

Prometheus et Grafana une solution de supervision et de métrologie

X/Stra réunion du 08/04/2021

Introduction

- Supervision, Avertissement, Métrologie
- Métriques => time series database (TSDB)
- Outils conventionnels (Nagios, Centreon)
 - interfaces d'un autre âge
 - orientés administrateurs systèmes
 - prise en compte des DEV ou BUISINES TEAM difficile
- Nouveaux besoins
 - ASR, DEV, TEAM, Responsable de groupe, Direction
 - Indicateurs métier (suivi de commande, prix moyen d'un panier d'achat, pourcentage de requêtes réussies, consommation du stockage, ...)
 - Tableaux de bord

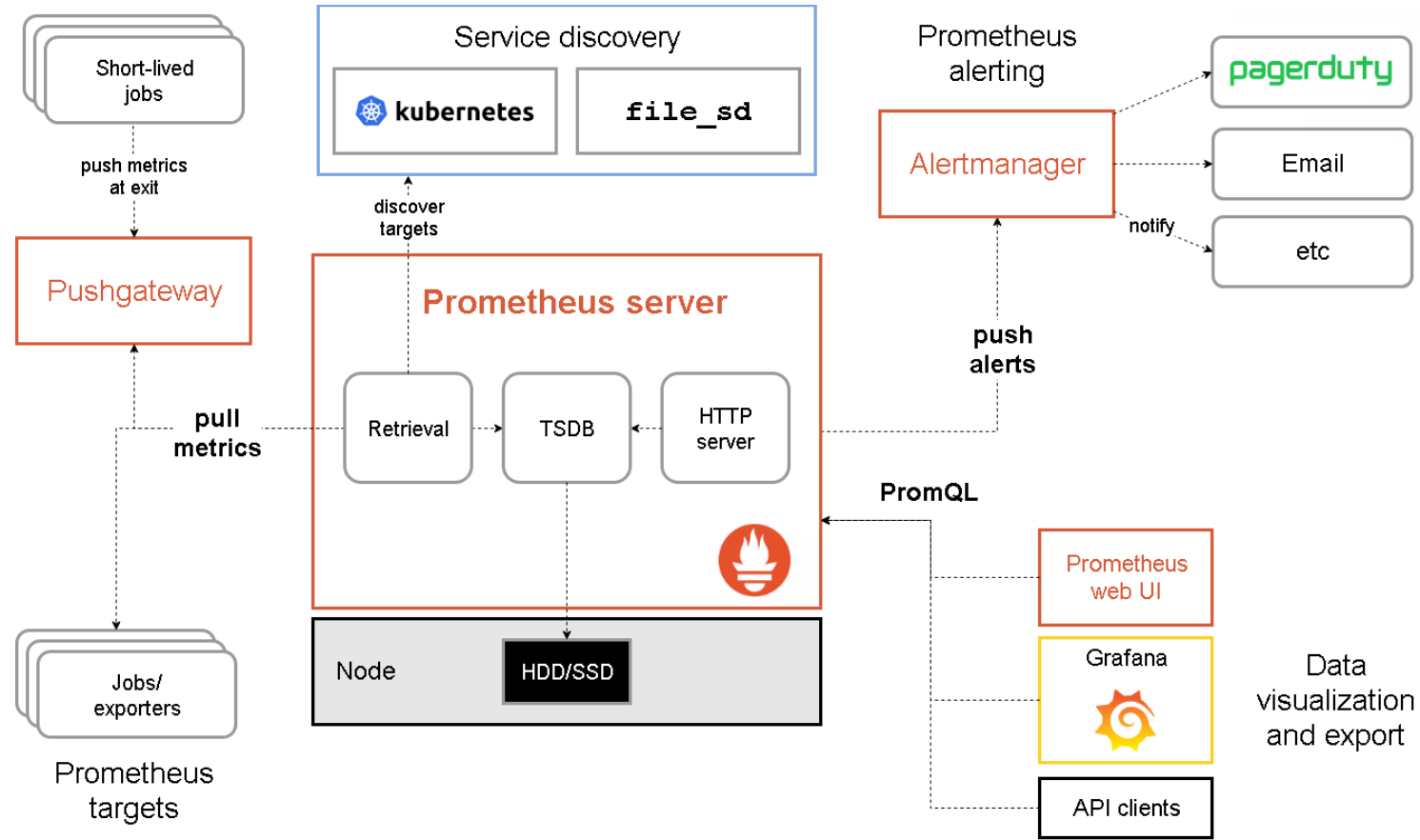
Comparatif des solutions open source

	Prometheus	Grafana	Elastic	Telegraf
Fonction	Collecte et stockage de métriques, gestion des alertes	Outil de visualisation	Collecte, NoSQL, Visualisation	Collecte de métriques
Société d'origine	SoundCloud	Grafana Labs	Elastic	InfluxData
Modèle de licence	Apache 2.0	Apache 2.02	Apache 2.0	MIT License
Ecosystème	CEPH, Kubernetes, OpenStack, RabbitMQ, PostgreSQL, Nginx, ...	Prometheus, InfluxDB Elastic, ...	Stack : Elasticsearch, Logstash, Kibana et Beats	Stack Tick : Telegraf, InfluxDB, Chronograf et Kapacitor
Points forts	Popularité, communauté active, adoubé par la Cloud native computing foundation	Facilité de prise en main, nombreux plugins, dashboards prêts à l'emploi, version entreprise	traiter une grosse volumétrie de fichiers logs	Ecosystème riche en plugins, grand historique des métriques
Traitement	Prometheus Anomaly Detector		x-pack (licence)	Anomaly detection, Machine learning

Prometheus - Grafana

- Prometheus, Node_exporter, Aletmanager
 - Ensemble d'outils, Go, multiple OS, API (http,https)
 - TSDB efficace, affichage des métriques, PromQL, service-discovery
- Grafana
 - web app, Go, support de plugins
 - Affichage de multiples métriques, tableaux de bord, Alerte notifications
 - Multiple datasource : prometheus, graphite, influxdb, Elastisearch
 - Authentification ldap, gestion des groupes et des rôles

Prometheus - Grafana infrastructure

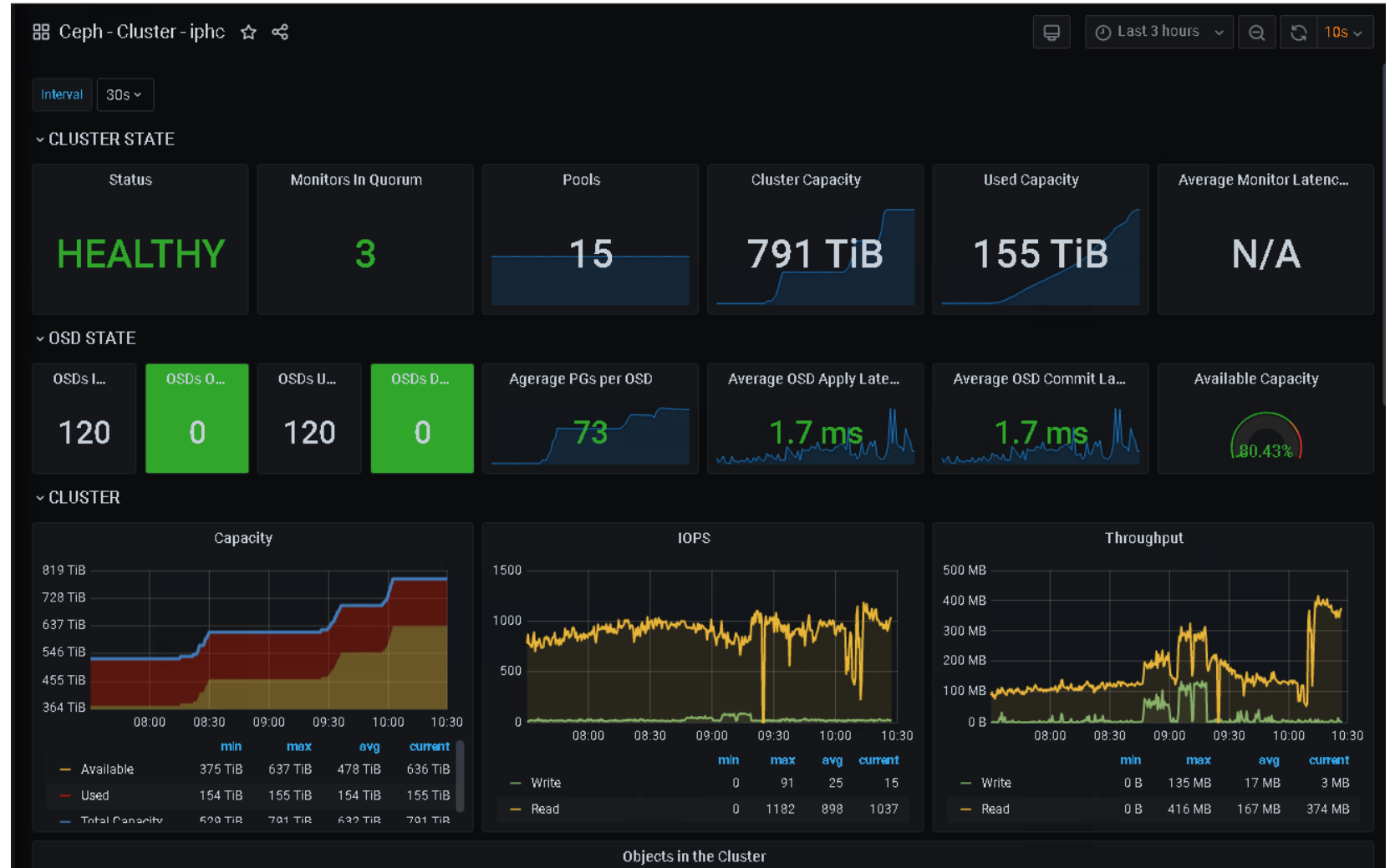


PromQL : Prometheus Query Language

- Permet de faire des requêtes sur des métriques et des labels
- Différent de SQL, garde l'idée du « where » et du « group by »
- Le facteur temps : offset (décalage) et vector ([5m])
- Exemples :
 - `node_network_receive_bytes_total{device=~"eth0|lo"}`
 - `count(count(node_cpu_seconds_total) by (cpu,instance)) by (instance)`
 - `sum(rate(http_requests_total[5m] offset 5m))`
 - `node_filesystem_avail{fstype!~"tmpfs|fuse.lxcfs|squashfs"} / node_filesystem_size{fstype!~"tmpfs|fuse.lxcfs|squashfs"}`
- Documentation : <https://timber.io/blog/promql-for-humans/>

Dashboard CEPH

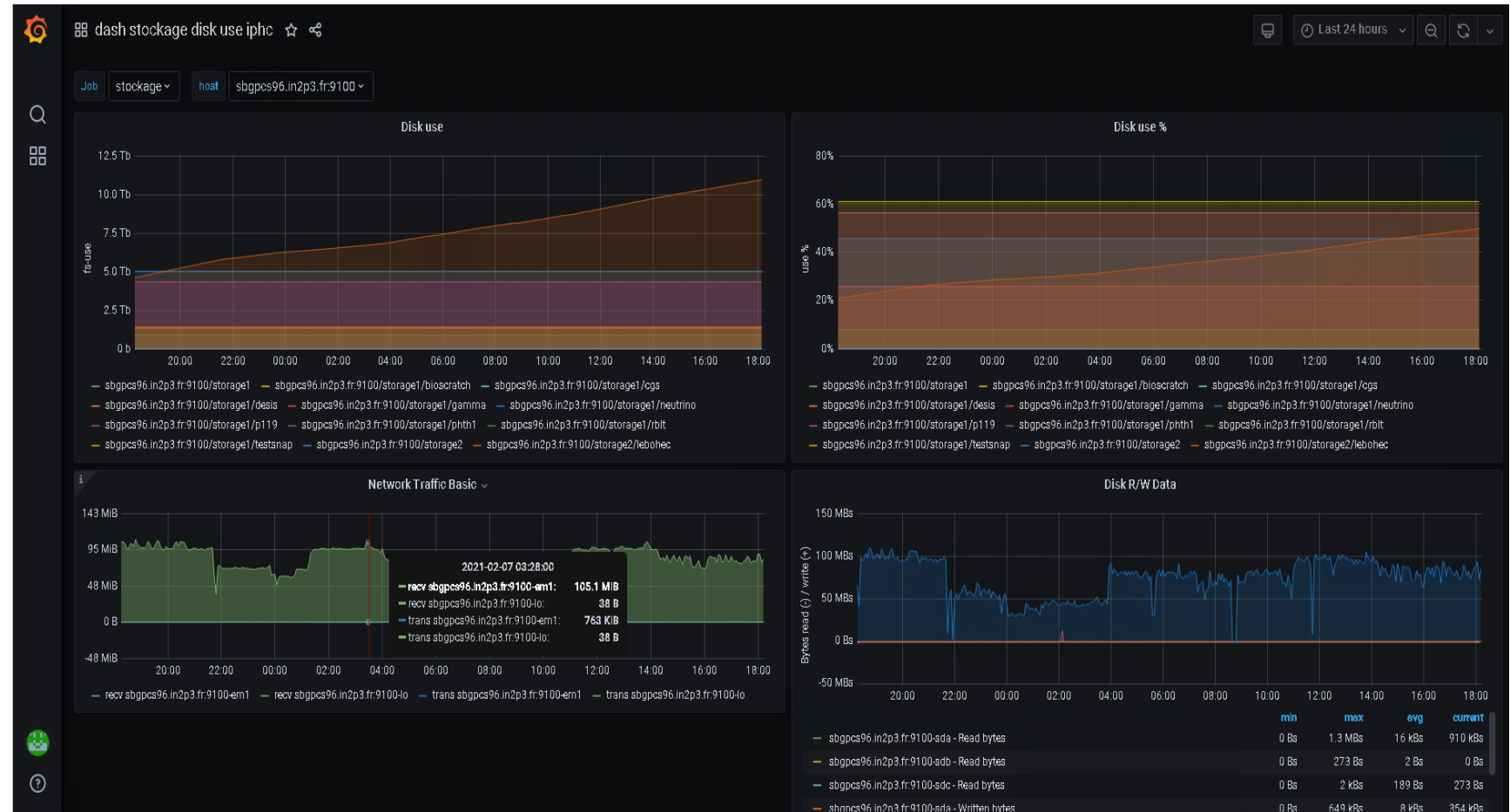
- Cluster : Etat, Capacity, Débit, IOPS, opérations Read, Write, Rebalance
- Pools : Débit, IOPS, Opérations Read, Write, reconstruction
- <https://grafana.com/grafana/dashboards/7056>



Dashboard Stockage

- Capacité par volume
- Pourcentage par volume
- Réseau in out
- Temps d'accès read write
- Tag par groupe ou fonction
- Liste des serveurs par tag

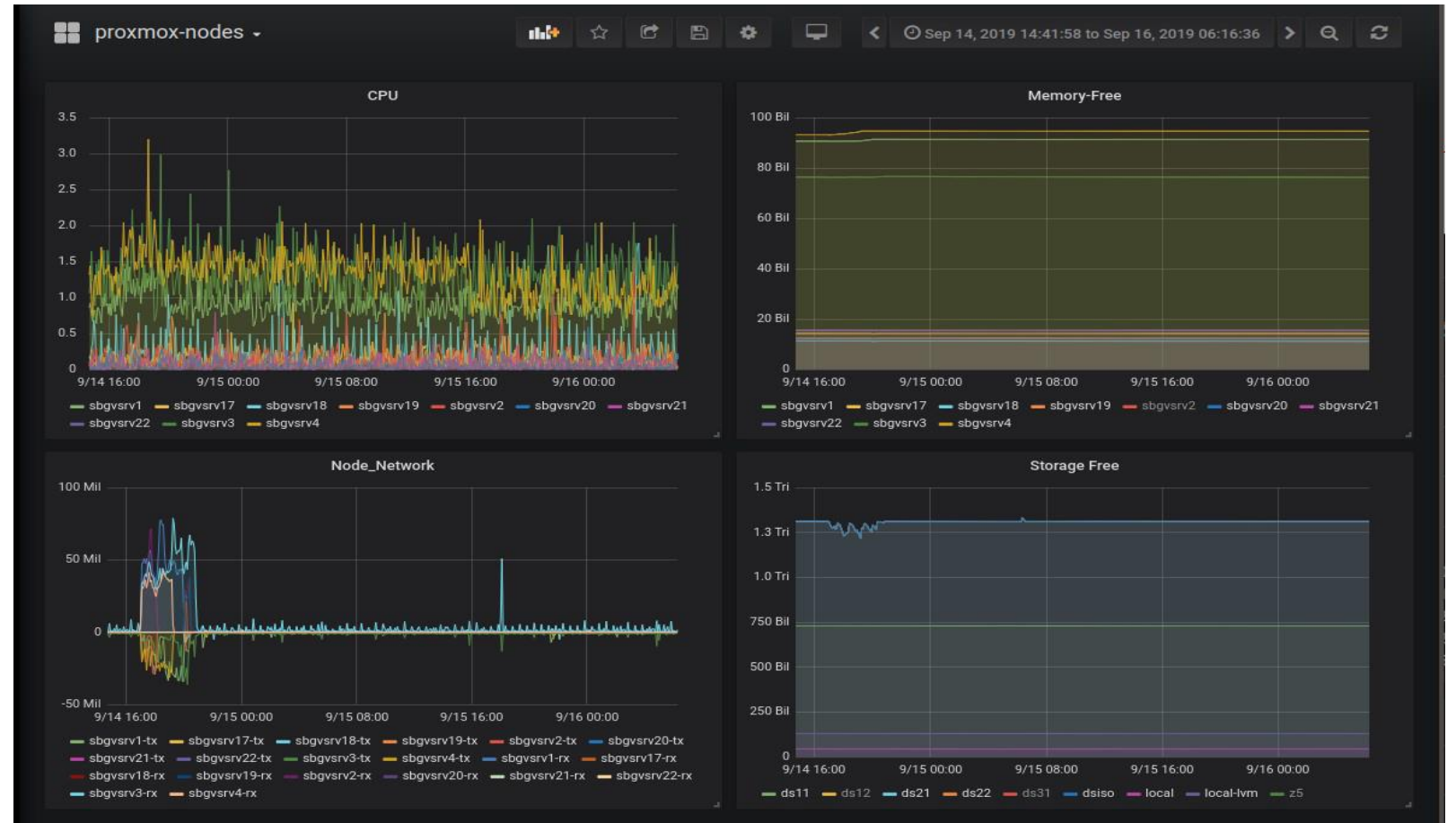
Fait maison



Dashboard Proxmox nodes

Node exporter : Prometheus exporter for hardware and OS metrics exposed by *NIX kernels, written in Go with pluggable metric collectors.

https://github.com/prometheus/node_exporter

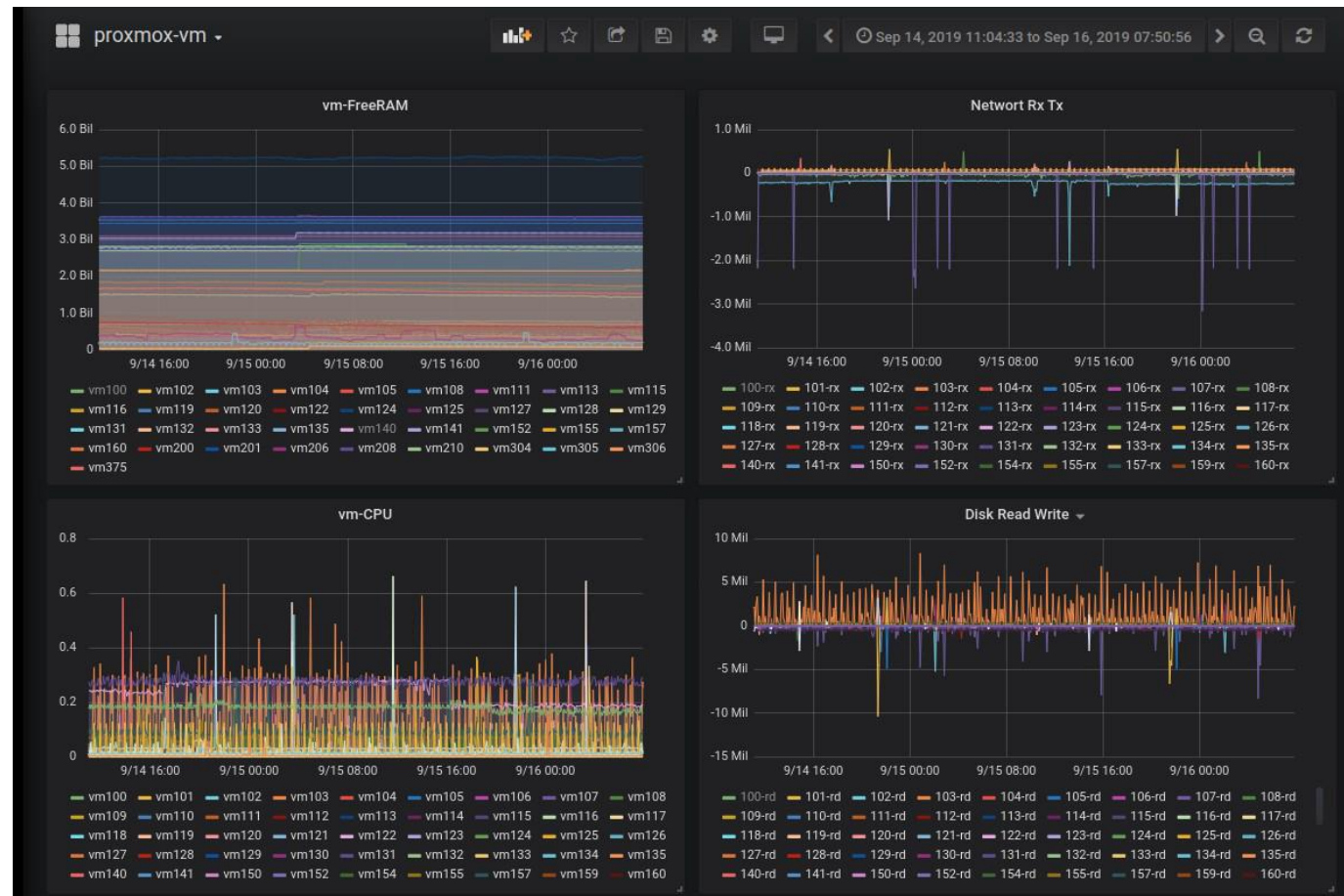


Dashboard Proxmox VMs

External Metric Server API
Graphite ou InfluxDB

Cpu, freeram, network tx-rx,
disque read write, ...

https://pve.proxmox.com/wiki/External_Metric_Server



Dashboard Cloud openstack

Canonical prometheus-openstack-exporter
Exposes high level OpenStack metrics to Prometheus.

Version Docker ou script python

Métrique :

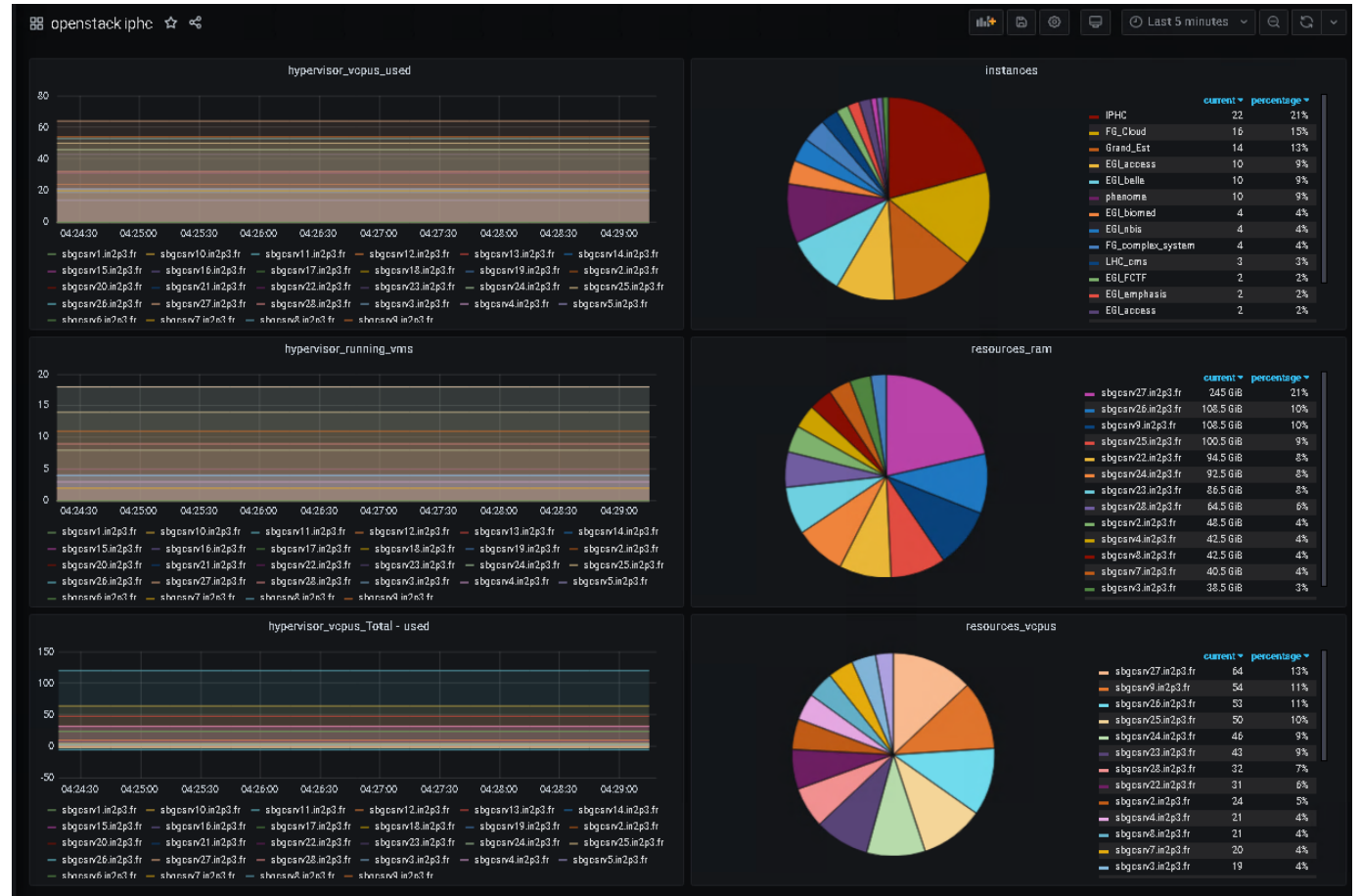
hypervisor: vCPUs, memory, disk, vms, capacity

Nova: instance, IP, ram, core

neutron : size, ip_usage

<https://github.com/CanonicalLtd/prometheus-openstack-exporter>

Faire: ressource utilisée / Total



Autres Dashboards

- SGDB : MySQL, PostgreSQL, NoSQL
- Web server : Apache, Nginx
- Auth : Ldap, kerberos
- Gestion de file d'attente : rabbitMQ, condor
- Et encore plus
<https://grafana.com/grafana/dashboards>

Alerte & Notifications

- Prometheus
 - Alert manager, règles PromQL
 - Avertissement par mail ou etc
 - Géré par les admins prometheus
 - <https://prometheus.io/docs/alerting/latest/alertmanager/>
- Grafana
 - Règles grafana, mais mêmes métriques
 - Notification : mail, slack, telegram
 - Géré par les Editor, créateur des tableaux de bord
 - <https://grafana.com/docs/grafana/latest/alerting/notifications/>

Auth & Rôles

- Grafana Auth
 - local, LDAP, AD, SAML, OAuth, ...
- Grafana Rôles : Admin, Editor, Viewer, None
- Filtre Dashboard
 - Users LDAP => admin
 - Groupe LDAP => viewer , other: none

Conclusion

- Travail en cours à l'IPHC
 - Règle de déploiement, automatisation
 - Différents services : web, grille, cloud, stockage, auth, réseau, jetons
 - Métriques : sélections, temps de rétention, labels
 - Dashboard : pour qui et accès
 - Traitement : Détections d'anomalies
- Besoin X/Stra
 - Séance d'échanges sur les configurations, café supervision ?
 - Formation
- Autres ?

Documentation

- <https://prometheus.io/docs/introduction/overview/>
- <https://grafana.com/grafana/>
- <https://www.elastic.co/fr/elastic-stack>
- <https://www.elastic.co/fr/what-is/open-x-pack>
- <https://www.influxdata.com/time-series-platform/>
- <https://blog.debugo.fr/tuto-monitoring-prometheus-partie-1-installation-prometheus/>
- <https://github.com/AICoE/prometheus-anomaly-detector>
- <https://thanos.io/tip/thanos/getting-started.md/>