Slurm referencia architektúra az ELKH Cloud-on

Rusznák Attila

SZTAKI

Slurm

Mi az a Slurm?

ELKH Cloud

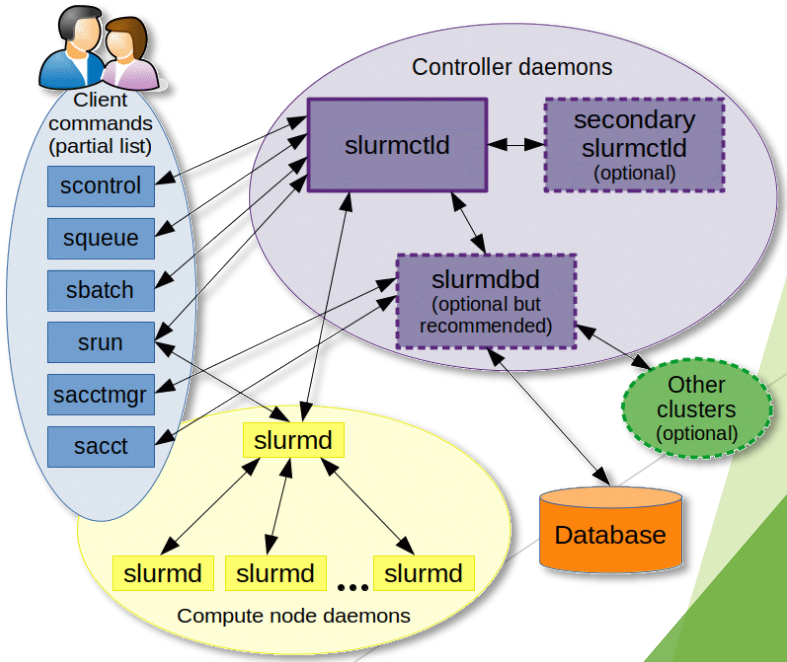
- **SLURM:** Slurm Workload Manager (Simple Linux Utility for Resource Management)
- A Slurm egy nyílt forráskódú, hibatűrő és nagymértékben skálázható fürtkezelő és feladatütemező rendszer nagy, illetve kis Linux fürtök számára.
  - Szuperszámítógépeken, klasztereken alkalmazzák
  - Többnyire teljesen önálló
- 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.

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

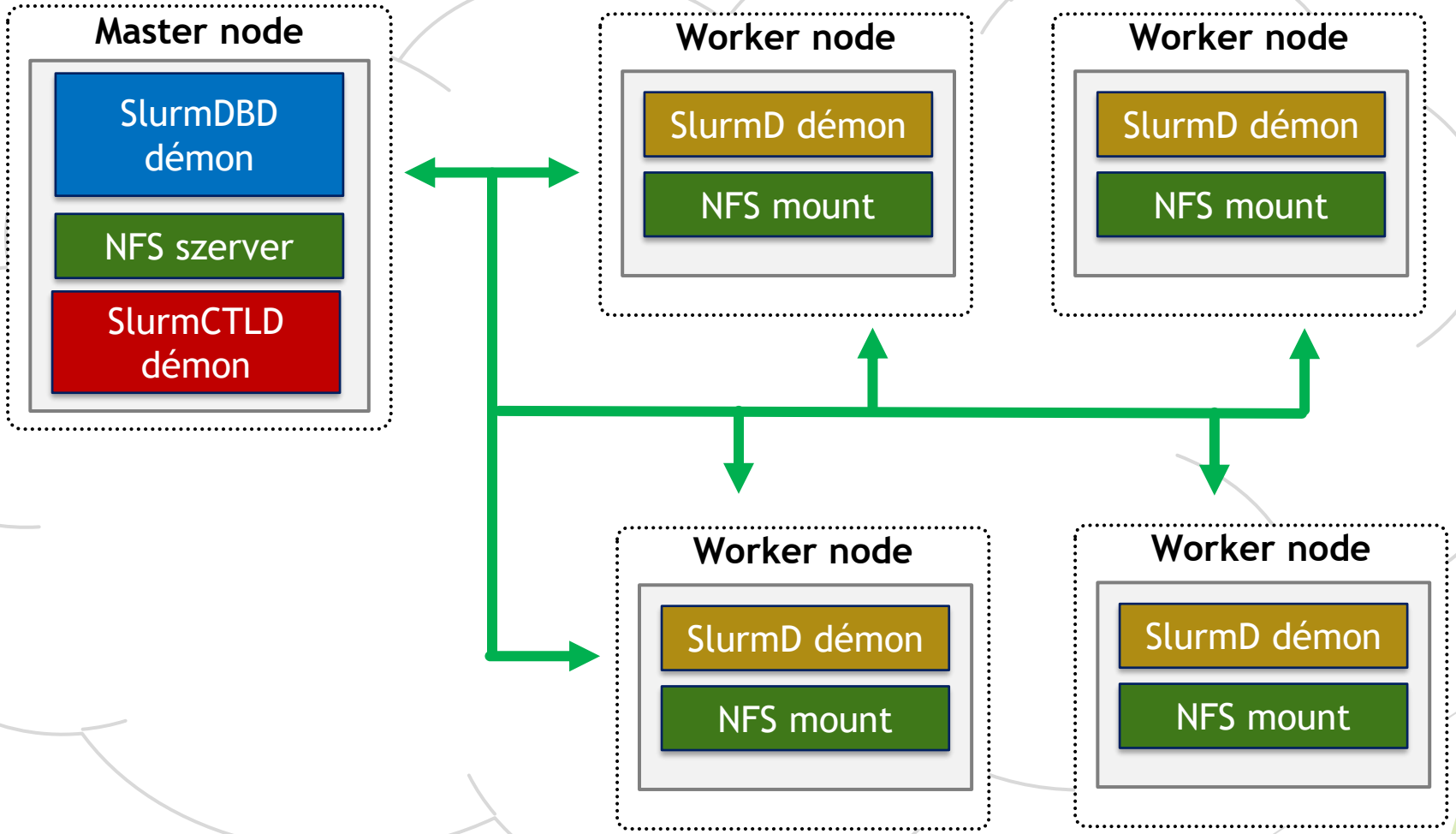
Azonos jelentéssel bírnak:

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



A Slurm referencia architektúra bemutatása

A Slurm architektúrája



ELKH Cloud

NFS: Network File Server

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-dataavenue.yaml**
8. Lépés: **Infrastruktúra használata**

0-1. lépés

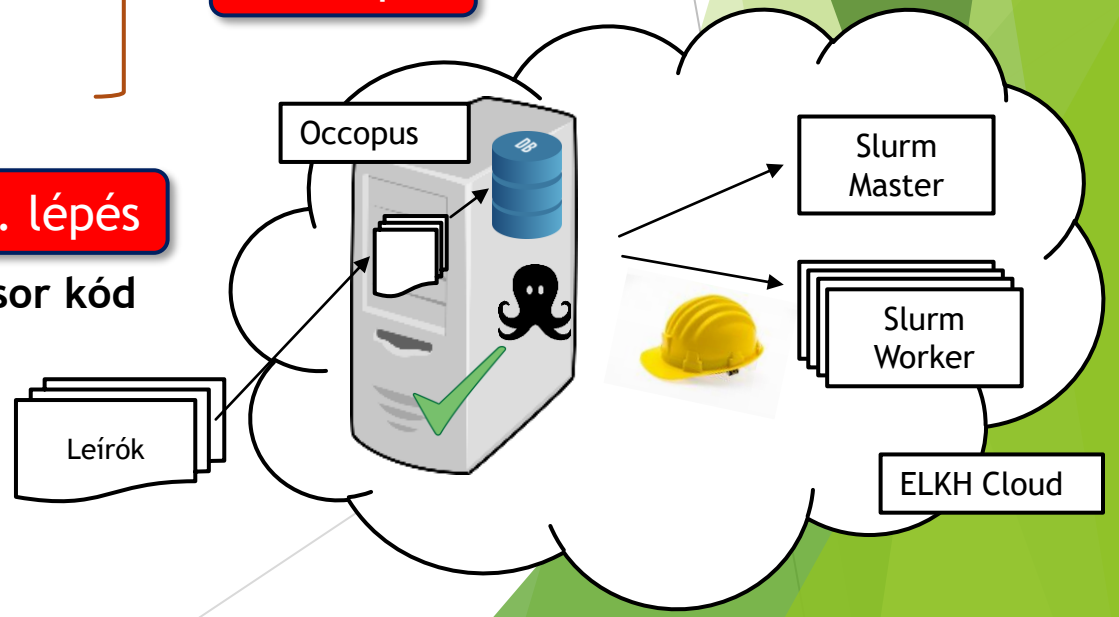
Csak első alkalommal kell beállítani.

Referencia architektúránként az Occopus-os gépen1x kell beállítani.

2-4. lépés

5-7. lépés

1-1 sor kód



2. lépés: A leírók letöltése a VM-re

ELKH Cloud

Csatlakozás | **Szolgáltatások** | Hírek | GYIK | Projektek | Dokumentumok | Publikációk | Kapcsolat | Fórum

Felhasználást segítő szolgáltatá

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 - Microsof*
- Kubernetes klaszter
- Apache Hadoop klaszter kiépítése
- Apache Spark klaszter RStudio stack-el
- Apache Spark klaszter Python stack-el *(Frissítés: ELKH Cloud - Mi*
- TensorFlow, Keras, Jupyter Notebook stack
- TensorFlow, Keras, Jupyter Notebook GPU stack *(Frissítés: E támogatással)*
- Horovod klaszter
- Kafka klaszter
- Slurm klaszter

Slurm klaszter

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

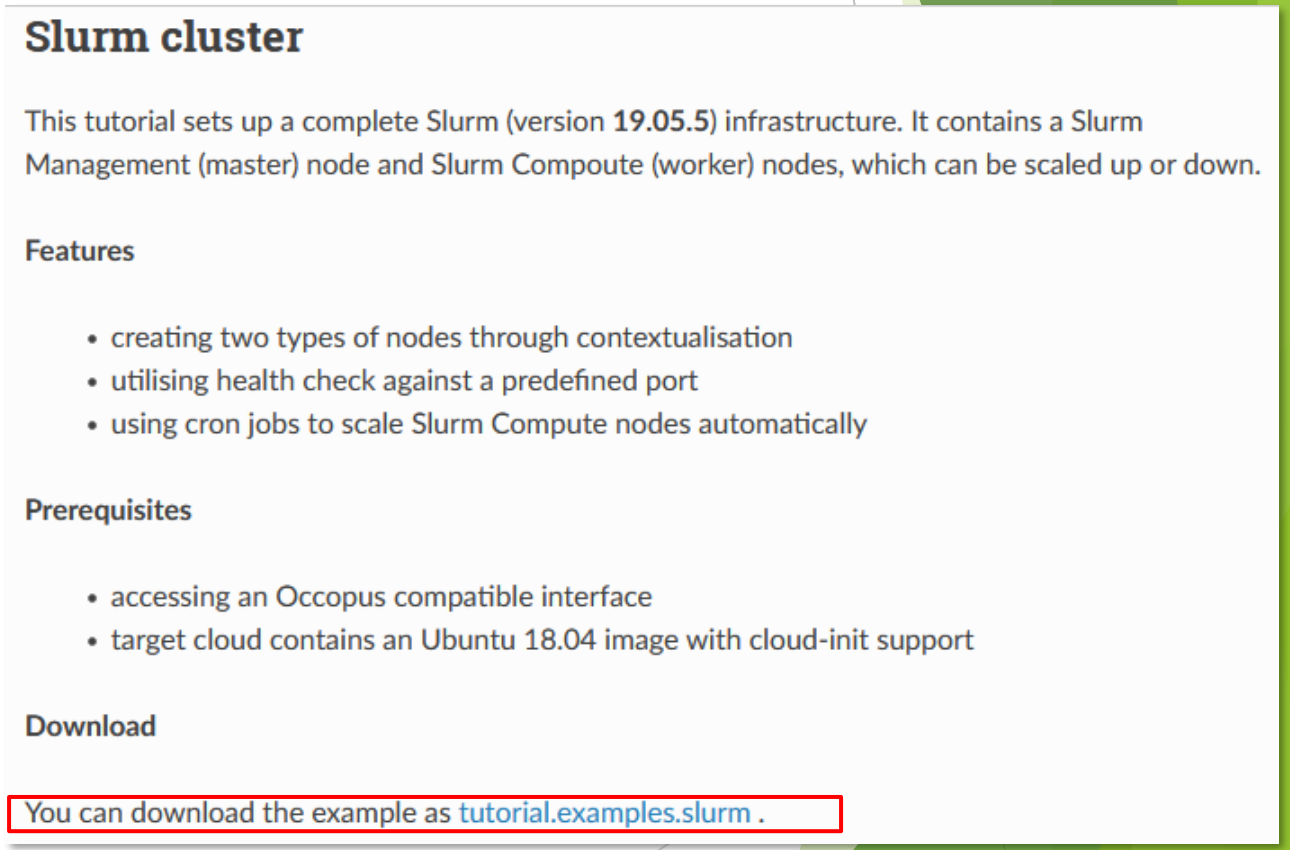
2. lépés: A leírók letöltése a VM-re

ELKH Cloud

A Slurm leírása

- Az Occopus weblapján:

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



Slurm cluster

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.

Features

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

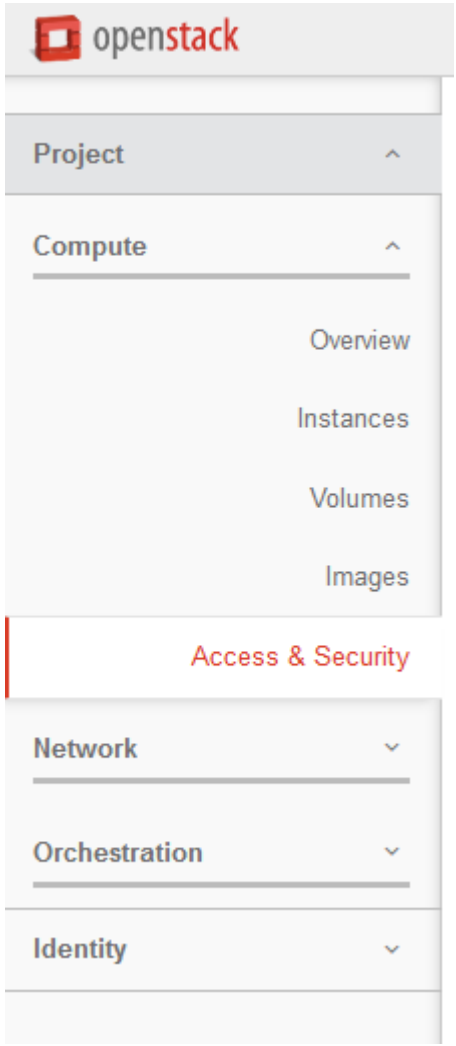
Prerequisites

- 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: A tűzfalszabályok beállítása



Access & Security

/ Manage Security Group Rules: Slurm (9162c4f5-8e42-47a8-be19-0087811ae601)

Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix
Egress	IPv6	Any	Any	::/0
Egress	IPv4	Any	Any	0.0.0.0/0
Ingress	IPv4	TCP	22 (SSH)	0.0.0.0/0
Ingress	IPv4	TCP	111	0.0.0.0/0
Ingress	IPv4	TCP	2049	0.0.0.0/0
Ingress	IPv4	TCP	6817 - 6819	0.0.0.0/0

Kimenő forgalom

SSH hozzáférés

RPCbind hozzáférés

NFS hozzáférés

Slurm hozzáférés

4. lépés: A leírók testreszabása

Infrastruktúra leíró fájl (**slurm-cluster/infra-slurm-cluster.yaml**)

Nova erőforrás szekció:

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

nodes:
  - &D
    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, *D ]
```

Itt lehet megadni a skálázhatóságot
https://occopus.readthedocs.io/en/latest/user-doc-createinfra.html#node-description

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

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

4. lépés: A leírók testreszabása

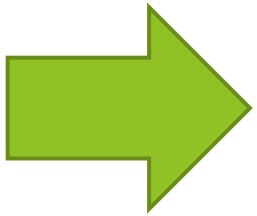
Csomópont definíciós fájl (**slurm-cluster/nodes/node_definition.yaml**)

Ajánlott operációsrendszer: 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_master.yaml
    health_check:
      ports:
        - 6819
      timeout: 1200

...
```



```
'node_def:spark_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: johndoe-key
      security_groups: [Slurm]
      floating_ip: yes
    contextualisation:
      type: cloudinit
      context_template: !yaml_import
        url: file://cloud_init_master.yaml
...
```

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

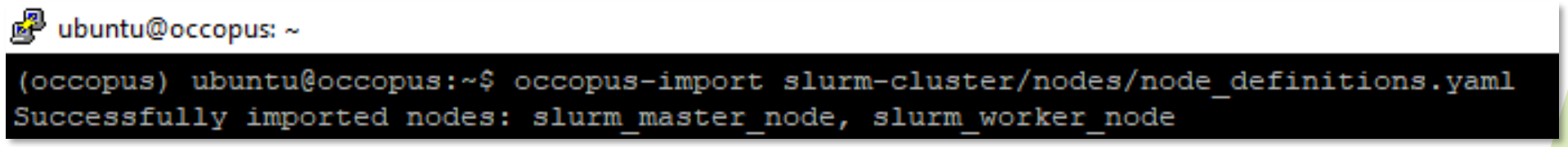
5-6. lépés: Aktiválás és importálás

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

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

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

- `(occopus) ubuntu@occopus:~$ occopus-import slurm-cluster/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: Az infrastruktúra kiépítése

Parancs a klaszter kiépítéséhez:

- `ubuntu@occopus:~$ occopus-build --parallelize slurm-claster/infra-slurm-cluster.yaml`

A parallelize segítségével a worker node-ok párhuzamosan építhetőek ki.

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

** 2021-06-13 11:13:21,391      Creating node 'slurm-master'/'00366ffa-2c0b-4e84-9ab0-a55ffd99edf'
...

** 2021-06-13 11:21:04,184      Health checking for node 'slurm-master'/'00366ffa-2c0b-4e84-9ab0-a55ffd99edf'
** 2021-06-13 11:21:05,360        Checking node reachability (00366ffa-2c0b-4e84-9ab0-a55ffd99edf):
** 2021-06-13 11:21:05,371          192.168.10.214 => ready
** 2021-06-13 11:21:05,371        Checking port availability (00366ffa-2c0b-4e84-9ab0-a55ffd99edf):
** 2021-06-13 11:21:05,373          6819 => ready
** 2021-06-13 11:21:05,373      Health checking result: ready
** 2021-06-13 11:21:05,376      Node 'slurm-worker'/'950c6e99-6aad-427d-a52d-470a153886ae' is ready.
** 2021-06-13 11:21:05,409      Creating node 'slurm-worker'/'950c6e99-6aad-427d-a52d-470a153886ae '
```

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created
occopus-slurm-cluster-d5427066-slurm-worker-9988aaee	Ubuntu 20.04 LTS Cloud Image - 20200518		m1.medium	attila-key	Build	nova	Scheduling	No State	0 minutes
occopus-slurm-cluster-d5427066-slurm-worker-d7ed8b10	Ubuntu 20.04 LTS Cloud Image - 20200518		m1.medium	attila-key	Build	nova	Scheduling	No State	0 minutes

7. lépés: Az infrastruktúra kiépítése

Az Occopus log üzenete kiépítés közben:

```
(occopus) ubuntu@occopus:~$ occopus-build slurm-cluster/infra-slurm-cluster.yaml
Using default configuration file: '/home/ubuntu/.occopus/occopus_config.yaml'
Using default authentication file: '/home/ubuntu/.occopus/auth_data.yaml'
** 2021-06-28 17:07:56,598      Starting up; PID = 14451
** 2021-06-28 17:07:56,606      [SchemaCheck] WARNING: missing "scaling" parameter in node 'slurm-master', using default scaling (single instance)
** 2021-06-28 17:07:56,614      Submitted infrastructure: 'f60f2ca0-6345-443b-b327-1cbf1c11f32e'
** 2021-06-28 17:07:56,614      Start maintaining the infrastructure f60f2ca0-6345-443b-b327-1cbf1c11f32e
** 2021-06-28 17:07:56,714      Creating node 'slurm-master'/'00366ffa-2c0b-4e84-9ab0-a555ffd99edf'
** 2021-06-28 17:08:04,631      Waiting for node 'slurm-master'/'00366ffa-2c0b-4e84-9ab0-a555ffd99edf' to become ready. No timeout.
** 2021-06-28 17:16:15,587      Health checking for node 'slurm-master'/'00366ffa-2c0b-4e84-9ab0-a555ffd99edf'
** 2021-06-28 17:16:16,680        Checking node reachability (00366ffa-2c0b-4e84-9ab0-a555ffd99edf):
** 2021-06-28 17:16:16,701          192.168.10.214 => ready
** 2021-06-28 17:16:16,702        Checking port availability (00366ffa-2c0b-4e84-9ab0-a555ffd99edf):
** 2021-06-28 17:16:16,703          6819 => pending
** 2021-06-28 17:23:37,922      Service on node 'slurm-master'/'00366ffa-2c0b-4e84-9ab0-a555ffd99edf' is down for 441.216 seconds! (Timeout for restart: 600s)
** 2021-06-28 17:23:48,418      Health checking for node 'slurm-master'/'00366ffa-2c0b-4e84-9ab0-a555ffd99edf'
** 2021-06-28 17:23:49,463        Checking node reachability (00366ffa-2c0b-4e84-9ab0-a555ffd99edf):
** 2021-06-28 17:23:49,473          193.224.59.182 => ready
** 2021-06-28 17:23:49,474        Checking port availability (00366ffa-2c0b-4e84-9ab0-a555ffd99edf):
** 2021-06-28 17:23:49,475          6819 => ready
** 2021-06-28 17:23:49,476      Health checking result: ready
** 2021-06-28 17:23:49,478      Node 'slurm-master'/'00366ffa-2c0b-4e84-9ab0-a555ffd99edf' is ready.
** 2021-06-28 17:23:49,488      Creating node 'slurm-worker'/'950c6e99-6aad-427d-a52d-470a153886ae'
** 2021-06-28 17:24:02,308      Waiting for node 'slurm-worker'/'950c6e99-6aad-427d-a52d-470a153886ae' to become ready. No timeout.
```

8. lépés: Az infrastruktúra használata

A Slurm Master-re kapcsolódjunk rá SSH-n keresztül külső IP cím segítségével.

A Slurm Master-en nem fog elindulni a slurmctld démon, amíg legalább egy worker nem csatlakozott!

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@occopus-slurm-cluster-f60f2ca0-slurm-master-00366ffa:~$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug*       up   infinite      2   idle occopus-slurm-cluster-f60f2ca0-slurm-worker-702a7cf9,occopus-slurm-cluster-f60f2ca0-slurm-worker-950c6e99
ubuntu@occopus-slurm-cluster-f60f2ca0-slurm-master-00366ffa:~$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
ubuntu@occopus-slurm-cluster-f60f2ca0-slurm-master-00366ffa:~$
```

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

8. lépés: Az infrastruktúra használata

Job létrehozása a Slurm-ben:

```
#!/bin/bash
#
#SBATCH --job-name=test
#SBATCH --output=res.txt
#
#SBATCH --ntasks=1
#SBATCH --time=10:00
#SBATCH --mem-per-cpu=100

srun hostname
srun sleep 60
```

Mentsük el: submit.sh

Ez a job 1db CPU-t kér 10 percre, 100 MB RAM-mal az alapértelmezett várólistán.

A `--job-name` paraméter segítségvel nevezhetjük el a job-okat.
Az `--output` pedig annak a fájlnak adja meg a nevét, ahová a kimenetet mentjük.
A job lefuttatja az `srun hostname` job step-et, azaz a unix hostname parancsot azon a node-on ahol kérték a CPU allokációt.
A következő job step elindítja a `srun sleep` parancsot 1 percre.

8. lépés: Az infrastruktúra használata

További hasznos Slurm parancsok:

- **sbatch:** job-ok beküldésére való parancs
- **sstat:** segítségével közel valós idejű információkat kaphatsz a futó programodról (memóriafogyasztás stb.)

Job beküldése a Slurm-ben:

- `sbatch submit.sh`

A Slurm visszaadja a beküldött job ID-ját:

- `Submitted batch job 2`

Az ID-vel megnézhetjük a részleteket:

- `Sstat –j 2`

```
ubuntu@occopus-slurm-cluster-f60f2ca0-slurm-master-00366ffa:~$ sudo sbatch submit.sh
Submitted batch job 2
ubuntu@occopus-slurm-cluster-f60f2ca0-slurm-master-00366ffa:~$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug*       up   infinite      1    mix occopus-slurm-cluster-f60f2ca0-slurm-worker-702a7cf9
debug*       up   infinite      1   idle occopus-slurm-cluster-f60f2ca0-slurm-worker-950c6e99
```

Slurm Master node

```
ubuntu@occopus-slurm-cluster-f60f2ca0-slurm-worker-702a7cf9:~$ ls
res.txt
ubuntu@occopus-slurm-cluster-f60f2ca0-slurm-worker-702a7cf9:~$ cat res.txt
occopus-slurm-cluster-f60f2ca0-slurm-worker-702a7cf9
```

Slurm Worker node

Fontos! Ha nem készítünk saját felhasználót, akkor csak sudo segítségével működnek a slurm utasítások!

8. lépés: Az infrastruktúra használata

A Slurm klaszter alapvető működése:

- Az NFS tárhelyen található egy slurm mappa
  - `cd /storage/slurm`
  - Az NFS a Master tárhelyét használja, de mindegyik node eléri
- Ebben található egy slurm.conf fájl
  - A fájl módosításával mind a master, mind a worker-ek lemásolják maguknak és újraindítják magukon a szolgáltatásokat. **Csak saját felelősségre módosítsuk ezt a fájlt!**

Hivatalos Slurm dokumentáció:

- https://slurm.schedmd.com

A slurm.conf fájl dokumentációja:

- https://slurm.schedmd.com/slurm.conf.html

Felhasználókezelés a Slurm-ben:

- https://slurm.schedmd.com/sacctmgr.html

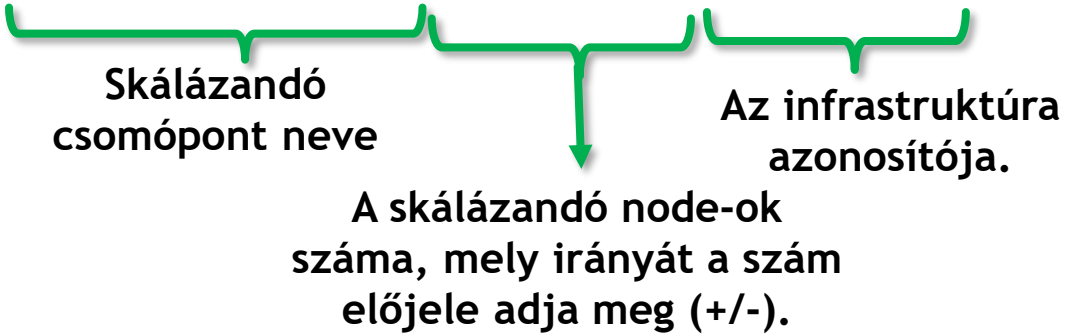
Az infrastruktúra skálázása

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`



A skálázási kérelem végrehajtása:

- `occopus-maintain – i INFRA_ID`

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
  - down állapotba kerülnek
  - majd down* állapotba kerülnek
  - végül törlésre kerülnek a node listából

Az infrastruktúra törlése

Az Occopus a kiépítés végén fontos információkat közöl:

```
** 2021-06-18 17:35:16,475     Submitted infrastructure: 'f60f2ca0-6345-443b-b327-1cbf1c11f32e'
** 2021-06-18 17:35:16,552     List of nodes/instances/addresses:
** 2021-06-18 17:35:16,552     slurm-master:
** 2021-06-18 17:35:16,552       00366ffa-2c0b-4e84-9ab0-a555ffd99edf:
** 2021-06-18 17:35:16,552         193.224.59.182
** 2021-06-18 17:35:16,553     slurm-worker:
** 2021-06-18 17:35:16,553       702a7cf9-51c5-4f87-b9bf-aec174214992:
** 2021-06-18 17:35:16,553         192.168.10.216
** 2021-06-18 17:35:16,553       950c6e99-6aad-427d-a52d-470a153886ae:
** 2021-06-18 17:35:16,554         192.168.10.215
```

Az infrastruktúra ID-ja lekérdezhető a következő Occopus utasítással is:

- `occopus-maintain -l`

Az infrsatruktúra törlése az egyedi ID alapján:

- `occopus-destroy –i f60f2ca0-6345-443b-b327-1cbf1c11f32e`

Az infrastruktúra törlése

```
(occopus) ubuntu@occopus:~$ occopus-destroy -i f60f2ca0-6345-443b-b327-1cbf1c11f32e
Using default configuration file: '/home/ubuntu/.occopus/occopus_config.yaml'
Using default authentication file: '/home/ubuntu/.occopus/auth_data.yaml'
** 2021-06-28 18:48:08,243      Starting up; PID = 15385
** 2021-06-28 18:48:08,254      Start dropping infrastructure f60f2ca0-6345-443b-b327-1cbf1c11f32e
** 2021-06-28 18:48:08,374      Dropping node 'slurm-master'/'00366ffa-2c0b-4e84-9ab0-a555ffd99edf'
** 2021-06-28 18:48:09,322      Dropping node 'slurm-worker'/'702a7cf9-51c5-4f87-b9bf-aec174214992'
** 2021-06-28 18:48:10,152      Dropping node 'slurm-worker'/'950c6e99-6aad-427d-a52d-470a153886ae'
** 2021-06-28 18:48:11,056      Finished dropping infrastructure f60f2ca0-6345-443b-b327-1cbf1c11f32e
```

Összefoglalás

- A Slurm referencia architektúra működése
- A referencia architektúra kiépítése
- A Slurm használata és a legfontosabb parancsok

- ELKH Cloud technikai támogatás:

  info@science-cloud.hu

Köszönöm a figyelmet!