



# perfSONAR Testpoint & Archive

*Helm-based deployment on NMaaS*



**Besiki Abuladze**

System Administrator · GRENA

*07 May 2026*

# Agenda

**01**

## Helm Charts

Testpoint & Archive Helm charts and the official perfSONAR images they wrap

**02**

## Architecture

How testpoint instances measure between each other and ship results to the archive

**03**

## Deployment on NMaaS

Current Testpoint and Archive instances running on `vnoc.nmaas.eu`

**04**

## pSConfig

Sending measurements from Testpoint to Archive — config and where to place it

**05**

## Live Demo

Walk through the running deployment and Grafana dashboards

# Helm Charts & Official Images



Both charts live in the perfSONAR GitHub organisation and pull the official perfSONAR images — no fork, no rebuild.

## perfsonar-testpoint

*Runs measurements (throughput, latency, traceroute)*

GITHUB REPOSITORY

**perfsonar/perfsonar-testpoint-docker**

OFFICIAL IMAGE

**perfsonar/testpoint**

**Helm chart** deploys the official image on Kubernetes.

## perfsonar-archive

*Stores results — OpenSearch + Logstash + Apache proxy*

GITHUB REPOSITORY

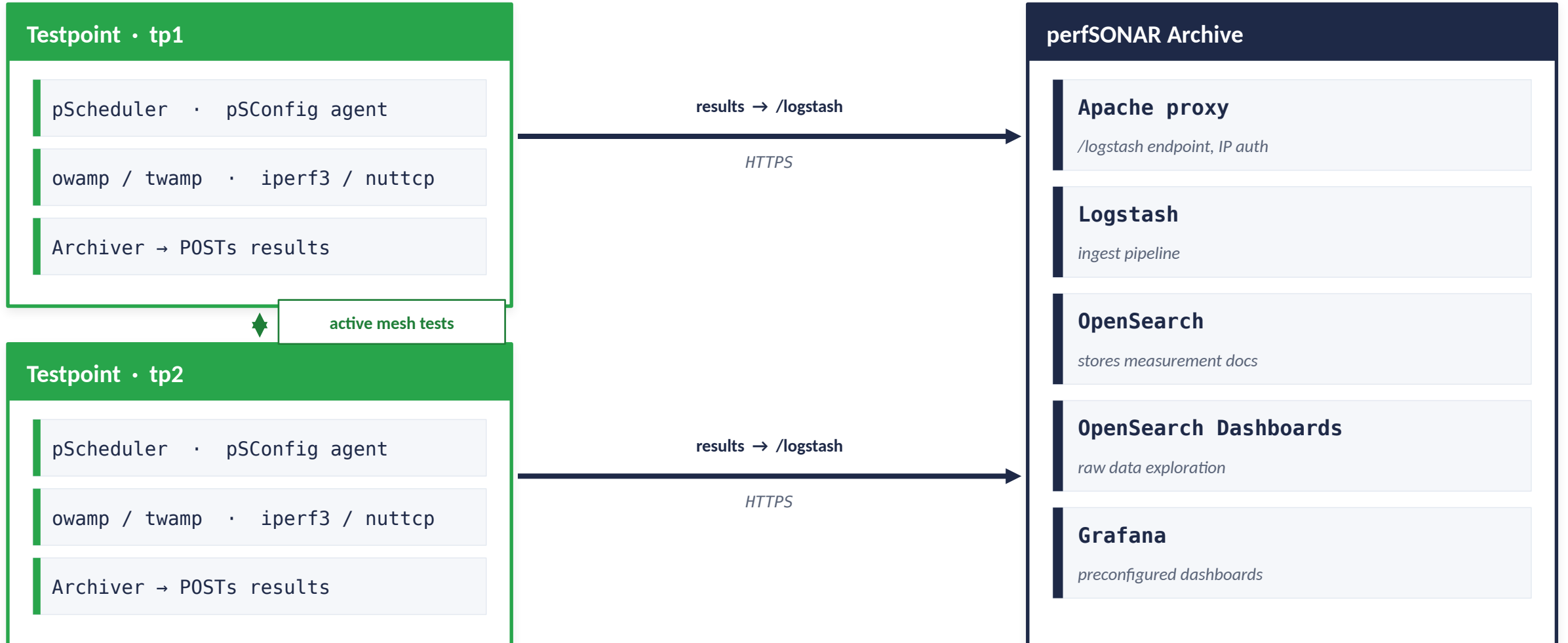
**perfsonar/perfsonar-archive-docker**

SERVICE COMPONENT IMAGES • ghcr.io

ghcr.io/perfsonar/opensearch-node  
ghcr.io/perfsonar/logstash  
ghcr.io/perfsonar/opensearch-dashboards  
ghcr.io/perfsonar/grafana

**Helm chart** orchestrates these images as one Archive deployment.

# Architecture



# Deployed on NMaaS

●  
**Test deployment** currently running on `vnoc.nmaas.eu` — *Kubernetes cluster, Helm-managed releases.*

| ARCHIVE                                                                                                                                                                                                                                                  | TESTPOINT                                                                                                                                                                                                 | TESTPOINT                                                                                                                                                                     |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>perfsonar-archive</b></p> <div> <div>IP</div> <div>Archive</div> </div> <p><b>endpoint:</b> <code>/logstash</code></p> <p>Central archive instance — receives results from all testpoints, exposes Grafana dashboards.</p> <div>Helm release</div> | <p><b>perfsonar-testpoint (tp1)</b></p> <div> <div>IP</div> <div>Testpoint1</div> </div> <p>Measurement point #1 — runs throughput, latency and traceroute tests against tp2.</p> <div>Helm release</div> | <p><b>perfsonar-testpoint (tp2)</b></p> <div> <div>IP</div> <div>Testpoint2</div> </div> <p>Measurement point #2 — peer of tp1 in the mesh group.</p> <div>Helm release</div> |

Mesh: **tp1** ↔ **tp2** · Results POSTed to: `https://archive/logstash`

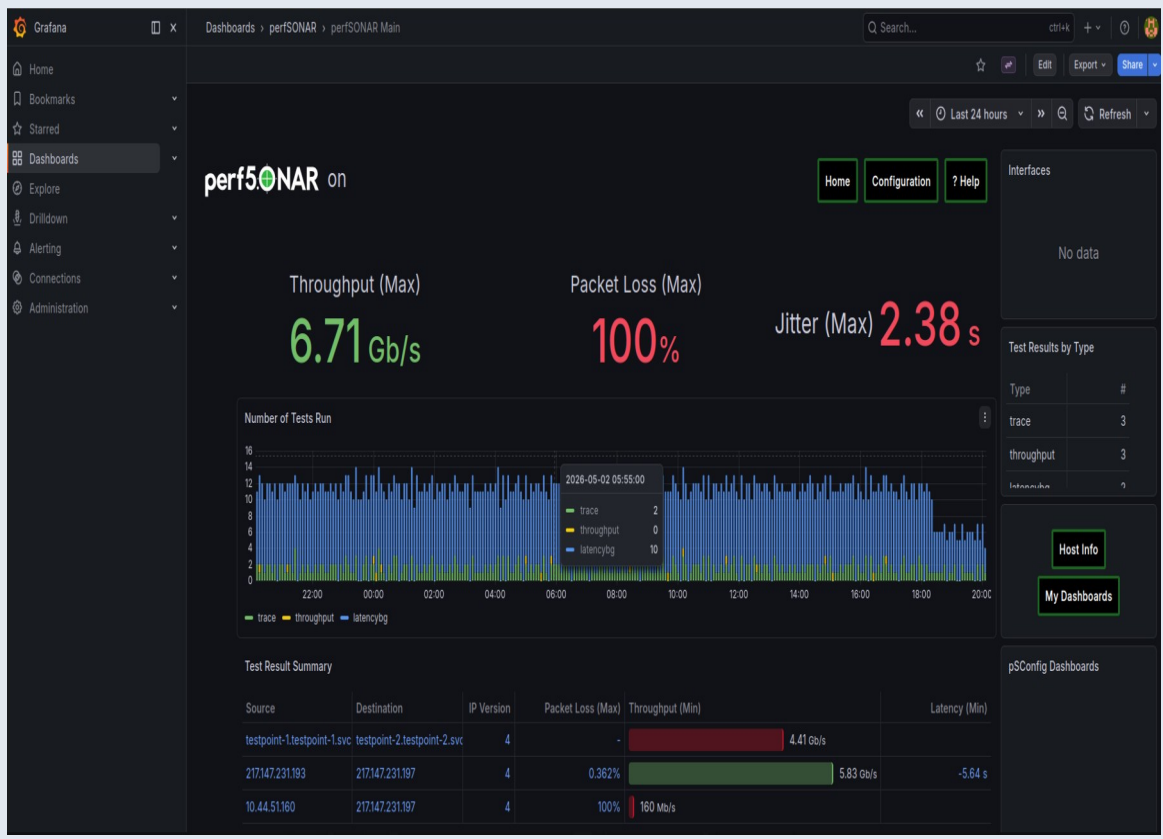
# pSConfig & Grafana Output

```
psconfig.json

{
  "archives": {
    "central_archive": {
      "archiver": "http",
      "data": {
        "_url": "https://Archive/logstash",
        "verify-ssl": false
      }
    }
  },
  "addresses": {
    "tp1": { "address": "Testpoint1" },
    "tp2": { "address": "Testpoint2" }
  },
  "groups": {
    "mesh_testpoints": {
      "type": "mesh",
      "addresses": [
        { "name": "tp1" },
        { "name": "tp2" }
      ]
    }
  }
}
```

highlighted = the only fields you change (archive URL + testpoint IPs)

**PLACE FILE AT**  
**/etc/perfsonar/psconfig/pscheduler.d/**  
 pscheduler agent picks it up automatically — no restart needed.



Grafana — pre-configured perfSONAR dashboard

## GET EXAMPLE TEMPLATE • curl

```
$ curl -o psconfig.json \
  https://raw.githubusercontent.com/perfsonar/psconfig/master/
  psconfig/perfsonar-psconfig/doc/skeleton.json
```

● LIVE

# Demo

*perfSONAR Testpoint & Archive on [vnoc.nmaas.eu](https://vnoc.nmaas.eu)*

●  
*pSConfig template walkthrough*

●  
*Test execution & archive ingestion*

●  
*Grafana dashboards*