

Projects for Volunteers (Older)

- [P1: Kibana Dashboard for VSPERF Logs.](#)
- [P2: Enhanced Data Collection and Analytics: Collectd and Jupyter.](#)
- [P3: Kubernetes Metrics Collection and Analysis – COMPLETED](#)
- [P4: All-In-One TestVNF \(Forwarding, Trafficgen, Stress-gen\(cpu, memory, storage\) and Metrics\) With dynamic configuration and management.](#)
- [P5: Alerts for NFV Environments](#)
- [Student Volunteer Letters/Certificates](#)

P1: Kibana Dashboard for VSPERF Logs.

This project aims to create custom kibana dashboard analyzing and synthesizing VSPERF logs/events. For every test-run, VSPERF generates following logs:

1. vsperf-overall
2. vswitchd
3. trafficgen
4. host-cmds
5. guest-cmds

These logs have useful information that can help user to understand the test behavior, and analyze the performance results.

P2: Enhanced Data Collection and Analytics: Collectd and Jupyter.

User can configure collectd as the metrics collection tool, when the test is run. Using collectd provides user with extremely useful information to analyze. VSPERF, as part of its results container, includes Jupyter notebook doing minimal analytics by consuming collectd metrics, apart from other information.. This project aims to improve this analytics by adding following features:

1. Per-Queue metrics and Visualization
2. CPU-Steal analysis
3. Collect and analyze context-switches in /proc/pid/status/
4. QPI Transaction information analysis

P3: Kubernetes Metrics Collection and Analysis – COMPLETED

Enhance VSPERF's metrics collection process (collectd) and Analysis with Kubernetes/container metrics.

This project was completed by [Parth Yadav](#)

P4: All-In-One TestVNF (Forwarding, Trafficgen, Stress-gen(cpu, memory, storage) and Metrics) With dynamic configuration and management.

This project aims to create VM, which includes different opensources (as shown below) and provides ability to manage (configure and control) over the network.

[blocked URL](#)

P5: Alerts for NFV Environments

This project aims to create Alerting System based on Prometheus-Alertmanager and Elasticsearch for NFV.

With collectd (metrics) and fluentd (logs) on the client-side and Prometheus (metrics) and Elasticsearch (logs) on sever-side - there is a need unified alert management system.

The work involves:

1. Creating NFV-Specific Alerting Rules in Prometheus.
2. Creating NFV-specific Alerting rules in Elasticsearch.

3. Configuration Notifications in both AlertManager and Elasticsearch.
4. Build Unified alert-notification handler - to support useful Closed-loop automation usecases

Expected Skill: Python/Go

Student Volunteer Letters/Certificates

[Letter for Parth Yadav, 28 Jan 2020](#)