

DEVOPS

2. forduló



A kategória támogatója: EPAM

RENDELKEZÉSRE ÁLLÓ IDŐ:

10:00

Ismertető a feladathoz

Fontos információk

Ha kifutsz az adott feladatlap kitöltésére rendelkezésre álló időből, a felület **automatikusan megpróbálja beküldeni** az addig megadott válaszokat

A kérdésekre **mindig van helyes válasz!** Ha csak egy helyes válasz van az adott kérdésre, radio button-os választási lehetőségeket fogsz látni.

Olyan kérdés viszont nincs, amelyre az összes válasz helyes!

Egyéb információkat a [versenyszabályzatban](#) találsz!

Felhasznált idő: 02:07/10:00

Elért pontszám: 0/19

1. feladat 0/3 pont

Linux rendszerben melyik parancs nem fájlkezelésre szolgál

Válasz

- ☐ cat
- ☒ htop
- ☐ diff
- ☐ touch

Magyarázat

A http parancs a rendszer folyamatok monitorozására szolgál.

2. feladat 0/3 pont

Milyen protokollokon keresztül lehet továbbítani http forgalmat?

Válaszok

☒ TLS

☐ SIP

☒ TCP

☐ IMAP

Magyarázat

A Session Initiation Protocol (SIP) egy kapcsolat felépítési protokoll az internetes telefóniában (VoIP), míg az IMAP egy email lekérési szabvány protokollja (lásd még POP3).

3. feladat 0/3 pont

Linux parancssorban melyik kilépési kód jelenti a kiadott parancs sikeres lefutását? (válasz egy szám)

Válaszok

A helyes válasz:

0

Magyarázat

A sikeres futást 0 exit kód jelöli azon megfontolásból, hogy ez egy egyedi kitüntetett eset míg a többi kódot (≤ 255) konvencionálisan hibák típusainak jelzésére tartják fent, amiből nyilván több lehet.

4. feladat 0/4 pont

Egy Dockerfile-ban felülről lefelé mi az optimális sorrendje egy Java-s alkalmazás futtatásához a következő komponenseket tekintve:

1. Java Runtime Environment
2. az alkalmazás jar file
3. az alkalmazásunk által közvetlenül használt egyéb Java-s dependenciák

Válasz

- ☐ 1, 2, 3
- ☒ 1, 3, 2
- ☐ 2, 3, 1
- ☐ 3, 1, 2

Magyarázat

Célszerű az image-et úgy rétegezni, hogy a ritkán változó részeket helyezzük a Dockerfile elejére, ezáltal jelentős idő és hálózati forgalom megtakarítást tudunk elérni akár image build akár push-pull helyzetekben.

5. feladat 0/3 pont

Milyen natív paranccsal támogatja a git a regressziós hibák felderítését (tegyük fel, hogy a master-ben már előfordul a hiba, de van egy korábbi verziód, ahol még működött a kód és a kettő között jelentős számú commit van)?

Válasz

- ☐ Commit-ról commit-ra lépkedve a git regression magas szintű paranccsal
- ☒ Bináris kereséssel a git bisect magas szintű paranccsal
- ☐ Véletlenszerű commit-okat kiválasztva a git random magas szintű paranccsal
- ☐ Nem támogatja

Magyarázat

git bisect paranccsal bináris keresést lehet végezni és az egyes teszt futásokra jelezni, hogy azok jól vagy hibásan futottak, így $\log_2(n)$ időben megtalálható a commit, ahol a hiba először jelentkezett.

6. feladat 0/3 pont

Milyen nyelveken írhatunk Terraform konfigurációt?

Válaszok

☐ XML

☒ JSON

☐ YAML

☒ HCL

Magyarázat

HCL (Hashicorp Configuration Language) mellett JSON is használható, bár ritka. Akkor hasznos, ha a kód részeket programból állítjuk elő ahol a JSON kezelő könyvtárak jellemzően jól támogatottak.

[Legfontosabb tudnivalók](#)

[Kapcsolat](#)

[Versenyszabályzat](#)

[Adatvédelem](#)

© 2022 Human Priority Kft.

KÉSZÍTETTE

Megjelenés

 Világos 