Central Management Server Setup

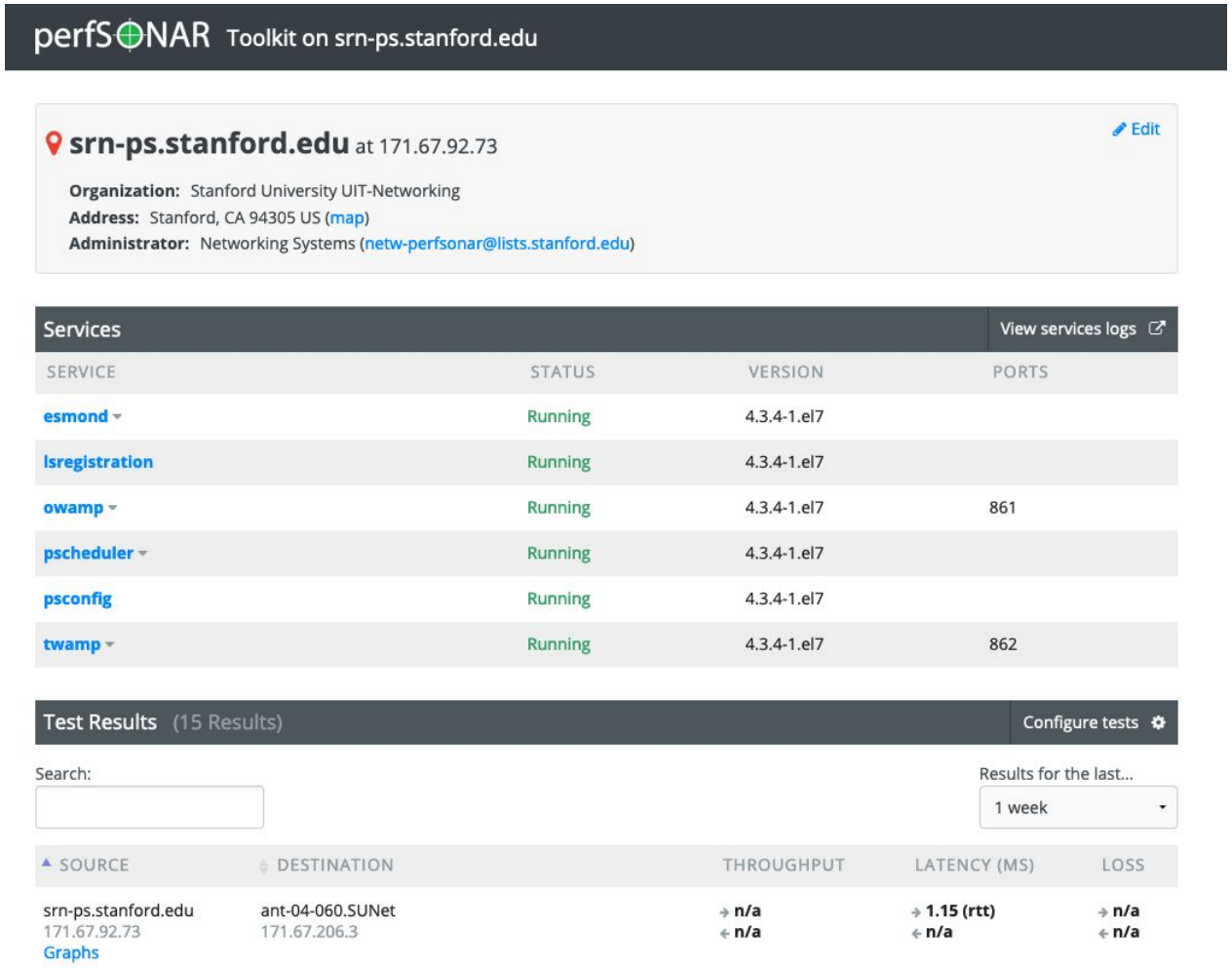
perfSONAR Sampler

2nd European perfSONAR User Workshop

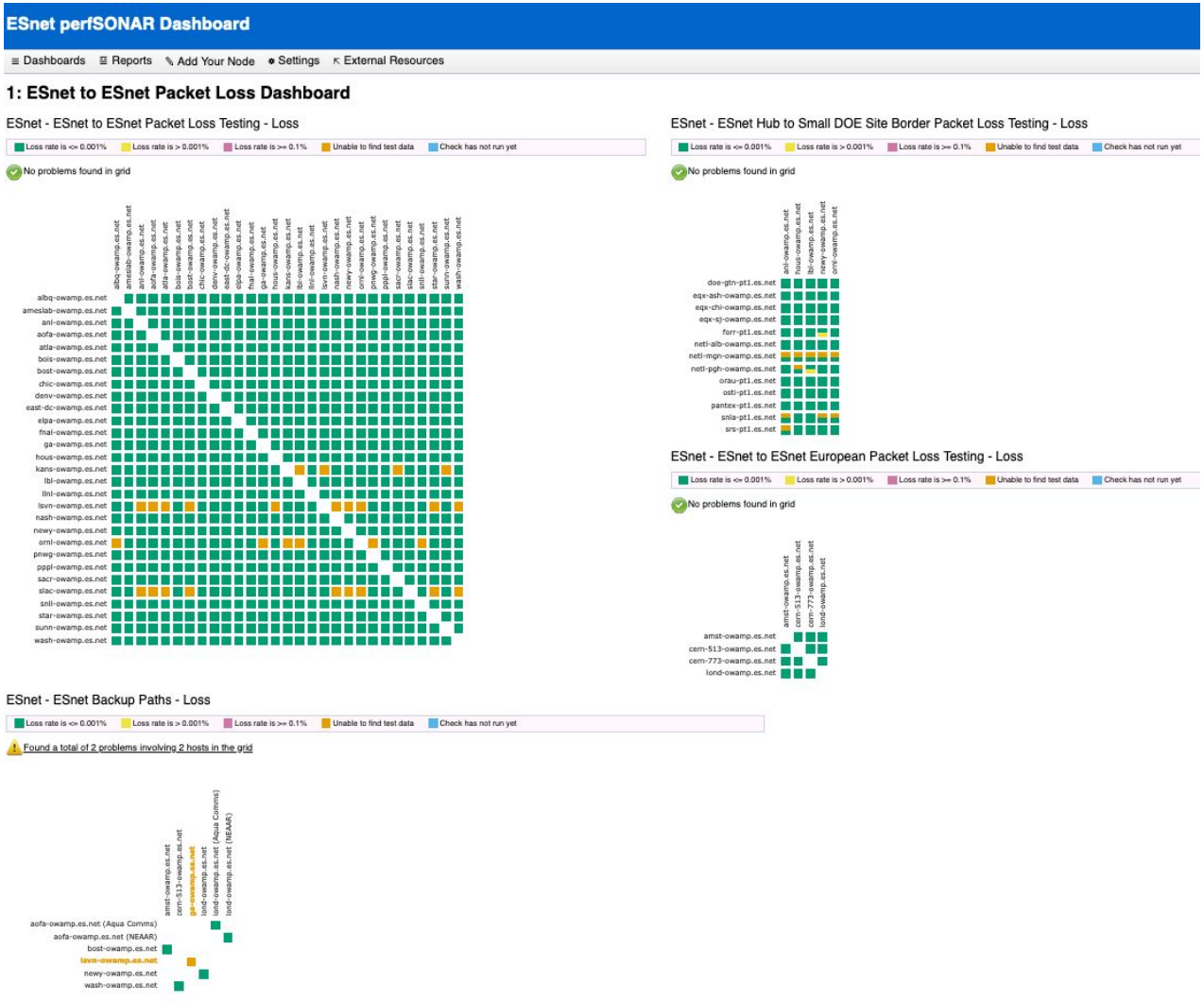
April 14 & 15, 2021, Online

Presenter: Phil Reese, preese@stanford.edu

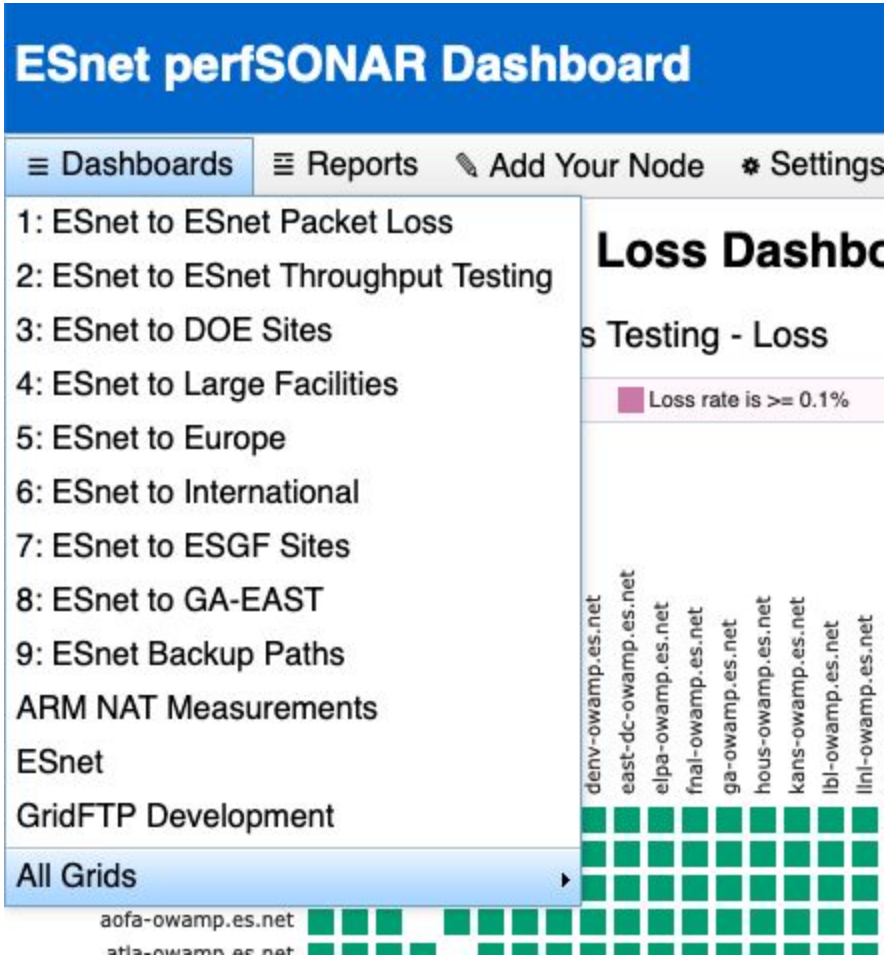
How most of us are introduced to perfSONAR



What is often the next view of perfSONAR



While the image is daunting, this makes things look even more complex



Way too many dashboards and sights to take in.

How to move from the 1x1 perfSONAR to the ESnet view and multiple dashboards?

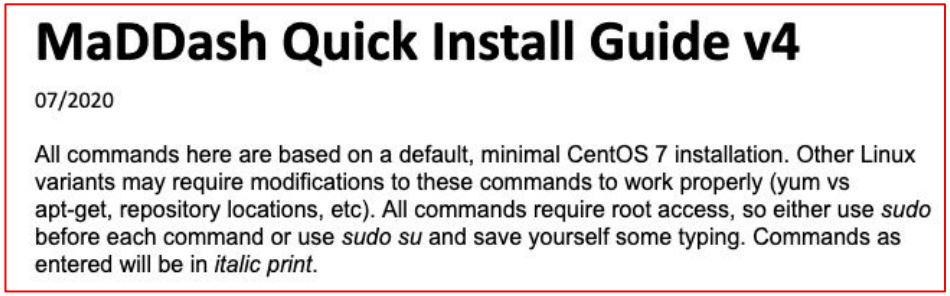
The ever popular docs.perfsonar.net web site has all the answers!

It actually does, but it is a long read to move from that that simple 'all in one' Toolkit, to a multi-dashboard MaDDash grid web site.

The perfSONAR team realized this and a few years ago Andy Lake pulled together a 2-3 page readable document for how to set up a Central Management web site and how to get the agents testing and reporting results to the Central Management system.

This is included in the main docs.perfsonar.net table of contents in one place only, the main page, a link from "MaDDash Quick Install Guide**".**

The link opens a Google Doc file and has a step by step process for how to get a MaDDash system going.



MaDDash Quick Install Guide v4

07/2020

All commands here are based on a default, minimal CentOS 7 installation. Other Linux variants may require modifications to these commands to work properly (yum vs apt-get, repository locations, etc). All commands require root access, so either use *sudo* before each command or use *sudo su* and save yourself some typing. Commands as entered will be in *italic print*.

What to expect from this presentation

The first paste of commands into the terminal window is a quick run through of all the commands discussed at a github project site:

https://github.com/preese/perfSONAR-Sampler

The Github project shows an implementation of Andy's document using a set of VMs and some tools to quickly bring all the pieces together.

If all goes well, the pasted commands will build 6 edge nodes and one MaDDash server, it will show a mesh dashboard and a disjoint dashboard. All of this will take place on a single NUC and should complete in ~15 minutes.

We'll talk about the JSON files and other features of the project while it installs in the background.

Hardware and IP chart used

Chart for perfSONAR-Sampler

Second host interface name: enp0s20f0u2

Address segment for the IPs in use: 192.168.1.0/24

Name	IP#	Mac Addr
ps1	192.168.1.210	52:54:00:73:6f:41
ps2	192.168.1.211	52:54:00:73:6f:42
ps3	192.168.1.212	52:54:00:73:6f:43
md	192.168.1.213	52:54:00:73:6f:44
dj1	192.168.1.214	52:54:00:73:6f:45
dj2	192.168.1.215	52:54:00:73:6f:46
dj3	192.168.1.216	52:54:00:73:6f:47

When you make changes to the above chart, plan to make similar edits to:

top section of **Vagrantfile**

top sections of both **mesh.json** and **disjoint.json**

the **hosts** file

Then the IP for that MaDDash server is used in several files.

Testpoint-MaDDashbuild.yml 2 instances of 192.168.1.213 and one use of the address segment

Maddash-dj.yml 2 instances of 192.168.1.213

More on the IP chart used

Chart for perfSONAR-Sampler

Second host interface name: enp0s20f0u2

Address segment for the IPs in use: 192.168.1.0/24

Name	IP#	Mac Addr
ps1	192.168.1.210	52:54:00:73:6f:41
ps2	192.168.1.211	52:54:00:73:6f:42
ps3	192.168.1.212	52:54:00:73:6f:43
md	192.168.1.213	52:54:00:73:6f:44
dj1	192.168.1.214	52:54:00:73:6f:45
dj2	192.168.1.215	52:54:00:73:6f:46
dj3	192.168.1.216	52:54:00:73:6f:47

When you make changes to the above chart, plan to make similar edits to:

top section of **Vagrantfile**

top sections of both **mesh.json** and **disjoint.json**

the **hosts** file

Then the IP for that MaDDash server is used in several files.

Testpoint-MaDDashbuild.yml 2 instances of 192.168.1.213 and one use of the address segment

Maddash-dj.yml 2 instances of 192.168.1.213

Review of the JSON files

~/Documents/PS-Talk/mesh.json (no function selected)

```
{
    "_meta": {
        "display-name": "1 PSproj"
    },

    "archives": {
        "Home_esmond_archive": {
            "archiver": "esmond",
            "data": {
                "measurement-agent": "{% scheduled_by_address %}",
                "url": "http://192.168.1.213/esmond/perfsonar/archive/"
            }
        }
     },

    "addresses": {
        "ps1": { "address": "192.168.1.210" },
        "ps2": { "address": "192.168.1.211" },
        "ps3": { "address": "192.168.1.212" }
    },

    "groups": {
        "Home_group_mesh": {
            "type": "mesh",
            "addresses": [
                { "name": "ps1" },
                { "name": "ps2" },
                { "name": "ps3" }
            ]
        }
     },
    "tests": {
        "test_throughput": {
            "type": "throughput",
            "spec": {
                "source": "{% address[0] %}",
                "dest": "{% address[1] %}",
                "duration": "PT20S"
            }
        },
        "test_rtt": {
            "type": "rtt",
            "spec": {
                "source": "{% address[0] %}",
                "dest": "{% address[1] %}",
                "count": 10,
                "interval": "PT1S",
                "length": 1000
            }
        },
        "test_latencybg": {
            "type": "latencybg",
            "spec": {
                "source": "{% address[0] %}",
                "dest": "{% address[1] %}",
                "flip": "{% flip %}"
            }
        },
        "test_trace": {
            "type": "trace",
            "spec": {
                "source": "{% address[0] %}",
                "dest": "{% address[1] %}"
            }
        }
    },
    "schedules": {
        "Home_schedule_PT2H": {
            "repeat": "PT2H",
            "sliprand": true,
```

~/Documents/PS-Talk/disjoint.json (no function selected)

```
{
    "_meta": {
        "display-name": "2 DJ"
    },

    "archives": {
        "Home_esmond_archive": {
            "archiver": "esmond",
            "data": {
                "measurement-agent": "{% scheduled_by_address %}",
                "url": "http://192.168.1.213/esmond/perfsonar/archive/"
            }
        }
     },

    "addresses": {
        "ps1": { "address": "192.168.1.210" },
        "ps2": { "address": "192.168.1.211" },
        "ps3": { "address": "192.168.1.212" },
        "dj1": { "address": "192.168.1.214" },
        "dj2": { "address": "192.168.1.215" },
        "dj3": { "address": "192.168.1.216" }

    },

    "groups": {
         "Home_group_disjoint": {
            "type": "disjoint",
            "a-addresses": [
                    { "name": "dj1" },
                    { "name": "ps1" }
                ],
             "b-addresses": [
                    { "name": "ps2" },
                    { "name": "ps3" },
                    { "name": "dj2" },
                    { "name": "dj3" }
                ]
        }
     },

    "tests": {
        "test_throughput": {
            "type": "throughput",
            "spec": {
                "source": "{% address[0] %}",
                "dest": "{% address[1] %}",
                "duration": "PT20S"
            }
        },
        "test_rtt": {
            "type": "rtt",
            "spec": {
                "source": "{% address[0] %}",
                "dest": "{% address[1] %}",
                "count": 10,
                "interval": "PT1S",
                "length": 1000
            }
        },
        "test_latencybg": {
            "type": "latencybg",
            "spec": {
                "source": "{% address[0] %}",
                "dest": "{% address[1] %}",
                "flip": "{% flip %}"
            }
        },
        "test_trace": {
            "type": "trace",
```

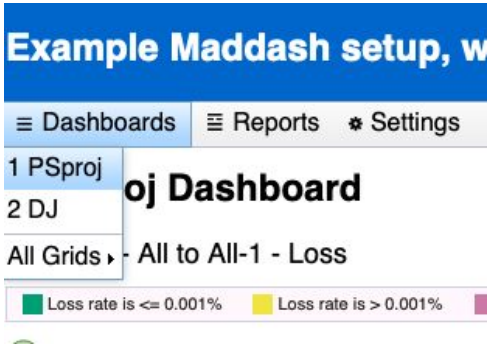
Review of some project files, Vagrantfile

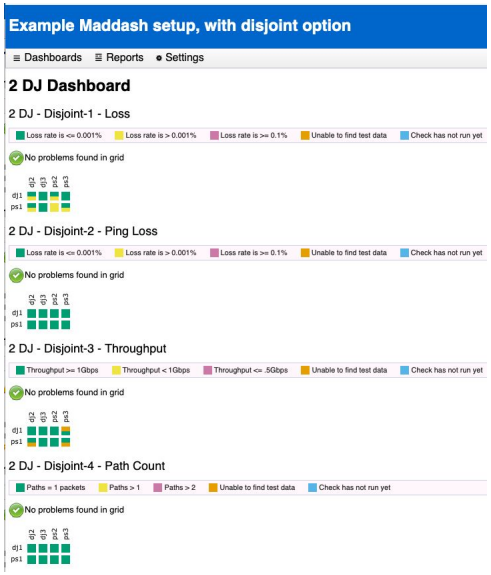
```
# -*- mode: ruby -*-
# vi: set ft=ruby :

nodes = [
  {:hostname => 'ps1', :ip => "192.168.1.210", :mac => "52:54:00:83:6f:41", :dev => "enp0s20f0u2" },
  {:hostname => 'ps2', :ip => "192.168.1.211", :mac => "52:54:00:83:6f:42", :dev => "enp0s20f0u2" },
  {:hostname => 'ps3', :ip => "192.168.1.212", :mac => "52:54:00:83:6f:43", :dev => "enp0s20f0u2" },
  {:hostname => 'md', :ip => "192.168.1.213", :mac => "52:54:00:83:6f:44", :dev => "enp0s20f0u2" },
  {:hostname => 'dj1', :ip => "192.168.1.214", :mac => "52:54:00:83:6f:45", :dev => "enp0s20f0u2" },
  {:hostname => 'dj2', :ip => "192.168.1.215", :mac => "52:54:00:83:6f:46", :dev => "enp0s20f0u2" },
  {:hostname => 'dj3', :ip => "192.168.1.216", :mac => "52:54:00:83:6f:47", :dev => "enp0s20f0u2" }
]

Vagrant.configure("2") do |config|
   nodes.each do |node|
    config.vm.define node[:hostname] do |config|
      config.vm.box = "centos/7"
      config.vm.network :public_network, :dev => node[:dev], :mode => "bridge", :mac => node[:mac], :ip => node[:ip]
    config.vm.provision "file", source: "/home/vagrant/.ssh/id_ed25519.pub", destination: "/home/vagrant/.ssh/authorized_keys"
    config.vm.hostname = node[:hostname]
      config.vm.provider :libvirt do |domain|
          domain.memory = 2048
          domain.cpus = 1
          domain.storage_pool_name = "default"
        end
      end
    end
end
```

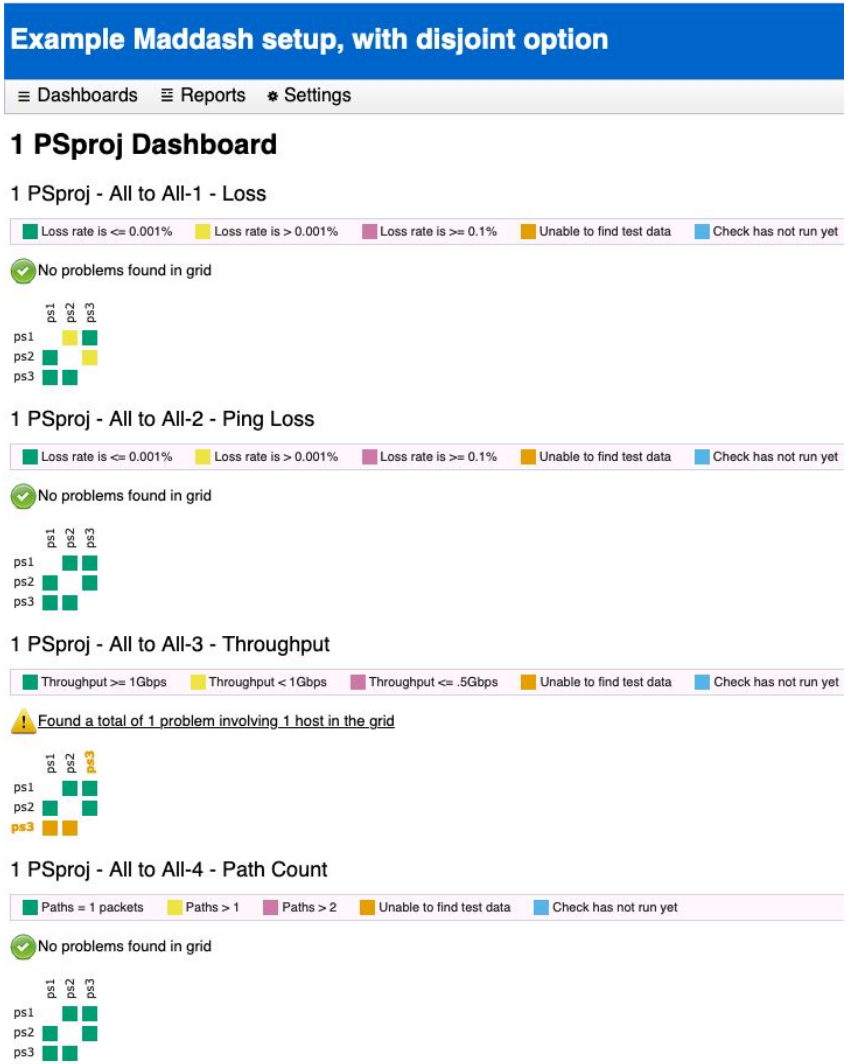
Review of some project files, Ansible

```
---

- hosts: all
  become: yes
  tasks:
    - name: Install EPEL&PS repo to all VMs
      yum:
        name:
          - epel-release
          - http://software.internet2.edu/rpms/el7/x86_64/latest/packages/perfSONAR-repo-0
        state: present

- hosts: ps,dj
  gather_facts: False
  become: yes
  tasks:
    - name: For ps & dj VMs, add perfsonar rpms to all testpoint edge nodes
      yum:
        name:
          - perfsonar-testpoint
        state: present

    - name: For ps & dj VMs, install remote URL and archives, even before the .json is pub
      command: 'psconfig remote  add --configure-archives "http://192.168.1.213/psconfig/m

    - name: For ps & dj VMs, restart psconfig-pscheduler-agent
      service:
        name: psconfig-pscheduler-agent
        state: restarted

- hosts: md
  gather_facts: False
  become: yes
  tasks:
    - name: For md VM and remaining lines, add perfsonar-centralmanagment rpm to MaDDash S
      yum:
        name:
          - perfsonar-centralmanagement
        state: present

    - name: Start FW service if not running
      service:
        name: firewalld
        state: started

    - name: Enable FW service
      service:
        name: firewalld
        enabled: yes

    - name: Add http to firewall
      firewalld:
        service: http
        permanent: yes
        immediate: yes
        state: enabled

    - name: Add https to firewall
      firewalld:
        service: https
        permanent: yes
        immediate: yes
        state: enabled
```

What we'll end up with







The unexpected...