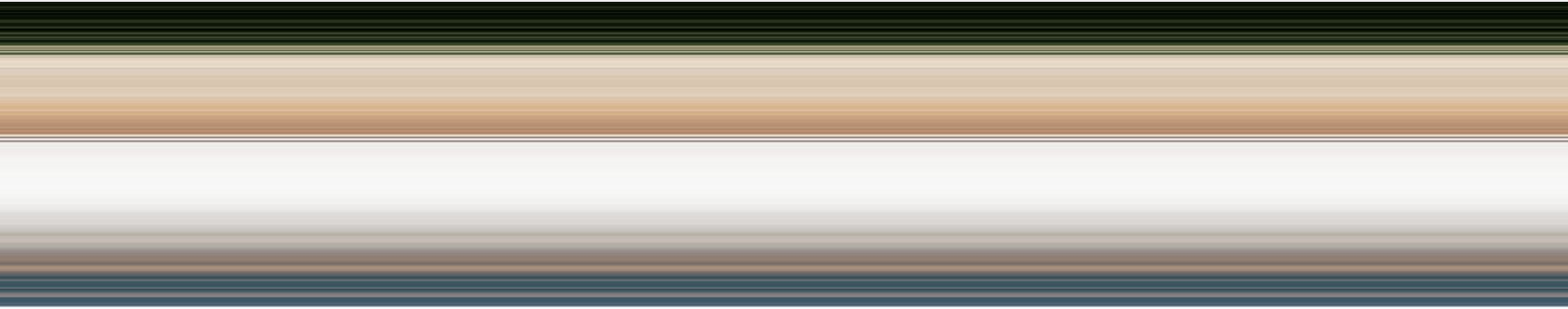
CARMINE DI MONACO - LINUX DAY 2023 MILANO

SUSE

Ambienti Effimeri per Pull Requests

Un Viaggio da GitHub alla nostra macchina senza passare per Kubernetes



Ambienti effimeri per ogni PR - Perché'?

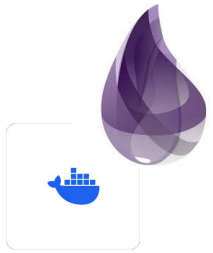
- Avere un ambiente di "staging" per ogni feature che sviluppiamo
- Mostrare ad altre funzioni e al PO i progressi delle features
- Integrarci con workload reali di test, per ogni singola feature
- Avere un feedback cycle sulla UI più' immediato

Nella prima implementazione ci siamo concentrati sulle applicazioni Elixir, in particolare sul progetto web

Cos'e' Trento?

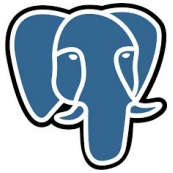
Suite di monitoring e policy checking per workload SAP

web



wanda

agent



demo.trento-project.io

Opzione K - Cluster Kubernetes

PRO

Letteratura e documentazione ampiamente presente online

Il problema si cala perfettamente nello strumento

Implementazione “naive”, un namespace per ogni PR

Implementazione più’ complessa, setup multitenant con strumenti dedicati

Utilizzo di operator per orchestrare le dipendenze in modo dichiarativo

X

CONTRO

Costi infrastruttura elevati

Know-How specialistico per amministrare il cluster

Know-How specialistico per orchestrare le dipendenze all’interno di un cluster

Opzione F - Firecracker MicroVM

PRO

API per provisioning e gestione delle MicroVM

Basso footprint delle microVM

Integrazione con diversi “frontend” che rendono dichiarativo il deploy e la gestione delle VM

Integrazione con l’ecosistema container

Utilizzo di operator per orchestrare le dipendenze in modo dichiarativo

X

CONTRO

Non adatto a tutti i tipi di workload, set minimale di funzionalità per ridurre al minimo consumo di risorse’

Dipendenze da gestire in un modo diverso

Necessità’ di produrre e gestire le immagini delle MicroVm

Opzione E - Una EC2 per singola PR

PRO

Facile provisioning e configurazione delle macchine, utilizzando terraform e ansible/salt

Costi prevedibili

Amministrazione ordinaria

Configurazione delle dipendenze automatizzabile

X

CONTRO

Costi prevedibili ma proibitivi

DImensione delle macchine non banale

Amministrazione ordinaria ma di molteplici macchine

Opzione S - Singola Macchina - Molteplici Istanze

PRO

Facile provisioning e configurazione utilizzando ansible/salt

Costi fissi, con possibilità' di riservare la singola istanza

Facile configurazione delle dipendenze

Amministrazione ordinaria di una singola macchina

OK

CONTRO

Singola macchina, single point of failure per tutti gli ambienti di tutte le pr

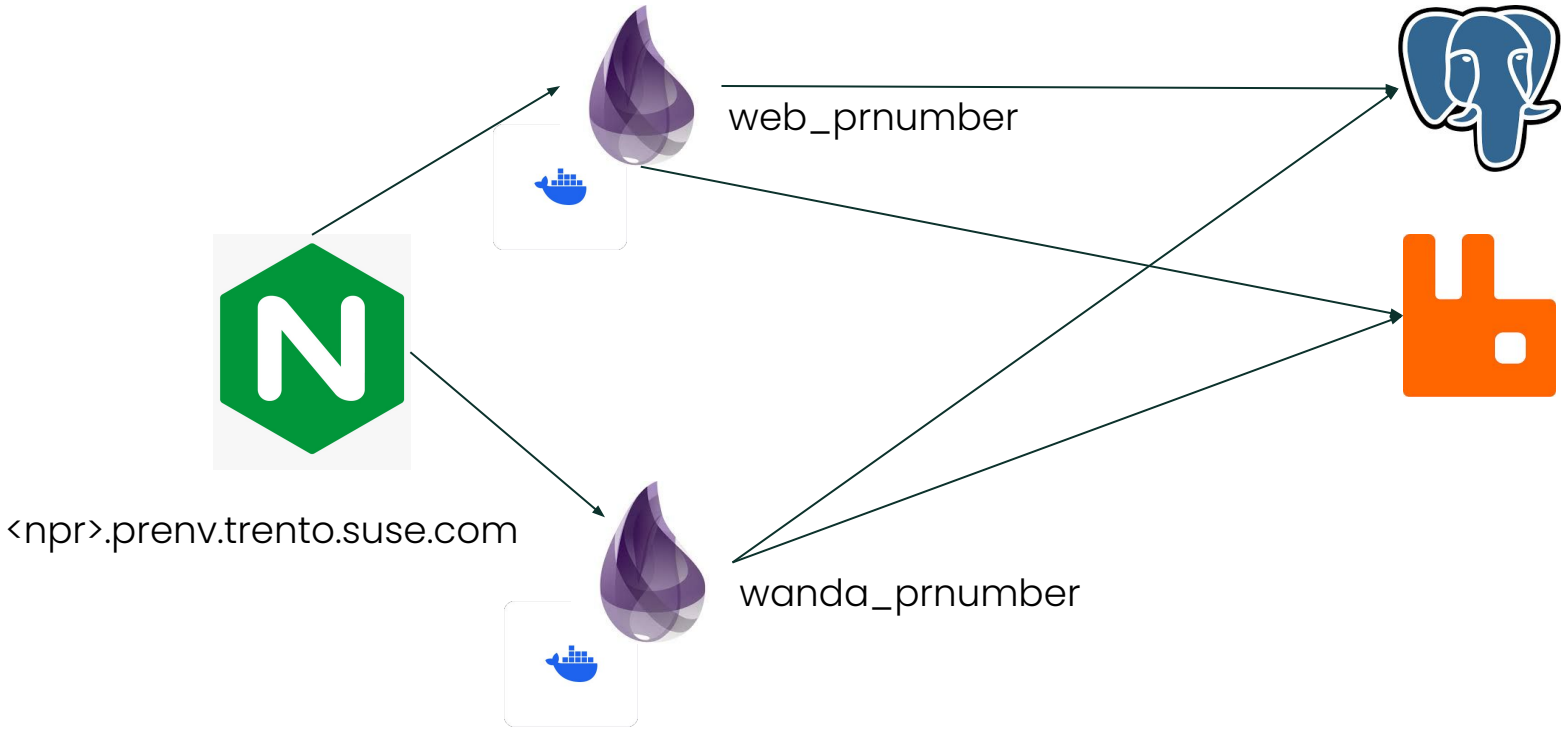
Pianificazione e dimensionamento della macchina non banale

La soluzione scelta

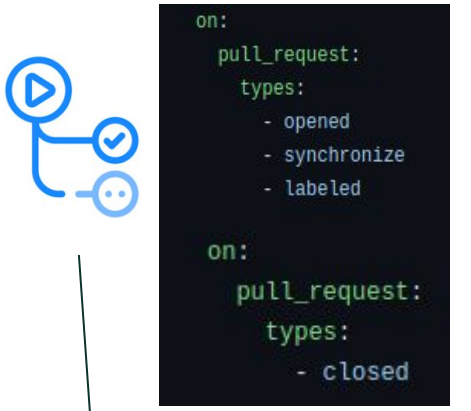
```
on:
  pull_request:
    types:
      - opened
      - synchronize
      - labeled

on:
  pull_request:
    types:
      - closed
```

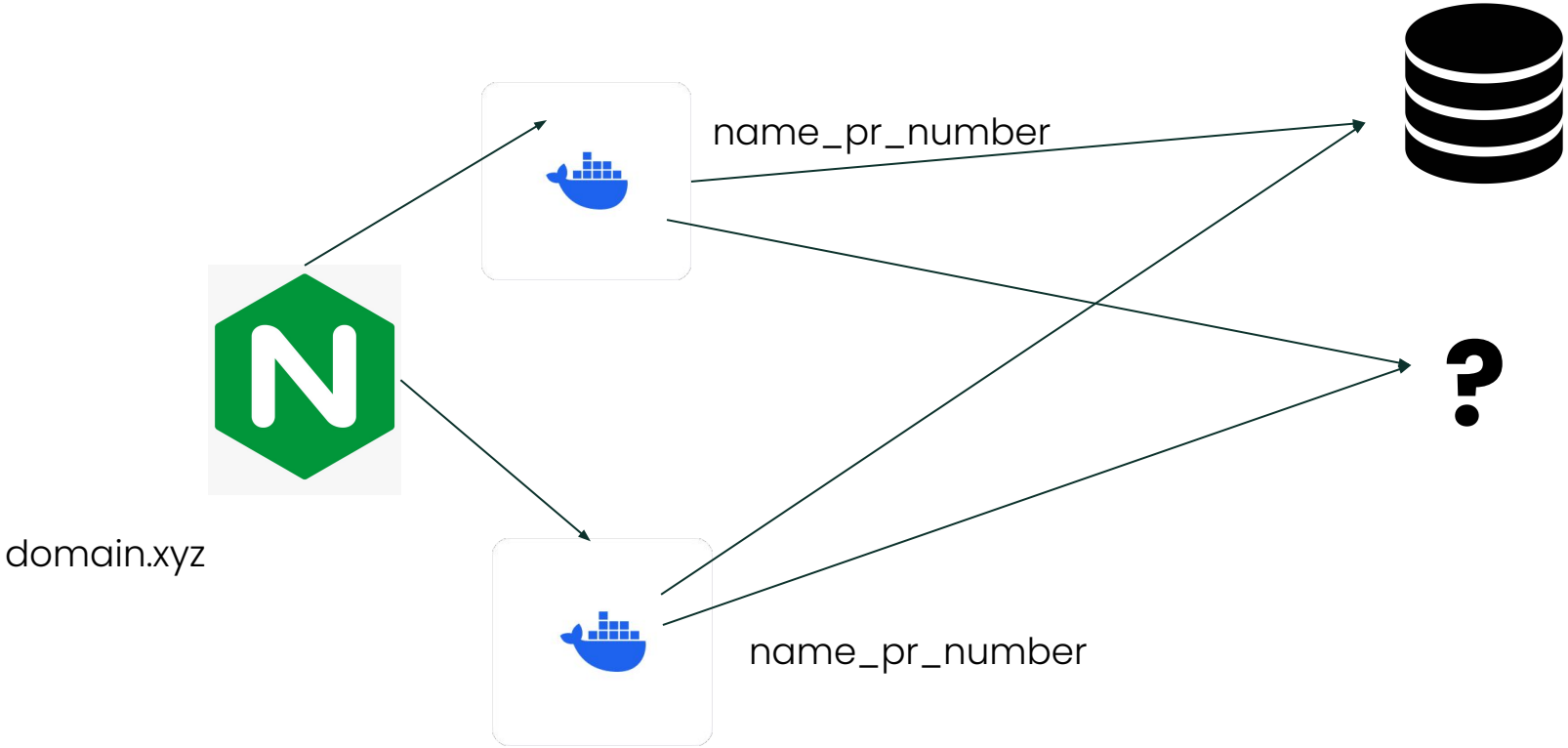
“env” label



Generalizziamo la soluzione



PR Label



Demo - Golang Api

https://github.com/CDimonaco/ephimeral-pr-env-demo

Who Am I

Carmine Di Monaco

Staff Software Engineer @ SUSE

Trento Team
Business Critical Linux

Podcast Host @ Gitbar.it

Contatti

carmine.dimonaco@gmail.com

carmine.dimonaco@suse.com

@cdimonaco Github

t.me/gitbar

Grazie