Institut Català de la Salut

PLEC DE PRESCRIPCIONS TÈCNIQUES PARTICULARS PER A LA CONTRACTACIÓ DELS SERVEIS D'OPERACIONS I MANTENIMENT DE LA PLATAFORMA DE COMPUTACIÓ D'ALGORITMES D'INTEL·LIGÈNCIA ARTIFICIAL DE L'ICS

Institut Català de la Salut

Contingut

Objecte del Contracte 3
Arquitectura actual 3
Serveis centralitzats 4
SUSE Rancher 4
GitLab 4
Clusters en producció o inferència 5
Clusters de Kubernetes 5
Emmagatzematge MinIO 5
Entorn de desenvolupament i entrenament 5
Cluster de K8s 6
Emmagatzematge MinIO 6
Frontend 6
Monitorització 7
Catàleg de serveis 7
Operacions i manteniment 7
Suport al desenvolupament d'algorismes IA 7
Requeriments 8

Institut Català de la Salut

Objecte del Contracte

L'objecte del contracte es el manteniment, actualització i ampliació a nous usuaris i projectes de la plataforma d'entrenament i de desplegament d'algoritmes d'Intel·ligència artificial de l'ICS. No entra dintre d'aquest servei les actuacions de manteniment de la infraestructura física, la gestió dels servidors i els serveis de còpies de seguretat no son objecte de la licitació.

Arquitectura actual

En el desenvolupament d'algorismes d'IA, i en particular en el cas d'aplicacions mèdiques, és essencial la reproductibilitat, és a dir, que en l'entorn de producció o inferència el software es comporti exactament igual que en el de desenvolupament o entrenament. Per això és molt important eliminar les possibles diferències a tots els nivell, a nivell de versions del sistema operatiu, de compiladors i intèrprets dels llenguatges de compilació, de les llibreries de tercers de les que en depèn, etc.

A més, cal que el sistema es pugui desplegar en múltiples Hospitals i centres de l'ICS, i sigui escalable per tal de donar suport a un increment del seu ús diari.

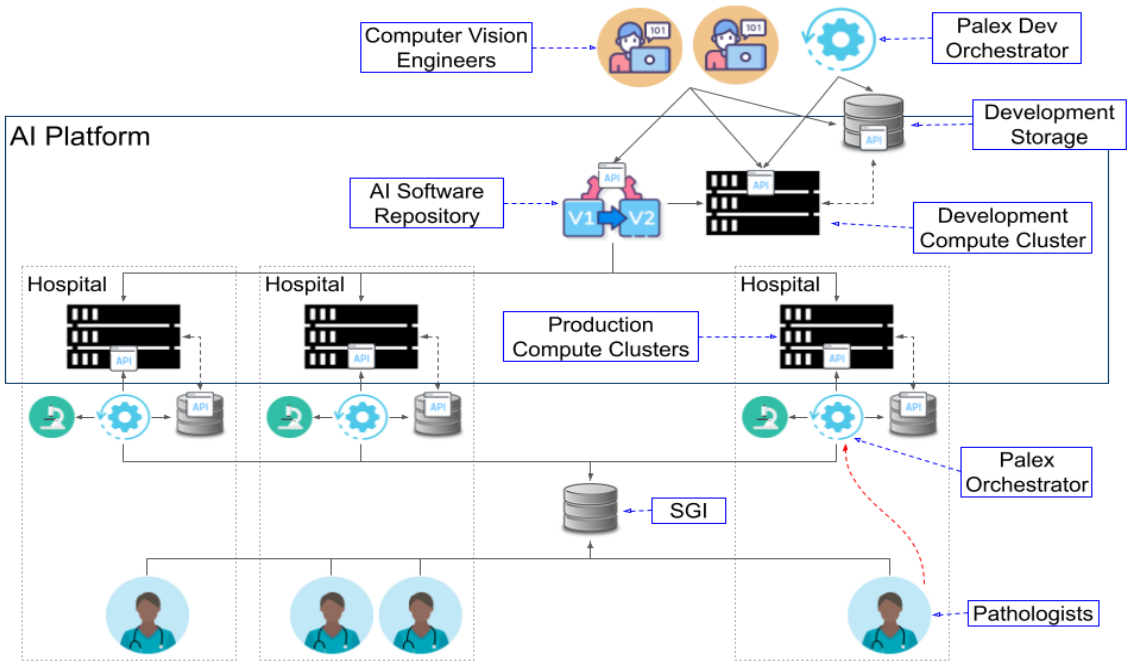
Per aconseguir aquesta reproductibilitat, escalabilitat i desplegament distribuït, l'arquitectura actual de desenvolupament i computació d'algorismes de IA del ICS es basa en l'ús de contenidors Docker, l'orquestrador SUSE Rancher amb Kubernetes (K8s), i la plataforma de DevOps GitLab.

L'ús de Docker permet encapsular totes les dependències del software per sobre del *kernel* del sistema operatiu garantint-ne la reproductibilitat, l'ús de Kubernetes l'escalabilitat tant a nivell de càrrega com dels recursos de hardware, l'orquestrador SUSE Rancher la distribució de múltiples clusters de Kubernetes en múltiples Hospitals i centres mèdics, i el GitLab el repositori privat i centralitzat, tant a nivell de codi font com a nivell de imatges de contenidors.

A nivell d'emmagatzematge de dades, la tecnologia principal és MinIO, que aporta un servei privat compatible amb el protocol S3 amb alta disponibilitat i rendiment.

La plataforma actual també consta d'un mecanisme de monitorització activa basat en OpenTelemetry, Prometheus i Grafana per garantir.ne el servei i avaluar-ne l'ús i saturació de recursos hardware.

Finalment, la plataforma actual també inclou mecanismes ad-hoc per tal de facilitar al màxim l'entrenament d'algorismes IA incorporant dos mode de funcionament dels contenidors: desenvolupament i producció. El mode de producció és el que usaran els usuaris finals en els clústers de computació en producció, i el mode de desenvolupament és el que podran usar els desenvolupadors dels algoritmes en el cluster de computació de desenvolupament i que inclou mecanismes que emulen el comportament habitual dels clusters d'alta computació (HPC) habituals.



Aquesta arquitectura porta en funcionament des del 2021 aproximadament i cal realitzar tasques d'evolució i actualitzacions que han d'anar coordinats en el dia a dia de la producció.

Serveis centralitzats

La consola d'administració federada de clusters de Kubernetes (SUSE Rancher) i el repositori de codi i contenidors (GitLab) són els serveis centrals de la plataforma. La resta són elements distribuïts i replicats que depenen d'aquests dos serveis.

SUSE Rancher

SUSE Rancher ofereix una plataforma centralitzada per desplegar i gestionar clústers de Kubernetes (K8s) de manera federada, facilitant l'administració d'entorns multi-clúster, en aquest cas multi-hospital. Permet gestionar de forma eficient la infraestructura K8s amb eines d'automatització, monitoratge i seguretat integrada, tot estandarditzant la gestió per a equips DevOps i de TI.

GitLab

La solució *on-premise* de GitLab permet gestionar repositoris de codi amb control de versions i col·laboració en equip, tot incloent eines per a la revisió de codi i gestió de branques de desenvolupament i de producció. També ofereix un registre de contenidors integrat per a l'emmagatzematge i gestió d'imatges Docker.

A nivell de CI/CD, GitLab facilita la configuració de *pipelines* automatitzades per a compilar, testejar i desplegar aplicacions, integrant-se amb repositoris i contenidors per agilitzar el desenvolupament i la posada en producció.

En particular l'arquitectura inclou un mecanisme de CI/CD usant Kaniko amb les pipelines de GitLab per generar automàticament imatges Docker versionades. Les pipelines de Kaniko s'executen en *gitlab-runners* desplegats dins d'un entorn Kubernetes (K8s), i les imatges de

contenidors automàticament generades són les que es podran desplegar i usar a tots els clústers de producció. Les imatges de contenidors estan versionades i segueixen la mateixa política de versions que el codi font.

Clusters en producció o inferència

A cada Hospital o centre de salut on es vulgui tenir un servei local que executi els algoritmes IA hi haurà un cluster de K8s i un servei d'emmagatzematge MinIO locals.

Clusters de Kubernetes

Actualment tots els clústers de K8S de producció consisteixen en una controlador virtualitzat que rep les peticions de computació / inferència, i un node físic amb 1 GPU (NVIDIA TESLA T4).

Tots aquests clusters estan administrats des de una única la consola de SUSE Rancher centralitzada, i estan configurats per descarregar les imatges de contenidors del GitLab centralitzat. Això permet que una nova versió d'un algoritme IA afegit al GitLab pugui estar disponible de forma automàtica tots i cadascun dels clústers de producció dels hospitals. Aquests processos o jobs executen dia i nit tasques d'inferència d'objectes emmagatzemats en cada hospital provinents dels escàners.

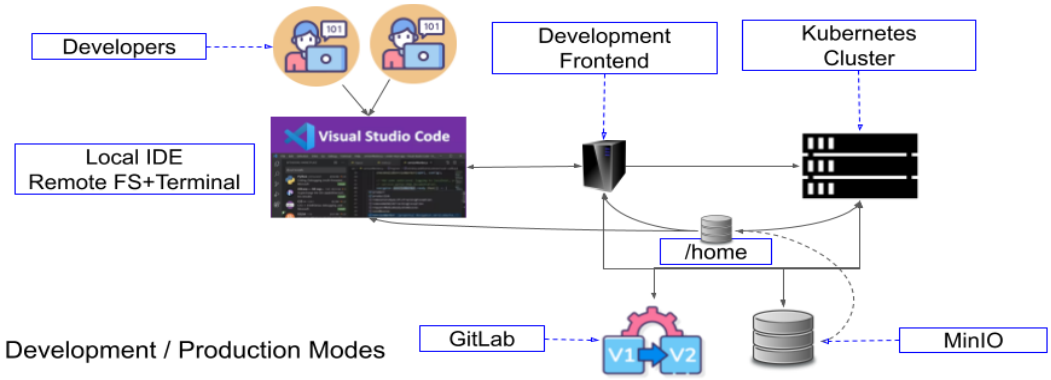
Emmagatzematge MinIO

El servei d'emmagatzematge de cada hospital es on es guarden les dades necessàries per l'execució d'algoritmes IA en aquell centre, així com els resultats obtinguts. Gràcies a aquest processat basat en la arquitectura *edge computing*, ens estalviem la transferència d'una gran quantitat de dades per la xarxa troncal del ICS.

Entorn de desenvolupament i entrenament

L'entorn de desenvolupament (o entrenament) únic és equivalent als entorns de producció (o inferència) del hospitals explicats anteriorment. També consta de un el cluster de K8s i el servei d'emmagatzematge MinIO locals, i el K8s també està configurat i administrat de manera centralitzada amb el mateix SUSE Rancher i GitLAB.

La diferència principal és l'existència d'una màquina *frontend* Linux a on múltiples desenvolupadors s'hi poden connectar via SSH i que conté un entorn equivalent al de un entorn d'alta computació (HPC) permeten l'ús compartit i interactiu dels recursos de computació (GPUs).



- Dev Container
  - Container will mount /src at run time (from /home in frontend)
  - Container will depend on /home on the running worker
  - Used only by Developers for quick development
- Prod Container
  - Container will copy /code at build time (from GitLab)
  - Always same /code at run-time
  - Used by Pathologist and the CI/CD on GitLab
  - Commands + YAML VS Python API

Cluster de K8s

Actualment el clúster de K8s de desenvolupament consisteix en un node virtual controlador i un node físic amb 5 GPUs (NVIDIA TESLA T4). Al igual que els clústers de producció / inferència, el cluster de desenvolupament i entrenament també està administrat i configurat amb els serveis centralitzat de SUSE Rancher i GitLab.

La diferència principal és que té una configuració propia i més dinàmica de *namespaces* per tal de permetre l'ús compartit dels recursos de computació entre múltiples desenvolupadors i alhora la privacitat entre els diferents equips de treball.

Per tal d'optimitzar recursos, aquest cluster també s'utilitza per executar les *pipelines* de CI/CD de GitLab que mitjançant Kaniko generen les imatges de contenidors.

Emmagatzematge MinIO

Per entrenar una IA amb precisió i fiabilitat, es requereixen grans quantitats de dades (datasets) que permetin identificar patrons i millorar l'eficiència dels models.

El servei MinIO del cluster de desenvolupament és equivalent als de producció, però té una configuració propia i més dinàmica de *namespaces* per tal de permetre l'ús compartit de les dades entre múltiples desenvolupadors i alhora la privacitat entre els diferents equips de treball.

Frontend

Aquesta màquina virtual proporciona l'accés als usuaris a través d'una consola SSH i pot servir de pasarela per eines web com pot ser *Jupyter Notebook*. Un cop dins, els desenvolupadors poden executar processos i interactuar amb el clúster de K8s i el MinIO de desenvolupament d'una manera semblant a la que ho farien amb un cluster d'alta computació (HPC).

En particular hi troben unes eines desenvolupades ad-hoc per tal de poder arrancar i accedir interactivament amb containers en "mode de desenvolupament" on hi ha totes les

dependències del seu software però el seu codi font es monta de via NFS i per tant poden desenvolupar-lo i testejar-lo de la manera més còmode possible.

Monitorització

Actualment es disposa d'un servei que centralitza la monitorització de sistemes K8s mitjançant les eines "kube-state-metrics" i "OpenTelemetry". Les dades de monitorització es recopilen i emmagatzemen en un servidor "Prometheus", on es registren mètriques tant a nivell de màquina com de "job" dins de Kubernetes (K8s).

Un servei Grafana permet un seguiment en temps real de l'estat dels recursos, com ara l'ús de CPU, memòria, i estat dels pods, proporcionant així una visió integral de la salut del clúster. A més, el servei inclou una funcionalitat de gestió d'alertes que envia notificacions automàtiques en cas de problemes o anomalies. Aquestes alertes es poden canalitzar per correu electrònic, Slack o Google Chat, garantint que els equips responsables rebin informació crítica de manera immediata. Les eines de visualització disponibles permeten identificar ràpidament els problemes i optimitzar el rendiment i la fiabilitat de les aplicacions dins del clúster.

Catàleg de serveis

Operacions i manteniment

- Monitorització activa de tota la infraestructura
  - Clústers de producció / inferència
  - Cluster de desenvolupament / entrenament
  - GitLab
- Actualització de tota la infraestructura per evitar qualsevol tipus de vulnerabilitat
- Evolucionar la plataforma amb tecnologies obsoletes i substituir per noves alternatives
- Coordinació amb tots els actors implicats
  - Servei de hosting (PALEX)
  - Connectivitat (CTTI)
  - Usuaris (ICS + proveïdors externs, com ara iThinkUPC, T-Systems)
- Suport i desplegament de possibles ampliacions de hardware de la plataforma
- Suport en la presa de decisions i definició d'arquitectures que donin cabuda a terceres empreses que vulguin convergir en la iniciativa
- Suport a la migració de serveis en maquinari nou un cop aquest ha superat el cicle de vida

Suport al desenvolupament d'algorismes IA

- Consultoria per l'ús dels clústers de producció o inferència
- Consultoria per l'ús del cluster de desenvolupament o entrenament
- Consultoria per la integració, contenerització i desenvolupament de nous algoritmes IA i iniciatives per part de nous grups
- Millora de les eines ad-hoc per millorar facilitar l'ús del cluster de desenvolupament.

Institut Català de la Salut

Requeriments

- Coneixements i experiència demostrada amb Docker.
- Coneixements i experiència demostrada amb administració de cluster de K8s.
- Coneixements i experiència demostrada amb SUSE Rancher.
- Coneixements i experiència demostrada amb administració de GitLab on-premise.
- Coneixements i experiència demostrada amb administració de MinIO, RustFS, Garage i Ceph.
- Coneixements i experiència demostrada amb Kaniko.
- Coneixements i experiència demostrada en entorns d'alta computació (HPC), en especial amb algoritmes IA i ús de GPUs.
- Coneixements i experiència demostrada en entorns híbrids d'entrenament IA (HPC amb Slurm) i inferència IA (serveis amb K8s), especialment amb Slinky com a connector.
- Coneixements i experiència demostrada en entorns de serveis de desplegament de serveis de models en vLLM.
- Coneixements i experiència demostrada en entorns de visualització remota integrada en clusters kubernetes: Open OnDemand.
- Coneixements i experiència demostrada en desplegament i gestió d'eines d'anotació de dades per a entrenaments d'algoritmes de IA: LabelStudio, eines de la suite de OMERO i Diffgram.
- Coneixements i experiència demostrada amb administració de xarxes.
- Coneixements i experiència demostrada amb OpenTelemetry, Prometheus i Grafana.

Barcelona,