RdS
RICERCA DI SISTEMA

ENEA
AGENZIA NAZIONALE PER LE NUOVE TECNOLOGIE, L'ENERGIA E LO SVILUPPO ECONOMICO SOSTENIBILE

MINISTERO DELL'AMBIENTE E DELLA SICUREZZA ENERGETICA

Ricerca di Sistema elettrico

Servizi informatici per il potenziamento della piattaforma CI-RES (LA2.15)

Stefano Sibilia

SPARKFABRIK

RdS_PTR22-24_PR1.7_LA2.15_369

Servizi informatici per il potenziamento della piattaforma CI-RES

Stefano Sibilia, Sparkfabrik srl

LA2.15 - CI-RES: Realizzazione della Piattaforma di servizi e miglioramento della fruibilità della piattaforma ed apertura ad utenti esterni istituzionali e operatori

Dicembre 2024

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dell'Ambiente e della Sicurezza Energetica - ENEA Piano Triennale di Realizzazione 2022-2024

Obiettivo: Decarbonizzazione

Progetto: Tema di ricerca 1.7 – Tecnologie per la penetrazione efficiente del vettore elettrico negli usi finali

Linea di attività: LA2.15

Responsabile del Progetto: Claudia Meloni, ENEA

Responsabile del Work Package: Claudia Meloni, ENEA

Responsabile Linea di Attività: Gregorio D'Agostino, ENEA

Mese inizio previsto: 19

Mese inizio effettivo: 19

Mese fine previsto: 36

Mese fine effettivo: 36

Il presente documento descrive le attività di ricerca svolte all'interno contratto dal titolo: "Servizi Informatici per il potenziamento della piattaforma CI-RES"

Indice

1 Risultati attesi 4
2 Risultati ottenuti 5
3 Prodotti attesi 6
4 Prodotti sviluppati 7
5 Analisi degli scostamenti su attività e risultati 8
6 Sintesi delle attività svolte 9
7 Dettaglio delle attività svolte 10
2.1 Deploy di DJAM (Django Authentication Module) 10
2.2 Deploy di GeoNode 10
2.3 Deploy di GeoServer 10
2.4 Deploy di OpenQuake 11
2.5 Deploy di WNRT 11
2.6 Deploy dell'Interfaccia Web per WNRT 11
2.7 Configurazione NGINX Ingress Controller 12
2.8 Infrastruttura CI/CD e Componenti Riutilizzabili 12
2.9 Componenti GitLab Standardizzati 12
14 Contributo delle eventuali consulenze alle attività sopra descritte 13
15 Pubblicazioni scientifiche 14
16 Eventi di disseminazione 15

1 Risultati attesi

Reimplementazione della piattaforma CI-RES seguendo i principi moderni dell'architettura a microservizi, orchestrata attraverso un cluster Kubernetes (ICSk8s).

L'architettura si articola su tre livelli principali, ciascuno con responsabilità ben definite:

1. **Livello Back-end**:
    - Dedicato all'elaborazione dei servizi di calcolo e dei workflow operativi
    - Gestione delle operazioni computazionalmente intensive
    - Elaborazione dei dati e analisi del rischio
    - Implementazione degli algoritmi di simulazione
2. **Livello Core**:
    - Gestione centralizzata dei dati e dei servizi fondamentali
    - Coordinamento delle comunicazioni tra i diversi moduli
    - Gestione della persistenza e della consistenza dei dati
    - Implementazione dei servizi di base condivisi
3. **Livello Front-end**:
    - Implementazione delle interfacce utente
    - Visualizzazione dei dati e delle analisi
    - Gestione delle interazioni con gli utenti
    - Presentazione dei risultati delle elaborazioni

L'infrastruttura tecnologica si basa su un stack moderno e robusto che include:

- Cluster Kubernetes (ICSk8s) per l'orchestrazione dei servizi containerizzati
- PROXMOX come piattaforma di virtualizzazione enterprise
- CEPH per la gestione distribuita e resiliente dello storage
- GitLab on-premise per il controllo completo del ciclo di vita del software

2 Risultati ottenuti

Implementazione su piattaforma proxmox/kubernetes dei seguenti applicativi con integrazione di opportune pipeline di deploy per produzione e sviluppo.

- Djam
- Geonode
- Geoserver
- Openquake
- WNRT
- Interfaccia web WNRT

Contestualmente sono stati realizzati dei component di ci/cd riutilizzabili per futuri progetti.

3 Prodotti attesi

Di seguito i prodotti attesi:

- pipeline e implementazione Djam su kubernetes
- pipeline e implementazione Geonode su kubernetes
- pipeline e implementazione Geoserver su kubernetes
- pipeline e implementazione Openquake su kubernetes
- pipeline e implementazione WNRT su kubernetes
- pipeline e implementazione Interfaccia web WNRT su kubernetes.

4. Prodotti sviluppati

Sono stati sviluppati tutti i prodotti previsti:

- pipeline e implementazione Djam su kubernetes
- pipeline e implementazione Geonode su kubernetes
- pipeline e implementazione Geoserver su kubernetes
- pipeline e implementazione Openquake su kubernetes
- pipeline e implementazione WNRT su kubernetes
- pipeline e implementazione Interfaccia web WNRT su kubernetes

Di seguito i prodotti sviluppati, in più oltre la previsione originale:

- component Gitlab per build docker
- component Gitlab per deploy Helm Chart
- Helm chart generico.

5. Analisi degli scostamenti su attività e risultati

Non ci sono stati scostamenti di rilevo.

6. Sintesi delle attività svolte

Porting della piattaforma CI-RES in ambiente kubernetes/proxmox al fine di aumentare la resilienza del servizio grazie alle peculiarità di kubernetes.
Contemporaneamente l'ambiente permette di semplificare, riducendo i tempi di sviluppo e deploy, variazioni dell'architettura software.

7. Dettaglio delle attività svolte

7.1 Deploy di DJAM (Django Authentication Module)

Il modulo DJAM rappresenta il componente centrale per la gestione dell'autenticazione e dell'autorizzazione all'interno della piattaforma CI-RES. L'implementazione, realizzata come microservizio nel cluster ICSk8s, è stata progettata per garantire massima sicurezza e affidabilità.

Il sistema, basato sul framework Django, è stato configurato per fornire:

- Un sistema centralizzato di autenticazione per tutti i servizi della piattaforma
- Gestione granulare dei permessi e dei ruoli utente
- Integrazione seamless con gli altri componenti dell'infrastruttura
- Scalabilità orizzontale per gestire picchi di richieste
- Alta disponibilità attraverso meccanismi di replica e failover

○ Deploy di GeoNode

GeoNode rappresenta il cuore della gestione dei dati geografici all'interno della piattaforma. La sua implementazione come servizio Kubernetes è stata ottimizzata per gestire grandi volumi di dati geospaziali mantenendo alte performance e affidabilità.

L'implementazione si caratterizza per:

- Un sistema robusto di gestione dei dati geografici e metadati associati
- Integrazione profonda con GeoServer per la pubblicazione dei servizi cartografici
- Sistema distribuito di caching per ottimizzare le performance
- Gestione degli accessi integrata con DJAM per garantire la sicurezza dei dati
- Interfacce standardizzate per l'interoperabilità con altri sistemi GIS

○ Deploy di GeoServer

Il deployment di GeoServer, componente fondamentale per la pubblicazione dei servizi cartografici, è stato realizzato con particolare attenzione alle performance e alla stabilità. Il servizio è stato configurato per operare in modo ottimale all'interno dell'infrastruttura Kubernetes, garantendo:

- Pubblicazione efficiente dei servizi geografici secondo standard OGC
- Gestione ottimizzata delle risorse computazionali
- Sistema robusto di persistenza dei dati attraverso CEPH
- Integrazione completa con il sistema di autenticazione DJAM
- Configurazione avanzata per il load balancing e la scalabilità

7.2 Deploy di OpenQuake

OpenQuake, sistema specializzato per il calcolo del rischio sismico, è stato integrato nel cluster ICSk8s con particolare attenzione alle esigenze computazionali. L'implementazione si distingue per:

- Ottimizzazione delle risorse di calcolo attraverso scheduling intelligente
- Integrazione profonda con il sistema di storage distribuito CEPH per la gestione efficiente dei dati
- Sistema avanzato di gestione delle code di calcolo
- Monitoraggio dettagliato delle performance e dello stato dei calcoli
- Controllo degli accessi tramite integrazione con DJAM

7.3 Deploy di WNRT

Il deployment di WNRT (Wide Networks Resilience Tool), strumento specializzato per la simulazione delle reti di distribuzione, rappresenta un elemento chiave nella piattaforma. L'implementazione come servizio Kubernetes è stata ottimizzata per garantire:

- Esecuzione efficiente delle simulazioni in ambiente containerizzato
- Gestione automatizzata delle risorse computazionali
- Sistema robusto di input/output attraverso l'integrazione con CEPH
- Scalabilità dinamica basata sul carico di lavoro delle simulazioni
- Logging centralizzato per il monitoraggio delle operazioni
- Gestione affidabile dello stato delle simulazioni

7.4 Deploy dell'Interfaccia Web per WNRT

La predisposizione dell'ambiente per l'interfaccia di gestione remota di WNRT è stata realizzata con particolare attenzione alla futura implementazione da parte del cliente. L'architettura predisposta include:

- Infrastruttura base configurata secondo best practice Kubernetes
- Sistema di autenticazione integrato con DJAM per la gestione sicura degli accessi
- Interfacce API RESTful per la comunicazione efficiente con il core WNRT
- Configurazione degli endpoint necessari per le operazioni di gestione
- Struttura modulare per facilitare future estensioni

7.5 Configurazione NGINX Ingress Controller

L'implementazione dell'Ingress Controller NGINX rappresenta un componente cruciale per la gestione del traffico in ingresso al cluster. Il sistema è stato configurato per garantire:

- Gestione sicura e centralizzata dei certificati ENEA

- Implementazione di regole di routing avanzate per i diversi servizi
- Politiche di sicurezza stringenti per il controllo degli accessi
- Configurazione ottimizzata del load balancing
- Monitoraggio dettagliato del traffico
- Gestione efficiente delle connessioni SSL/TLS

7.6 Infrastruttura CI/CD e Componenti Riutilizzabili

Un aspetto fondamentale del progetto è stata l'implementazione di una robusta infrastruttura CI/CD, basata su componenti generici e riutilizzabili disponibili nel repository dedicato (https://10.196.1.1/infrastructure/ci-cd/components/).

7.7 Componenti GitLab Standardizzati

- Sistema automatizzato per build e push delle immagini Docker
- Template standardizzati per le pipeline CI/CD
- Helm chart generico per standardizzare i deployment Kubernetes
- Configurazioni predefinite per i vari ambienti di deployment

Questo approccio ha portato significativi benefici:

a) Standardizzazione completa dei processi di deployment
b) Riduzione significativa della duplicazione del codice
c) Semplificazione delle attività di manutenzione
d) Coerenza garantita nelle configurazioni tra i diversi servizi
e) Accelerazione nei tempi di implementazione di nuovi servizi
f) Maggiore affidabilità dei processi di deployment

8 Contributo delle eventuali consulenze alle attività sopra descritte

Non ci si è avvalsi di nessuna consulenza.

9 Pubblicazioni scientifiche

L’attività non prevede la creazione di pubblicazioni.

10 Eventi di disseminazione

Nessun evento previsto.