



## Fejlesztői kapacitások meghatározására vonatkozó módszertan



## Tartalomjegyzék

1	Bevezetés.....	3
2	Erőforrások menedzselése .....	3
2.1	Felhasználók száma alapján történő erőforrás menedzselés.....	3
2.1.1	Jira .....	4
2.1.2	BitBucket .....	4
2.1.3	Bamboo .....	4
2.1.4	Confluence .....	4
2.1.5	Nexus.....	5
2.1.6	Jira ServiceDesk .....	5
2.2	Fizikai erőforrások mennyisége alapján történő erőforrás menedzselés .....	5
2.2.1	OpenShift .....	5
2.2.1.1	Fizikai erőforrások .....	5
2.2.1.2	Erőforrások hozzáférése .....	6
2.2.1.3	Erőforrások mérése.....	7
3	Kapacitás kezelési módszertan.....	7
3.1	Szempontrendszer .....	7
3.2	Kapacitásainak meghatározása és beállítása.....	8
3.3	A rendszer használatának monitorozása .....	8
3.4	A szempontrendszernek való megfelelés.....	9
4	Szakmailag indokolt mértékű fejlesztői kapacitások .....	10
4.1	Ingyenes szolgáltatáscsomagok .....	10
4.2	Csatlakozáskor igényelhető ingyenes szolgáltatások.....	10
4.3	További ingyenes szolgáltatások.....	10

## 1 Bevezetés

A dokumentum célja az ÁAFP-hez rendelkezésre álló műszaki infrastruktúra hatékony felhasználását biztosító koncepció bemutatása. A dokumentum fő fejezeteiben

- ismerteti az erőforrások elosztásának, a felhasznált erőforrások monitorozásának, korlátozásának és skálázásának technikai lehetőségét
- bemutatja a platform aktuális kihasználtságán, valamint az alkalmazás-fejlesztési projektek várható kifutásán és csatlakozásán alapuló kapacitáskezelési módszertant
- meghatározza a vonatkozó kormányrendeletben előírtak szerint az ingyenesen igénybe vehető szakmailag indokolt mértékű fejlesztői kapacitásokat.

A magasabb szolgáltatás szinteket biztosító prémium szolgáltatás csomagok a pilot időszak futtatási tapasztalatai alapján a későbbiekben kerülhetnek meghatározásra.

## 2 Erőforrások menedzselése

A platform erőforrásaihoz való hozzáférést az alábbi módokon tudjuk szabályozni:

- időben
- felhasználók (nevesített / konkurens) számában
- használható fizikai erőforrások mennyiségében

Az időben történő szabályozás a platform szempontjából egy külső szabályozás, ami folyamat szinten van biztosítva: a felhasználók nem egyszerre férnek hozzá az erőforrásokhoz, hanem időben elosztva. A dokumentum a továbbiakban nem foglalkozik ezzel a módszerrel.

A továbbiakban a felhasználók száma és a használható fizikai erőforrások mennyisége szempontjából kerülnek bemutatásra az erőforrás menedzsment folyamatok.

### 2.1 Felhasználók száma alapján történő erőforrás menedzselés

A fejezetben felsorolásra kerülnek azok a rendszerek, ahol lehetőség van a felhasználók száma alapján történő erőforrás menedzselésre.

Meghatározásra kerül, hogy az egyes rendszerek milyen módszerrel licenszelik a felhasználókat (nevesített / konkurens), hogy mennyi a rendelkezésre álló licenkek száma, illetve hogy van-e lehetőség a felhasznált licenkek számának mérésére. Szintén meghatározásra kerül, hogy van-e lehetőség tenantonként korlátozni a hozzáférések számát, illetve mérni a felhasznált licenkek számát.

### 2.1.1 Jira

Erőforrások hozzáférése	
Felhasználó alapú licenszelés típusa	Nevesített / Konkurens
Felhasználó licenzek száma (összesen)	500/2000
Tenantonként korlátozható a felhasználók száma?	Igen/Nem
Erőforrások mérése	
Mérhető a belépett felhasználók száma (összesen)?	Igen/Nem
Tenantonként mérhető a felhasználók száma?	Igen/Nem

### 2.1.2 BitBucket

Erőforrások hozzáférése	
Felhasználó alapú licenszelés típusa	Nevesített / Konkurens
Felhasználó licenzek száma (összesen)	100/2000
Tenantonként korlátozható a felhasználók száma?	Igen/Nem
Erőforrások mérése	
Mérhető a belépett felhasználók száma (összesen)?	Igen/Nem
Tenantonként mérhető a felhasználók száma?	Igen/Nem

### 2.1.3 Bamboo

Erőforrások hozzáférése	
Felhasználó alapú licenszelés típusa	Nevesített / Konkurens
Agent licenzek száma (összesen)	5/25
Tenantonként korlátozható a felhasználók száma?	Igen/Nem
Erőforrások mérése	
Mérhető a belépett felhasználók száma (összesen)?	Igen/Nem
Tenantonként mérhető a felhasználók száma?	Igen/Nem

### 2.1.4 Confluence

Erőforrások hozzáférése	
Felhasználó alapú licenszelés típusa	Nevesített / Konkurens

Felhasználó licenzek száma (összesen)	500/2000
Tenantonként korlátozható a felhasználók száma?	Igen/Nem
Erőforrások mérése	
Mérhető a belépett felhasználók száma (összesen)?	Igen/Nem
Tenantonként mérhető a felhasználók száma?	Igen/Nem

### 2.1.5 Nexus

Erőforrások hozzáférése	
Felhasználó alapú licenzelés típusa	Nevesített / Konkurens
Felhasználó licenzek száma (összesen)	OSS
Tenantonként korlátozható a felhasználók száma?	Igen/Nem
Erőforrások mérése	
Mérhető a belépett felhasználók száma (összesen)?	Igen/Nem
Tenantonként mérhető a felhasználók száma?	Igen/Nem

### 2.1.6 Jira ServiceDesk

Erőforrások hozzáférése	
Felhasználó alapú licenzelés típusa	Nevesített / Konkurens
Felhasználó licenzek száma (összesen)	50 v 100 v 250 (igény alapján)
Tenantonként korlátozható a felhasználók száma?	Igen/Nem
Erőforrások mérése	
Mérhető a belépett felhasználók száma (összesen)?	Igen/Nem
Tenantonként mérhető a felhasználók száma?	Igen/Nem

## 2.2 Fizikai erőforrások mennyisége alapján történő erőforrás menedzselés

A fejezetben felsorolásra kerülnek azok a rendszerek, ahol lehetőség van a fizikai erőforrások mennyisége alapján történő erőforrás menedzselésre.

### 2.2.1 OpenShift

#### 2.2.1.1 Fizikai erőforrások

Az OpenShift rendszerben jelenleg erőforrás alatt az elérhető cpu, memory és fizikai tárhely mennyiséget értjük.

Jelenleg a rendszer (20 worker node együttesen) az alábbi fizikai erőforrás kapacitásokkal rendelkezik:

- 168 vcpu
- 2560 GB memory
- 6680 GB tárhely

A későbbiekben a rendszer kapacitása bővíthető, a fent felsorolt értékeket frissíteni kell.

#### 2.2.1.2 Erőforrások hozzáférése

Az OpenShift rendszerben az erőforrás kezelés a szeparációhoz hasonlóan project alapon kerül kialakításra. A rendszer lehetővé teszi, hogy minden egyes projectben korlátozzuk a fizikai erőforrások (cpu, memory) használatát. Ez a korlátozás úgy történik, hogy a project alatt futó konténerek által használt erőforrás igények összeadódnak, és ha ez az összesített erőforrás igény meghaladja a project szinten meghatározott kvótát, akkor adott konténer nem fog elindulni. Erre a célra az alábbi erőforrás típust használjuk:

- ResourceQuota

A project szintű kvóták megadásánál minden egyes fizikai erőforrás típushoz az alábbi két érték kerül beállításra:

- requests.\* (lefoglalt mennyiség)
- limits.\* (maximum mennyiség)

Ilyenkor minden egyes fizikai erőforrás típusnál a project alatt futó konténereknél megadott request értékek összeadódnak és ezek az eredmények kerülnek összehasonlításra a project szintű kvótaként megadott értékekkel.

Példa a project szintű kvóta megadására:

**apiVersion: v1**

**kind: ResourceQuota**

**metadata:**

**name: compute-resources**

**spec:**

**hard:**

**requests.cpu: "0.1"**

**requests.memory: 1Gi**

`requests.ephemeral-storage: 2Gi`

`limits.cpu: "0.2"`

`limits.memory: 2Gi`

`limits.ephemeral-storage: 4Gi`

A megoldás lehetőséget ad a túlfoglalásra, azaz magasabb limits.\* értékekkel az egyes projectek alatt futó konténerek több erőforrást is használhatnak mint ami garantáltan ki lett osztva nekik, ezáltal lehetőség adódik a rendszer kihasználatlan kapacitásainak csökkentésére.

Mivel az egyes tenantok projectek mentén vannak szétválasztva, a project szintű kvóták egyben tenant szintű kvótákat is jelentenek, így minden egyes tenant esetében lehetőség adódik a fizikai erőforrások hozzáféréseinek szabályozására.

### 2.2.1.3 Erőforrások mérése

Az egyes namespace-ek alatti erőforrás felhasználást a Prometheus rendszer monitorozza az alábbiak szerint:

- Adott project alatt futó konténerek által használt cpu
- Adott project alatt futó konténerek által használt memória
- Teljes clusteren storage kihasználtság
- Disk I/O műveletek forgalma
- Cluster és podok közötti hálózati forgalom

A fenti mérések a kapacitás kezelési módszertanban ismertetett folyamathoz biztosít adatokat.

## 3 Kapacitás kezelési módszertan

### 3.1 Szempontrendszer

A kapacitás kezelési módszertan kialakításnál a szolgáltató és az ügyfél oldali szempontokat is figyelembe kell venni.

Szolgáltató oldali szempontok:

- A platform kihasználtsága a lehető legmagasabb legyen
- A lehető legtöbb ügyfelet lehessen kiszolgálni
- A platform kapacitás-bővítési igények időben jelentkezzenek

Ügyfél oldali szempontok:

- A szolgáltatást folyamatosan igénybe lehessen venni
- A szolgáltatást jó minőségben lehessen igénybe venni

- A szolgáltatás teljesítse a vállalt erőforrás garanciákat
- Legyen lehetőség a szolgáltatás szint bővítésére

### 3.2 Kapacitásainak meghatározása és beállítása

Egy a lenti erőforrásokkal rendelkező rendszernek 20 ügyfelet kell kiszolgálnia egy időben.

- 6 cpu
- 48 GB memória
- 250 GB tárhely

A fentiekből meghatározhatóak a felhasználók számára és a fizikai erőforrások használatára vonatkozó kezdeti garantált erőforrások. Ez ügyfelenként:

- 5 nevesített (vagy 2 konkurens) felhasználó
- 300 milicore CPU
- 2.4 GB memória
- 5 GB tárhely

A garantált kvóták mellett korlátozzuk a felhasználható erőforrások maximális mértékét (1.5x):

- 9 nevesített (vagy 3 konkurens) felhasználó (ahol támogatott)
- 450 milicore CPU
- 3.6 GB memória
- 5 GB tárhely

A fenti értékek beállításra kerülnek a rendszerekben.

### 3.3 A rendszer használatának monitorozása

Tenantonként monitorozásra kerül a felhasználók száma és az igénybe vett fizikai erőforrások mennyisége. Az alábbi szituációk állhatnak elő:

Elemzés eredménye	Beavatkozás
Az aktív felhasználók száma nem éri el a garantált felhasználók számának a felét	Garantált felhasználók számának felezése
Az aktív felhasználók száma meghaladja (konkurens esetben), vagy eléri a garantált felhasználók számát.	Garantált felhasználók számának kétszerezése
A felhasznált fizikai erőforrások mennyisége nem éri el a garantált érték felét	Garantált fizikai erőforrás kvóta felezése
A felhasznált fizikai erőforrások mennyisége meghaladja a garantált értéket	Garantált fizikai erőforrás kvóta kétszerezése

Mielőtt módosításra kerülnek a kvóták az alábbi két dolgot kell megvizsgálni:



Elemzés eredménye	Beavatkozás
A tenantok számára garantált kvóták összege jelentősen kevesebb mint a platform kapacitása.	További tenantok beengedése a platformra.
A tenantok számára garantált kvóták összege megközelíti a platform kapacitását.	Platform kapacitás bővítés

Érdemes lehet tartalékot képezni a kiosztható kvótákban, mert a bővítés mindig időigényes, az átmeneti időszakban ez a tartalék biztosíthatja a folytonosságot. Ez a tartalék lehet 1 Közepes csomagnyi erőforrás, így a következő felhasználót még be lehet engedni egy Közepes csomaggal a platform bővítése nélkül. Ebben az esetben ezt egy egy tartalék kvótát figyelembe kell venni a erőforrások kezdeti szétosztásánál (+1 tenant).

Ezzel folyamatosan kivédhető az a szituáció, amikor egy alacsony fordulatszámon működő projekt időben nagyon elhúzódik, és leköti a platform erőforrásait.

### 3.4 A szempontrendszernek való megfelelés

A felvázolt módszertan megfelel a korábban említett szolgáltató és az ügyféloldali szempontrendszernek.

Szolgáltató oldali szempontok:

- **A platform kihasználtsága a lehető legmagasabb legyen:**  
A módszertan nem engedi hogy egyes tenantoknál kihasználatlan dedikált kapacitások ragadjanak be.
- **A lehető legtöbb ügyfelet lehessen kiszolgálni:**  
A módszertan kihasználatlan kapacitások esetén lehetőséget ad további tenantok csatlakozására.
- **A platform kapacitásbővítési igények időben jelentkezzenek:**  
A módszertan jelzi ha szükség van a platform kapacitás bővítésére.

Ügyfél oldali szempontok:

- **A szolgáltatást folyamatosan igénybe lehessen venni:**  
A kvóták finomhangolása nincs hatással a szolgáltatás elérhetőségére.
- **A szolgáltatást jó minőségben lehessen igénybe venni:**  
A módszertan érzékeli és korrigálja ha egy tenant esetében nem elegendő az allokált kvóta.
- **A szolgáltatás teljesítse a vállalt erőforrás garanciákat** A garantált erőforrások tekintetében nem alkalmazunk túlfoglalást.
- **Legyen lehetőség a szolgáltatás szint emelésére:**  
A módszertan a kvóták növelésén keresztül biztosítja a szolgáltatás szint emelését.

## 4 Szakmailag indokolt mértékű fejlesztői kapacitások

Egy fejlesztői kapacitás szakmailag akkor indokolt, ha ténylegesen ki van használva. Azaz:

- Egy fejlesztési projektben résztvevő fejlesztők száma indokolt, ha a megadott számú fejlesztő ténylegesen dolgozik a projekten és használja a platformot.
- Egy szoftver rendszer teszteléséhez szükséges fizikai erőforrás igény indokolt, ha a rendszer tesztelése ténylegesen folyik és a rendszer kihasználja az igényelt fizikai erőforrást.

### 4.1 Ingyenes szolgáltatáscsomagok

Az ÁAFK-ban ingyenesen az alábbi táblázatban szereplő kapacitásokat és erőforrásokat biztosítja az ügyfelei számára:

Név	Garantált				Maximális			
	User (fő)	CPU (mcore)	RAM (GB)	Tárhely (GB)	User (fő)	CPU (mcore)	RAM (GB)	Tárhely (GB)
XS - Kisebb	1	75	0.6	1	2	113	0.9	1
S - Kis	3	150	1.2	3	4	225	1.8	3
M - Közepes	5	300	2.4	5	9	450	3.6	5
L - Nagy	10	600	4.8	10	19	900	7.2	10
XL - Nagyobb	20	1200	9.6	20	30	1800	14.4	20

### 4.2 Csatlakozáskor igényelhető ingyenes szolgáltatások

- **S – Kis:** A Közepes csomag kvótáinak fele
- **M - Közepes:** Egy átlagos alkalmazás fejlesztéséhez és futtatásához szükséges kvóták
- **L - Nagy:** A Közepes csomag kvótáinak duplája

### 4.3 További ingyenes szolgáltatások

A rendszernek a fenti csomagokban szereplő kapacitásokhoz viszonyított alul vagy túl használata esetén további két csomag is elérhető. Ezekbe a csomagokba csak "átkerülni" lehet, csatlakozáskor nem lehet kiválasztani:

- **XS - Kisebb:**  
A Kis csomag kvótáinak fele. Ha valaki ezt a csomagot is alulhasználja, akkor is itt marad. Meg kell különböztetni azt az esetet, amikor valaki egyáltalán nem használja a rendszert.

- **XL - Nagyobb:**

A Nagy csomag kvótáinak duplája. Ha valaki ezt a csomagot túlhasználja, akkor is itt marad. Ha még nagyobb kapacitásra van szüksége, akkor érdemes elgondolkodni egy dedikált rendszeren.