

# DEVOPS

4. forduló



A kategória támogatója: EPAM

## Ismertető a feladathoz

**A 4. forduló után elérhetőek lesznek a helyezések %-os formában: azaz kiderül, hogy a kategóriában a versenyzők TOP 20% - 40% -60% -ához tartozol-e!**

Szeretnénk rá felhívni figyelmedet, hogy a játék nem Forma-1-es verseny! Ha a gyorsaságod miatt kilököd a rendesen haladó versenyzőket, kizárást vonhat maga után!

Felhasznált idő: 00:00/10:00

Elért pontszám: 0/7

## 1. feladat 0/1 pont

CloudFront egyedi SSL tanúsítvány esetén melyik AWS régió AWS Certificate Manager-ben (ACM) kell a tanúsítványt tárolni?

### Válasz

- ☐ Az adott CloudFront disztribúció régiójában, mert az ACM régióspecifikus
- ☒ us-east régióban  
Ez a válasz helyes, de nem jelölted meg.
- ☐ Bármelyik régióban, mert az ACM-et a CloudFront globális szolgáltatás, és bármelyik régióban eléri az adott ACM-ben tárolt tanúsítványt
- ☐ us-west régióban

### Magyarázat

Mivel a CloudFront globális szolgáltatás a hozzá tartozó egyedi tanúsítványt az AWS US EAST (us-east-1) régiójának az ACM-jébe kell tenni

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/cnames-and-https-requirements.html>

## 2. feladat 0/1 pont

Git verziókezelő használata esetén érdemes bizonyos fájl típusokra (pl. bináris fájlok) kivételt definiálni, hogy azokat ne kezelje a git. (pl. "git add" műveletkor)

Hogy lehet ezt megtenni?

### Válaszok

- ☒ `.git/info/exclude` fájlban  
Ez a válasz helyes, de nem jelölted meg.
- ☒ `.gitignore` fájlban  
Ez a válasz helyes, de nem jelölted meg.
- ☒ a `'core.excludesfile'` konfigurációs beállításnál megadott fájlban  
Ez a válasz helyes, de nem jelölted meg.
- ☐ a `GITEXCLUDES` környezeti változóban

### Magyarázat

Mind a 3 helyen megadható a kizárandó fájlok neve, névmintája. Lásd `gitignore(5)`.

## 3. feladat 0/1 pont

IaC toolok esetén visszatérő kérdés, hogy az érzékeny adatokat hogyan kezeljük. Terraform esetén milyen lehetőségeid vannak ezen érzékeny adatok elrejtésére a parancsok outputjából?

### Válaszok

- ☒ Az adott változók jelölése sensitive markerrel  
Ez a válasz helyes, de nem jelölted meg.
- ☒ Sensitive függvény használatával  
Ez a válasz helyes, de nem jelölted meg.
- ☐ Egyik sem, mert a local state használata esetén az állapotot plain-text JSON-ban tárolja
- ☐ Csak a változók jelölése sensitive markerrel

### Magyarázat

Terraform v0.15-től a sensitive function is elérhető ritkább használati esetekre, például ha fileból olvasunk fel érzékeny adatokat.

## 4. feladat 0/1 pont

Prometheusban az alábbi feltétel használatával milyen eredményre lennénk kíváncsiak?

**`"((node_memory_MemTotal_bytes - node_memory_MemFree_bytes - node_memory_Cached_bytes - node_memory_Buffers_bytes) / (node_memory_MemTotal_bytes )) * 100 > 85"`**

### Válasz

- ☐ Ahol 85% a szabad memória
- ☒ Ahol a memóriahasználat nagyobb, mint 85%  
Ez a válasz helyes, de nem jelölted meg.
- ☐ A lekérdezés hibás
- ☐ Ahol a puffer memória mérte nagyobb, mint a teljes memória 85%

## Magyarázat

Lekérdezés első része kivonja a szabad, cache és puffer memóriát az összes memóriából. Majd, hogy megkapjuk a százalékos értékét, az eredményt elosztjuk a teljes memória nagyságával és megszorozzuk 100-al.

## 5. feladat 0/1 pont

Helm és Kustomize összehasonlítása esetén melyik állítás **igaz**?

### Válasz

- ☐ A Helm közvetlenül cluster-be deploy-ol, ezért nem képes statikus YAML fájlok generálására
- ☒ A Kustomize nem tartalmaz template engine-t a resource-ok testreszabásához  
**Ez a válasz helyes, de nem jelölted meg.**
- ☐ Egy Helm chart nem tud kiterjeszteni egy már meglévő Helm chart-ot
- ☐ Egy Kustomization nem képes kiterjeszteni egy már meglévő Kustomization-t

## Magyarázat

Bar a Helm általában cluster-be történő deployment-tel van használva, de tartalmaz egy "helm template" alparancsot, ami az átadott paraméterek alapján statikus Kubernetes YAML-t generál.

Ez így igaz, a kizárólag kustomize strukturális transzformációk útján képes változtatásokat eszközölni már meglévő YAML manifest-eken (pl. strategic merge vagy jsonpatch formájában).

## 6. feladat 0/1 pont

Az alább felsorolt deployment stratégiák közül melyiket választanád, ha az üzleti oldal az alábbi igényeket támasztja egy vertikálisan jól skálázható alkalmazás új verziójának telepítésekor?

- Hiba esetén egyszerű visszaállást biztosítson az előző verzióra
- Hiba esetén próbáljuk minimalizálni a hibát esetlegesen észlelő felhasználók számát
- Fontos szempont a költség, így lehetőség szerint tartsuk alacsonyan a telepítés során felhasznált extra erőforrásokat
- Lehetőleg ne legyen leállás az alkalmazás frissítése során

### Válasz

- ☐ Blue-green deployment
- ☒ Canary deployment  
**Ez a válasz helyes, de nem jelölted meg.**
- ☐ Mindent leállítunk egy olyan időszakban, amikor nem nagyon használják a szolgáltatást, majd frissítés után újraindítjuk a rendszert
- ☐ A feltételek sajnos nem teljesíthetők egyszerre, így szükséges az üzleti oldal követelményeinek megváltoztatása

## Magyarázat

Canary deployment lehetőséget biztosít arra, hogy az alkalmazás új verziója csak a felhasználók egy kis csoportja által legyen elérhető, így ebben a környezetben lehetőségünk van megvizsgálni az alkalmazás működését, és hiba esetén leállítani a

## 7. feladat 0/1 pont

AWS Cloudformation-ben melyik a javasolt megoldás, ha szeretnénk biztosítani, hogy az Autoscaling Group-unkbán megfelelő számú instance legyen általunk elvárt állapotban?

### Válasz

- ☐ Autoscaling Group-nak az Amazon EC2 status check-ek használatával
- ☒ CreationPolicy cfn-helper scripttel  
**Ez a válasz helyes, de nem jelölted meg.**
- ☐ CreationPolicy DependsOn attribútummal
- ☐ WaitCondition használatával

### Magyarázat

CreationPolicy attribútum cfn-helper scriptekkel való használata javasolt az AWS által erre a helyzetre.

Az EC2 status check, mint health check nem az általunk elvárt állapotot reprezentálja. Mind a CreationPolicy, mind a DependsOn attribútumok és a CreationPolicy már jelöli a resource-ot, amin dependálunk.

WaitCondition-t stack-en kívüli resource-ok koordinálására javasolja AWS-en.

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-waitcondition.html>

