Slurm referencia architektúra az ELKH Cloudon

Emődi Márk

emodi.mark@sztaki.hu

Bemutatkozás

Emődi Márk

SZTAKI PERL – Tudományos segédmunkatárs, kutató

emodi.mark@sztaki.hu

ELKH Cloud

Mi az a Slurm?

ELKH Cloud

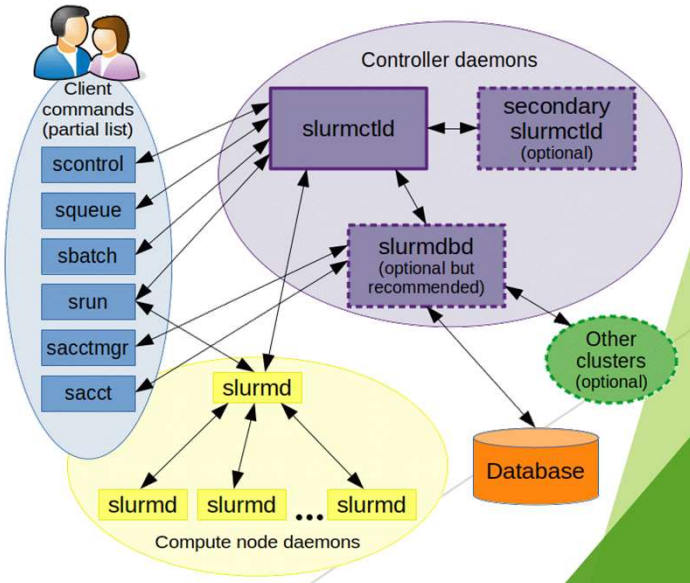
- **SLURM:** Slurm Workload Manager (Simple Linux Utility for Resource Management)
- A Slurm egy nyílt forráskódú, hibatűrő és nagymértékben skálázható fürtkezelő és feladatütemező rendszer.
  - Szuperszámítógépeken, klasztereken alkalmazzák
- Három fő funkciója van:
  - Kizárólagos erőforrást képes rendelni adott felhasználóhoz adott időre
  - Keretrendszert (parancsokat) biztosít a job-ok indításához, törléséhez és felügyeletéhez.
  - Nyomon követi az összes feladatot, hogy mindenki hatékonyan használhassa az összes erőforrást anélkül, hogy egymást hátráltatnák.
- Alaprendszer bővítményekkel bővíthető
  - MPI, Konténeres környezet, saját logikán működő ütemező, stb.

A Slurm architektúrája

- **slurmd démon:** minden egyes (worker) node-on fut
- **slurmctld démon:** a master node-on fut (management node)
- **slurmdbd démon:** adatbázis, mely a felhasználókezelésért felel (opcionális)
- **Slurm parancsok:** a slurm-ön belüli parancsok a teljes fürtön belül (master és worker) node-okon egyaránt futtathatóak

Azonos jelentéssel bírnak:

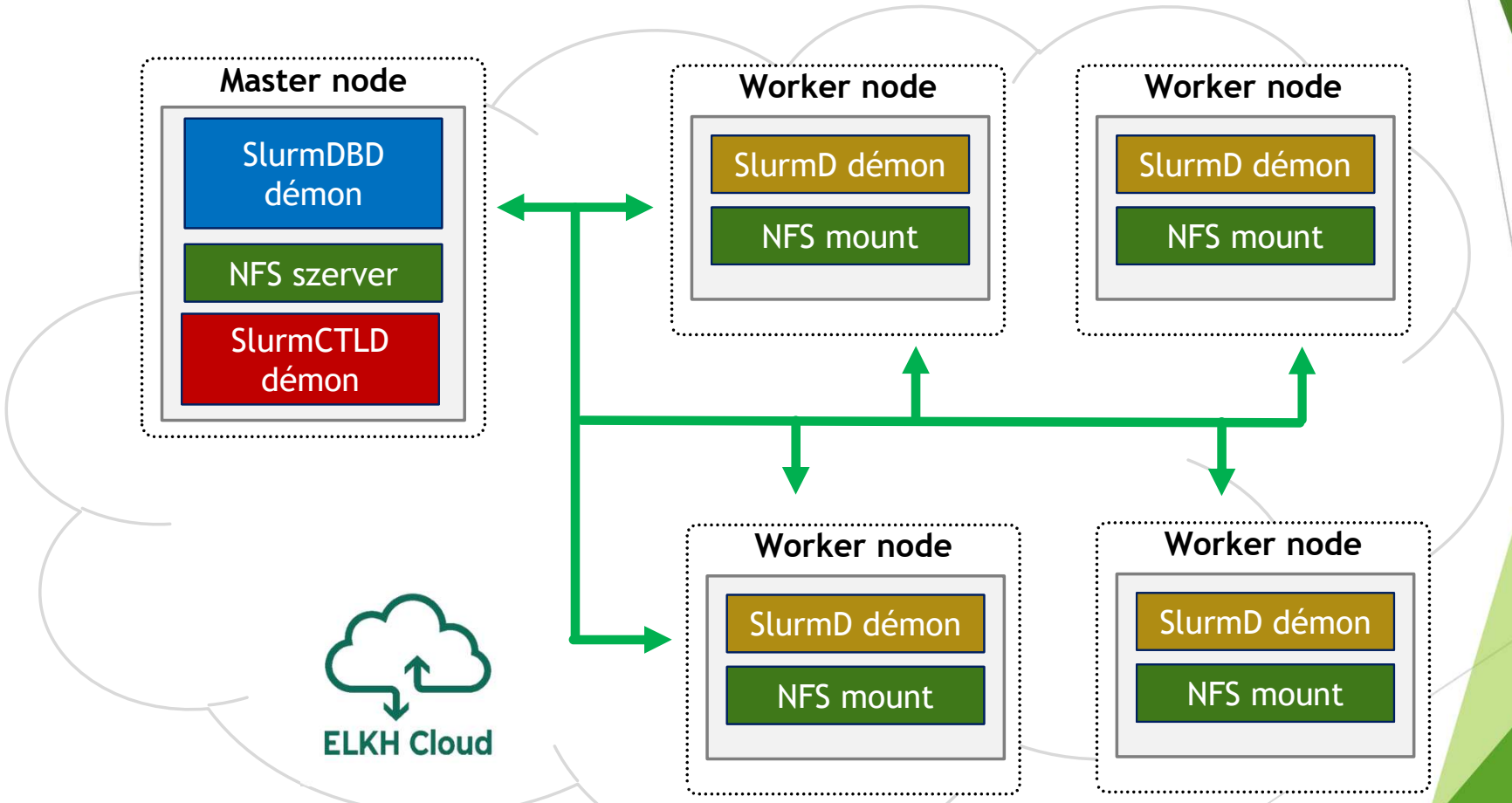
- Master node = Management node = Controller node
- Worker node = Compute node



Referencia architektúrák célja

- Komplex virtuális infrastruktúrák kiépítése automatizált módon az ELKH Cloud-on. Az alábbi kritériumoknak magas minőségben való megfelelés:
  - Elosztott
  - Hibatűrő
  - Biztonságos
  - Skálázható
- Egyszerűsített felhasználás tesztrendszer és élesrendszer felépítésére
  - Minimális LINUX és felhő ismereti tudás szükséges!

A Slurm architektúrája



NFS: Network File Server

Occopus leírók

Infrastruktúra leíró (infrastructure description)

- Csomópontok
- Változók
- Skálázás
- Függőségek

```
infra_name: slurm-cluster
user_id: somebody@somewhere

nodes:
  - &M
    name: slurm-master
    type: slurm_master_node
  - &S
    name: slurm-worker
    type: slurm_worker_node
    scaling:
      min: 2
      max: 10
variables:
  mungeversion: 0.5.13-2build1
  slurmversion: 19.05.5-1
dependencies:
  -
    connection: [ *S, *M ]
```

ELKH Cloud

Occopus leírók

Infrastruktúra leíró (infrastructure description)

- Csomópontok
- Változók
- Skálázás
- Függőségek

Csomópont leíró (Node definition)

- Erőforrás definiálás
- Kontextualizáció
- Állapot ellenőrzés
- Konfigurációs menedzsment

```
'node_def:slurm_master_node':
  -
    resource:
      type: nova
      endpoint: replace_with_endpoint_of_nova_interface_of_your_cloud
      project_id: replace_with_projectid_to_use
      user_domain_name: Default
      image_id: replace_with_id_of_your_image_on_your_target_cloud
      network_id: replace_with_id_of_network_on_your_target_cloud
      flavor_name: replace_with_id_of_the_flavor_on_your_target_cloud
      key_name: replace_with_name_of_keypair_or_remove
      security_groups:
        - replace_with_security_group_to_add_or_remove_section
      floating_ip: add_yes_if_you_need_floating_ip_or_remove
      floating_ip_pool: replace_with_name_of_floating_ip_pool_or_remove
    contextualisation:
      type: cloudinit
      context_template: !yaml_import
        url: file://cloud_init_slurm_master.yaml
    health_check:
      ports:
        - 8080
      timeout: 1000
...
```

```
...
'node_def:slurm_worker_node':
  -
    resource:
      type: nova
      endpoint: replace_with_endpoint_of_nova_interface_of_your_clou
      project_id: replace_with_projectid_to_use
      user_domain_name: Default
      image_id: replace_with_id_of_your_image_on_your_target_cloud
      network_id: replace_with_id_of_network_on_your_target_cloud
      flavor_name: replace_with_id_of_the_flavor_on_your_target_clou
      key_name: replace_with_name_of_keypair_or_remove
      security_groups:
        - replace_with_security_group_to_add_or_remove_section
    contextualisation:
      type: cloudinit
      context_template: !yaml_import
        url: file://cloud_init_slurm_worker.yaml
    health_check:
      ping: False
```

Occopus leírók

Infrastruktúra leíró (infrastructure description)

- Csomópontok
- Változók
- Skálázás
- Függőségek

Csomópont leíró (Node definition)

- Erőforrás definiálás
- Kontextualizáció
- Állapot ellenőrzés
- Konfigurációs menedzsment

Cloud-init fájlok

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

Megoldás használatának lépései

Felhasználó feladatköre:

0. Lépés: **Előkészítés** (ELKH Cloud projekt, Üres Ubuntu VM elindítás)
1. Lépés: **Occopus telepítés/konfigurálás a virtuális gépen**
2. Lépés: **Leírók letöltése a virtuális gépre**
   Occopus/ELKH Cloud weboldala
3. Lépés: **Tűzfalszabályok létrehozása**
   ELKH Cloud OpenStack felületén
4. Lépés: **Leírók személyre szabása a virtuális gépen**
5. Lépés: **Occopus aktiválása**
   $ source ~/occopus/bin/activate
6. Lépés: Leírók importálása Occopus számára
   $ occopus-import nodes/node_definitions.yaml
7. Lépés: **Infrastruktúra kiépítése**
   $ occopus-build --parallelize infra-slurm-cluster.yaml
8. Lépés: **Infrastruktúra használata**

(9. Lépés: **Infrastruktúra leállítása**)

0-1. lépés

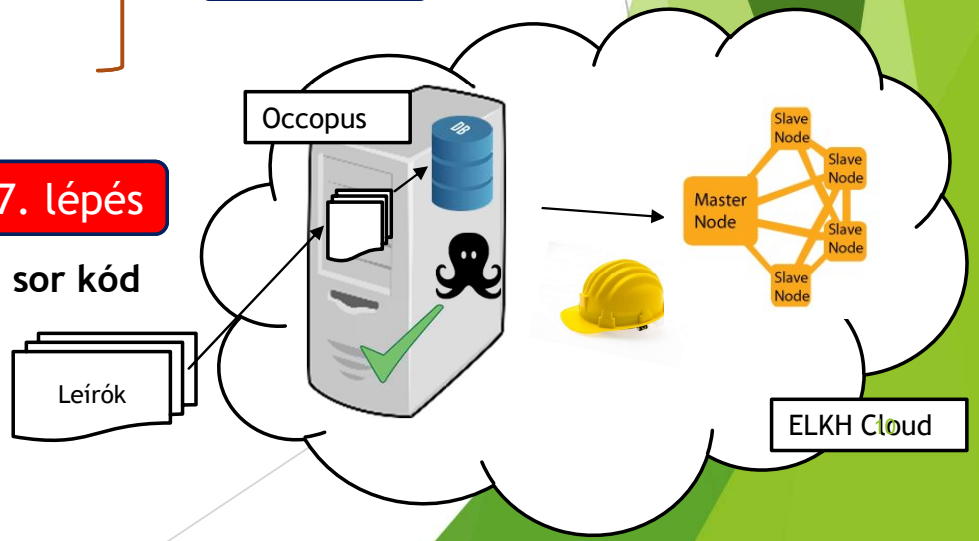
Csak első alkalommal kell beállítani.

Referencia architektúránként az Occopus-os gépen 1x kell beállítani.

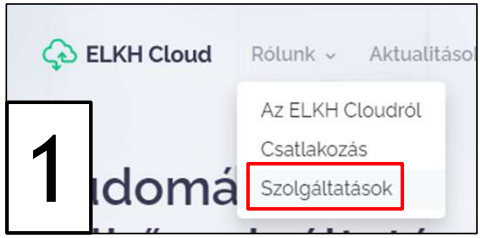
2-4. lépés

5-7. lépés

1-1 sor kód



2. lépés: leírók letöltése



ELKH Cloud

Felhasználást segítő szolgáltatások

2

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

Slurm klaszter

3

Slurm referencia architektúra

Áttekintés:

A Slurm az egy nyílt forráskódú, hibatűrő és jól skálázható klaszterkezelő és job ütemező rendszer, melyet kis- illetve nagyméretű Linux alapú fürtökhöz készítettek. A Slurm működéséhez nincs szükség kernelmódosításokra és többnyire önállóan működik. Mint workload menedzser, a Slurm három fő funkcióval rendelkezik:

- Az erőforrásokhoz (compute node-ok / worker-ek) kizárólagos vagy nem kizárólagos hozzáférést rendel a felhasználók számára a munkavégzés idejére.
- Kész megoldást kínál az allokált node-ok halmazán (általában párhuzamos módon) a munka kezdeti, végrehajtási és monitorozási fázisában egyaránt.
- Különválasztja az erőforrásokért folyó vitát a folyamatban lévő munka kezelésétől.

Használati és telepítési útmutató:

https://occopus.readthedocs.io/en/latest/tutorial-building-clusters.html#slurm-cluster

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

2. lépés: leírók letöltése

Slurm klaszter leírása:
https://occopus.readthedocs.io/en/latest/tutorial-building-clusters.html#slurm-cluster

Slurm cluster

Slurm is an open source, fault-tolerant, and highly scalable cluster management and job scheduling system for large and small Linux clusters. Slurm requires no kernel modifications for its operation and is relatively self-contained. As a cluster workload manager, Slurm has three key functions:

- First, it allocates exclusive and/or non-exclusive access to resources (compute nodes) to users for some duration of time so they can perform work.
- Second, it provides a framework for starting, executing, and monitoring work (normally a parallel job) on the set of allocated nodes.
- Finally, it arbitrates contention for resources by managing a queue of pending work.

This tutorial sets up a complete Slurm (version **19.05.5**) infrastructure. It contains a Slurm Management (master) node and Slurm Compoute (worker) nodes, which can be scaled up or down.

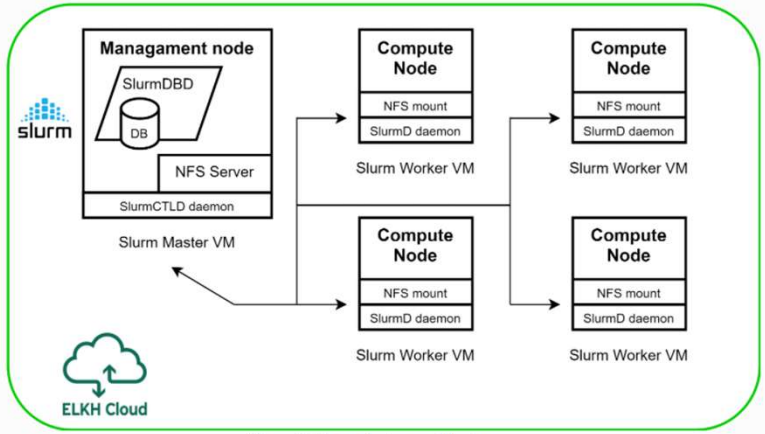


Figure 2. Slurm cluster architecture

Features

- creating two types of nodes through contextualisation
- utilising health check against a predefined port
- using cron jobs to scale Slurm Compute nodes automatically

Prerequisites

- accessing an Occopus compatible interface
- target cloud contains an Ubuntu 18.04 image with cloud-init support

Download

You can download the example as tutorial.examples.slurm .

3. lépés: Tűzfalszabályok összegzése

Ajánlott tűzfalszabály Slurm referencia architektúrákra esetén

	Direction	IP Protocol	Port Range	Remote IP Prefix
1	Egress	Any	Any	0.0.0.0/0
2	Egress	Any	Any	::/0
3	Ingress	TCP	1 - 65535	Your IP address 123.45.67.89/32
4	Ingress	TCP	1 - 65535	192.168.0.0/24
5	Ingress	TCP	22 (SSH)	0.0.0.0/0
6	Ingress	TCP	6819	Occopus VM - Floating IP

Kimenő forgalom

(opcionális)
Pl. lekérdezés
(harmadik féltől)

$ curl ifconfig.me
92.21.242.26

Slurm cluster oldali kommunikáció

SSH hozzáférés

Occopus állapot ellenőrzés

4. lépés: Leírók személyre szabása – klaszter méretének beállítása

Ha szükséges, frissítse a Slurm dolgozó (worker) csomópontok számát!

Ehhez szerkessze át az infra-slurm-cluster.yaml fájlt és módosítsa a „min” és „max” paramétereket a „scaling” kulcsszó alatt.

- A skálázás az az intervallum, amelyben a csomópontok száma megváltozhat (min, max). Jelenleg a minimális érték 2-re van állítva (ami az indításkor a kezdeti szám lesz), és a maximális értéke 10.
- Ne feledje, hogy az Occopusnak legalább egy csomópontot el kell indítania minden egyes csomóponttípusból, hogy az infrastruktúra megfelelően működjön, valamint a skálázás csak a Slurm dolgozó (worker) csomópontokon alkalmazható ebben a példában!

Haladó felhasználók itt tudják átírni az eltérő verziójú Ubuntu rendszerekbe beépített package telepítők verziószámait. További információk: https://pkgs.org/download/slurmctld

```
nodes:
  - &M
    name: slurm-master
    type: slurm_master_node
  - &S
    name: slurm-worker
    type: slurm_worker_node
    scaling:
      min: 2
      max: 10
variables:
  mungeversion: 0.5.13-2build1
  slurmversion: 19.05.5-1
dependencies:
  -
    connection: [ *S, *M ]
```

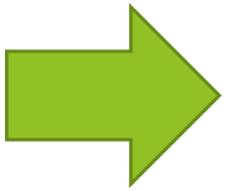
4. lépés: Leírók személyre szabása

Csomópont definíciós fájl (**nodes/node_definition.yaml**)

Támogatott operációs rendszer: Ubuntu 20.04

Nova erőforrás szekció:

```
'node_def:slurm_master_node':
  -
    resource:
      type: nova
      endpoint: replace_with_endpoint_of_nova_interface_of_your_cloud
      project_id: replace_with_projectid_to_use
      user_domain_name: Default
      image_id: replace_with_id_of_your_image_on_your_target_cloud
      network_id: replace_with_id_of_network_on_your_target_cloud
      flavor_name: replace_with_id_of_the_flavor_on_your_target_cloud
      key_name: replace_with_name_of_keypair_or_remove
      security_groups:
        - replace_with_security_group_to_add_or_remove_section
      floating_ip: add_yes_if_you_need_floating_ip_or_remove
      floating_ip_pool: replace_with_name_of_floating_ip_pool_or_remove
    contextualisation:
      type: cloudinit
      context_template: !yaml_import
        url: file://cloud_init_slurm_master.yaml
    health_check:
      ports:
        - 6819
      timeout: 1000
...
```



```
'node_def:slurm_master_node':
  -
    resource:
      type: nova
      endpoint: https://sztaki.cloud.mta.hu:5000/v3
      project_id: caml6db63ddf47a98045ef9c726vgqbp
      user_domain_name: Default
      image_id: zgsf1dc3-b6d5-4b15-942e-61e0ef2l8dk
      network_id: 3yqqqe1c-858c-4047-a48a-e2fab0nd547
      flavor_name: 3
      key_name: key_name
      security_groups:
       - f7d8dc12-fd7a-4d69-ba8d-c1f11c9b5b73
      floating_ip: yes
    contextualisation:
      type: cloudinit
      context_template: !yaml_import
        url: file://cloud_init_slurm_master.yaml
    health_check:
      ports:
        - 6819
      timeout: 1000
...
```

https://occopus.readthedocs.io/en/latest/user-doc-collecting-resources.html#openstack-horizon-nova

5-6. lépés: Occopus aktiválása

Az 5. lépésben aktiváljuk az Occopus virtualenv-et (ha még nem történt meg):

```
ubuntu@occo:~/slurm-cluster$ source $HOME/occopus/bin/activate

(occopus) ubuntu@occo:~/slurm-cluster$
```

A 6. lépésben importáljuk be a leíró mappájából a megfelelő fájlt:

```
(occopus) ubuntu@occo:~/slurm-cluster$ occopus-import nodes/node_definitions.yaml
Successfully imported nodes: slurm_master_node, slurm_worker_node
```

Fontos!

Minden módosításkor újra kell importálni a leíró fájlokat a 6. lépés szerint.

A nodes mappában lévő további fájlokat csak saját felelősségre szerkesszék.

7. lépés: infrastruktúra kiépítése

Az alábbi parancs segítségével megkezdhetjük a klaszter felépítését:
`$ occopus-build --parallelize infra-slurm-cluster.yaml`

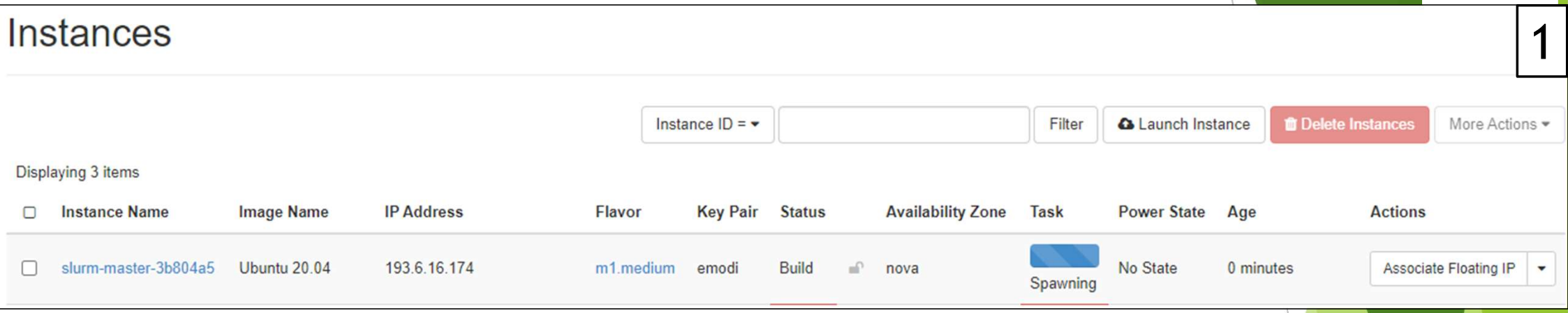
Tipp: occopus-build **--parallelize** infra-slurm-cluster.yaml (párhuzamos VM kiépítés, az egymástól független VM-ek esetében)

```
$ occopus-build --parallelize infra-slurm-cluster.yaml

** 2021-02-17 17:07:21,391      Creating node 'slurm-master'/'0b8269fe-78ec-47fc-8cdb-7195209e5123’

...

** 2021-02-17 17:25:04,184      Health checking for node 'slurm-master'/'0b8269fe-78ec-47fc-8cdb-7195209e5123'
** 2021-02-17 17:25:05,360        Checking node reachability (0b8269fe-78ec-47fc-8cdb-7195209e5123):
** 2021-02-17 17:25:05,371          193.224.59.67 => ready
** 2021-02-17 17:25:05,371        Checking port availability (0b8269fe-78ec-47fc-8cdb-7195209e5123):
** 2021-02-17 17:25:05,373          6819 => ready
** 2021-02-17 17:25:05,373      Health checking result: ready
** 2021-02-17 17:25:05,376      Node 'slurm-master'/'0b8269fe-78ec-47fc-8cdb-7195209e5123' is ready.
** 2021-02-17 17:25:05,409      Creating node 'slurm-worker'/'50c7cfe9-4c3c-4928-81e6-d58323b1e2b2'
** 2021-02-17 17:25:05,413      Creating node 'slurm-worker'/'98a82e45-6bf2-4cb9-9ead-953be4435bd7'
```

7. lépés: infrastruktúra kiépítése

ELKH Cloud

1



2

7. lépés: infrastruktúra kiépítése

ELKH Cloud

3

Instances

Instance ID = | Filter | Launch Instance | Delete Instances | More Actions

Displaying 5 items

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
slurm-worker-d8911ce	Ubuntu 20.04	193.6.16.158	m1.medium	emodi	Active	nova	None	Running	0 minutes	Create Snapshot
slurm-worker-f284834	Ubuntu 20.04	193.6.16.159	m1.medium	emodi	Active	nova	None	Running	0 minutes	Create Snapshot
slurm-master-3b804a5	Ubuntu 20.04	193.6.16.174	m1.medium	emodi	Active	nova	None	Running	3 minutes	Create Snapshot

7. lépés: infrastruktúra kiépítése

- Sikeres lefutás után a virtuális gépek IP címei, node ID-jai, valamint az infrastruktúra azonosítója megjelenik a log üzenetek alján, listába szedve.

- Az infrastruktúra azonosítója elmenthető, vagy lekérdezhető az occopus-maintain parancs segítségével:

```
** 2021-02-17 17:26:33,041      Submitted infrastructure: 'c8553539-42c9-40b1-97bc-d53ab8947ca7'
** 2021-02-17 17:26:33,127      List of nodes/instances/addresses:
** 2021-02-17 17:26:33,127      slurm-worker:
** 2021-02-17 17:26:33,128        50c7cfe9-4c3c-4928-81e6-d58323b1e2b2:
** 2021-02-17 17:26:33,128          198.124.39.182
** 2021-02-17 17:26:33,128        98a82e45-6bf2-4cb9-9ead-953be4435bd7:
** 2021-02-17 17:26:33,128          198.124.39.144
** 2021-02-17 17:26:33,129      slurm-master:
** 2021-02-17 17:26:33,129        0b8269fe-78ec-47fc-8cdb-7195209e5123:
** 2021-02-17 17:26:33,129          198.124.39.67
c8553539-42c9-40b1-97bc-d53ab8947ca7
```

8. lépés: Az infrastruktúra használata

- A Slurm Master-re kapcsolódjunk rá SSH-n keresztül külső IP cím segítségével.
- Néhány alapvető parancs a Slurm-ben:
  - sinfo: áttekintést ad a fürt által kínált erőforrásokról
    (akkor működik, ha van min. 1db worker)
  - squeue: az erőforrások jelenleg mely job(ok)-hoz vannak hozzárendelve
- Az sinfo alapvetően a rendelkezésre álló partíciókat listázza.
  Egy partíció compute node-ok egy logikai csoportját foglalja magában.

```
ubuntu@slurm-master-61f21c0:~$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug*       up   infinite      2   idle slurm-worker-6c7eabe,slurm-worker-dff61eb
```

- Az ábrán a mi partíciónk a debug*. A csillag az alapértelmezett partícióra utal.

```
root@slurm-master-9b63b65:/storage# srun -J test.job -n 2 --ntasks-per-node 1 -o /storage/%j.txt hostname
root@slurm-master-9b63b65:/storage# ls
2.txt  slurm
root@slurm-master-9b63b65:/storage# cat 2.txt
slurm-worker-bfce264
slurm-worker-2cdcff9
```

Az infrastruktúra skálázása

Az Occopus segítségével az infrastruktúrák felfelé vagy lefelé skálázhatóak.

- **Felfelé skálázás:** amikor egy vagy több új node-ot adunk a klaszterhez
- **Lefelé skálázás:** amikor egy vagy több meglévő node-ot törlünk a klaszterből

Skálázási kérelem az Occopus-ban:

- `occopus-scale -n slurm-worker -c COUNT -i INFRA_ID`

Skálázandó csomópont neve

A skálázandó node-ok száma, mely irányát a szám előjele adja meg (+/-).

Az infrastruktúra azonosítója.

A skálázási kérelem végrehajtása:

- `occopus-maintain -i INFRA_ID`

A cél infrastruktúra azonosítója.

Az infrastruktúra skálázása

Felfelé skálázáskor a Slurm referencia architektúra működése:

- Az Occopus létrehozza a kért worker node-okat
- A worker node-ok telepítése és beállítása után csatlakoznak a master node-hoz
- A master újraindítja a Slurm szolgáltatásokat, majd megjelenik az új node

Lefelé skálázáskor a Slurm referencia architektúra működése:

- Az Occopus törli a kért worker node-okat
- A master pár perc múlva érzékeli, hogy nem érhetőek el a worker node-ok
  - törlésre kerülnek a node listából

Infrastruktúra törlése

- Az infrastruktúra lebontásához szükségünk lesz az infrastruktúra ID-ra. Az ID-nak a beszerzésére az alábbi módokon van lehetőség:
  - Az infrastruktúra felépítés végénél az Occopus kiírja:

```
** 2021-02-17 17:26:33,127      slurm-worker:
** 2021-02-17 17:26:33,128        50c7cfe9-4c3c-4928-81e6-d58323b1e2b2:
** 2021-02-17 17:26:33,128          198.124.39.182
** 2021-02-17 17:26:33,128        98a82e45-6bf2-4cb9-9ead-953be4435bd7:
** 2021-02-17 17:26:33,128          198.124.39.144
** 2021-02-17 17:26:33,129      slurm-master:
** 2021-02-17 17:26:33,129        0b8269fe-78ec-47fc-8cdb-7195209e5123:
** 2021-02-17 17:26:33,129          198.124.39.67
c8553539-42c9-40b1-97bc-d53ab8947ca7
```

  - Az Occopus által menedzselt infrastruktúrák lekérése: $ occopus-maintain -l

```
(occopus) ubuntu@occo:~$ occopus-maintain -l
Using default configuration file: '/home/ubuntu/.occopus/occopus_config.yaml'
Using default authentication file: '/home/ubuntu/.occopus/auth_data.yaml'
** 2021-02-19 07:45:09,861      Starting up; PID = 12555
List of active infrastructure:
c8553539-42c9-40b1-97bc-d53ab8947ca7
```

Infrastruktúra törlése

Az infrastruktúrát az **$ occopus-destroy -i <infraID>** paranccsal törölhetjük ki.

```
(occopus) ubuntu@occo:~$ occopus-destroy -i c8553539-42c9-40b1-97bc-d53ab8947ca7
Using default configuration file: '/home/ubuntu/.occopus/occopus_config.yaml'
Using default authentication file: '/home/ubuntu/.occopus/auth_data.yaml'
** 2021-02-17 19:26:21,112      Starting up; PID = 1787
** 2021-02-17 19:26:21,114      Start dropping infrastructure c8553539-42c9-40b1-97bc-d53ab8947ca7
** 2021-02-17 19:26:21,259      Dropping node 'slurm-worker'/'50c7cfe9-4c3c-4928-81e6-d58323b1e2b2'
** 2021-02-17 19:26:22,440      Dropping node 'slurm-worker'/'98a82e45-6bf2-4cb9-9ead-953be4435bd7'
** 2021-02-17 19:26:23,309      Dropping node 'slurm-master'/'0b8269fe-78ec-47fc-8cdb-7195209e5123'
** 2021-02-17 19:26:24,184      Finished dropping infrastructure c8553539-42c9-40b1-97bc-d53ab8947ca7
```

Összefoglalás

- A Slurm referencia architektúra működése
- A referencia architektúra kiépítése
- A Slurm használata és a legfontosabb parancsok
- A különböző működőképes Big Data/MI/konténer/workload menedzser környezetek létrehozását automatizáltan, az Occopus eszköz segítségével építjük ki
- A kiépítéshez nem szükséges mély informatikai, hálózati, vagy a szoftverek telepítéséhez és konfigurációjához szükséges tudás
- Az eddig összeállított környezetek (Hadoop, Spark, Tensorflow, Slurm, DataAvenue, Horovod, Kafka) referencia architektúra formájában elérhetők és kipróbálhatók az ELKH Cloud-on
- ELKH Cloud technikai support:

  info@science-cloud.hu

Köszönöm a figyelmet!