Bevezetés a referencia architektúrák alkalmazásába

Farkas Attila

farkas.attila@sztaki.hu

Bemutatkozás

Farkas Attila

- SZTAKI – tudományos segédmunkatárs, kutató
  - PERL – Párhuzamos és Elosztott Rendszerek Kutatólaboratórium

- farkas.attila@sztaki.hu

Tartalom

- Referencia architektúra
  - koncepció
  - megvalósítás
  - példák
- ELKH Cloud szolgáltatások
- Jövőbeli terveink

Referencia architektúra koncepció

ELKH Cloud

- Egyfajta PaaS felhő szolgáltatás
- Referencia architektúrák tartalmaznak minden szükséges építőelemet egy komplex szoftverrendszer felépítéséhez felhő alapú erőforrásokon
- Továbbá a nem funkcionális követelményeknek is eleget tesznek:
  - Skálázhatóság
  - Rendelkezésre állás
  - Konfigurálhatóság
  - Biztonság
- Mindezt jól definiált leírófájlok és egy felhő orkesztrátor segítségével

Occopus

- SZTAKI PERL labor által fejlesztett nyílt forráskódú hibrid orkesztrációs eszköz
- „OCCO” – „One Click Cloud Orchestrator”
- Orkesztráció: virtuális gépek (VM) és infrastruktúrák rugalmas létrehozása és menedzselése felhő rendszerekben (előtérbe kerül IaaS, IaaS rétegben)
- Kontextualizáció: gépek személyre szabása
  - Szoftvercsomagok telepítése, konfigurációja, finomhangolása
- Hordozható leírók
- Skálázási lehetőség
- Felhőfüggetlen megoldás

ELKH Cloud

openstack CLOUD SOFTWARE

CloudSigma

amazon web services

Microsoft Azure

CloudBroker

Open Nebula

Occopus leírók

Infrastruktúra leíró (infrastructure description)

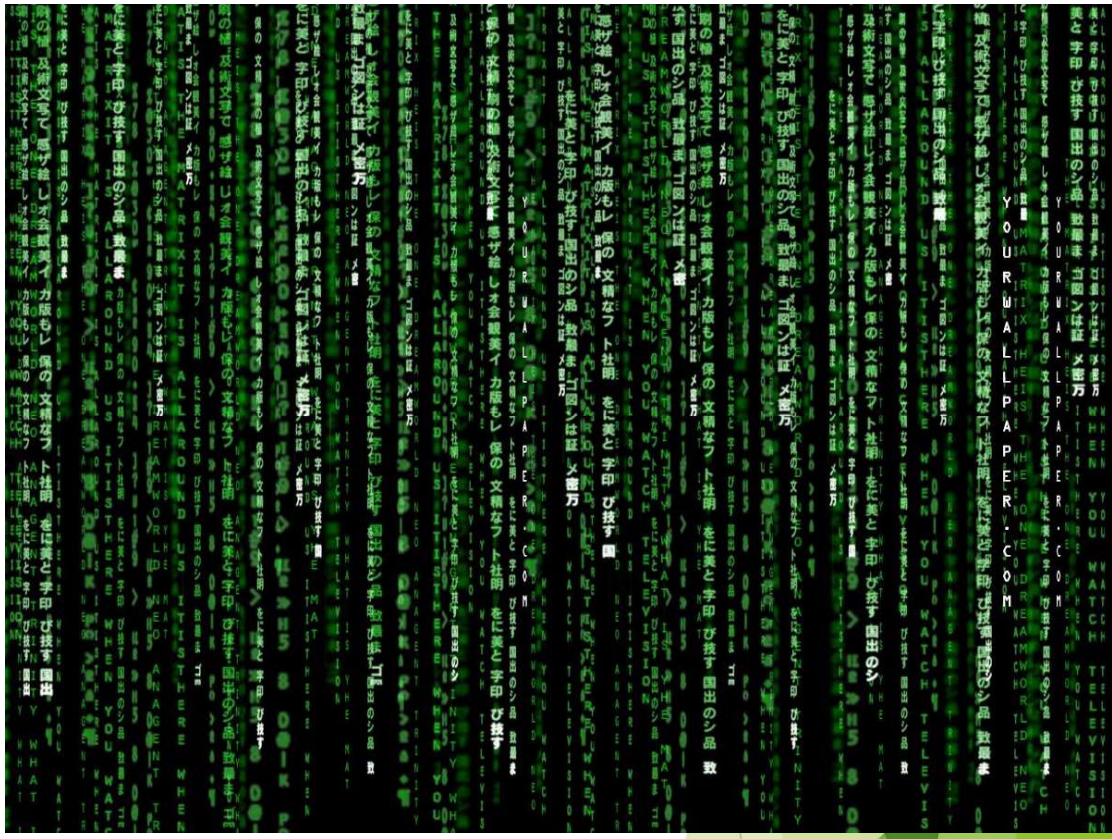
- Csomópontok
- Változók
- Skálázás
- Függőségek

Csomópont leíró (Node definition)

- Erőforrás definiálás
- Kontextualizáció
- Egészség-ellenőrzés
- Konfigurációs menedzsment

Cloud-init fájlok

- A virtuális gép konfiguráláshoz szükséges lépések:
  - Felhasználó kezelés
  - Komponensek telepítése/beállítása
  - Szolgáltatások telepítése/konfigurálása/indítása
  - Szolgáltatások elindítása



Referencia architektúra használata

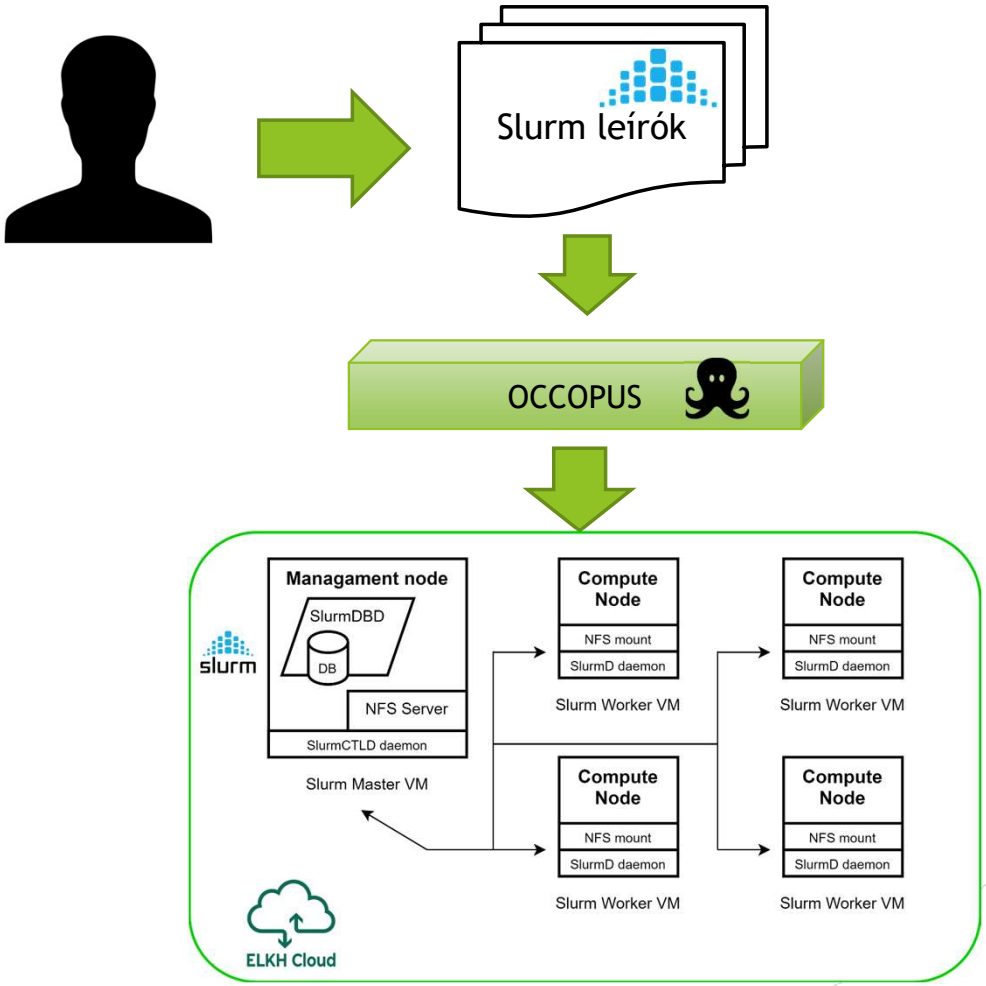
1. Occopus telepítése
2. Leírók letöltése
3. Tűzfalszabályok létrehozása
4. Leírók személyre szabása
5. Occopus aktiválása
6. Leírók importálása
7. Klaszter kiépítése
8. Infrastruktúra használata

1x (Aktuális VM-en az aktuális referencia architektúra esetén)

egy-egy sor kód

Rövid és hosszú felhasználás is támogatott

Referencia architektúra használata #2

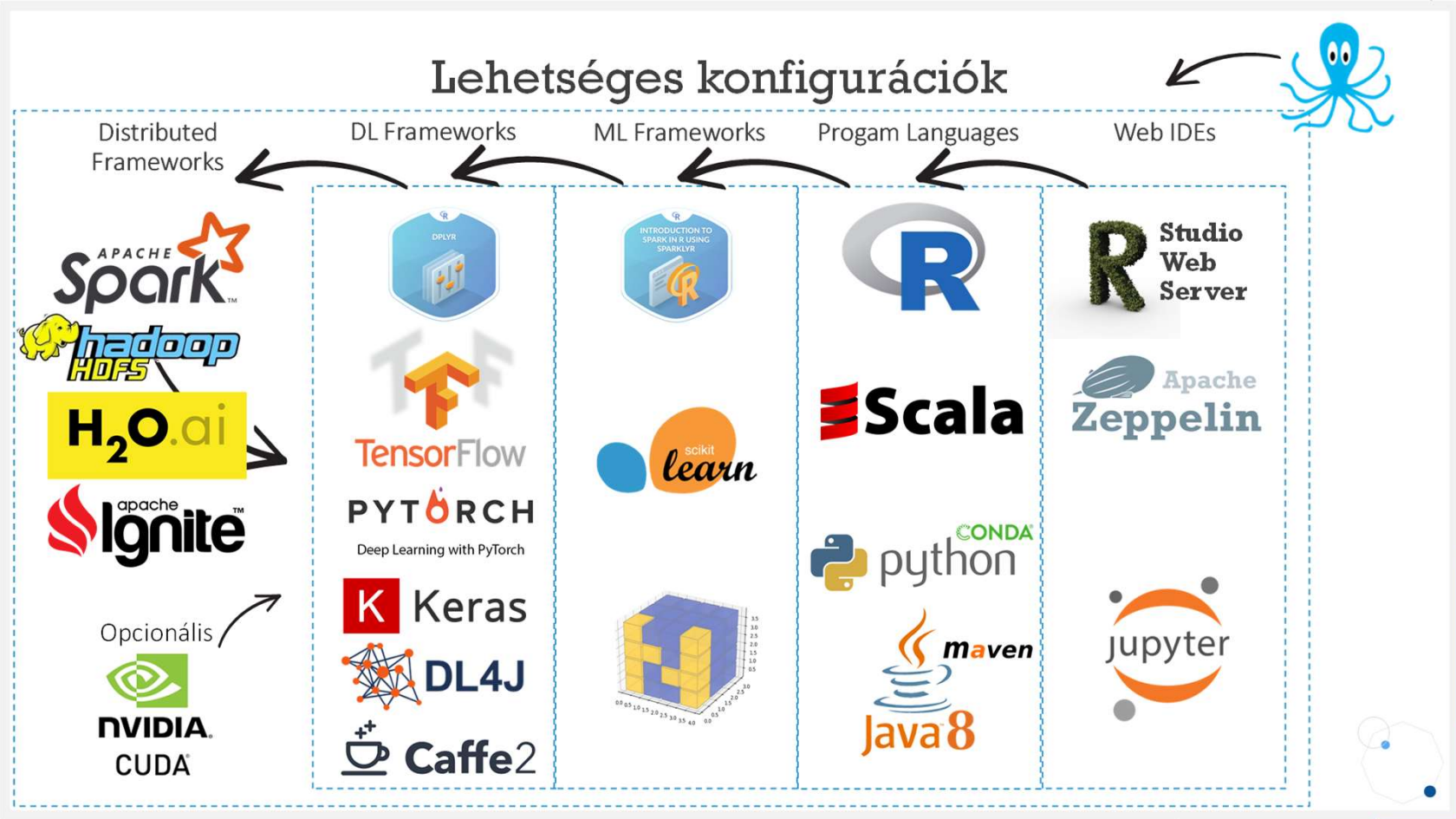


Referencia architektúrák csoportosítása

- Gépi tanulás támogatására
- Big Data és IoT platform felkínálása
- Konténer platform biztosítására
- Workload menedzser szolgáltatásra

Gépi tanulást támogató referencia architektúrák

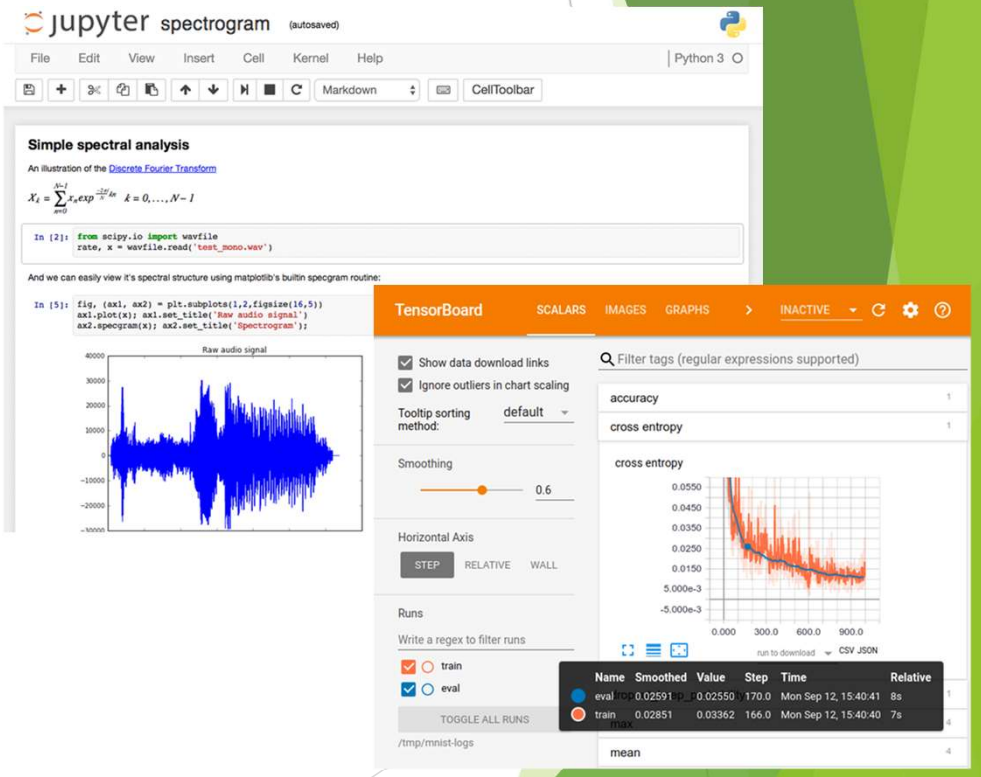
MI referencia architektúra



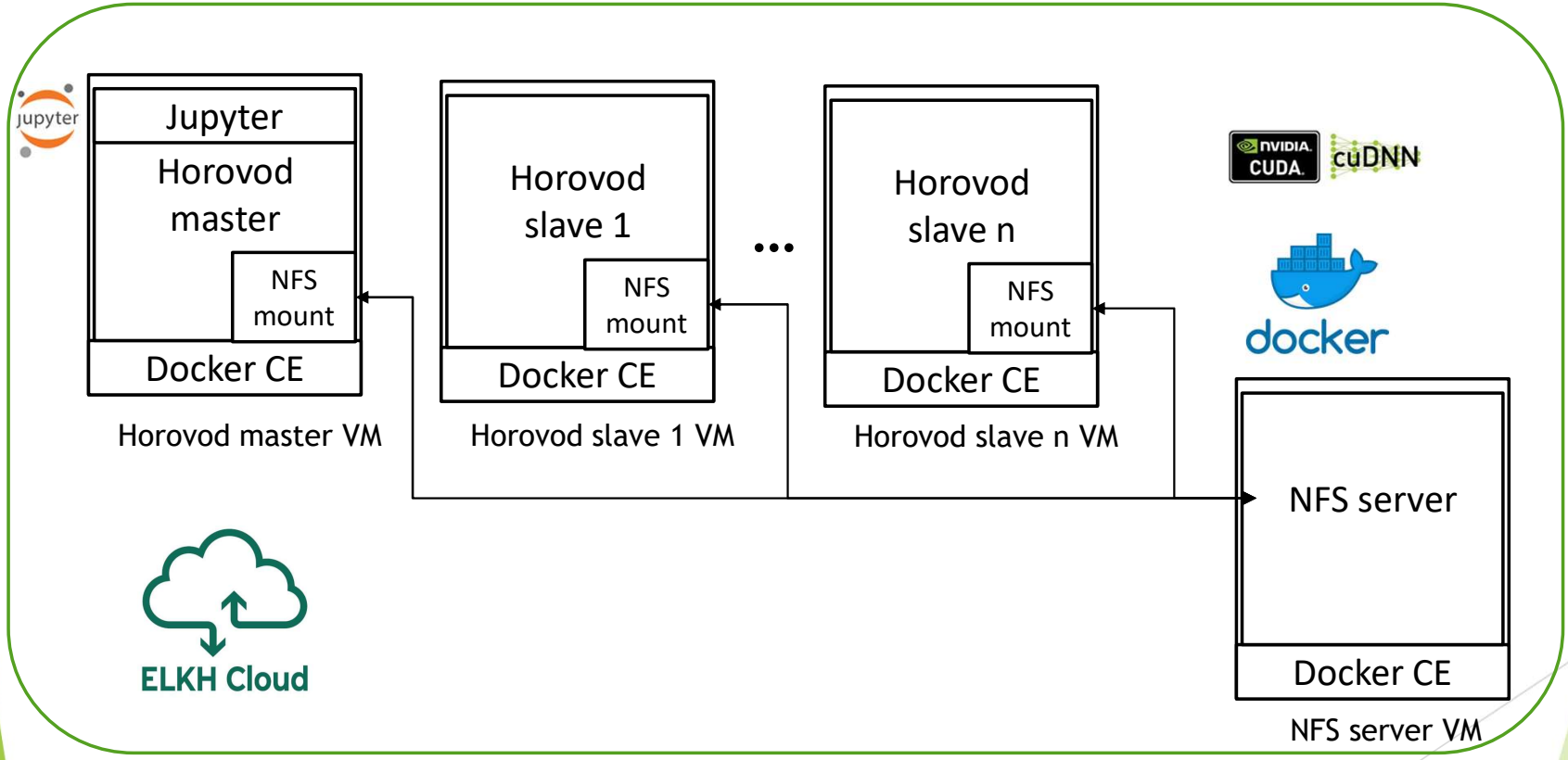
Tensorflow és Keras referencia architektúra

ELKH Cloud

- Jupyter Notebook / JupyterLab
- Keras
- TensorFlow + TensorBoard
- CUDA / cuDNN (BLAS, Eigen)
- Virtual Machine (VM) with CPU / GPU

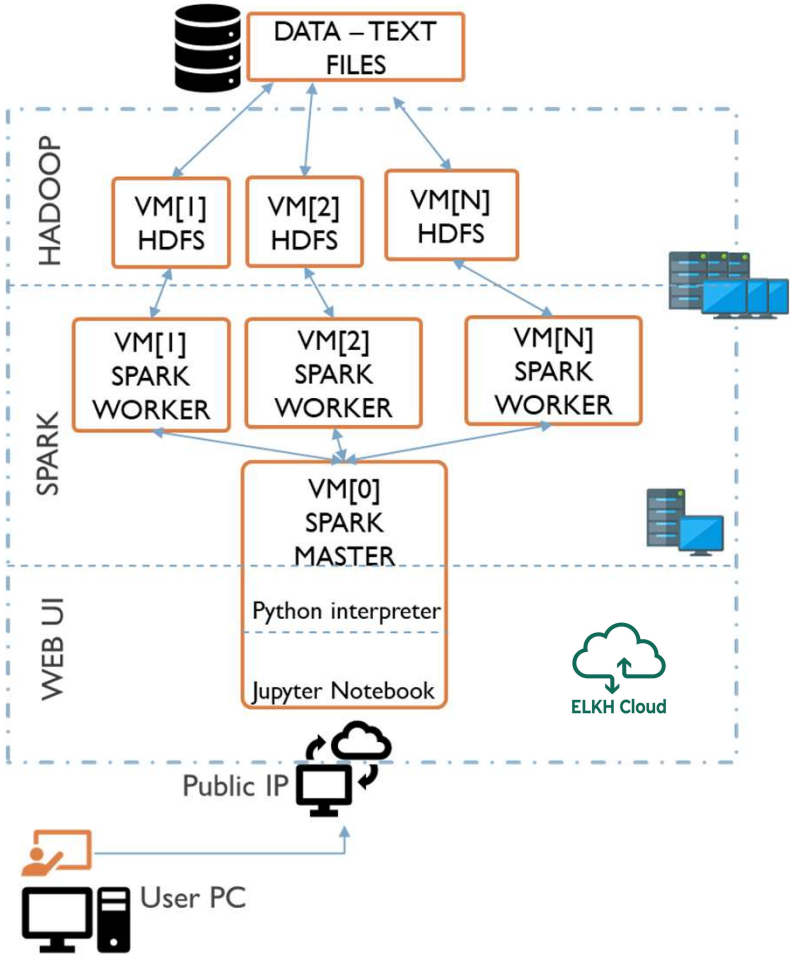


Horovod referencia architektúra



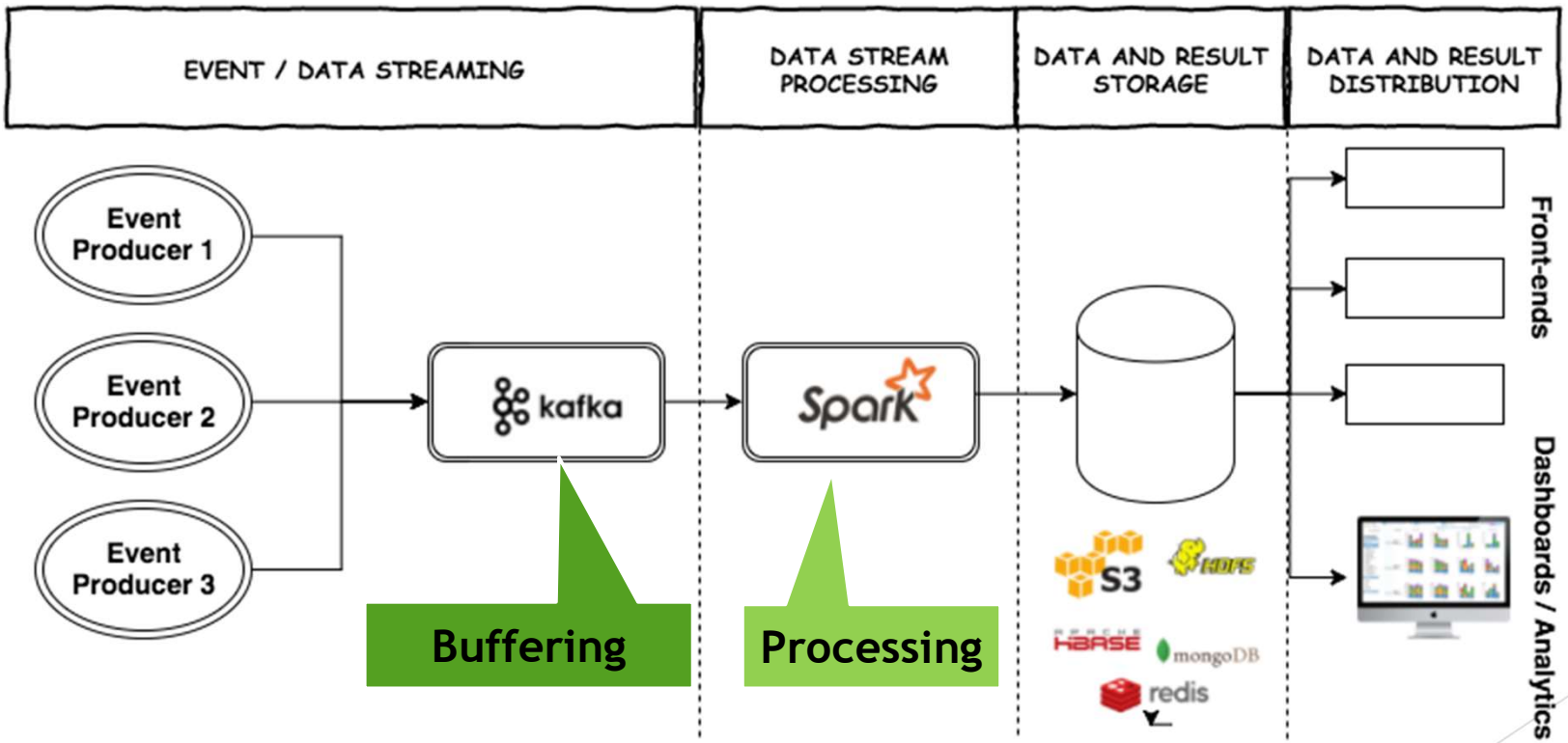
Big Data és IoT referencia architektúrák

Spark referencia architektúra



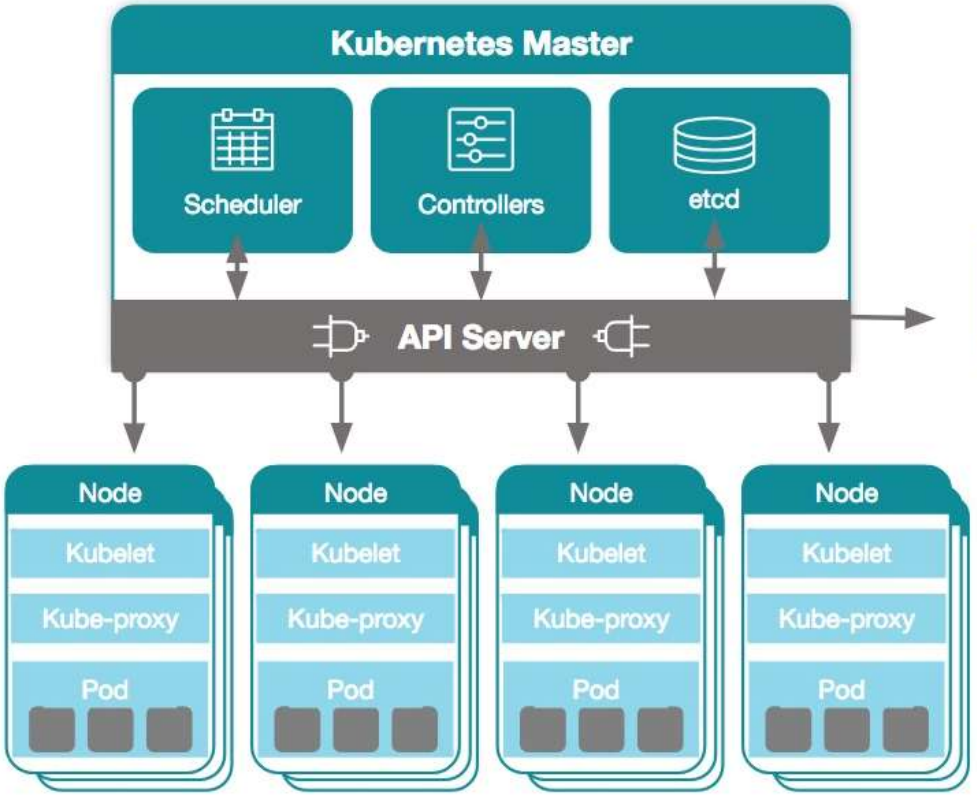
ELKH Cloud

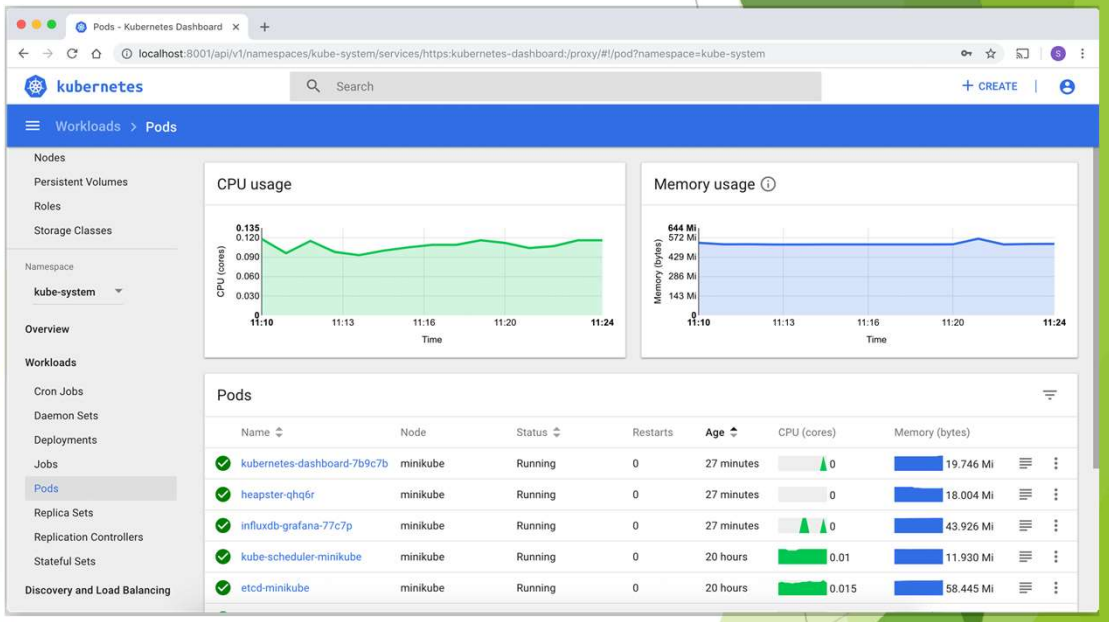
Kafka és Spark referencia architektúra



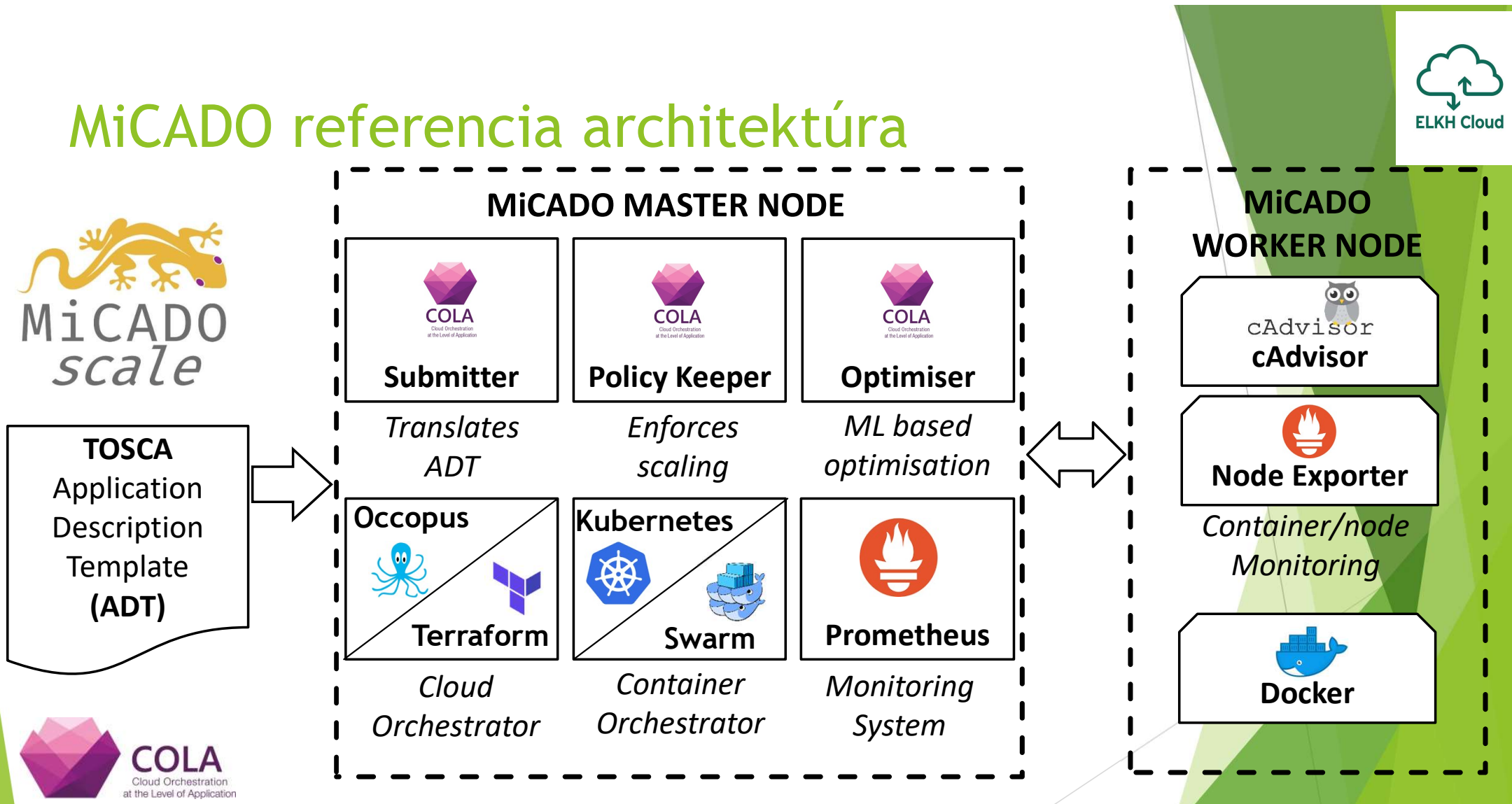
Konténer platformot biztosító referencia architketúrák

Kubernetes referencia architektúra



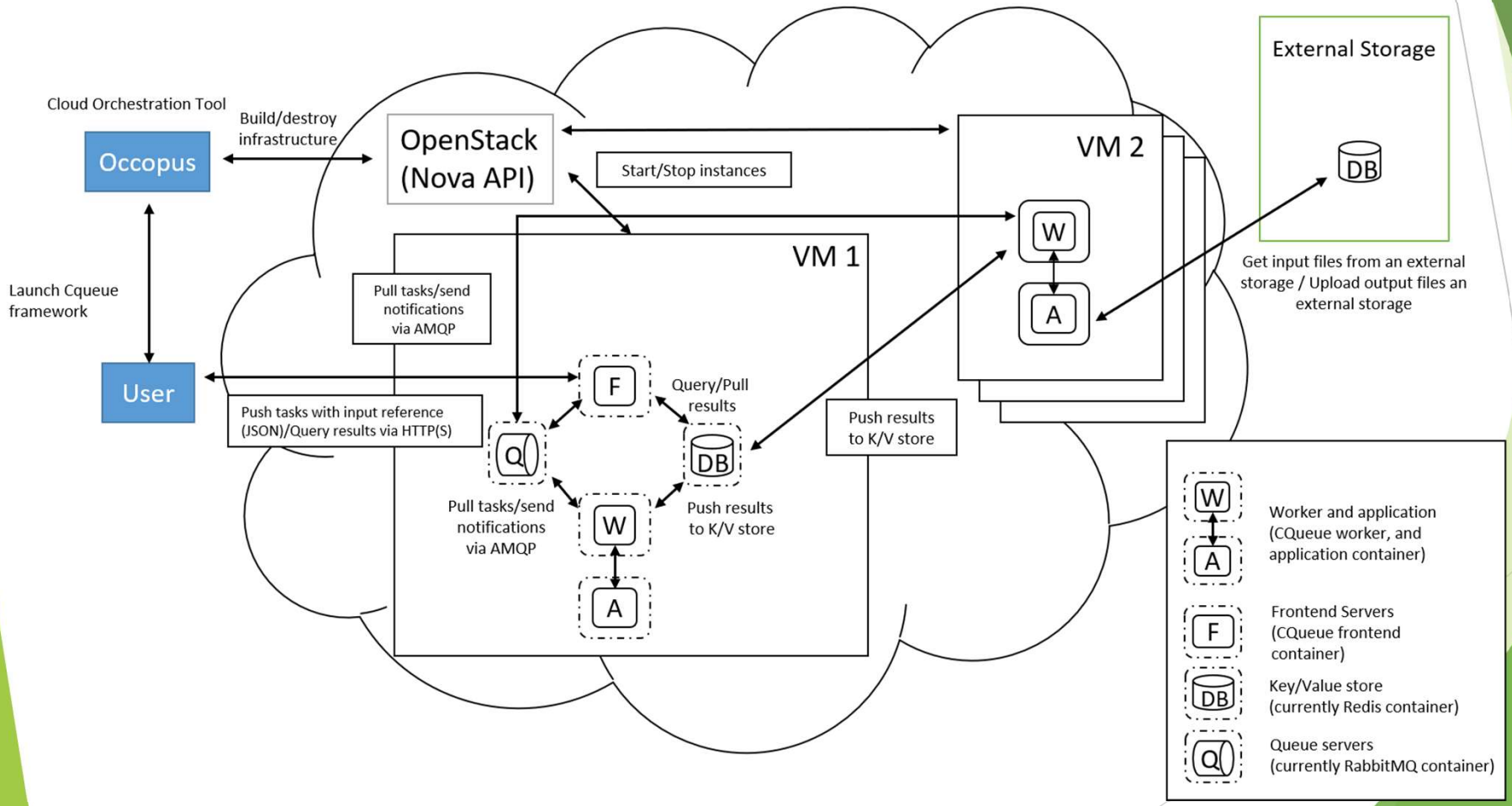


MiCADO referencia architektúra

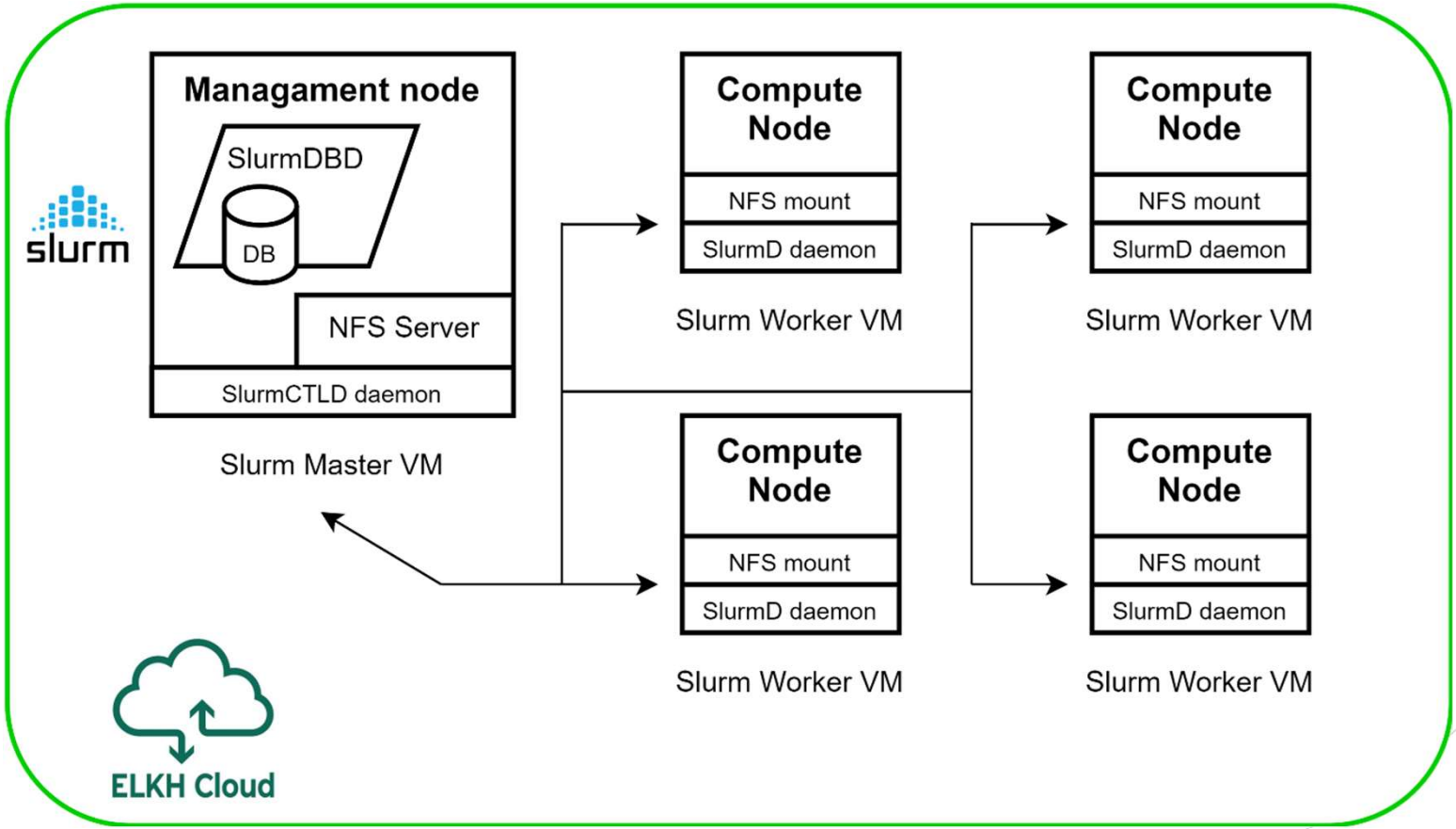


Workload menedzser referencia architketúrák

Cqueue referencia architektúra



SLURM referencia architektúra



További példák

A rendelkezésre álló referencia architektúrák és leírásuk:

- Occopus cloud orchestrator indítása
- JupyterLab
- DataAvenue
- Cloud alkalmazásokat támogató portál indítása
- Flowbster - Autodock Vina
- CQueue klaszter
- Docker-Swarm klaszter kiépítése *(Frissítés: ELKH Cloud - Microsoft Azure hibrid felhő támogatással)*
- Kubernetes klaszter
- Apache Hadoop klaszter kiépítése
- Apache Spark klaszter RStudio stack-el
- Apache Spark klaszter Python stack-el *(Frissítés: ELKH Cloud - Microsoft Azure hibrid felhő támogatással)*
- TensorFlow, Keras, Jupyter Notebook stack
- TensorFlow, Keras, Jupyter Notebook GPU stack *(Frissítés: ELKH Cloud - Microsoft Azure hibrid felhő támogatással)*
- Horovod klaszter
- Kafka klaszter
- Slurm klaszter

Forrás: https://science-cloud.hu/felhasznalast-segito-szolgaltatasok

Referencia architektúra előnyei

- Az újrahasznosítható leíróknak köszönhetően egyszerűbben és gyorsabban kiépíthetők bonyolultabb infrastruktúrák is
  - Manuális létrehozás esetén több időt és szakértelmet kíván
- A leírók előre konfiguráltak
  - Megfelelő szoftver verziók, integrált komponensesek, biztonsági konfiguráció
- A leírók előre teszteltek a nem funkcionális követelményekre is
- Jól hordozhatók különböző felhő platformok között

További terveink

- Szélesebb körben támogatott orkesztrátor használata (Terraform)
  - Hordozhatóság javítása
- Konfiguráció menedzsment eszköz alkalmazása a kiépítéshez (Ansible)
  - A kiépített referencia architektúra a teljes életciklusa alatt könnyebben menedzselhető
- Leírók automatikus tesztelésének kialakítása (GitLab CI)
  - Megbízhatóság növelése
- További referencia architektúrák kialakítása

Összefoglalás

- ELKH Cloud
  - IaaS felhő szolgáltatás a kutatók és egyetemek számára
- Referencia architektúra koncepció
  - PaaS felhő szolgáltatás
  - Újrahasznosítható és hordozható leírók
  - Nem funkcionális követelmények biztosítása
- Referencia architektúra példák
  - Gépi tanulást támogató referencia architektúrák
  - Big Data és IoT referencia architektúrák
  - Konténer platformot biztosító referencia architketúrák
  - Workload menedzser referencia architketúrák

Köszönöm a figyelmet!