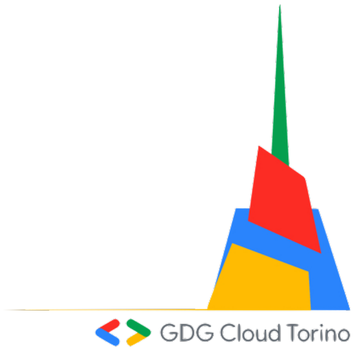
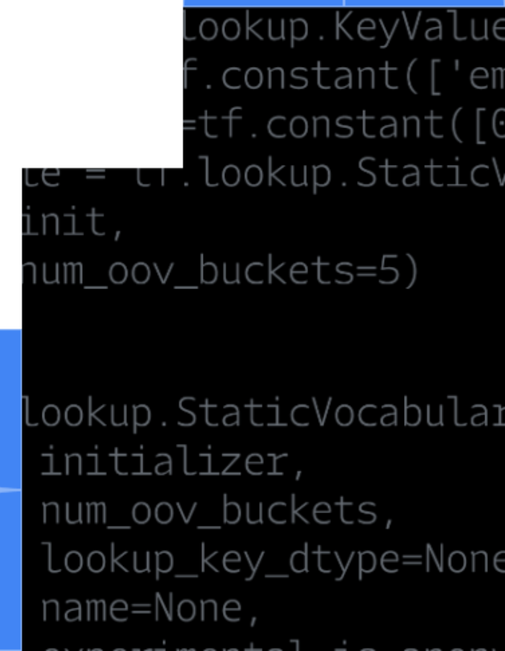
CloudConf

DevOps on Kubernetes: DIY in Open Source

Gianni Forlastro
Francesco Pirrone





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

tag Talent Garden · CloudConf · MUPIN · TOOLBOX · stickermule · LITUS AI manent

GDG Cloud Torino · Flutter Torino

CloudConf

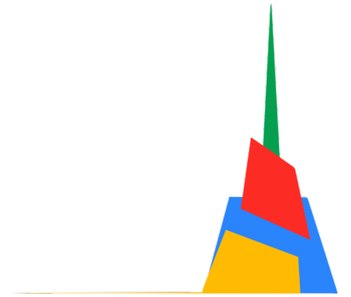
Platform basati su prodotti Open Source

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



CloudConf

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).

CloudConf

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.

CloudConf

Continuous Integration: Gitlab Runner

Codice sorgente: https://gitlab.com/gitlab-org/gitlab-runner
Linguaggio: GO

Licenza: MIT

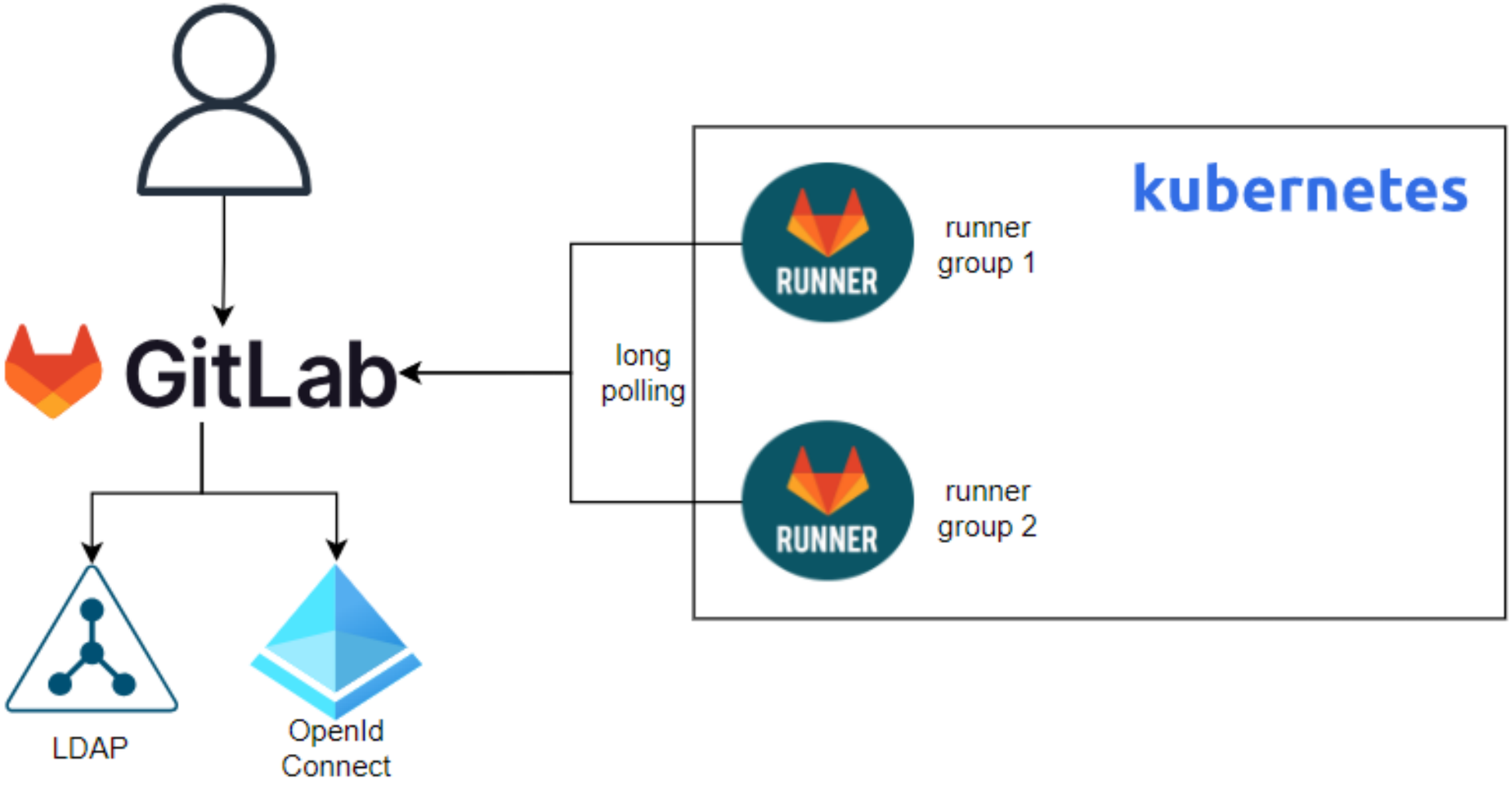
Si integra facilmente con **Gitlab** per eseguire delle build al momento del push remoto.
Le pipeline sono scritte tramite file yaml.

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...)

CloudConf

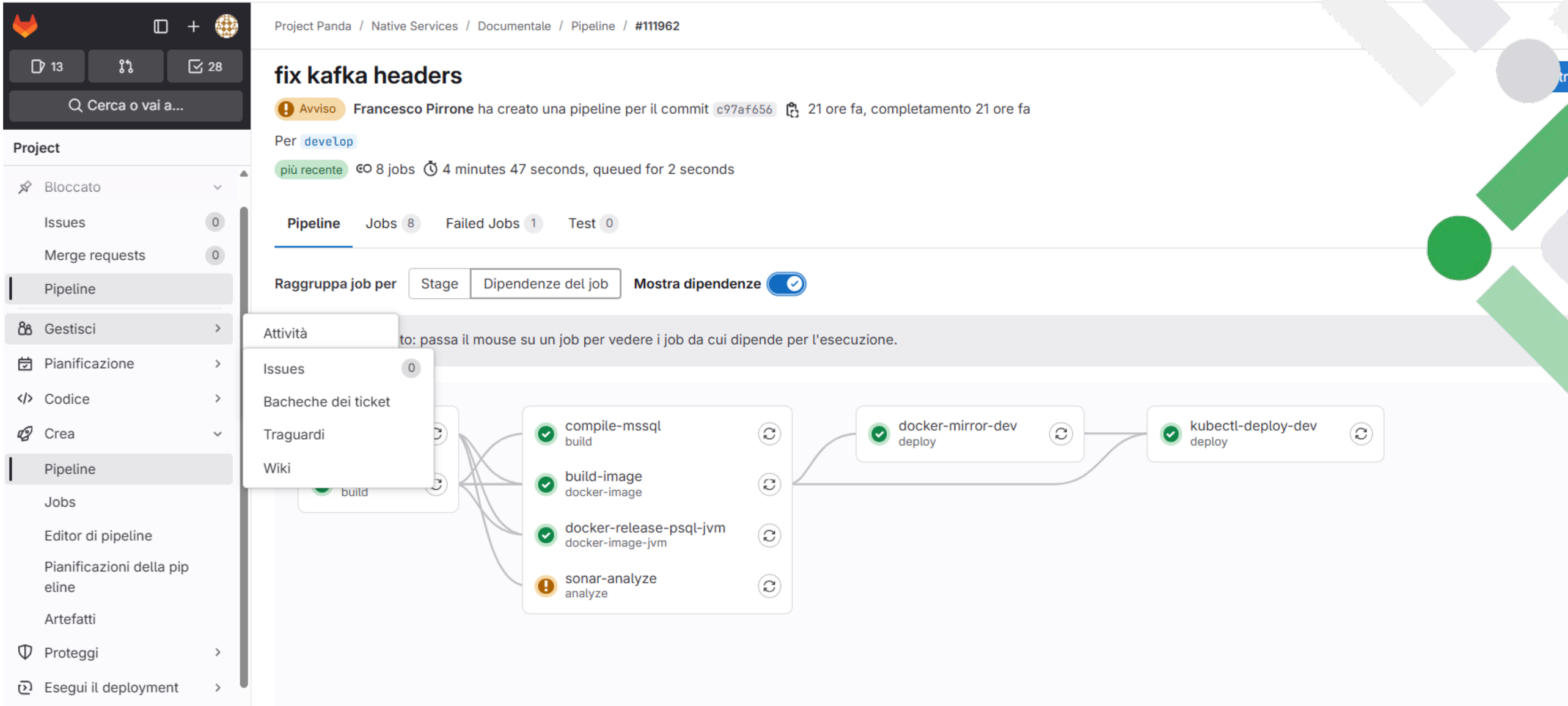
Gitlab Continuous Integration



GDG Cloud Torino

CloudConf

Gitlab Pipeline



CloudConf

Identity Management: Keycloak

Codice sorgente: https://github.com/keycloak/keycloak
Linguaggio: Java

Licenza: Apache 2.0

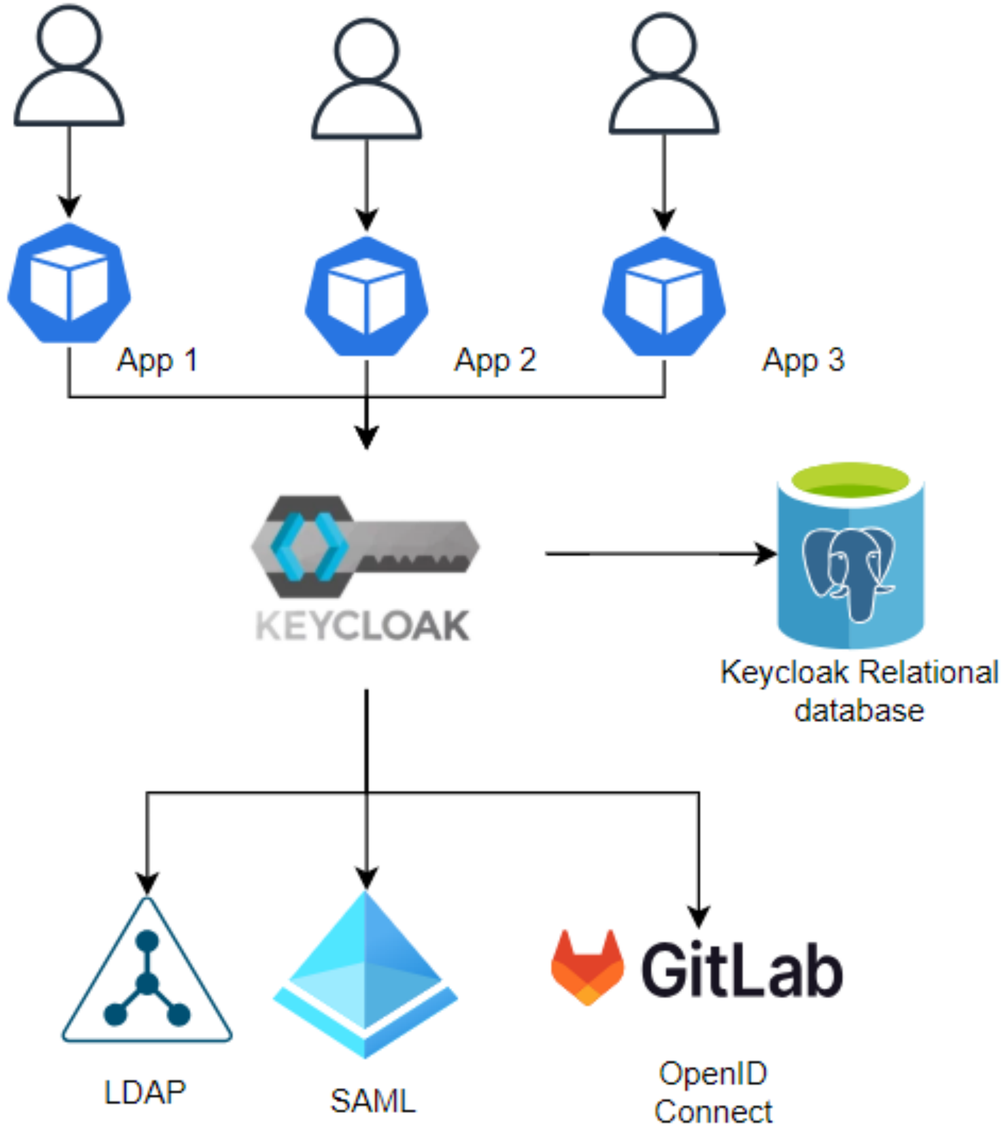
Progetto a livello di Incubating della Cloud Native Computing Foundation, è una soluzione di Identity e Access Management (IAM).

Tra le sue features principali troviamo:

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

CloudConf

IM: Keycloack



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

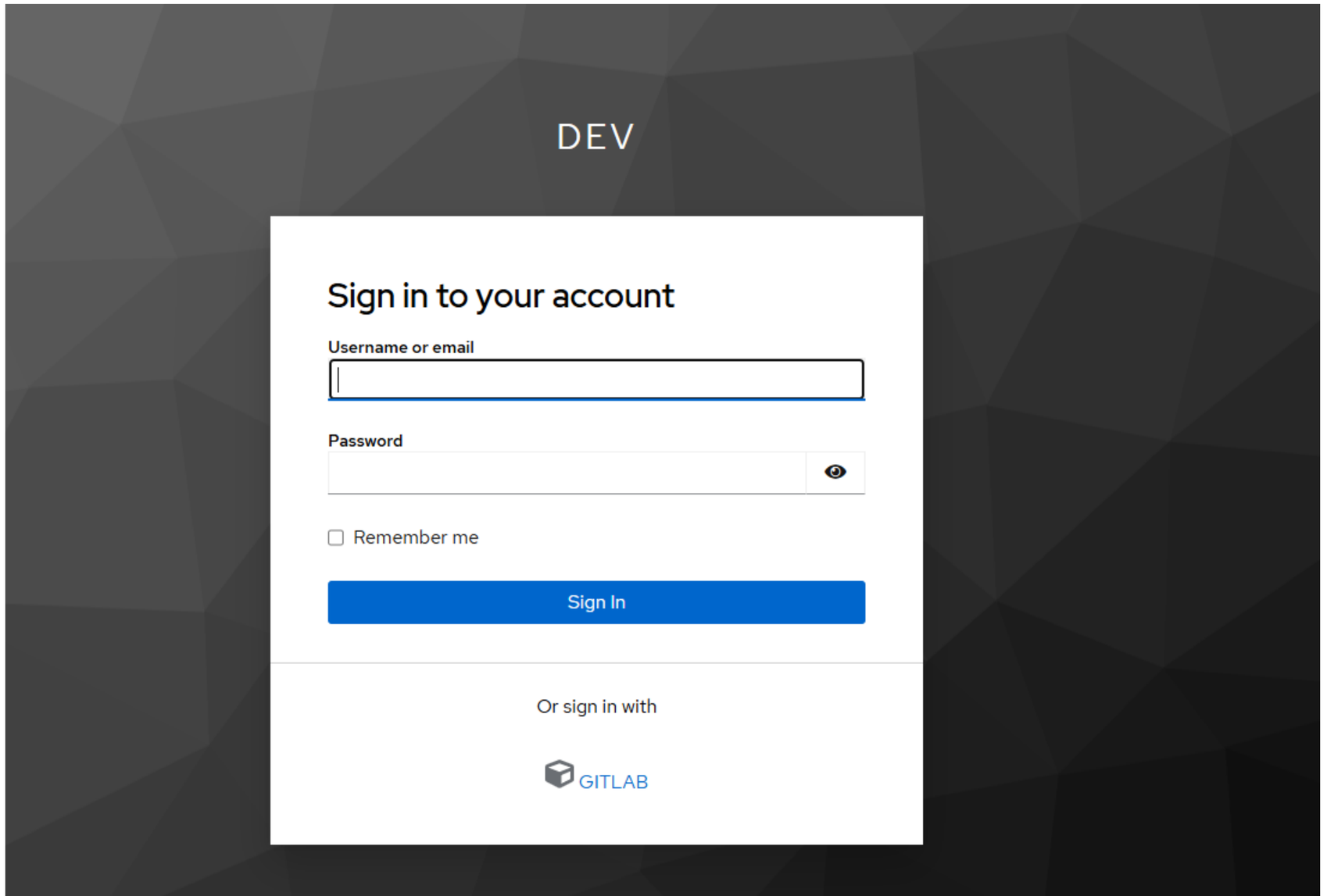
Le utenze possono essere presenti nel database di keycloak oppure è possibile federare con LDAP o altri sistemi di autenticazione basati su SAML o OpenID.

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.

CloudConf

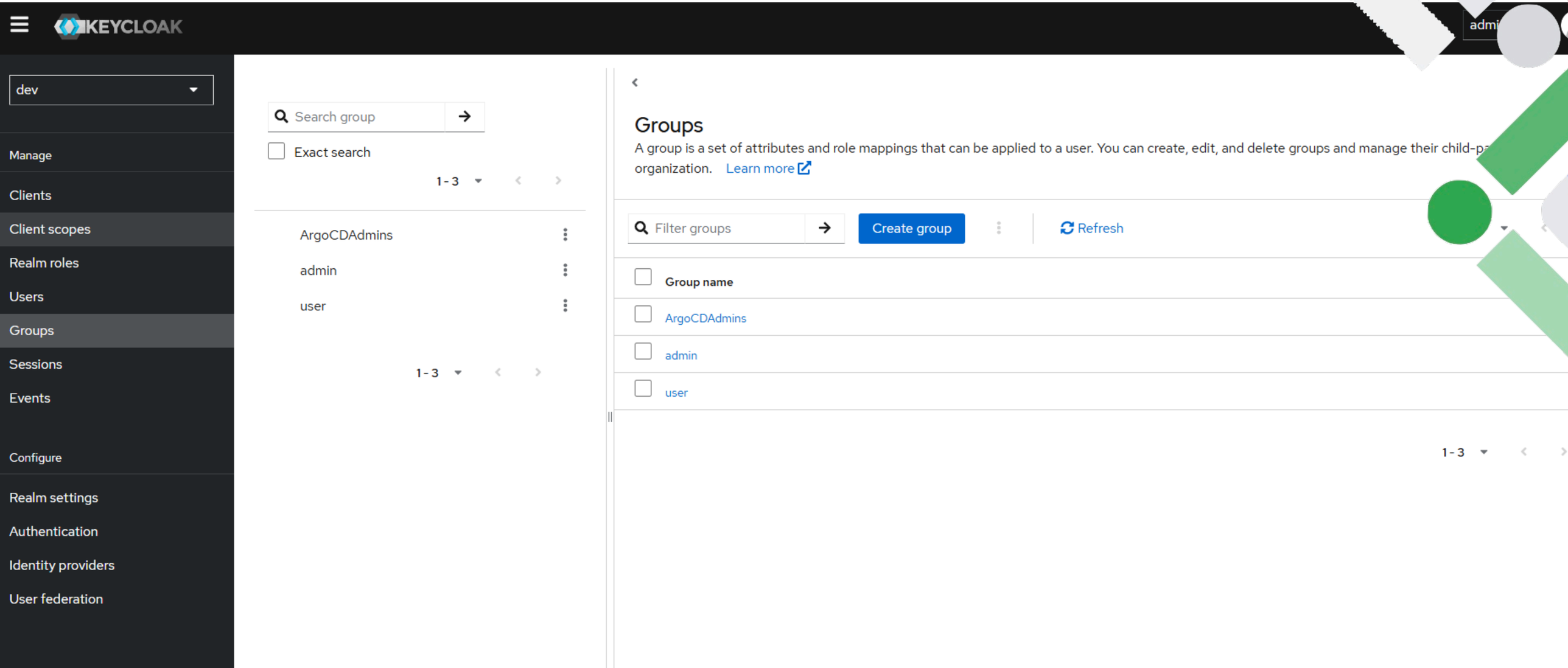
Keycloack : login (personalizzabile)



GDG Cloud Torino

CloudConf

Keycloack : gestione realm



CloudConf

Continuous Delivery : ArgoCD

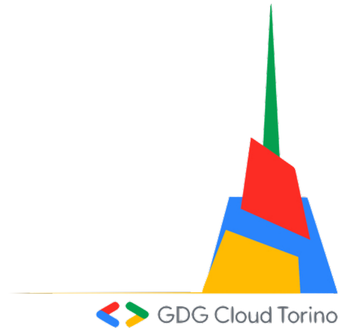
Codice sorgente: https://github.com/argoproj/argo-cd
Linguaggio: GO

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

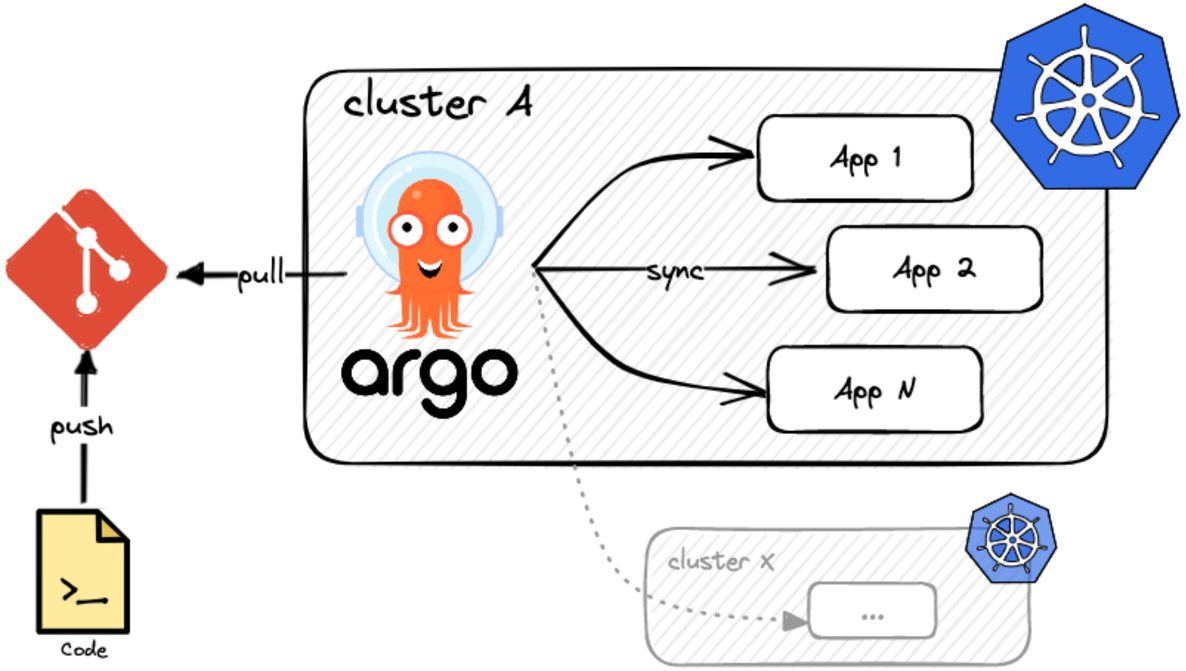


CloudConf

ArgoCD: GitOps

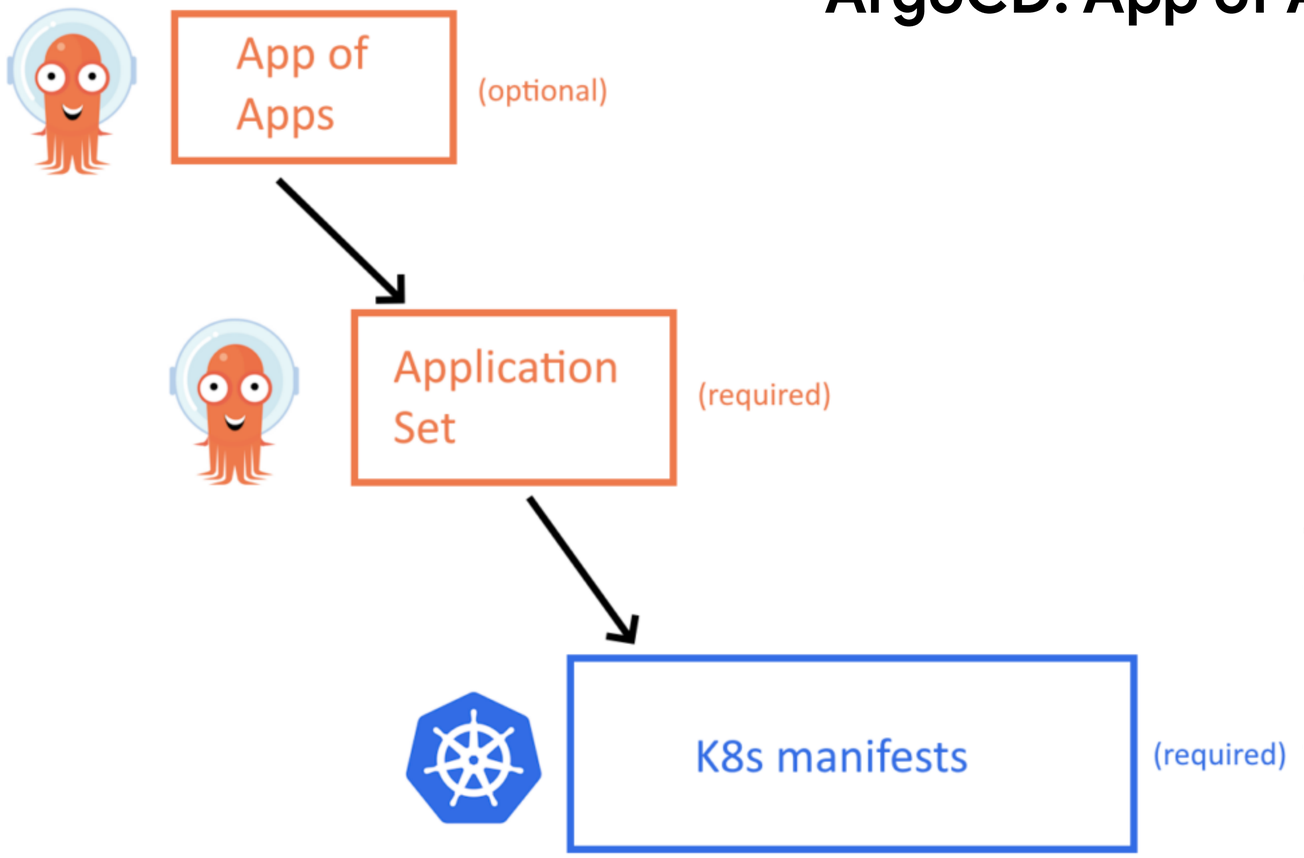
L’approccio GitOps permette di:

- 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



CloudConf

ArgoCD: App of Apps pattern



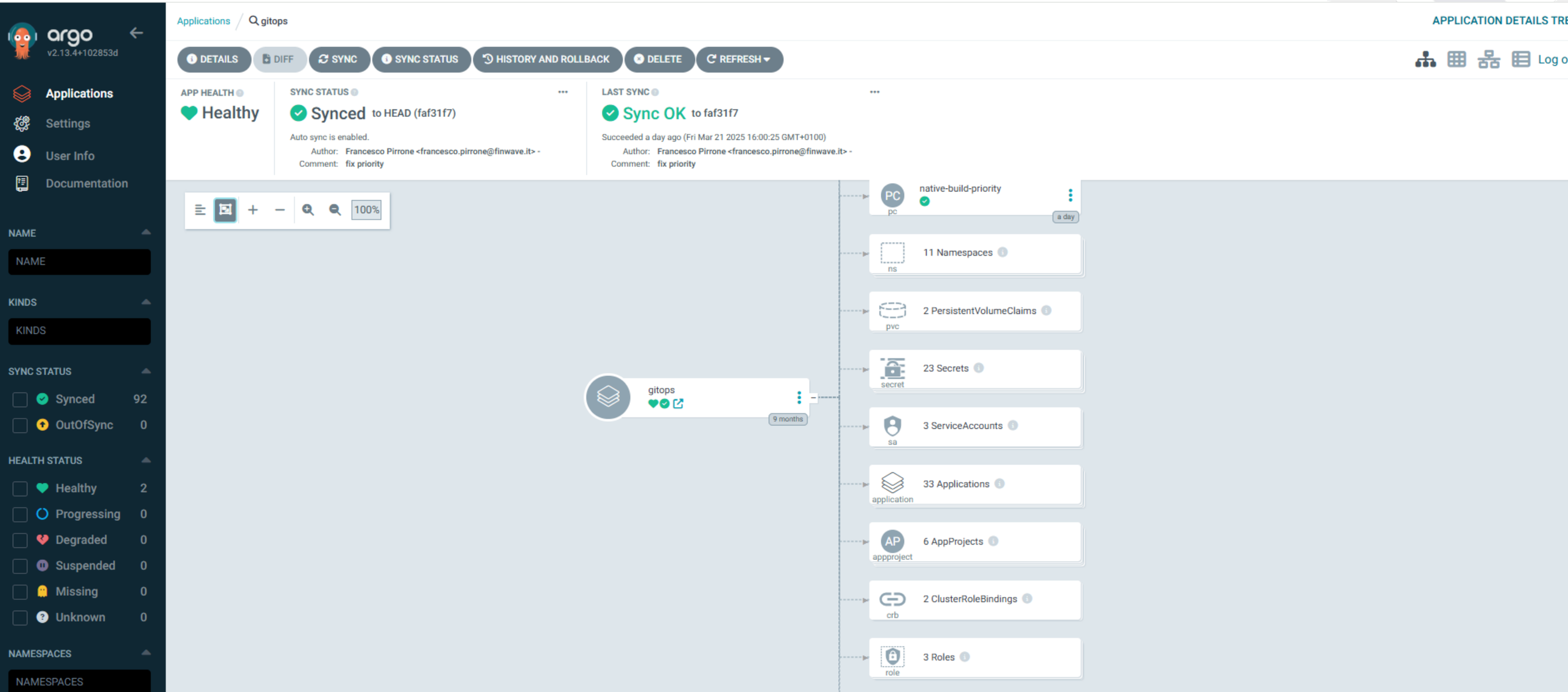
Consiste nel definire un’applicazione che definisce nel repository se stessa e tutte le altre.

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.

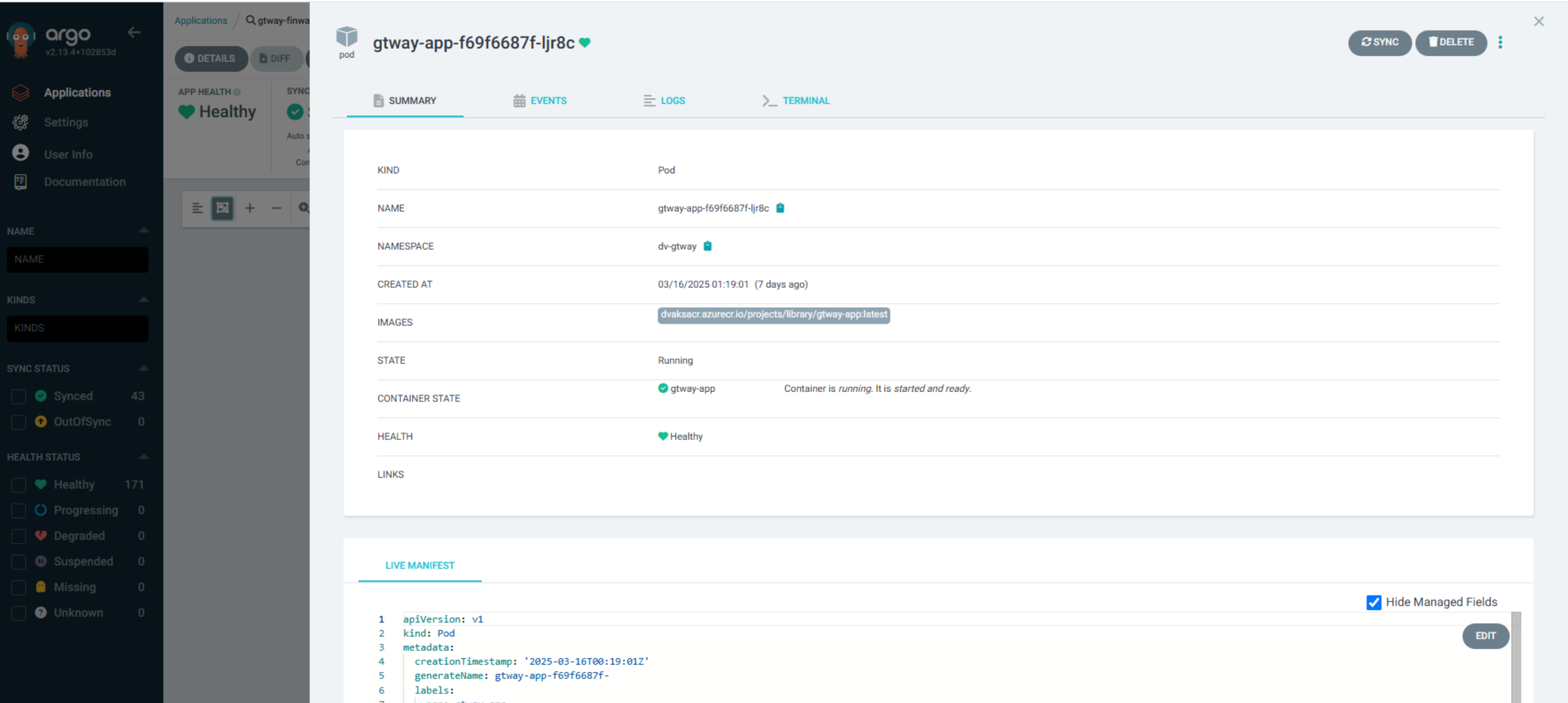
CloudConf

ArgoCD: application status



CloudConf

ArgoCD: POD drilldown



CloudConf

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.

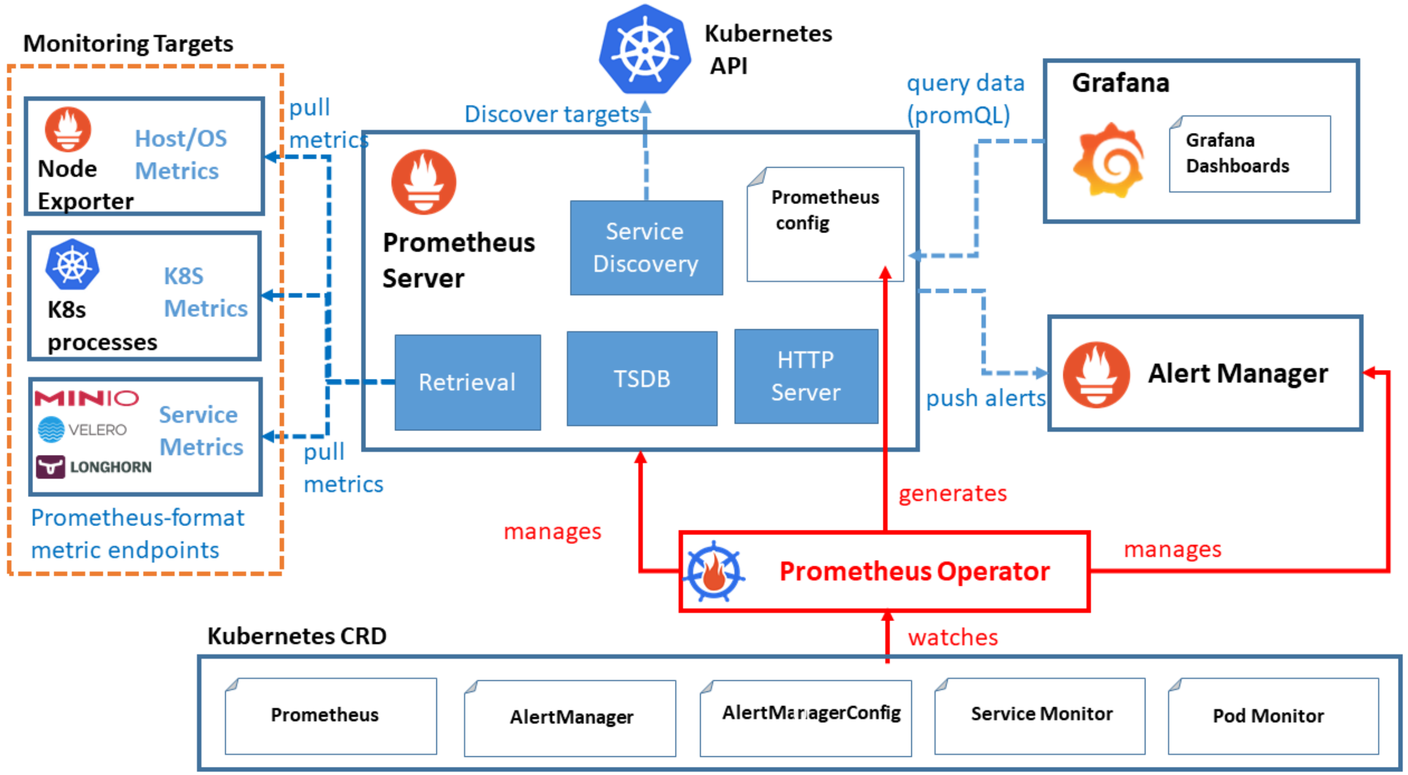
GDG Cloud Torino

CloudConf

Prometheus Operator Architecture

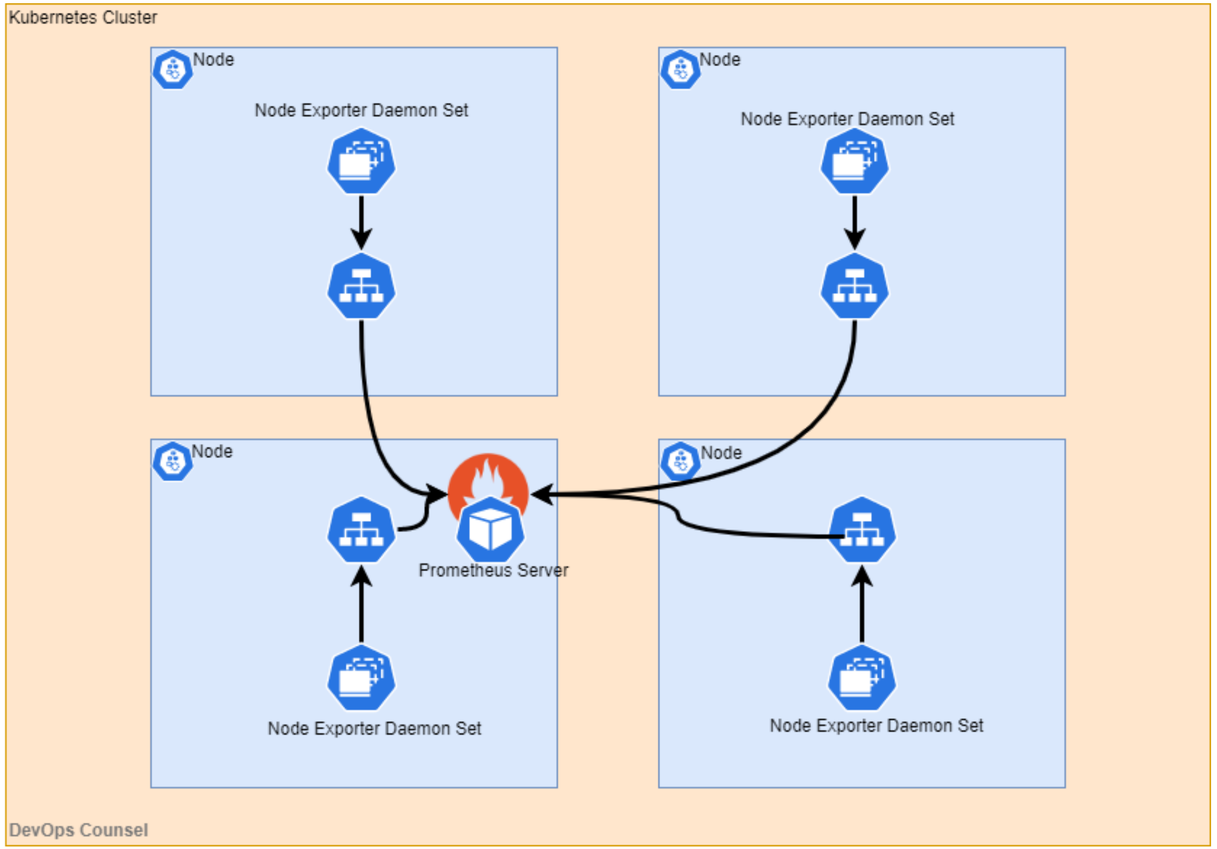
Mette a disposizione:

- Prometheus (il database Timeseries DB)
- Grafana
- Alert Manager per configurare notifiche basate su metriche
- Node Exporter per ottenere le metriche sistemistiche 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



CloudConf

Prometheus: Node Exporter



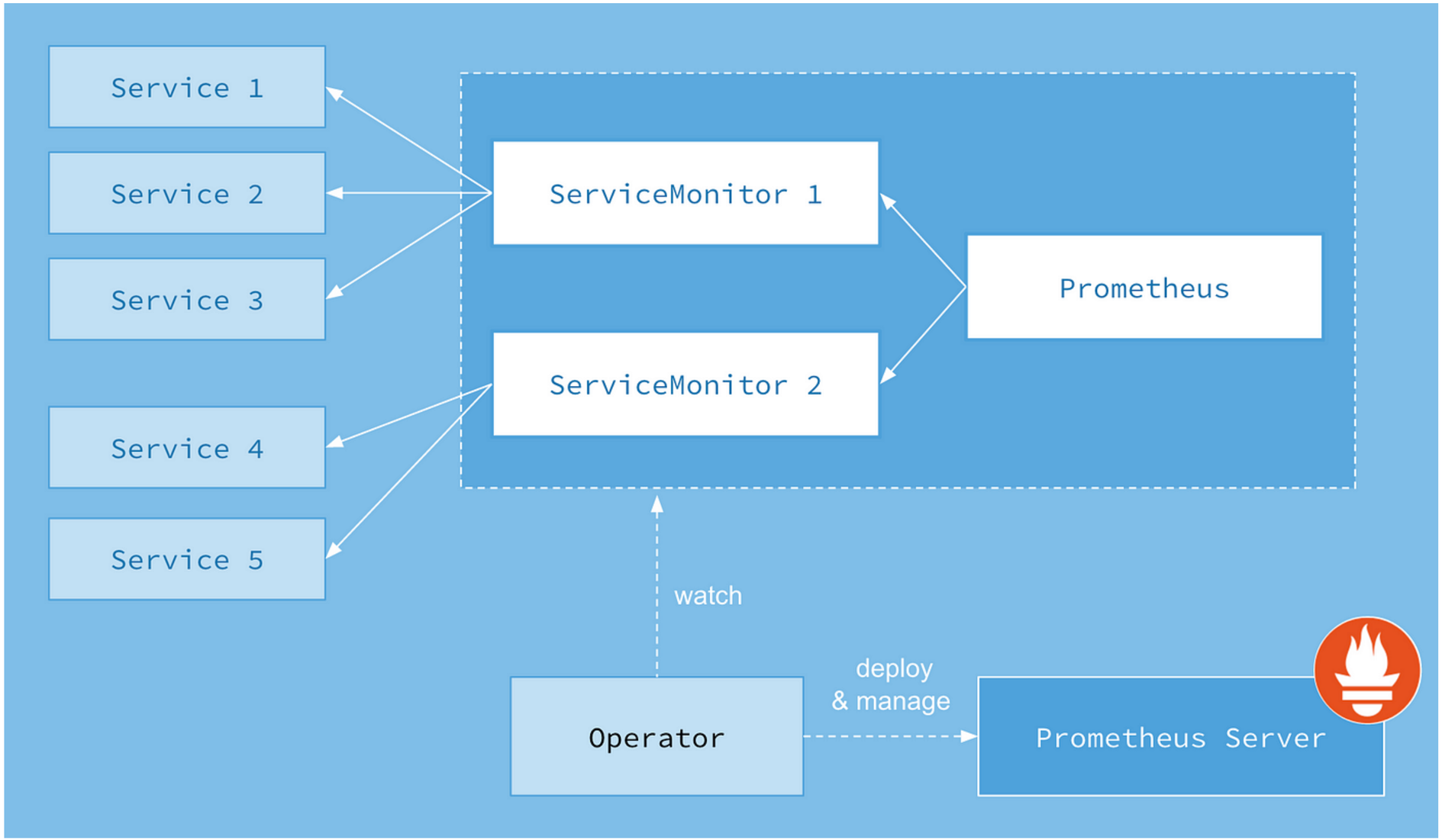
Viene definito un daemon set:

- in ogni nodo del cluster (a meno di taints) viene installato il pod Node Exporter
- il pod ha accesso alle informazioni sistemitiche del nodo (CPU,RAM) e di tutti i pod in esecuzione al suo interno
- Lo scraper di prometheus contatta periodicamente tutti i pod di tipo node exporter, recupera le metriche e le salva nel database

GDG Cloud Torino

CloudConf

Prometheus Operator: Service Monitor



GDG Cloud Torino

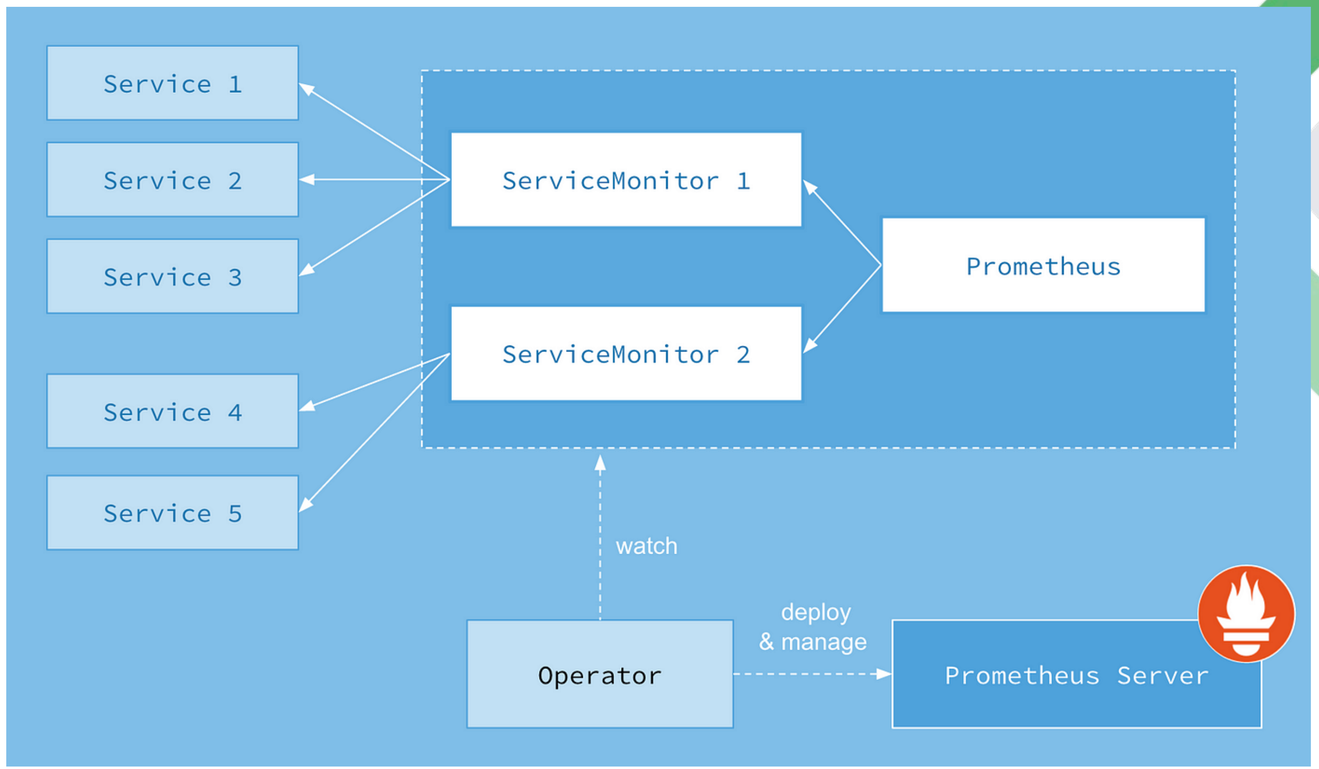
CloudConf

Prometheus Operator: Service Monitor

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

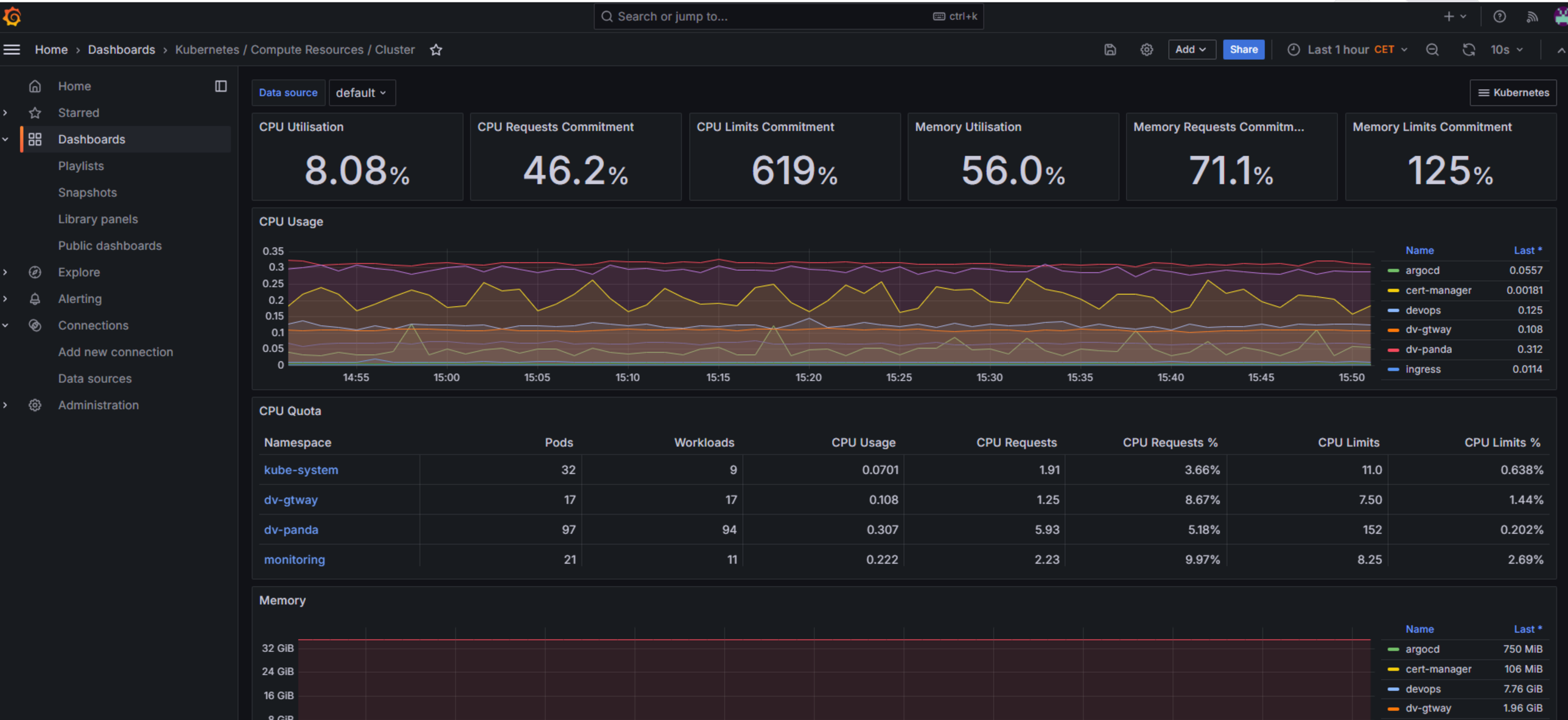
Supponendo di aver definito delle metriche custom esposti da una serie di pod in formato prometheus su un endpoint http e identificati nel cluster da un insieme di label.

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



CloudConf

Grafana: grafici



CloudConf

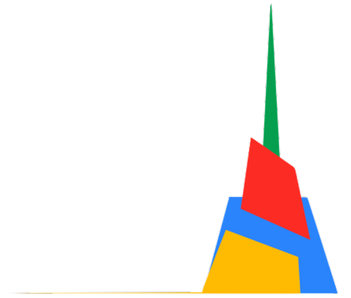
Observability logs: Opensearch

Codice sorgente: https://github.com/opensearch-project
Linguaggio: Java

Licenza: Apache 2.0

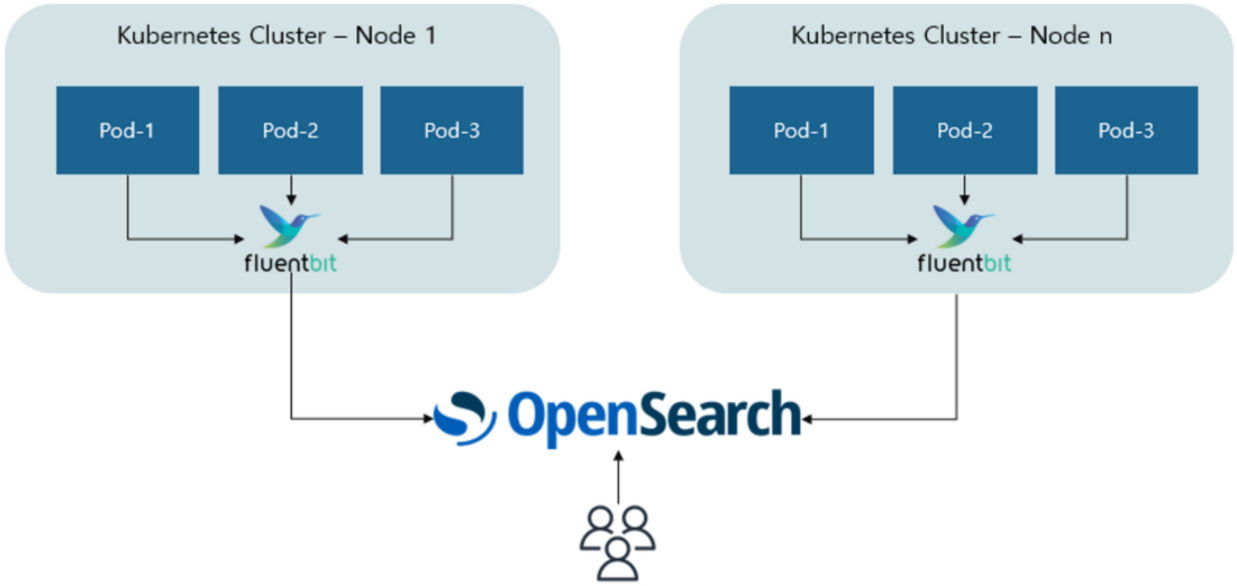
Opensearch (fork open source di Elasticsearch) offre una piattaforma scalabile per la raccolta, l'analisi e la visualizzazione dei log provenienti dalle applicazioni e dall'infrastruttura Kubernetes.
Sviluppato sotto l’egida della Linux Foundation, dopo il fork ha introdotto una serie di feature non presenti nella versione community tra cui il Single Sign On.

Essenziale per troubleshooting, analisi delle performance e security auditing.



CloudConf

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.

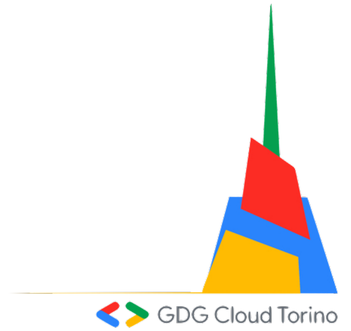
I pod Fluentbit sono abilitati a vedere i file di log di tutti i container del nodo in cui sono installati (anche se le applicazioni container utilizzano il console output, kubernetes conserva i loro log in file di testo). Il programma traccia fino a che punto sono stati letti da esso i file, ne può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

CloudConf

Opensearch

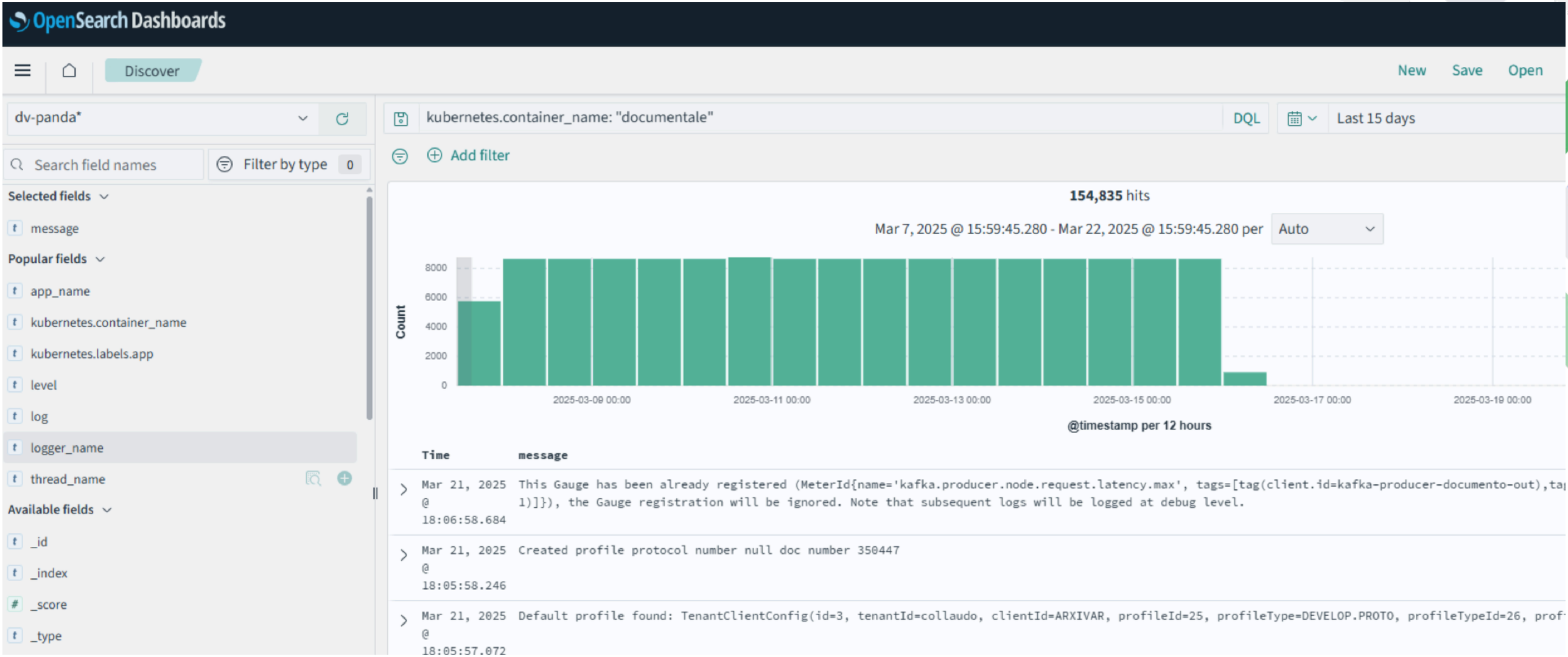
Essendo un database NoSQL offre alcuni diversi vantaggi nella gestione dei log:

- 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



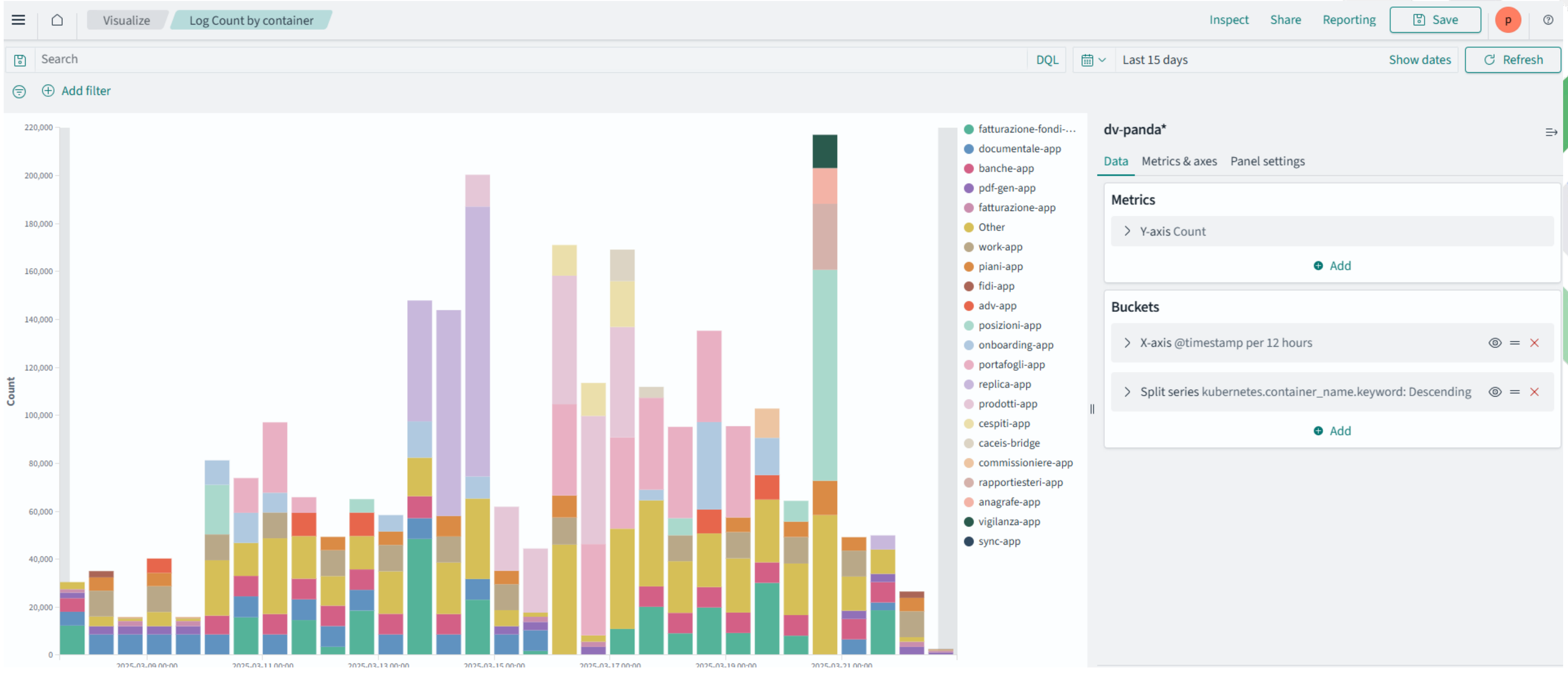
CloudConf

Opensearch: ricerca log



CloudConf

Opensearch: dashboard e grafici



CloudConf

Grazie!

Google Developer Group Cloud Torino

Devfest Torino

GDG Cloud Torino

Grazie!

CloudConf

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/

GDG Cloud Torino

Flutter Torino