RED HAT
OPEN SOURCE DAY
Europe, Middle East & Africa

redhat.

Monitoraggio delle applicazioni su Openshift con Prometheus

Luigi De Masi

extrared

#redhatosd

DevOps Pillars

- **C**ollaboration
- **A**utomation
- **L**ean
- **M**onitoring
- **S**haring

What is monitoring?

“Observe and check the progress or quality of (something) over a period of time, keep under systematic review.”

--

Oxford Dictionary

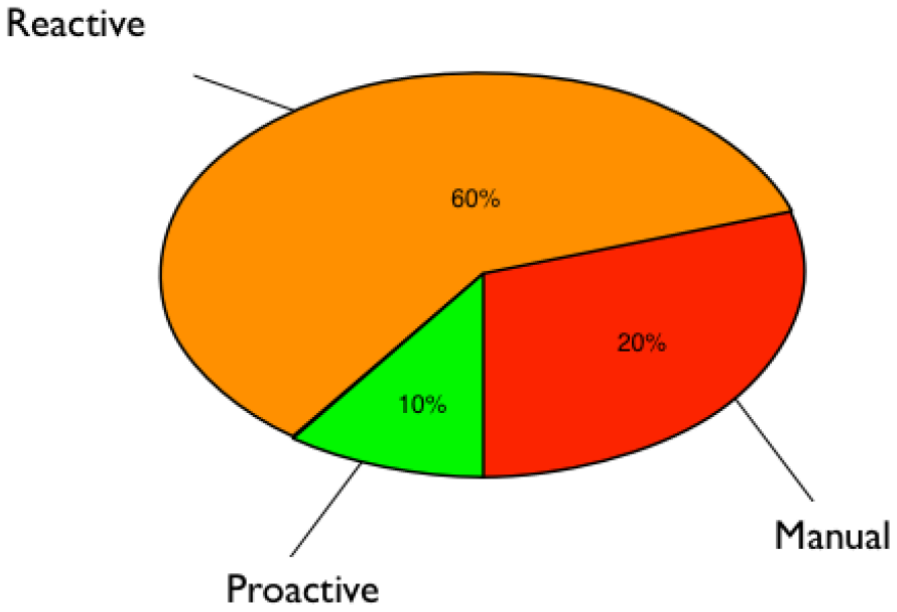
Infrastructure evolution

bare metal	• tipicamente ~10^2 host • configurazione statica • host fino all' EOL dell'HW • istanze pet
virtualizzazione	• tipicamente ~10^3 vm • configurazione semi-dinamica • istanze pet
cloud	• Il monitoraggio è considerato un asset IT e di Business • L' individuazione e la creazione di metriche fa parte del processo di sviluppo. • Focus sulla qualità del servizio e business KPI

Monitoring evolution

manuale	• Iniziato da un operatore e gestito tramite checklist • script semplici e processi non automatizzati. • Focus sull' uptime
reattivo	• Prevalentemente automatico con ancora alcuni task lasciati all' operatore. • Utilizzati tool come Nagios con controlli di default (cpu, memoria,network...) • Focus sulla disponibilità dei servizi
proattivo	• Il monitoraggio è considerato un asset IT e di Business • L' individuazione e la creazione di metriche fa parte del processo di sviluppo. • Focus sulla qualità del servizio e business KPI

Monitoring evolution



Monitoring has two customers

Information Technology as a customer

Offre ai responsabili dei servizi la possibilità di tenere sotto controllo la disponibilità e la salute del sistema.

Business as a customer

Offre ai business owner una traduzione tra le metriche generate e il business value del proprio sistema, fornendo al management degli indicatori su quali e quanti sono i servizi erogati e in che modo gli utenti li consumano

Prometheus

What is Prometheus?

- Prometheus è un tool open-source per il monitoraggio e l’ alerting
- Scritto in Go per monitoraggio dei sistemi di SoundCloud.
- Ispirato a BorgMon, fa lo scraping delle metriche delle applicazioni (pull)

Instrumentation

Client libraries:

- Go
- Java or Scala
- Python
- Ruby

- Bash
- C++
- Common Lisp
- Elixir
- Erlang
- Haskell
- Lua for Nginx
- Lua for Tarantool
- .NET / C#
- Node.js
- PHP
- Rust

Instrumentation: code

```
import io.prometheus.client.Counter;
class YourClass {
  static final Counter requests = Counter.build()
     .name("requests_total").help("Total requests.").register();

  void processRequest() {
    requests.inc();
    // Your code here.
  }
}
```

```
# HELP go_gc_duration_seconds A summary of the GC invocation durations.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0
go_gc_duration_seconds{quantile="0.75"} 0
go_gc_duration_seconds{quantile="1"} 0
go_gc_duration_seconds_sum 0
go_gc_duration_seconds_count 0
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 14
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 995200
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 995200
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.443801e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 0
```

Instrumentation: jmx exporter

```
##TURNING JMX ON
#Should retrieve local IP address
IP_ADDR=`ip route get 8.8.8.8 | awk '{print $NF; exit}'`
JAVA_OPTS="$JAVA_OPTS -Djava.rmi.server.hostname=$IP_ADDR"
JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.port=9696"
JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.authenticate=false"
JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.ssl=false"

##JMX Exporter Agent
JMX_DIR="/home/patman/jmx_exporter_zookeeper"
JAVA_OPTS="$JAVA_OPTS -javaagent:$JMX_DIR/jmx_prometheus_javaagent-0.9.jar=1234:
```

```
# HELP go_gc_duration_seconds A summary of the GC invocation durations.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0
go_gc_duration_seconds{quantile="0.75"} 0
go_gc_duration_seconds{quantile="1"} 0
go_gc_duration_seconds_sum 0
go_gc_duration_seconds_count 0
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 14
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 995200
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 995200
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.443801e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 0
```

Metrics pull in the cloud

Integrazione con i più popolari sistemi di cloud IaaS e PaaS, tramite i loro sistemi di service discovery:

- AWS EC2
- Google Cloud Engine,
- Azure,
- Kubernetes/Openshift
- Openstack

- Consul - Hashicorp
- Marathon - Mesos
- Nerve – Airbnb (zookeeper)
- Serverset (zookeeper)
- Triton
- DNS SRV query

Prometheus and Openshift

Discovery degli endpoint delle metriche recuperando gli oggetti kubernetes tramite API:

node:

- `__meta_kubernetes_node_name`: The name of the node object.
- `__meta_kubernetes_node_label_<labelname>`: Each label from the node object.
- `__meta_kubernetes_node_annotation_<annotationname>`: Each annotation from the node object.
- `__meta_kubernetes_node_address_<address_type>`: The first address for each node address type, if it exists.

pod:

- `__meta_kubernetes_namespace`: The namespace of the pod object.
- `__meta_kubernetes_pod_name`: The name of the pod object.
- `__meta_kubernetes_pod_ip`: The pod IP of the pod object.
- `__meta_kubernetes_pod_label_<labelname>`: The label of the pod object.
- `__meta_kubernetes_pod_annotation_<annotationname>`: The annotation of the pod object.
- `__meta_kubernetes_pod_container_name`: Name of the container the target address points to.
- `__meta_kubernetes_pod_container_port_name`: Name of the container port.
- `__meta_kubernetes_pod_container_port_number`: Number of the container port.
- `__meta_kubernetes_pod_container_port_protocol`: Protocol of the container port.
- `__meta_kubernetes_pod_ready`: Set to `true` or `false` for the pod's ready state.
- `__meta_kubernetes_pod_node_name`: The name of the node the pod is scheduled onto.
- `__meta_kubernetes_pod_host_ip`: The current host IP of the pod object.

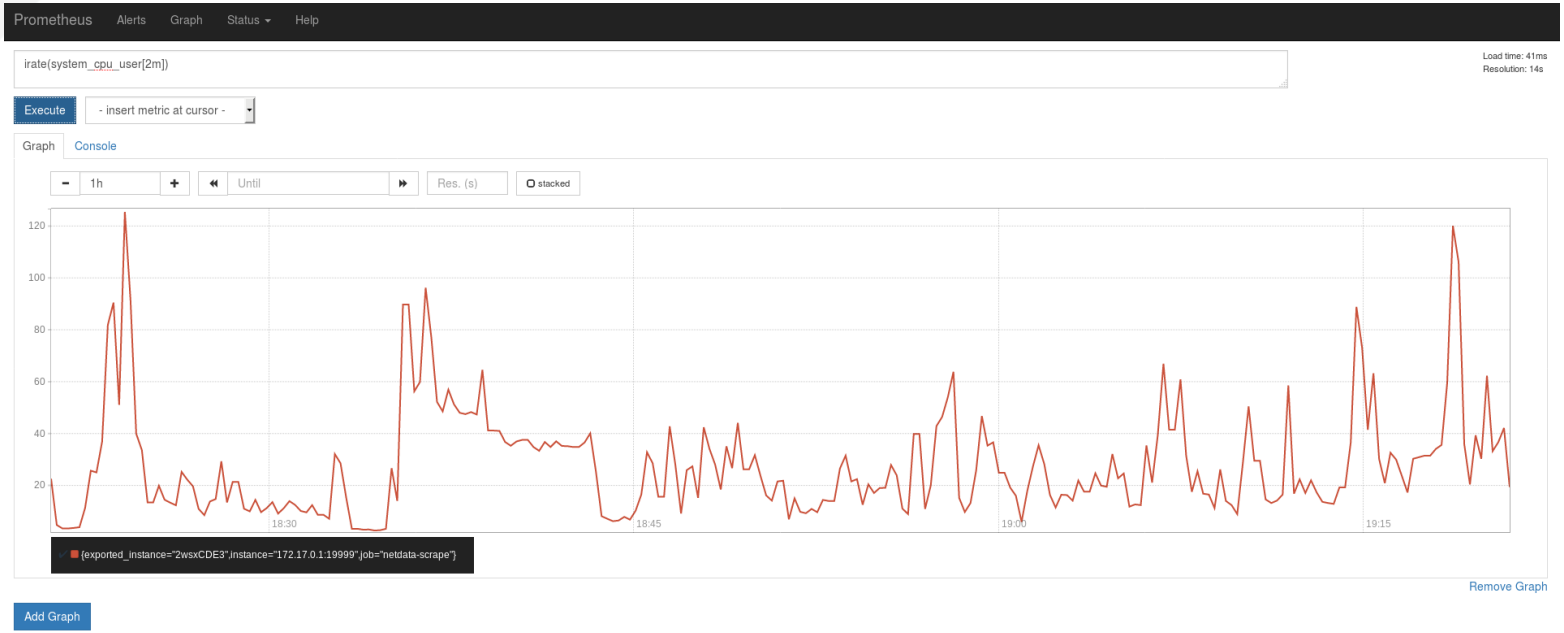
service:

- `__meta_kubernetes_namespace`: The namespace of the service object.
- `__meta_kubernetes_service_name`: The name of the service object.
- `__meta_kubernetes_service_label_<labelname>`: The label of the service object.
- `__meta_kubernetes_service_annotation_<annotationname>`: The annotation of the service object.
- `__meta_kubernetes_service_port_name`: Name of the service port for the target.
- `__meta_kubernetes_service_port_number`: Number of the service port for the target.
- `__meta_kubernetes_service_port_protocol`: Protocol of the service port for the target.

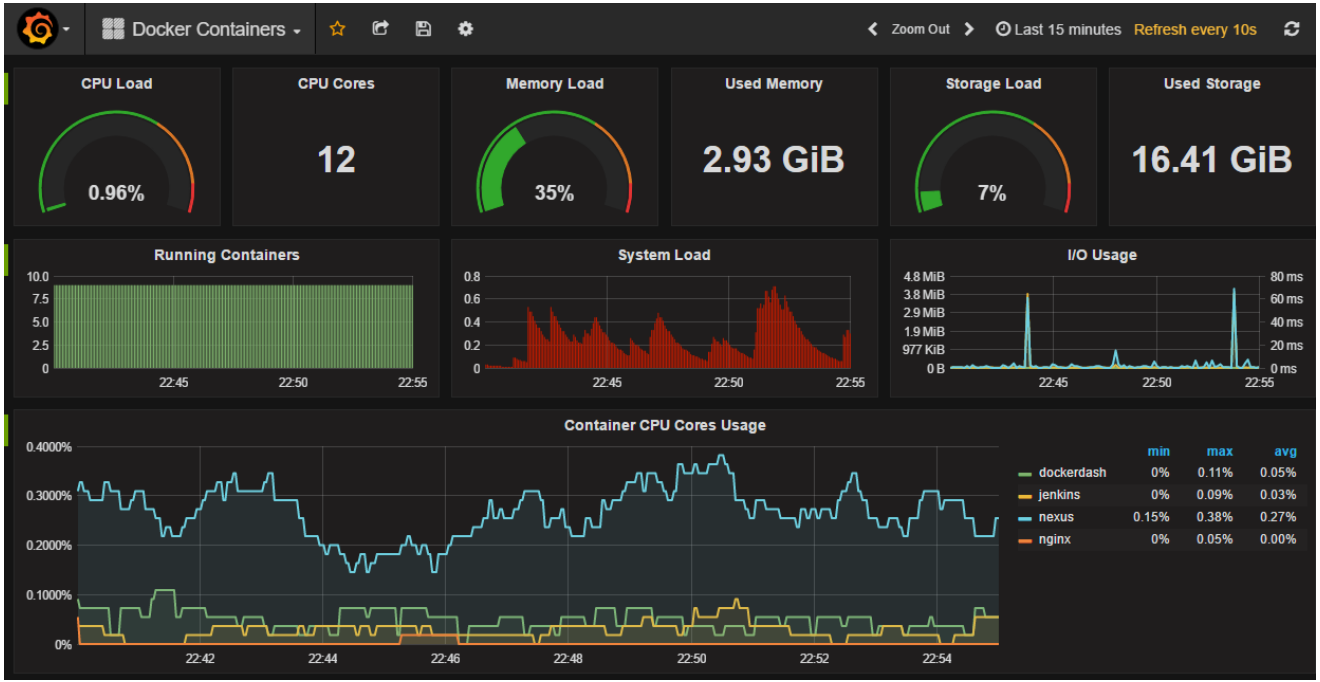
endpoint:
ingress:

- `__meta_kubernetes_namespace`: The namespace of the ingress object.
- `__meta_kubernetes_ingress_name`: The name of the ingress object.
- `__meta_kubernetes_ingress_label_<labelname>`: The label of the ingress object.
- `__meta_kubernetes_ingress_annotation_<annotationname>`: The annotation of the ingress object.
- `__meta_kubernetes_ingress_scheme`: Protocol scheme of ingress, `https` if TLS config is set. Defaults to `http`.
- `__meta_kubernetes_ingress_path`: Path from ingress spec. Defaults to `/`.

Metrics Visualization



Dashboarding with Grafana



OpenShift Roadmap

OpenShift Container Platform 3.4 (January)
- Kubernetes 1.4 & Docker 1.12
- Dynamic Storage provisioning & Storage QoS tiers
- Storage size quota and scopes
- Kubernetes Deployments & Job Scheduler API
- Pod Disruption Budget
- Eviction on File System Usage
- Usability enhancements & first-time user flows
- Build enhancements (performance, integrations)
- CNI integration for openshift-sdn
- Integration to external logging systems (Splunk)
- Registry enhancements

OpenShift Online & Dedicated
- OpenShift Dedicated on Google (Dec 8)
- OpenShift Online Dev Preview User Expansion

OpenShift Container Platform 3.6 (August)
- Kubernetes 1.6 & Docker 1.12
- New Application Services - 3Scale API Mgt OnPrem, SCL 2.4
- Web UX Project Overview enhancements
- Service Catalog/Broker & UX (Tech Preview)
- Ansible Service Broker (Tech Preview)
- Secrets Encryption (3.6.1)
- Signing/Scanning + OpenShift integration
- Storage - Integrated CNS for Reg and Installer, AWS EFS
- OverlayFS with SELinux Support (RHEL 7.4)
- User Namespaces (RHEL 7.4)
- System Containers for docker

OpenShift Online & Dedicated
- OpenShift Online Paid Tier GA (July)

Q1 CY2017

Q2 CY2017

OpenShift Container Platform 3.5 (April)
- Kubernetes 1.5 & Docker 1.12
- Kubernetes StatefulSets (Tech Preview)
- Java S2I for Cloud Native
- CD Pipeline enhancements
- Dynamic Provisioning for Azure block
- Network Policy (Tech Preview) & Multicast
- Opaque Integers (Tech Preview)
- Registry Image Handling Options
- SCC Usability
- SCL 2.3 shipped out of box

OpenShift Online & Dedicated
- OpenShift Online Free Tier GA (May/Summit)
- OpenShift.io Launch (May/Summit)

Q3 CY2017

Q4 CY2017

OpenShift Container Platform 3.7 (November)
- Kubernetes 1.7 & Docker 1.13
- Red Hat OpenShift Application Runtimes (GA)
- Service Catalog/Broker & UX (GA)
- Ansible Service Broker (GA)
- AWS Service Catalog
- Network Policy (GA)
- [illegible] (Tech Preview)
- Prometheus Real-Time Metrics (Tech Preview)
- [illegible]
- Performance Sensitive Apps & Spark (TBD)

OpenShift Online & Dedicated
- OpenShift Dedicated Enhancements
- OpenShift Online Enhancements

RED HAT
OPEN SOURCE DAY

Europe, Middle East & Africa

extrared

#redhatosd