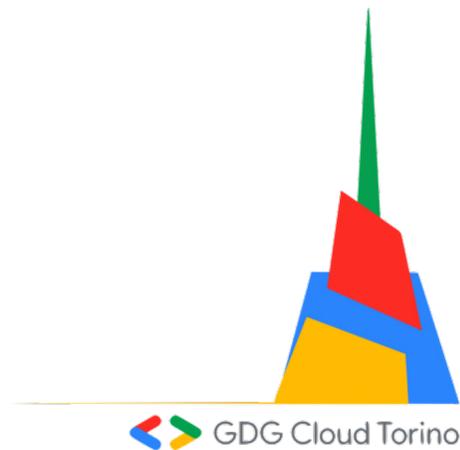




DevOps on Kubernetes: DIY in Open Source

Gianni Forlastro
Francesco Pirrone



```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
te = tf.lookup.StaticV  
init,  
num_oov_buckets=5)  
  
lookup.StaticVocabular  
initializer,  
num_oov_buckets,  
lookup_key_dtype=None  
name=None,  
experimental_is_ope
```

GIANNI FORLASTRO

Google Developer Expert

CTO - Securities & Funds Dept. @ finwave

Community Lead

@ GDG Cloud Torino

@ Flutter Torino

FRANCESCO PIRRONE

Full Stack Dev @ finwave

Community Lead

@ GDG Cloud Torino

@ Flutter Torino



La nostra community collabora con



Platform basati su prodotti Open Source

PRO:

- risparmio sui costi di licenza e slegato dai volumi
- permette di personalizzare la piattaforma secondo le proprie esigenze
- accesso libero ai sorgenti
- educativo e divertente
- omogeneità piattaforma tra cloud provider/onpremise

CONTRO:

- nessun supporto esterno
- difficile da scalare
- attività di manutenzione a carico
- evolutive dipendono dalle community

Orchestrazione Container: Kubernetes Engine



Codice sorgente: <https://github.com/kubernetes/kubernetes>

Linguaggio: GO

Licenza: Apache 2.0

Kubernetes (abbreviato K8s) è un sistema open-source di orchestrazione e gestione di container.

Inizialmente sviluppato da Google, adesso è mantenuto da Cloud Native Computing Foundation.

Funziona con molti sistemi di containerizzazione su interfacce CRI (Container Runtime Interface).

Distributed Version Control: Gitlab CE GitLab

Codice sorgente: <https://gitlab.com/rluna-gitlab/gitlab-ce>

Linguaggio: Ruby

Licenza: MIT

Installabile su macchine virtuali tramite pacchetti Linux sulle varie distribuzioni, su Kubernetes tramite Operator o Helm chart, oppure tramite template nei market place dei vari cloud provider.

Permette la configurazione di Single Sign On sia tramite protocollo LDAP, OpenId, SAML2 che a sua volta può essere sfruttato come identity provider.

Continuous Integration: Gitlab Runner



Codice sorgente: <https://gitlab.com/gitlab-org/gitlab-runner>

Linguaggio: GO

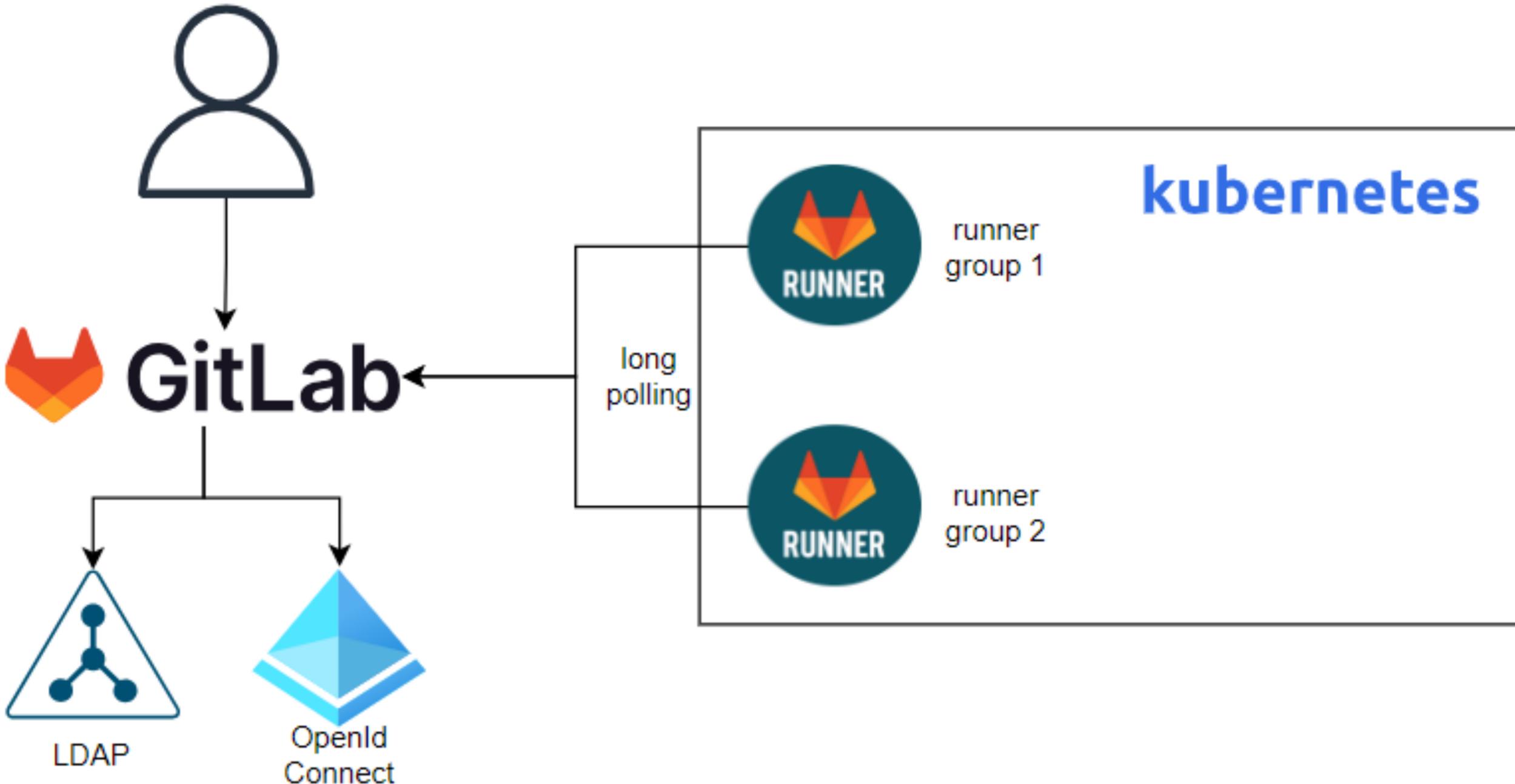
Licenza: MIT

Si integra facilmente con **Gitlab** per eseguire delle build al momento del push remoto.
Le pipeline sono scritte tramite file yaml.

I template delle pipeline possono essere importati nei vari progetti ed è facile sovrascrivere eventualmente step o variabili d'ambiente.

Supporta l'esecuzione di job remoti su macchine linux/windows; è facilmente scalabile tramite un cluster Kubernetes o VM Cloud oppure direttamente dal virtualizzatore (VMWare, VirtualBox...)

Gitlab Continuous Integration



Gitlab Pipeline

The screenshot shows the Gitlab Pipeline interface for a project named "fix kafka headers". The breadcrumb navigation at the top indicates the path: "Project Panda / Native Services / Documentale / Pipeline / #111962".

Project Sidebar: On the left, there is a sidebar with navigation options: "Project", "Bloccato", "Issues" (0), "Merge requests" (0), "Pipeline" (selected), "Gestisci", "Pianificazione", "Codice", "Crea", "Pipeline", "Jobs", "Editor di pipeline", "Pianificazioni della pipeline", "Artefatti", "Proteggi", and "Esegui il deployment".

Pipeline Details: The main content area shows the pipeline title "fix kafka headers" and a notification: "Avviso Francesco Pirrone ha creato una pipeline per il commit c97af656 21 ore fa, completamento 21 ore fa". Below this, it indicates the pipeline is for the "develop" branch and is the "più recente" (most recent) with 8 jobs, a total duration of 4 minutes 47 seconds, and a 2-second queue time.

Job Summary: A summary bar shows "Pipeline" (selected), "Jobs 8", "Failed Jobs 1", and "Test 0".

Job Grouping: Below the summary, there are buttons for "Raggruppa job per" (Stage, Dipendenze del job) and a "Mostra dipendenze" toggle which is currently turned on.

Job Graph: The main part of the interface displays a job graph. A tooltip is visible over a job, stating: "to: passa il mouse su un job per vedere i job da cui dipende per l'esecuzione." The graph shows the following jobs:

- build (partially visible)
- compile-mssql build (status: success)
- build-image docker-image (status: success)
- docker-release-psql-jvm docker-image-jvm (status: success)
- sonar-analyze analyze (status: failed)
- docker-mirror-dev deploy (status: success)
- kubectl-deploy-dev deploy (status: success)

Each job box includes a status icon (checkmark or warning), the job name, and a refresh icon.

Identity Management: Keycloak



Codice sorgente: <https://github.com/keycloak/keycloak>

Linguaggio: Java

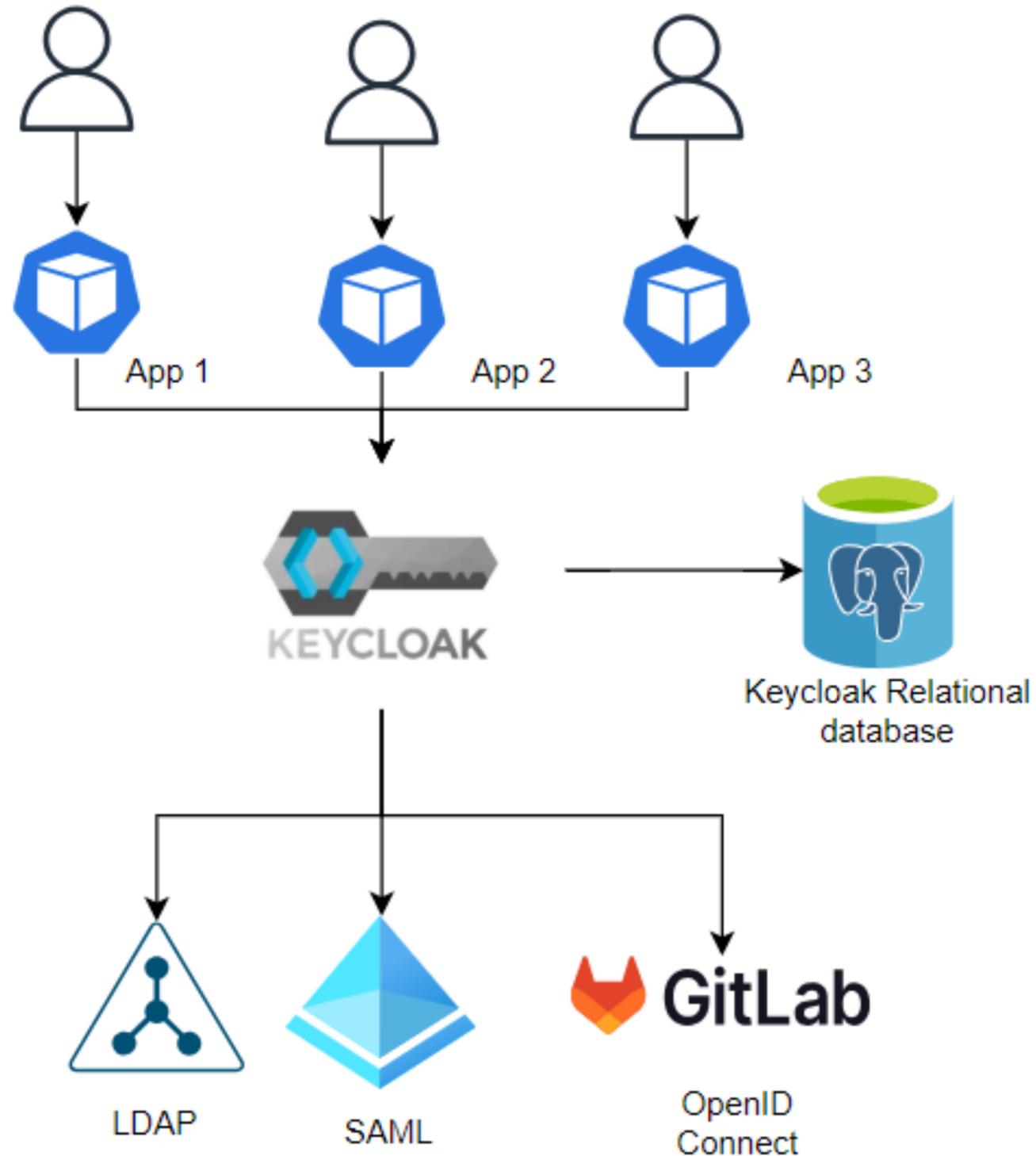
Licenza: Apache 2.0

Progetto a livello di Incubating della Cloud Native Computing Foundation, è una soluzione di Identity e Access Management (IAM).

Tra le sue features principali troviamo:

- supporto 2FA/passwordless con Webauthn
- Password policy
- Federazione con sistemi di autenticazione LDAP, OpenID, Oauth2, SAML2
- Estendibile tramite lo sviluppo di plugin
- Definire diversi Realm all'interno della stessa istanza

IM: Keycloak



Gli utenti si autenticano tramite protocollo OpenID Connect con le loro applicazioni verso Keycloak.

Le utenze possono essere presenti nel database di keycloak oppure è possibile federare con LDAP o altri sistemi di autenticazione basati su SAML o OpenID.

Inoltre possiamo avere diversi “realm” utenti, ognuno con federazioni e ruoli diversi.

Lo stesso utente potrebbe essere amministratore in un realm ed utente semplice in un altro.

Keycloak : login (personalizzabile)

DEV

Sign in to your account

Username or email

Password

 Remember me

Sign In

Or sign in with



Keycloak : gestione realm

The screenshot displays the Keycloak administration interface. On the left is a dark sidebar with a menu containing: Manage, Clients, Client scopes, Realm roles, Users, Groups (highlighted), Sessions, Events, Configure, Realm settings, Authentication, Identity providers, and User federation. The main content area is divided into two panels. The left panel shows a search for groups with the text 'Search group' and a search icon, an 'Exact search' checkbox, and a list of groups: 'ArgoCDAdmins', 'admin', and 'user'. The right panel is titled 'Groups' and includes a description: 'A group is a set of attributes and role mappings that can be applied to a user. You can create, edit, and delete groups and manage their child-organization. [Learn more](#)'. Below the description is a search bar 'Filter groups', a 'Create group' button, and a 'Refresh' button. A list of groups is shown with checkboxes: 'Group name', 'ArgoCDAdmins', 'admin', and 'user'. The top of the interface features a 'dev' dropdown menu and the 'KEYCLOAK' logo.

Continuous Delivery : ArgoCD



Codice sorgente: <https://github.com/argoproj/argo-cd>

Linguaggio: GO

Licenza: Apache 2.0

Progetto a livello di Graduated della CNCF, ArgoCD è uno strumento di Continuous Deployment (CD) basato su un approccio dichiarativo delle applicazioni e dello stato del cluster tramite la metodologia GitOps.

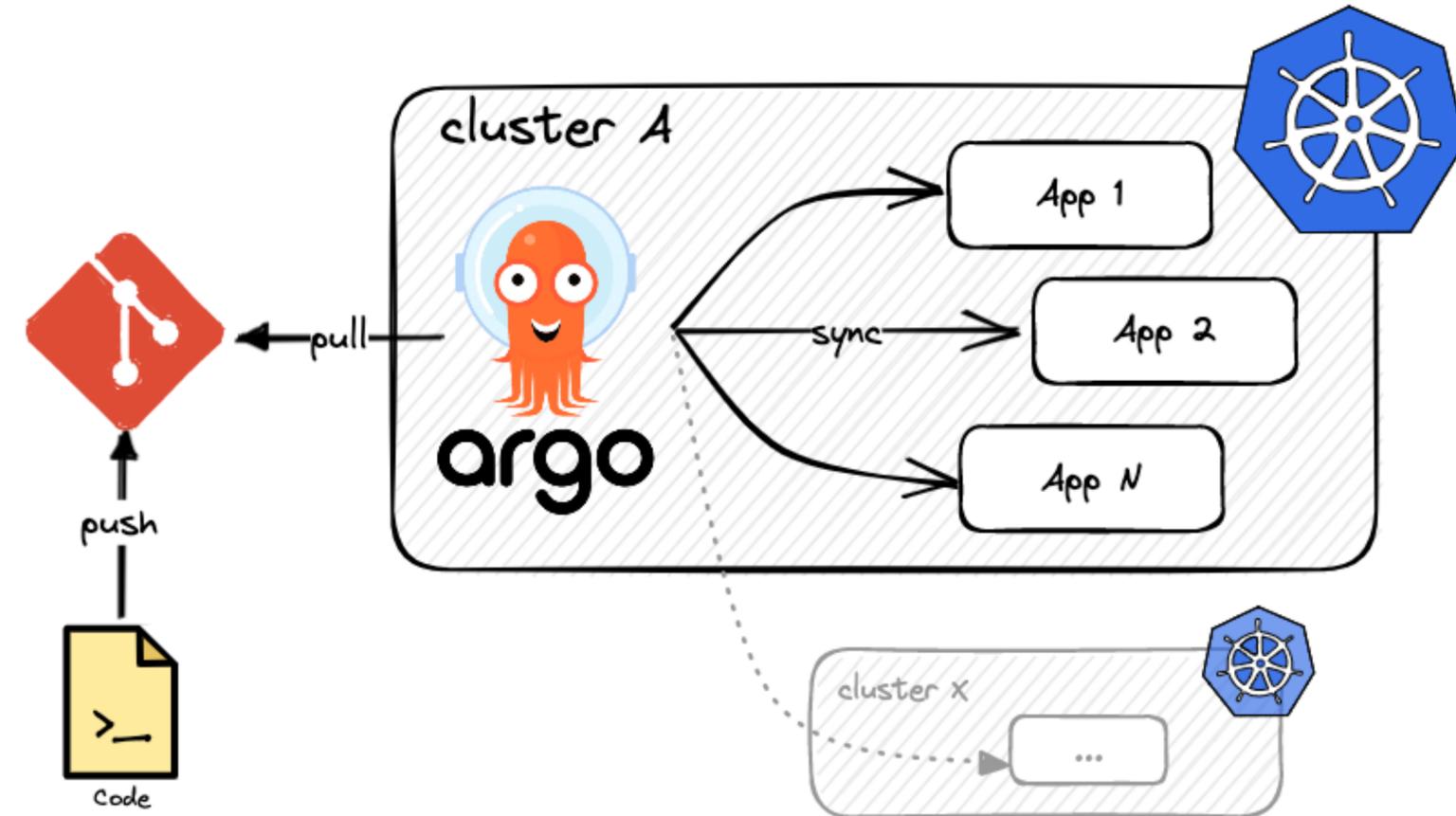
Tra i vantaggi dell'utilizzo vi è:

- l'utilizzo della sua UI come dashboard di Kubernetes
- la gestione più cluster Kubernetes da una singola installazione di ArgoCD
- associare le applicazioni a dei progetti e definire su di essi delle limitazioni su namespace, cluster e risorse Kubernetes associati
- ricevere notifiche dopo un rilascio ed il suo esito

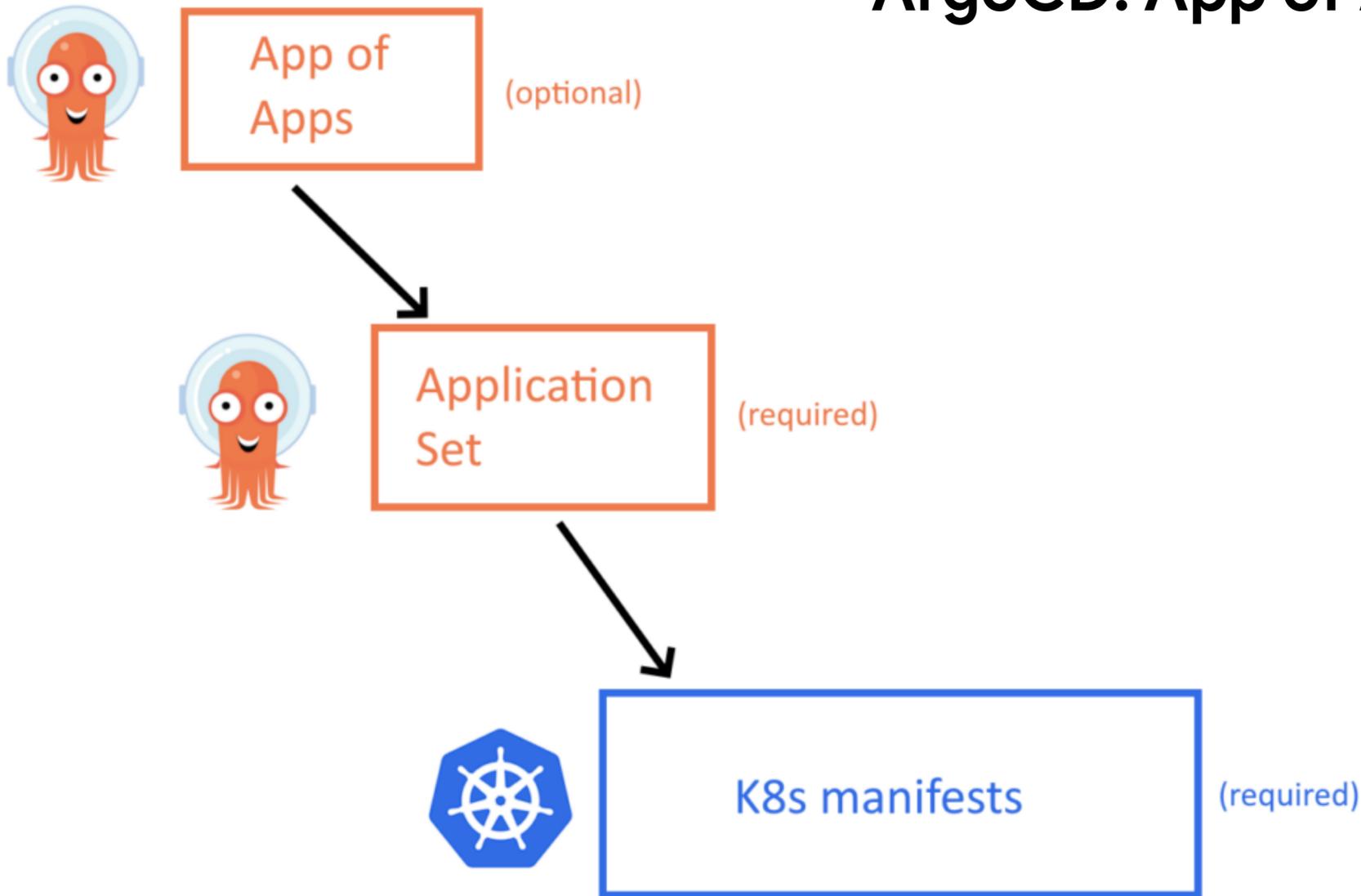
ArgoCD: GitOps

L'approccio GitOps permette di:

- ridurre le attività manuali di gestione del cluster
- tracciare le attività infrastrutturali tramite il commit log
- automatizzare i rilasci
- evitare il configuration drift tra le risorse del cluster e la configurazioni sul repository
- Disaster Recovery del cluster: in quanto è sempre possibile ripristinare lo stato (ma non i dati delle applicazioni) a partire dal repository



ArgoCD: App of Apps pattern



Consiste nel definire un'applicazione che definisce nel repository se stessa e tutte le altre.

In questo modo è possibile aggiungere altre applicazioni gestite da ArgoCD tramite il modello GitOps centralizzando il tutto in un unico repository.

Tramite **App of Apps** possiamo infatti gestire l'installazione stessa di ArgoCD, altre dipendenze del nostro cluster ed i relativi aggiornamenti senza accedere direttamente al Control Plane noi stessi.

ArgoCD: application status

The screenshot displays the ArgoCD web interface for an application named 'gitops'. The interface is divided into several sections:

- Header:** Shows 'Applications / gitops' and 'APPLICATION DETAILS TREE'.
- Navigation:** Includes buttons for 'DETAILS', 'DIFF', 'SYNC', 'SYNC STATUS', 'HISTORY AND ROLLBACK', 'DELETE', and 'REFRESH'.
- Summary Cards:**
 - APP HEALTH:** Shows a green heart icon and the word 'Healthy'.
 - SYNC STATUS:** Shows a green checkmark icon and 'Synced to HEAD (faf31f7)'. Below it, it states 'Auto sync is enabled.' and lists the author 'Francesco Pirrone' and comment 'fix priority'.
 - LAST SYNC:** Shows a green checkmark icon and 'Sync OK to faf31f7'. Below it, it states 'Succeeded a day ago (Fri Mar 21 2025 16:00:25 GMT+0100)' and lists the author 'Francesco Pirrone' and comment 'fix priority'.
- Resource List:** A central area showing a tree view of resources. A 'gitops' application icon is highlighted in the center. To its right, a list of resources is shown:
 - PC: native-build-priority (a day)
 - ns: 11 Namespaces
 - pvc: 2 PersistentVolumeClaims
 - secret: 23 Secrets
 - sa: 3 ServiceAccounts
 - application: 33 Applications
 - appproject: 6 AppProjects
 - crb: 2 ClusterRoleBindings
 - role: 3 Roles
- Left Sidebar:** Contains navigation links for 'Applications', 'Settings', 'User Info', and 'Documentation'. It also has filters for 'NAME', 'KINDS', 'SYNC STATUS', 'HEALTH STATUS', and 'NAMESPACES'. The 'SYNC STATUS' filter shows 92 'Synced' and 0 'OutOfSync' items. The 'HEALTH STATUS' filter shows 2 'Healthy', 0 'Progressing', 0 'Degraded', 0 'Suspended', 0 'Missing', and 0 'Unknown' items.

ArgoCD: POD drilldown

The screenshot displays the ArgoCD web interface. On the left is a dark sidebar with navigation options: Applications, Settings, User Info, and Documentation. Below these are filters for NAME, KINDS, SYNC STATUS (Synced: 43, OutOfSync: 0), and HEALTH STATUS (Healthy: 171, Progressing: 0, Degraded: 0, Suspended: 0, Missing: 0, Unknown: 0). The main content area shows the details for a Pod named 'gateway-app-f69f6687f-ljr8c' in the 'dv-gateway' namespace. The pod is in a 'Running' state and is 'Healthy'. The container 'gateway-app' is also running. The 'LIVE MANIFEST' section shows the pod's configuration, including its creation timestamp and labels. The interface includes a top navigation bar with 'SUMMARY', 'EVENTS', 'LOGS', and 'TERMINAL' tabs, and a right-hand toolbar with 'SYNC' and 'DELETE' buttons.

KIND	Pod
NAME	gateway-app-f69f6687f-ljr8c
NAMESPACE	dv-gateway
CREATED AT	03/16/2025 01:19:01 (7 days ago)
IMAGES	dvaksacr.azurecr.io/projects/library/gateway-app:latest
STATE	Running
CONTAINER STATE	✔ gateway-app Container is <i>running</i> . It is <i>started and ready</i> .
HEALTH	♥ Healthy
LINKS	

```
1 apiVersion: v1
2 kind: Pod
3 metadata:
4   creationTimestamp: '2025-03-16T00:19:01Z'
5   generateName: gateway-app-f69f6687f-
6   labels:
7     app: gateway-app
```

Metrics observability: Prometheus e Grafana



Prometheus Operator:

Codice sorgente: <https://github.com/prometheus-operator/prometheus-operator>

Linguaggio: GO

Licenza: Apache 2.0

Grafana:

Sorgente: <https://github.com/grafana/grafana>

Linguaggio: GO

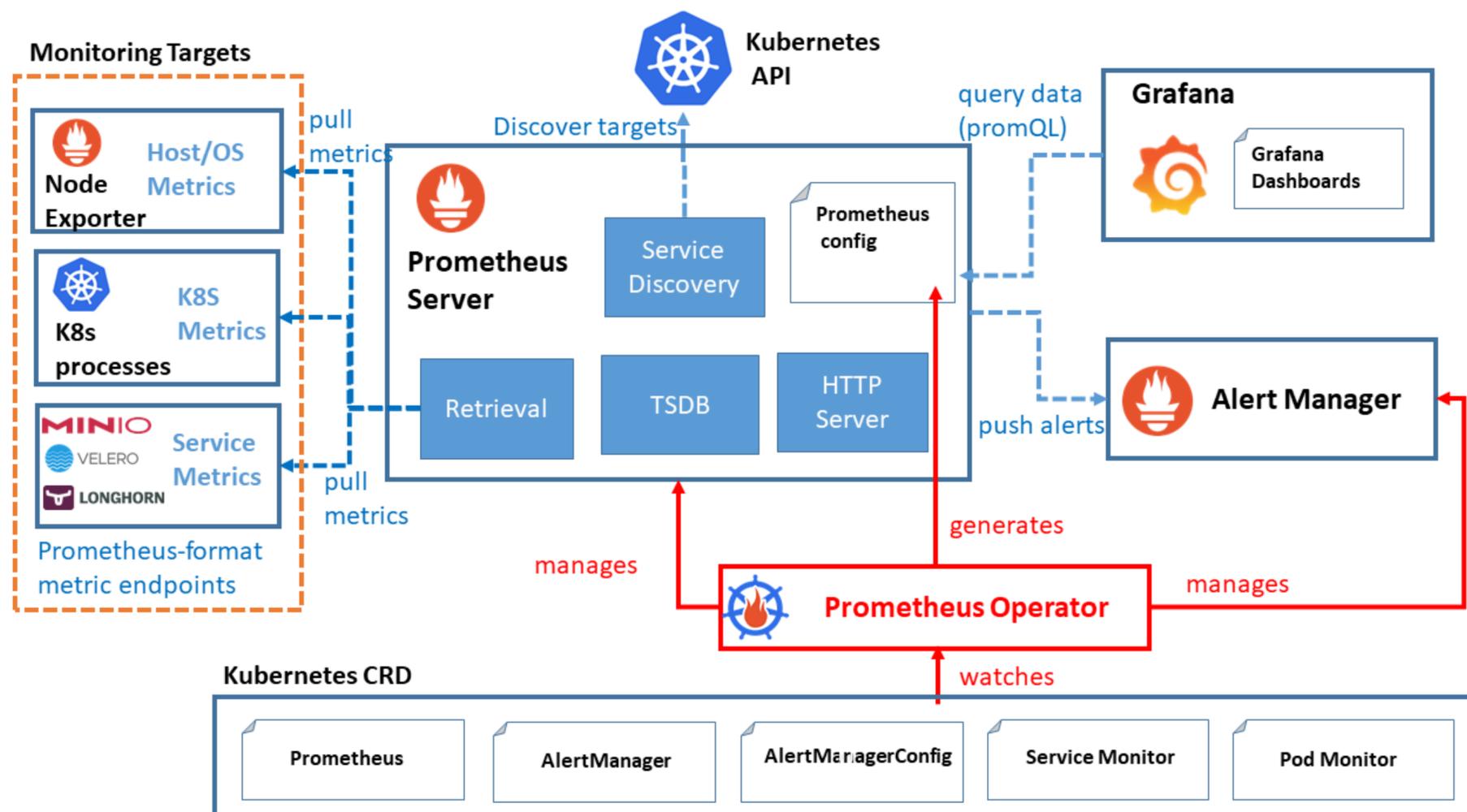
Licenza: AGPL 3.0

Prometheus Operator semplifica enormemente l'installazione, la configurazione e la gestione di Prometheus, il sistema di monitoring open source, all'interno di un cluster Kubernetes.

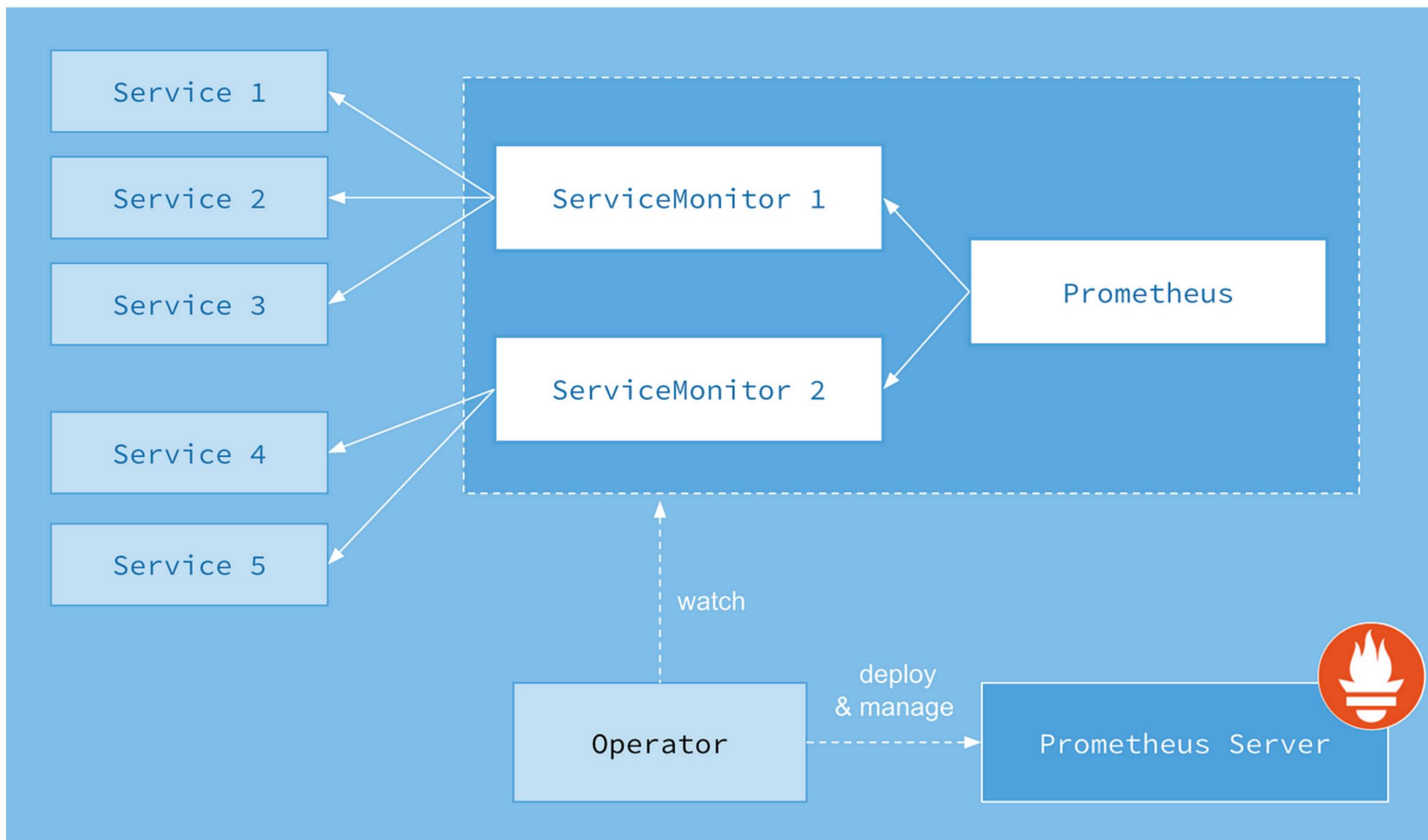
Prometheus Operator Architecture

Mette a disposizione:

- Prometheus (il database Timeseries DB)
- Grafana
- Alert Manager per configurare notifiche basate su metriche
- Node Exporter per ottenere le metriche sistemiche dei pod e dei nodi del cluster
- Svolge la funzione di Metric Server di Kubernetes permettendo di definire risorse di tipo HPA
- Fornisce CRDS per effettuare scraping di metriche aggiuntive



Prometheus Operator: Service Monitor

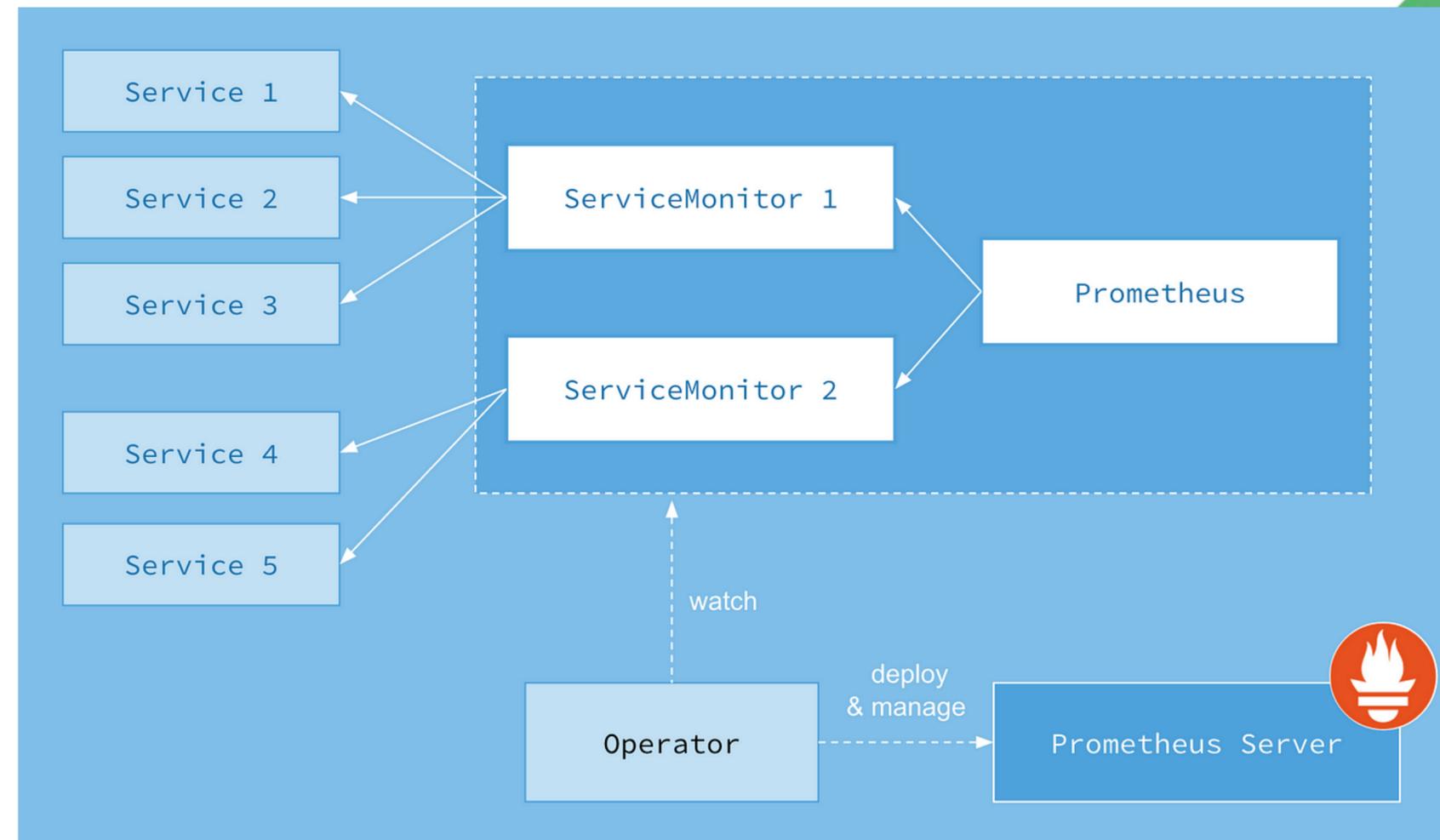


Prometheus Operator: Service Monitor

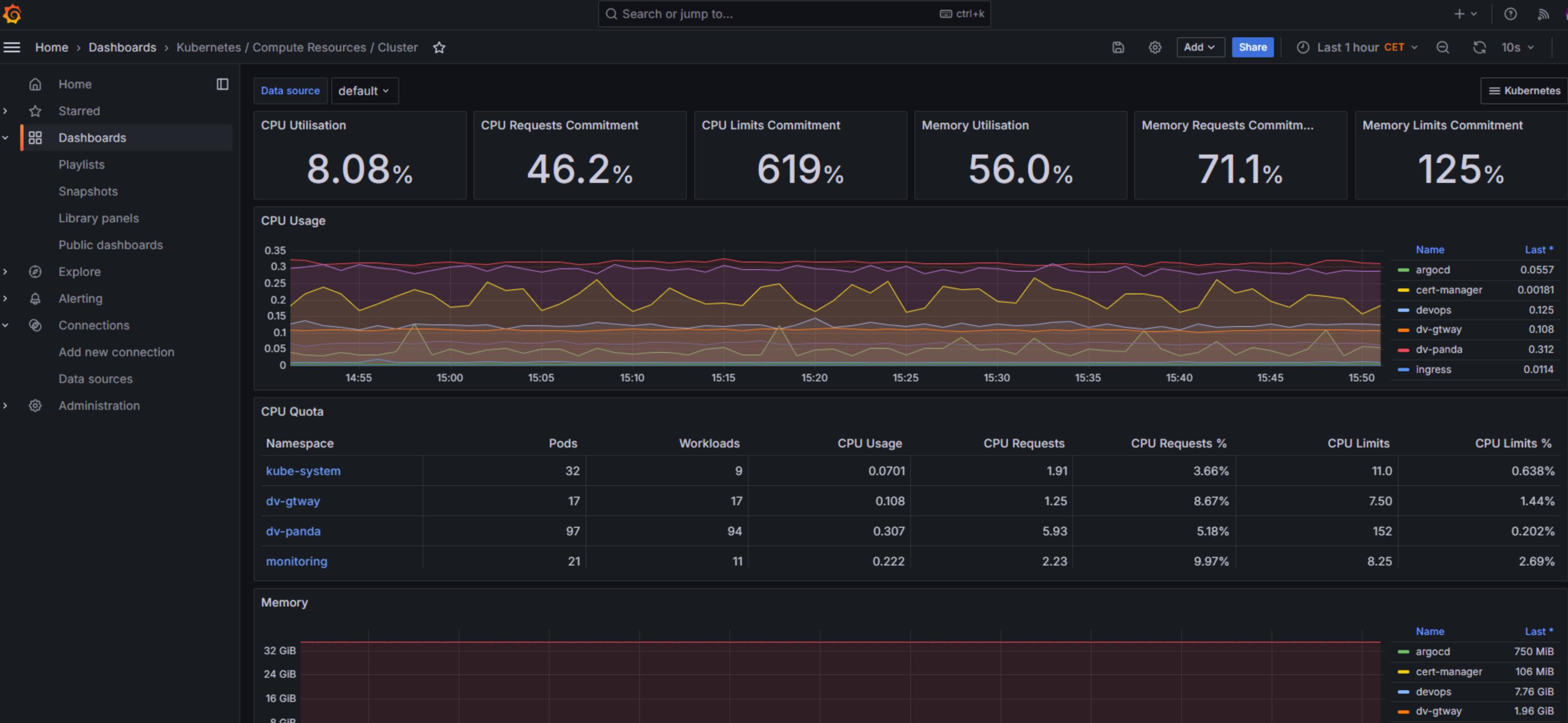
Più in generale permette di generalizzare il recupero di metriche tramite la definizione di ServiceMonitor (o PodMonitor).

Supponendo di aver definito delle metriche custom esposti da una serie di pod in formato prometheus su un endpoint http e identificati nel cluster da un insieme di label.

Possiamo definire un ServiceMonitor affinché su tutti i pod aventi le label specificate nella CRDS venga effettuata periodicamente una chiamata HTTP(S) sull'endpoint indicato per recuperare le metriche.



Grafana: grafici



Observability logs: Opensearch



Codice sorgente: <https://github.com/opensearch-project>

Linguaggio: Java

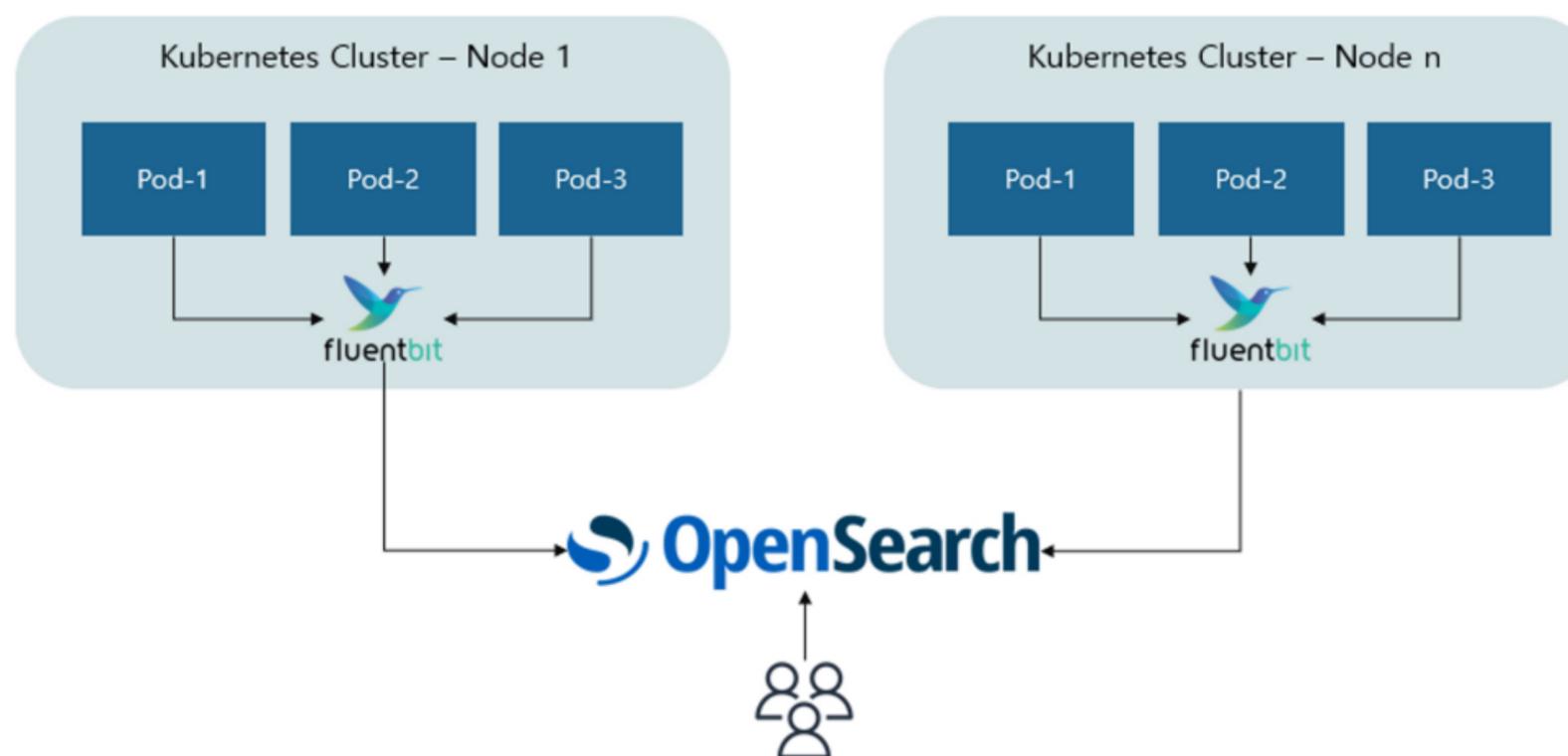
Licenza: Apache 2.0

Opensearch (fork open source di Elasticsearch) offre una piattaforma scalabile per la raccolta, l'analisi e la visualizzazione dei log provenienti dalle applicazioni e dall'infrastruttura Kubernetes.

Sviluppato sotto l'egida della Linux Foundation, dopo il fork ha introdotto una serie di feature non presenti nella versione community tra cui il Single Sign On.

Essenziale per troubleshooting, analisi delle performance e security auditing.

Opensearch & Fluentbit



Affinchè i log applicativi siano presenti su Opensearch è solitamente prassi installare un Daemonset. I più usati sono Fluentd o la sua versione light Fluentbit.

I pod Fluentbit sono abilitati a vedere i file di log di tutti i container del nodo in cui sono installati (anche se le applicazioni container utilizzano il console output, kubernetes conserva i loro log in file di testo). Il programma traccia fino a che punto sono stati letti da esso i file, ne può applicare dei filtri, trasformazioni ed infine caricarli su un sistema esterno come Kafka o Opensearch.

Opensearch

Essendo un database NoSQL offre alcuni diversi vantaggi nella gestione dei log:

- non è necessario avere uno schema fisso, ad esempio se i programmi scrivono i log in JSON possiamo utilizzare tutti i suoi campi per fare ricerche e dashboard senza alcuno sforzo
- tramite repliche e sharding delle collection, dette indici, possiamo scalare orizzontalmente i dati su più nodi Opensearch
- molto veloce per fare ricerche fulltext senza dover per forza specificare in quale campo effettuare la ricerca
- possibilità di definire policy di data retention sui dati (età, dimensione)
- possibilità di definire visualizzazioni e dashboard basati sui dati presenti nei log

Opensearch: ricerca log

OpenSearch Dashboards

Discover New Save Open

dv-panda* ▼ ↻ kubernetes.container_name: "documentale" DQL 📅 Last 15 days

🔍 Search field names 🔇 Filter by type 0 ⊕ Add filter

Selected fields ▼

- message

Popular fields ▼

- app_name
- kubernetes.container_name
- kubernetes.labels.app
- level
- log
- logger_name
- thread_name

Available fields ▼

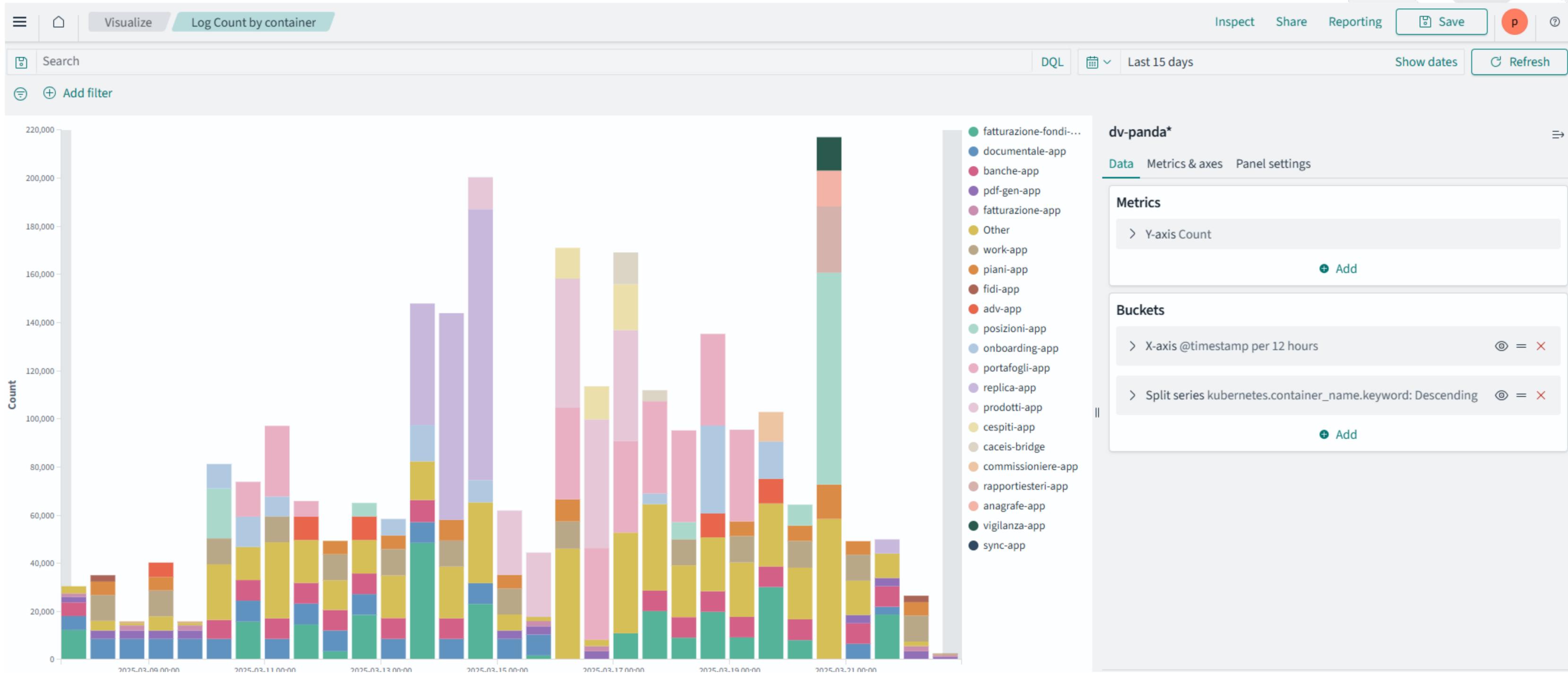
- _id
- _index
- _score
- _type

154,835 hits

Mar 7, 2025 @ 15:59:45.280 - Mar 22, 2025 @ 15:59:45.280 per Auto ▼

Time	message
> Mar 21, 2025 18:06:58.684	This Gauge has been already registered (MeterId{name='kafka.producer.node.request.latency.max', tags=[tag(client.id=kafka-producer-documento-out),tag(1)]}), the Gauge registration will be ignored. Note that subsequent logs will be logged at debug level.
> Mar 21, 2025 18:05:58.246	Created profile protocol number null doc number 350447
> Mar 21, 2025 18:05:57.072	Default profile found: TenantClientConfig(id=3, tenantId=collaudò, clientId=ARXIVAR, profileId=25, profileType=DEVELOP.PROTO, profileTypeId=26, prof

Opensearch: dashboard e grafici



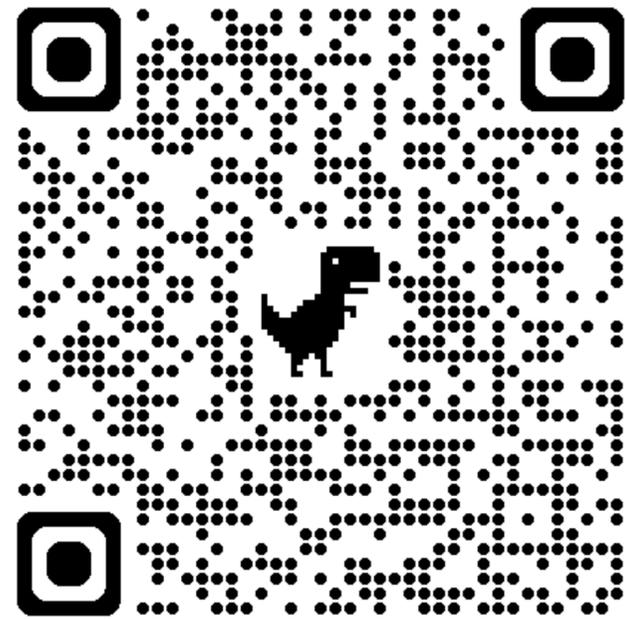


 Google Developer Group Cloud Torino

Devfest Torino



// **Grazie!**
CloudConf



Community Mixer 2023



Amsterdam

-  <https://www.linkedin.com/in/giovanni-forlastro/>
-  <https://twitter.com/mosquitoman81>
-  <https://www.instagram.com/mosquitoman81/>
-  <https://www.linkedin.com/in/francesco-pirrone-c01/>
-  <https://twitter.com/cfrancescop>
-  <https://www.instagram.com/cfrancescop/>

