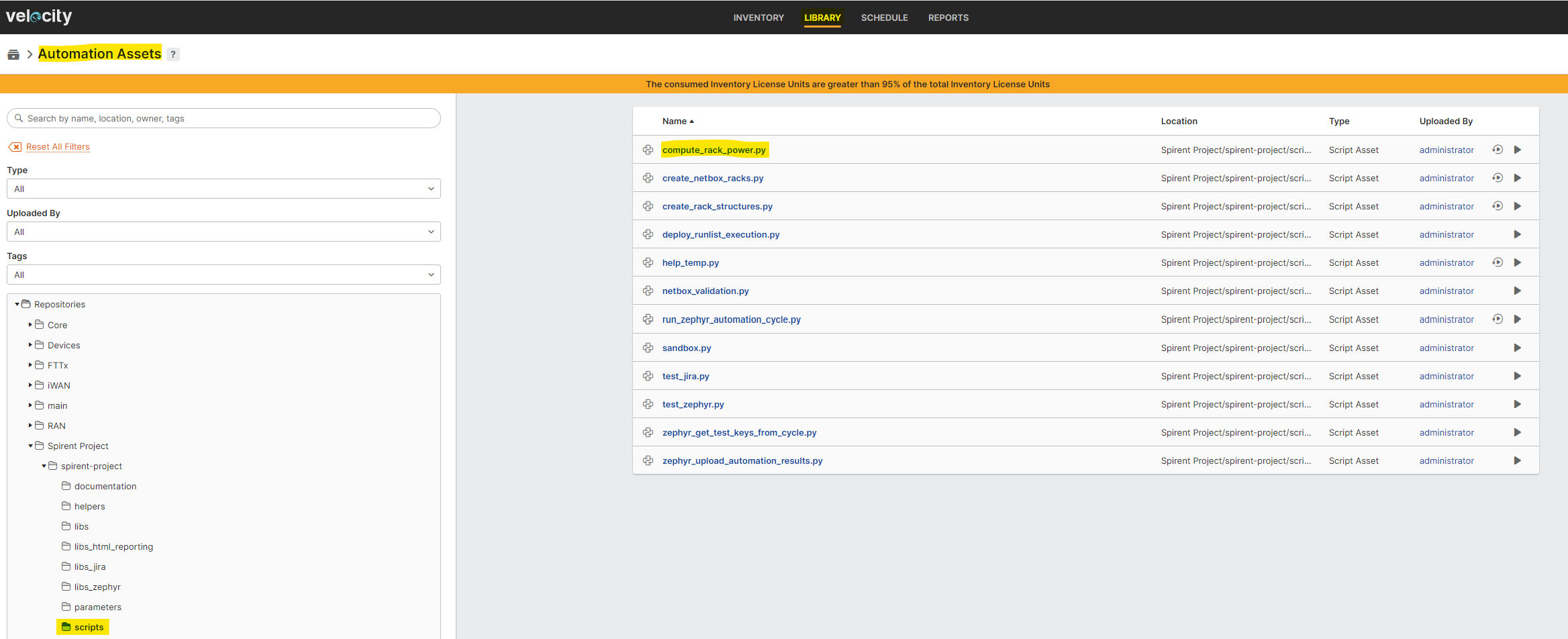
## General information

The script used for reading / writing power data is committed on Git on the *scripts* directory:

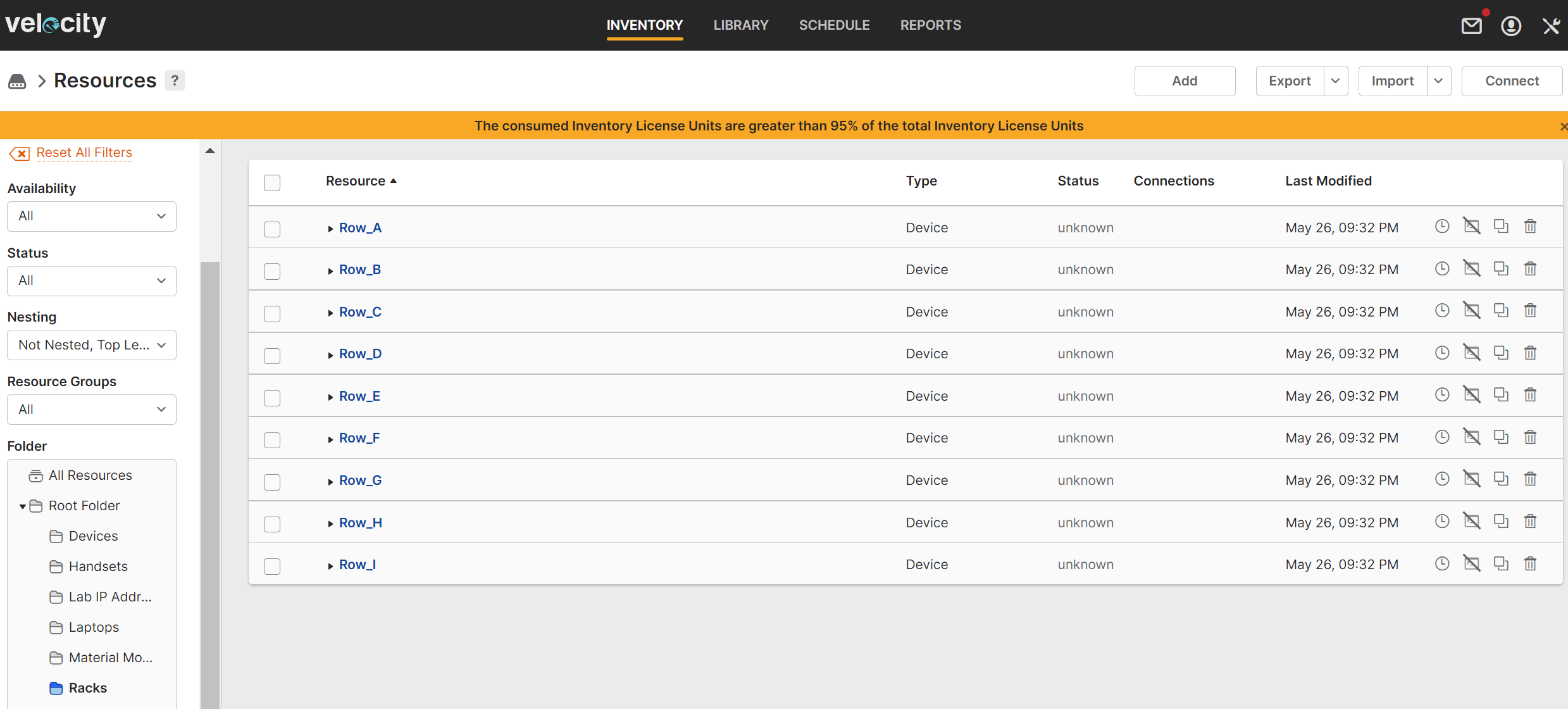


Script is run from Velocity, either scheduled or on demand. The agent that will run the script need to have Python installed and, specifically for this script, opensearchpy package.

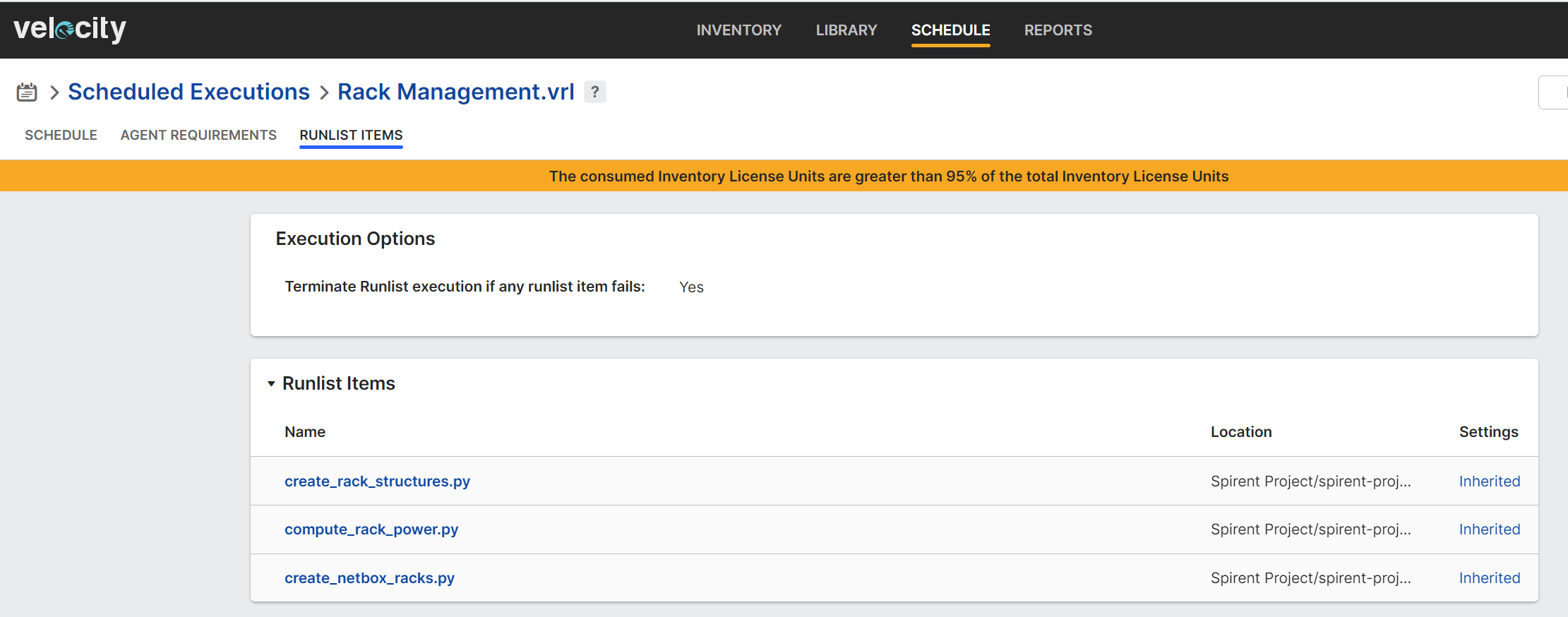
The script can be found in Velocity in Library / Automation assets and then browsing the repositories tree from the left side as in the picture below:



The power management script is looping to the rack structure that is created in the Resources tree:



This structure is created by the create\_rack\_structure.py script. Functionality of this script is described in a separate document. So, to be sure that all devices have been added in the rack structura and are taken into account for power data calculation, it is advised that the create\_rack\_structure script to be run before in a runlist such as the one in the picture below:



Currently, this runlist is scheduled to run nightly.

## Functionality

There are 3 type of power definitions used in the context of this script:

At Device level:

* Max Power Consumption

The maximum power consumed by a device. It is set by the user as a device property, measured in Watts. This information can usually be found on the device labels.

* PDU Reported Power

The power reported by a particular PDU, received from power driver after it runs, measured in Watts.

At Rack level:

* Max Installed Power

The maximum available power for a rack. It is set by the user as a rack property, measured in Watts.

* Max Power Consumption

The sum of the Maximum Power Consumption values from all devices from a rack, computed by the compute\_rack\_power.py script, measured in Watts.

* PDU Reported Power

The sum of the PDU Reported Power values from all PDUs from a rack, computed by compute\_rack\_power.py script, measured in Watts.

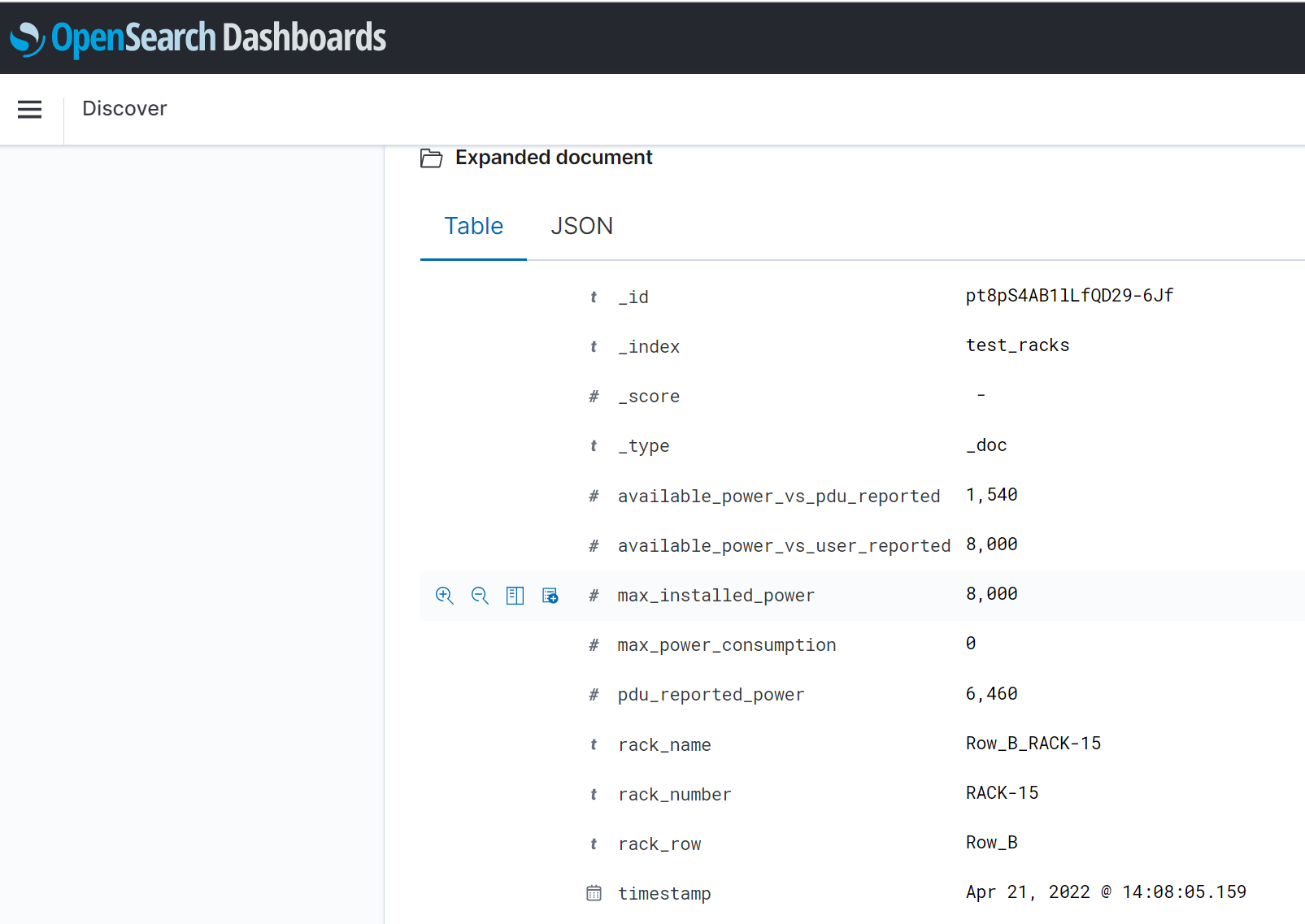
The compute\_rack\_power script functionalities are the following:

* Update Max Power Consumption at rack level for each rack in Velocity
* Update PDU Reported Power at rack level for each rack in Velocity
* Write Max Installed Power, Max Power Consumption and PDU Reported Power to OpenSearch database per each rack.
* Compute diff between Max Installed Power and Max Power Consumption / PDU Reported power to show Available Power in the Dashboards.

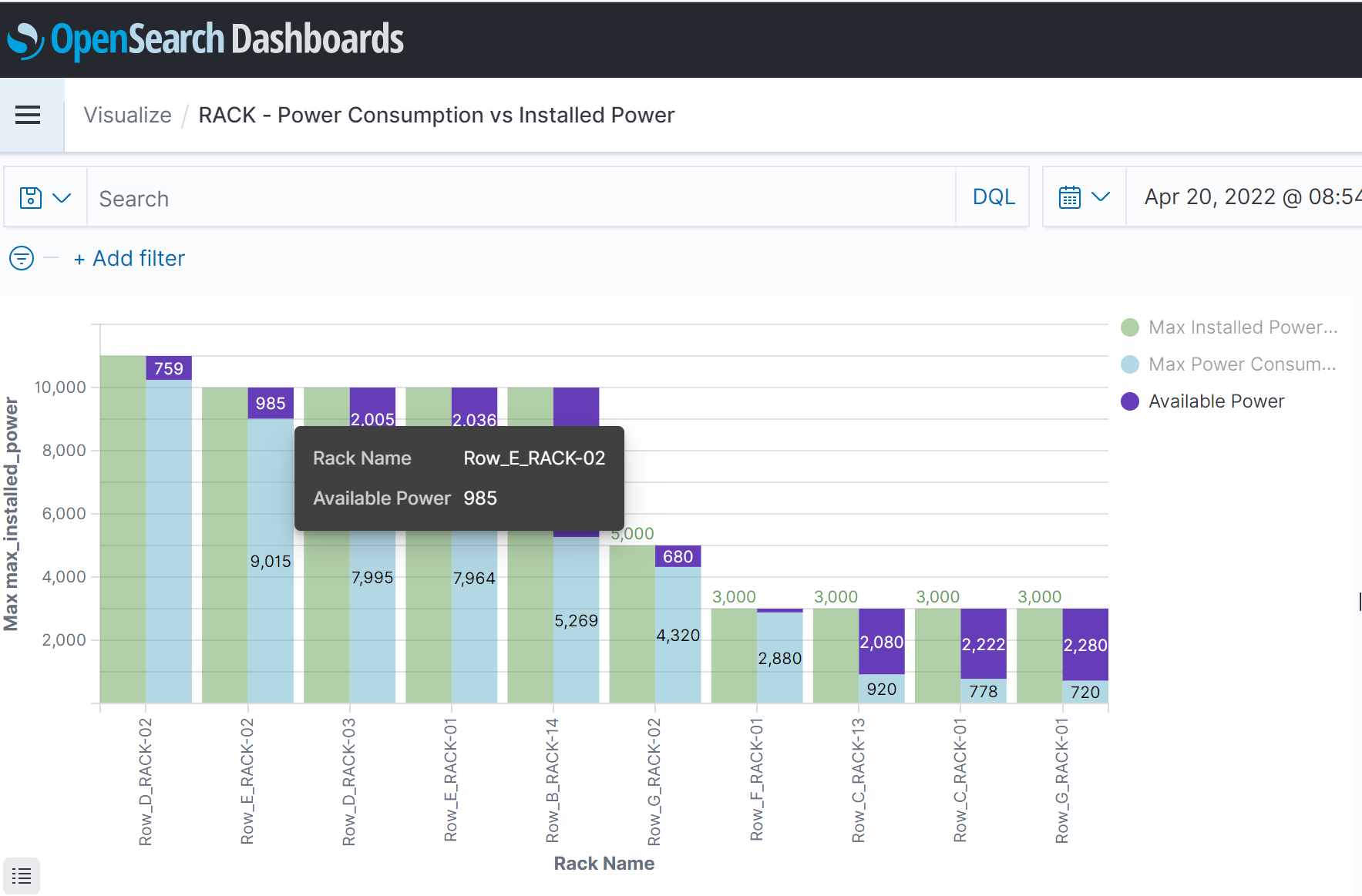
IMPORTANT NOTE: To have an accurate view on Power from lab, drivers need to run as often as possible and users have to input correct data at rack and device level.

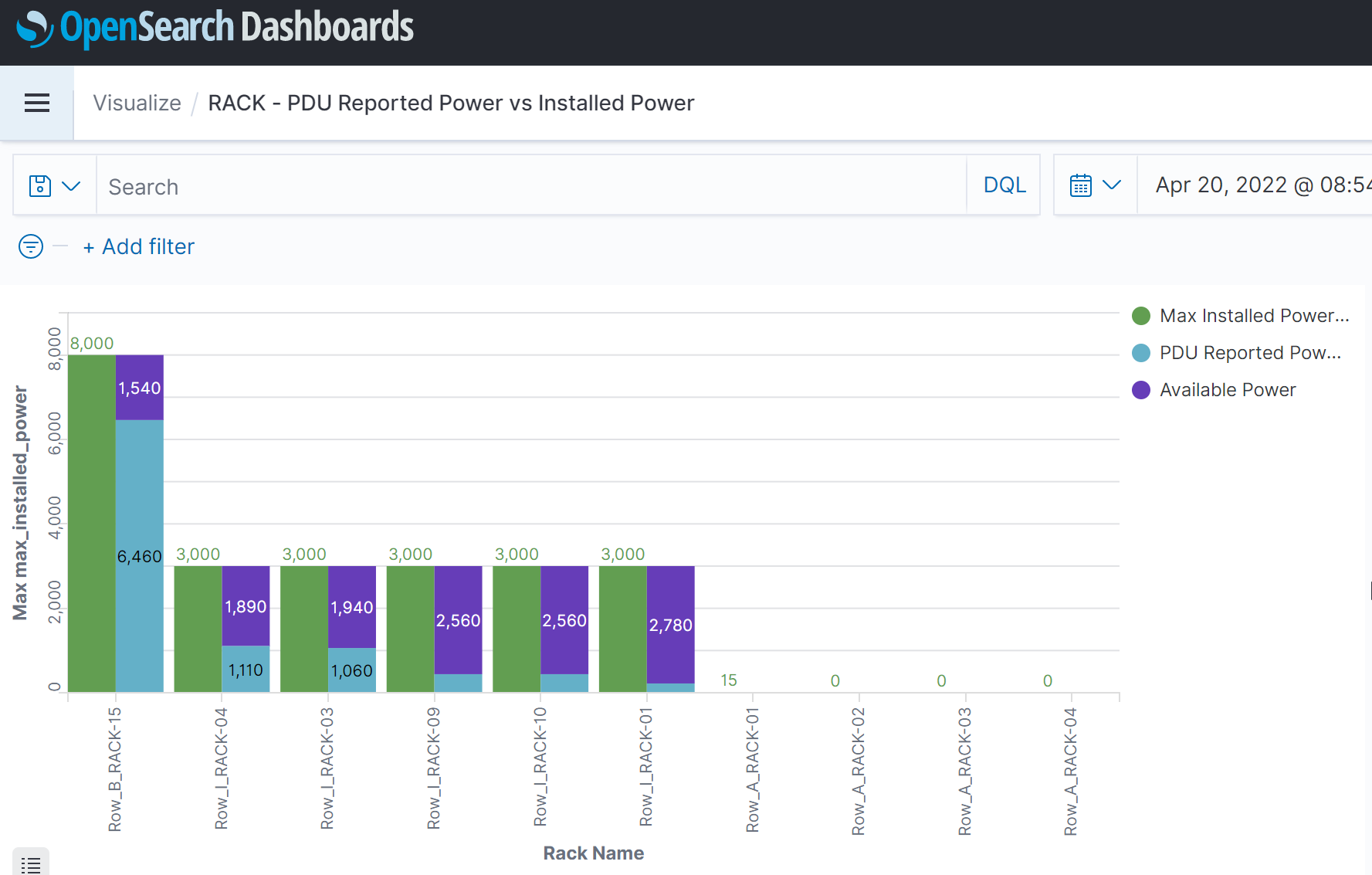
## OpenSearch Dashboards

Power information per each rack is written in OpenSearch database in the form of documents such this one:



Power data can be visualized and dashboarded depending what brings more value to the user. For example, below are 2 visualizations:





The visualizations can be added to a dashboard. Below is an example:

