



# FLOWBSTER – AUTODOCK VINA BEMUTATÓ

Az MTA Cloud felhasználói számára készült  
Flowbster – autodock vina bemutató v1.0

## Tartalom

|   |   |
|---|---|
| Áttekintés .....  | 2 |
| A Flowbster-ről .....   | 2 |
| Az AutoDock Vina alkalmazásról .....                                      | 2 |
| Lépések.....  | 3 |
| Occopus telepítése .....  | 3 |
| Letöltés .....  | 3 |
| 1. Erőforrás szekció kitöltése.....                                       | 3 |
| 2. Hitelesítés .....  | 3 |
| 3. Tűzfalszabály létrehozása .....  | 4 |
| 4. Gather szolgáltatás elindítása .....                                   | 4 |
| 5. Változók szerkesztése .....  | 4 |
| 6. Klaszter méretének beállítása .....                                    | 4 |
| 8. Felépítés .....  | 5 |
| 9. A felépült infrastruktúra .....  | 5 |
| 10. Input fájlok küldése .....  | 6 |
| 11. A folyamat ellenőrzése .....  | 6 |
| 12. Gather szolgáltatás leállítása .....                                  | 6 |
| 13. Infrastruktúra törlése .....  | 6 |
| Hibák, illetve a projekttel kapcsolatos további igények bejelentése ..... | 6 |

## Áttekintés

### A Flowbster-ről

A Flowbster egy új felhőorientált workflow-rendszer. Úgy tervezték, hogy hatékony pipeline-okat hozzon létre a felhőkben, amellyel a nagyon nagy adathalmazok is hatékonyan feldolgozhatók. A Flowbster workflow-t létre lehet hozni a célfelhőben virtuális infrastruktúraként, amelyen keresztül a feldolgozandó adatok áramlani tudnak. A Flowbster workflow igény szerint telepíthető a célfelhőben (MTA Cloud) a telepítés alapjául szolgáló Occopus orkesztrációs eszköznek segítségével. A Flowbster intuitív grafikus felhasználói felületet biztosít a végfelhasználó kutatók számára. Ez a felület lefedi az alacsony szintű felhőorientált rétegeket, és így a felhasználók az adatfeldolgozó alkalmazásuk üzleti logikájára koncentrálhatnak anélkül, hogy részletes ismeretre lenne szüksége az alapul szolgáló felhőalapú infrastruktúráról.

### Az AutoDock Vina alkalmazásról

Ebben az esetben a Flowbster-t használjuk fel a Vina workflow feldolgozásához szükséges infrastruktúra létrehozására. A telepítés a következő: egy VM működik generátorként, 5 VM működik Vina feldolgozó csomópontként, és végül egy VM működik Collector csomópontként. A workflow input fájllai magában foglalja a következőket: egy receptor molekulát, egy Vina konfigurációs fájlt és egy olyan molekulakészletet, amely a receptor molekulához kötődik. A generátor csomópont feladata, hogy a molekulák halmazát feldarabolja több részre. A Vina csomópontok feladata, hogy ezeket a részeket feldolgozzák, az adott részen minden egyes molekulán keresztül iterálják, a dokkolási szimuláció végrehajtásaként. A dokkolás eredménye magában foglalja az energia szintjét, és a legkisebb energiaszintű dokkolást. A Collector csomópont feladata, hogy az egyes molekulák feldolgozási eredményét a Vina csomópontok részéről kapja meg, és ezek közül kiválassza a legjobb 5 energiaszintet. A kísérlet futtatásához 3840 molekula molekula-készletet választottunk ki. Ezt a készletet 240 részre osztottuk, így mindegyik rész tartalmaz 16 molekulát a receptor molekula ellen.

#### Jellemzők:

- csomópontok létrehozása kontextualizáció által
- nova erőforrás használata
- egészség ellenőrzés használata előre definiált porttal és url-lel szemben (health check)
- skálázási paraméterek használata a worker csomópontok számának méretezéséhez

#### Előfeltételek:

- felhő elérése valamilyen Occopus kompatibilis felületen (pl. EC2, Nova, OCCI stb.)
  - Az Occopus eszköz képes az MTA Cloud-on virtuális gépeket indítani, ehhez a nova interfész használatát ajánljuk (ennek címe a felületen megtekinthető: Compute/Access & Security / API Access / Identity )
  - **Fontos: az Occopus eszköz használata jelenleg csak az MTA Cloud SZTAKI ágán működik!**
- a célfelhő tartalmaz egy felhőalapú támogatással rendelkező 14.04 alapú Ubuntu OS képfájlt
  - Ez a képfájl a Compute/Images/Public/Ubuntu 14.04 LTS Cloud image alatt megtalálható

## Lépések

### Occopus telepítése

Az automatikusan felépülő Flowbster architektúra kiépítése az Occopus eszköz segítségével fog megvalósulni, ezért elsőként telepítenünk kell az Occopus eszközt. Az Occopus orkesztrációs eszközt csupán egyetlen parancs segítségével is telepíteni tudja. A telepítéshez szükséges lépésekről, illetve az Occopus-ról bővebben [az alábbi linken tájékozódhat](#). Javasoljuk, hogy indítson el egy Ubuntu alapú virtuális gépet az MTA Cloud-ban, majd oda telepítse az Occopus eszközt.

### Letöltés

Az Occopus leírók alapján működik. Ezen leírók alapján fogja az Occopus felépíteni a cél felhőben az infrastruktúrát. A Flowbster telepítéséhez szükséges leírókat elkészítettük a felhasználók számára. Ezeket az alábbi linkről lehet letölteni: [tutorial.examples.flowbster-autodock.vina](https://tutorial.examples.flowbster-autodock.vina). Az Occopus-t futtató virtuális gépre telepítsük a leírókat.

**Megjegyzés:** Ebben bemutatóban nova felhő erőforrásokat használunk (az alap bemutató részben található nova oktatóanyagok alapján). Ugyanakkor nyugodtan használhat bármilyen Occopus – kompatibilis felhő erőforrást a csomópontok számára, de javasoljuk, hogy az összes csomópontot azonos felhőben építse fel.

### 1. Erőforrás szekció kitöltése

Nyissa meg a nodes/node\_definitions.yaml fájlt és szerkessze a csomópontok „node\_def:” címkével ellátott erőforrás szekcióját:

- Válasszon ki egy [Occopus kompatibilis erőforrás plugin-t](#)
  - Ez az MTA Cloud esetében a bemutatóban szereplő erőforrás plugin-nal azonos, nova erőforrás lesz
- Meghatározhatja [a plugin attribútumainak megfelelő listát](#)
- Követheti a bővítmény [attribútumainak értékeinek összegyűjtésére vonatkozó segítséget](#)
- A plugin-hez tartozó erőforrás sablonokhoz találhat [erőforrás plugin bemutatókat is](#)

Fontos, hogy indítás előtt a node definíciós fájlt személyre kell szabnia a felhasználónak. Ebben a fájlban megadjuk azokat az erőforrás azonosítókat, amelyeket használni fogunk, pld.: projekt ID, virtuális gép mérete stb. Ezeket az azonosítókat nem tudjuk megadni a felhasználó helyett, azonban azokat az MTA Cloud felületéről könnyedén össze lehet szedni. Ehhez részletes segítség [ezen linken](#), vagy az [alábbi dokumentációban](#) található. A példában található letölthető csomag a Nova plugin erőforrás sablont tartalmazza (ezt használjuk az MTA Cloud használatához is).

### 2. Hitelesítés

Az Occopus számára szükség van az MTA Cloudban használt felhasználónév/jelszó párosra, hogy az hitelesíteni tudja magát és képes legyen virtuális gépeket/infrastruktúrákat létrehozni egy adott projekt alatt. Kérjük, ellenőrizze, hogy az Occopus telepítése során helyesen adta-e meg a hitelesítési adatait! Az autentikációs adatok megadásához [az alábbi linken](#) talál segítséget.

### 3. Tűzfalszabály létrehozása

Az infrastruktúra komponensei egymáshoz kapcsolódnak, ezért több port tartományt is ki kell nyitni a virtuális gépek számára. Jelentkezzünk be az [MTA Cloud OpenStack felületére](#). A „Compute/Access&Security” menüpont alatt létrehozhatunk új tűzfalszabályt a „Create Security Group” gombra kattintva. Létrehozás után szerkeszteni tudjuk a tűzfalszabályt a „Manage Rules/Add Rule” gombra kattintva. Az alábbi portokat adjuk hozzá a tűzfalszabályhoz:

- TCP 5000 receiverport. Ezt a portot a csomópontok a bejövő kérések fogadására használják.

### 4. Gather szolgáltatás elindítása

Felhívjuk a figyelmét arra, hogy az eredmények fogadásához el kell indítani egy Gather szolgáltatást. (Flowbster része), amely összegyűjti az eredményeket (a legalacsonyabb energiaszintű dokkolási szimulációkat) a gyűjtőből (Collector – a munkafolyamat utolsó csomópontja). Indítsa el a Gather szolgáltatást a következő parancs segítségével:

```
scripts/flowbster-gather.sh -s
```

Alapértelmezés szerint a Gather szolgáltatás az 5001-es porton figyel.

**Megjegyzés:** A „scripts” nevű mappában található szkriptek futtatásához szükség van a Python 2.7-re. Alternatívaként aktiválja az Occopus virtuális környezetét (virtualenv)!

### 5. Változók szerkesztése

Szerkessze a „variables” szekciót az *infra-autodock-vina.yaml* fájlban. Állítsa be az alábbi értékeket:

- **gather\_ip** az ip címe annak a host-nak ahol a Gather szolgáltatás el lett indítva
- **gather\_port** az a port hol a Gather szolgáltatás figyel

```
gather_ip: &gatherip "<External IP of the host executing the Gather service>"
gather_port: &gatherport "5001"
```

### 6. Klaszter méretének beállítása

Ha szükséges, frissítse a VINA csomópontok számát. Ehhez szerkessze az *infra-autodock-vina.yaml* fájlt és módosítsa a „min” paramétert a „scaling” kulcsszó alatt. A skálázás az az intervallum, amelyben a csomópontok száma megváltozhat (min, max). Jelenleg a minimális érték 5-re van állítva (ami az indításkor a kezdeti szám lesz).

```
- &VINA
  name: VINA
  type: flowbster_node
  scaling:
    min: 5
```

## 7. Importálás

Importáljuk a személyre szabott leírókat az Occopus adatbázisába:

```
occopus-import nodes/node_definitions.yaml
```

Győződjön meg róla, hogy a megfelelő virtualenv aktiválva van! Amennyiben ezt még nem tette volna meg korábban, az alábbi parancs segítségével aktiválható az Occopus virtuális környezete:

```
source occopus/bin/activate
```

**Fontos:** Az Occopus akkor veszi fel a csomópont definíciókat az adatbázisból, amikor felépíti az infrastruktúrát, így mindig importálásra van szükség, ha a node definíciós fájl, vagy bármelyik (pld.: kontextualizációs) fájl megváltozik!

## 8. Felépítés

Indítsa el az infrastruktúra felépítését az alábbi paranccsal:

```
occopus-build infra-autodock-vina.yaml
```

## 9. A felépült infrastruktúra

Sikeres lefutás után a virtuális gépek IP címei, node ID -jai, valamint az infrastruktúra azonosítója megjelenik a log üzenetek alján, listába szedve. Az infrastruktúra azonosító elmenthető, vagy lekérdezhető az occopus-maintain parancs segítségével:

```
List of nodes/ip addresses:  
VINA:  
  <ip-address> (2f7d3d7e-c90c-4f33-831d-91e987e8e8b2)  
  <ip-address> (49bed8d2-94b0-4a7e-9672-744921dacac0)  
  <ip-address> (10664026-0b31-4848-9f7a-98f880f98be7)  
  <ip-address> (a0f5d091-aecc-488c-94f2-34e546f87832)  
  <ip-address> (285d7efd-84a7-4ed5-a6fa-73db47bc2e87)  
COLLECTOR:  
  <ip-address> (4ca11ad3-a6ec-411b-89e6-d516169df9c7)  
GENERATOR:  
  <ip-address> (9b8dc4f1-bed4-4d1c-ba9e-45c18ee2523d)  
30bc1d09-8ed5-4b7e-9e51-24ed881fc166
```

## 10. Input fájlok küldése

Miután felépült az infrastruktúra az input fájlokat elküldhetjük a workflow Generator csomópontjára (ellenőrizze a címét a csomópontnak az *occopus-build* parancs kimenetének végén). Az input fájlok elküldéséhez használja az alábbi parancsot a *flowbster-autodock-vina/inputs* könyvtárban:

```
../scripts/flowbster-feeder.sh -h <ip of GENERATOR node> -i input-  
description-for-vina.yaml -d input-ligands.zip -d input-receptor.pdbqt -  
d vina-config.txt
```

A `-h` paraméter a Generátor csomópont címe, `-i` az input leíró fájl és a `-d` kapcsolóval pedig megadhatjuk az adat fájl(oka)t.

**Megjegyzés:** A szkriptek könyvtárban található szkriptek futtatásához Python 2.7-re van szükség. Alternatívaként aktiválja az Occopus virtuális környezetét (virtualenv)!

**Megjegyzés:** Néhány percig eltarthat ameddig a folyamatok véget nem érnek. Kérjük, legyen türelemmel!

## 11. A folyamat ellenőrzése

A 10. lépésnél megkezdődött az adatfeldolgozás. Az egész feldolgozási idő a VINA csomópontok teljesítményétől függ. A VINA csomópontok 240 molekulacsomagot dolgoznak fel, amelyeket a Collector csomópont gyűjt össze. A feldolgozás folyamatát a Collector csomóponton ellenőrizheti a */var/flowbster/jobs/<munkafolyamat azonosító>/inputs* könyvtárban található fájlok számával. Amikor a fájlok száma elérte a 240-et, akkor a Collector csomópont egyesíti őket és egy csomagot küld a Gather csomópontra, amely a */tmp/flowbster/results* könyvtárban tárolódik.

## 12. Gather szolgáltatás leállítása

Miután befejeződött a molekulák feldolgozása, a Gather szolgáltatás leállítható:

```
scripts/flowbster-gather.sh -d
```

## 13. Infrastruktúra törlése

Végül, lebonthatja az infrastruktúrát az *occopus-build* által visszaadott infrastruktúraazonosító használatával:

```
occopus-destroy -i 30bc1d09-8ed5-4b7e-9e51-24ed881fc166
```

## Hibák, illetve a projekttel kapcsolatos további igények bejelentése

Az MTA Cloud szolgáltatással kapcsolatos kommunikáció és támogatás e-mail formájában történik. A közös e-mail cím: [info@cloud.mta.hu](mailto:info@cloud.mta.hu). Az erre az e-mail címre bejelentett hibákból, igényekből egy bejelentési űrlap generálódik, melyet az MTA Cloud csapat kijelölt tagja kezel.