

Cloud Engineering (IBM)

1. forduló

Ismertető a feladathoz

Az Imagination Wizard Kft. egy közepes méretű egészség-megőrzési kellékeket gyártó és forgalmazó cég amely szignifikáns üzleti potenciállal rendelkezik a Kelet Közép Európai régióban. Értékesítése az elmúlt 3 évben meghaladta az évi 600 millió eurót. Dolgozói 5 országban, 3 gyár és 4 háttértámogató szervezet egységében tevékenykedik az üzletkötők kiterjedt nemzetközi hálózata mellett.

A cég vezetése új termékek piacra bocsátását tervezi, amelytől az árbevétel és a piaci részesedés megduplázódását várja.

Elemzések egybehangzó iránymutatása alapján a siker egyik kulcsa a megfelelő informatikai támogatás, rendszerfejlesztés megvalósítása, ami rugalmas kiszolgálást, skálázható szolgáltatásokat, megbízható rendszereket, önálló szoftverfejlesztési folyamatirányítást nyújt. Ezen célkitűzések súlyozott prioritást kapnak korunk vírushelyzetre felkészült, rugalmas működtetést igénylő időszakában.

A te feladatod az lesz, hogy feltérképezd a jelenlegi IT környezetet, felhőalapú infrastruktúrát és kiszolgáló rendszert tervezz és implementálj, valamint a DevOps vívmányait megfontolva modern és minden igényt kielégítő szoftverfejlesztési folyamatot alakíts ki a cég számára.

**Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 10 perc került rögzítésre mint megoldáshoz felhasznált idő.**

1. feladat 0 / 2 pont

Hogyan mérjük fel a meglévő IT rendszereket?

- ☒ A szervezet belső IT szolgáltatáskatalógusának felhasználásával
- ☐ Az adatbázisok migrálásával
- ☒ Architektúrális dokumentációk elemzésével
- ☐ Pénzügyi igazgatóval való beszélgetéssel

Magyarázat a megoldáshoz

Első körben a belső IT katalógus adataira támaszkodva megbízható elemzéseket lehet készíteni és képet kapni a cég alkalmazás portfóliójáról, az alkalmazott technológiák listájáról és a rendszerek általános nagyságáról. Ezután az alkalmazások szintjén készített elemzések segítenek a pontosításban.

2. feladat 0 / 2 pont

Felhő alapú migrációt szem előtt tartva, mi a legjobb módja az IT rendszerek csoportosításának?

- ☒ Alkalmazott programozási nyelvek szerint
- ☒ Alkalmazott szerver technológiák szerint
- ☐ Alkalmazott kódsorok mennyisége szerint
- ☒ Adat bizalmasság szintje szerint
- ☐ Fejlesztők száma szerint

Magyarázat a megoldáshoz

A platform és a nyelv amin az alkalmazás vagy a megoldás készült a legfontosabb megszorító elem. Ezek csoportosítása kiváló tervezési alapot ad a migrációhoz, ami a projekt időbeli lefolyását és a migráció sorrendjét is sugallni tudja.

3. feladat 0 / 2 pont

Mik a felhő alapú szolgáltatás ismérvei (public cloud)?

- ☒ Felhasználó által választható szolgáltatások
- ☒ Gyors, automatizált erőforrás allokáció
- ☒ Erőforrás megosztás
- ☐ Nagy összegű, előzetesen fizetendő befektetést igényel
- ☐ A cég helyi hálózatán üzemeltethető
- ☒ Gyors, pillanatnyi igényhez igazodó erőforrás skálázás
- ☐ Programozási nyelvhez kötött

Magyarázat a megoldáshoz

Minden a piacon lévő fontos cloud szolgáltató a vevő (felhasználó) direkt hozzáférést biztosít az infrastruktúrához. A kiválasztott erőforrást web felületen keresztül, gyorsan lehet lefoglalni ahol az adatközpontot használó többi vevővel osztozunk ez erőforrásokon.

4. feladat 0 / 2 pont

Melyek a private cloud karakterisztikái?

- ☐ A bérelt szolgáltatáshoz tartozó szoftver és hardver komponenseken más vevők is futtatnak alkalmazásokat vagy tárolhatnak adatot
- ☒ A szolgáltatáshoz tartozó szoftver és hardver komponenseket kizárólag csak a vevő használja
- ☒ A cég saját szervertermében maga vagy egy harmadik szolgáltató is üzemeltetheti
- ☒ A szolgáltatás-spektrumot magában foglalja (CPU, adat, hálózat)

Magyarázat a megoldáshoz

A publikus felhővel ellentétben a magán felhő esetében az erőforrásokat kizárólag 1 vevő használja, ahol a vevő 1 vagy több folyamatot is futtathat. Ezeket az erőforrásokat üzemeltetheti egy külső szolgáltató (Amazon, Google, IBM) vagy a cég saját maga működteti (pénzintézetek)

5. feladat 0 / 2 pont

Melyik esetben nem tudjuk eltávolítani / eliminálni az összes saját szervert?

- ☐ Ha webszervereket is hosztolunk
- ☒ Ha van olyan belső szabályzat vagy külső törvényi kötelezettség, ami előírja a saját tulajdonú szerver üzemeltetését
- ☐ Ha az archivált data mennyisége több mint 100 TB
- ☐ Ha az alkalmazásoknak 200ms alatti válaszidőt kell garantálniuk

Magyarázat a megoldáshoz

Léteznek olyan iparágak és iparági szabályzatok ahol az adatok védelme előírja a pontos lokalizációt. Pl az adatot nem lehet egy adott ország határain kívül tárolni és/vagy más által is használt szerveren tárolni. Ebben az esetben ha a szolgáltatónak nincs adatközpontja az adott országban vagy nem tud a szabályzatnak megfelelni akkor mindenképp saját szervert kell üzemeltetni.

6. feladat 0 / 2 pont

Mi a legfontosabb célkitűzése a DevOps folyamat kialakításának?

- ☐ A fejlesztők munkájának hatékonyabbá, egyszerűbbé tétele modern eszközök segítségével korlátozás nélkül
- ☒ Összekapcsolni a fejlesztői és üzemeltetői csapatok folyamatait, gyakorlatait
- ☐ A legújabb technológiák fejlesztői stratégiába történő integrálása
- ☐ A fejlesztői és üzemeltető munka eloszlásának kiegyenlítése, az erőforrások optimalizálása

Magyarázat a megoldáshoz

A DevOps fő célja a fejlesztői és üzemeltetői csapatok folyamatainak, gyakorlatainak összehangolása, így megteremtve az automatizáció és skálázhatóság feltételeit.

7. feladat 0 / 2 pont

Gyors skálázhatóság, fenntartási költségek csökkentése és rugalmasság: a következők közül melyiknek jellemzői?

- ☐ Folyamatos szállítás
- ☐ Folyamatos telepítés
- ☒ Felhő környezet
- ☐ Folyamatos integráció

Magyarázat a megoldáshoz

A felhő környezet a fenntartási költségek csökkentését, skálázhatóságot és rugalmasságot tesz lehetővé.





Cloud Engineering (IBM)  
2. forduló

Ismertető a feladathoz

KERETTÖRTÉNET:

Az Imagination Wizard Kft. egy közepes méretű egészségmegőrzési kellékeket gyártó és forgalmazó cég amely szignifikáns üzleti potenciállal rendelkezik a Kelet Közép Európai régióban. Értékesítése az elmúlt 3 évben meghaladta az évi 600 millió eurót. Dolgozói 5 országban, 3 gyár és 4 háttértámogató szervezet egységében tevékenykedik az üzletkötők kiterjedt nemzetközi hálózata mellett.

A cég vezetése új termékek piacra bocsájtását tervezi, amelytől az árbevétel és a piaci részesedés megduplázódását várja.

Elemzések egybehangzó iránymutatása alapján a siker egyik kulcsa a megfelelő informatikai támogatás, rendszerfejlesztés megvalósítása, ami rugalmas kiszolgálást, skálázható szolgáltatásokat, megbízható rendszereket, önjáró szoftverfejlesztési folyamatirányítást nyújt. Ezen célkitűzések súlyozott prioritást kapnak korunk vírushelyzetre felkészült, rugalmas működtetést igénylő időszakában.

A te feladatod az lesz, hogy feltérképezd a jelenlegi IT környezetet, felhőalapú infrastruktúrát és kiszolgáló rendszert tervezz és implementálj, valamint a DevOps vívmányait megfontolva modern és minden igényt kielégítő szoftverfejlesztési folyamatot alakíts ki a cég számára.

Az előző lépésben megtörtént az IT környezet felmérése. Most az igények összegyűjtése és a koncepció kialakítása a feladat.

**Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 10 perc került rögzítésre mint megoldáshoz felhasználható idő.**

1. feladat 0 / 2 pont

A cég a teljes alkalmazásportfólióját átalakítja felhő alapú szolgáltatásra. Kiválasztott egy felhő szolgáltatót, de az átalakítás egy éven túlmutató projekt lett. Milyen szolgáltatásban van a cég amíg a projekt tart és a transzformáció folyamatban van?

- ☒ Hibrid felhő (Hybrid Cloud)
- ☐ Magán felhő (Private Cloud)
- ☐ Publikus felhő (Public Cloud)
- ☐ Több szereplős felhő (Multi Cloud)

Magyarázat a megoldáshoz

Hibrid felhő állapotnak nevezzük amikor egy cég saját adatközpontot is üzemeltet de mellette külső felhőszolgáltatást is igénybe vesz.

2. feladat 0 / 2 pont

Mikor jó FaaS (Function as a Service) platformot választani?

- ☐ Amikor fix költségű megoldást keresel hogy a költségek jól tervezhetőek legyenek
- ☐ Amikor számításiigényes feladatokat hajtasz végre
- ☒ Ha a fejlesztő nem akar vagy nem tud az infrastruktúrával minimálisan sem foglalkozni
- ☐ Amikor az alkalmazásod sok szöveges alapú adatot kezel és dolgoz fel

Magyarázat a megoldáshoz

A FaaS használata az össze futtatókörnyezet gondját leveszi a fejlesztő válláról. A fejlesztés folyamán kizárólag a programnyelvet kell meghatározni és közvetlenül a webfelületen kódolni. A szolgáltató ezt a kódot lefordítja, konténerizálja és futtatja.

3. feladat 0 / 2 pont

Milyen szoftverfejlesztési architektúra passzol legjobban a FaaS (Function as a Service) platformhoz?

- ☐ Adatelemzés központú
- ☒ Micro szervíz architektúra
- ☐ Kliens - szerver megoldás
- ☐ Monolitikus tervezés

Magyarázat a megoldáshoz

A FaaS platform kifejezetten kicsi, 1-1 funkció szintű kódolásra van optimalizálva ami legjobban a mikro service architektúrával passzol.

4. feladat 0 / 2 pont

A felhő alapú tervezésben újraírt alkalmazások melyik terhelés elosztást preferálják?

- ☒ Horizontális terhelés elosztás újabb pod-ok és kód futtató végpontok hozzáadásával egy konténerizált platformon
- ☐ Vertikális skálázás nagyobb (erősebb) szerver üzembe állításával

Magyarázat a megoldáshoz

A felhő alapú szolgáltatók mindig a horizontális terhelés elosztást preferálják, mert egy jól megírt program esetében egy megfelelő load balancer segítségével sokkal tovább skálázható mintha 1 szerver erőforrásait növelnénk.

5. feladat 0 / 2 pont

Az egyik most fejlesztendő új, 3 tier alkalmazás folyamatos adatelemzést fog készíteni. Az informatika csak az üzleti folyamatra és az adatra akar fókuszálni. Melyik platform az optimális választás?

- ☐ Saját szerver
- ☐ IAAS (Infrasturcture as a Service)
- ☒ PaaS (Platform as a Service)
- ☐ FaaS (Function as a Service)

Magyarázat a megoldáshoz

A PaaS platform esetében a szolgáltató kezeli a futtatókörnyezettől lefelé mindent (middleware, operációs rendszer, virtualizáció, hálózat, etc). A fejlesztő feladata az alkalmazás futókönyvezeéről (konténer) és az adatokról gondoskodni.

6. feladat 0 / 3 pont

Ahhoz, hogy egy a felhasználóval interakcióban lévő web front-end alkalmazásunk virtuálisan végtelen terhelést is képes legyen lekezelni, milyen karakterisztikákkal kell rendelkeznie?

- ☒ Terhelés elosztó használata (Workload Balancer)
- ☐ CI/CD folyamatok használata
- ☒ Alkalmazástól független adat csatornák (Message Queue) használata hogy az alkalmazás és az adat független legyen egymástól
- ☐ NoSQL adatbázis használata
- ☐ Mesterséges intelligencia használata
- ☒ Mikro szervíz architektúra használata

Magyarázat a megoldáshoz

Egy kiugró terheléssel rendelkező, a felhasználóval interakcióban lévő honlap (aliexpress, amazon.com, etc) esetében horizontális skálázással (workload balancer) kell dolgoznunk. Az alkalmazásnak függetlenül kell kezelnie az adatot minden másról és minden funkciót egyénileg skálázhatóvá kell tennie.

7. feladat 0 / 3 pont

Azonosítsd a lehetséges érveket amellett, hogy a cég a hagyományos rendszerfejlesztési életciklusról áttérjen a DevOps struktúrára.

- ☒ Megbízhatóbb, stabilabb frissítések költséghatékony módon
- ☐ A DevSecOps több hangsúlyt fektet a szoftver biztonságra mint a tradicionális rendszerfejlesztési struktúra
- ☐ A szabályozások és adatvédelmi törvények kötelezik a váltást
- ☒ Minőségi fejlesztések gyakoribb szállítása

Magyarázat a megoldáshoz

A DevSecOps célja, hogy gyakrabban szállítsunk jó minőségű szoftver fejlesztéseket. Megbízható és költséghatékony frissítéseket kell szállítani az üzleti igényeknek megfelelően.

8. feladat 0 / 3 pont

A felhő infrastruktúra milyen tulajdonságaival segíti a DevSecOps hatékony alkalmazását az alábbiak közül?

- ☒ Működési költségek csökkentése
- ☐ Automatikus archiválás
- ☒ Gyors skálázhatóság (Rapid elasticity)
- ☒ Rugalmasság
- ☐ Kommunikációs eszközök

Magyarázat a megoldáshoz

A felhő környezet nagyobb foku rugalmasságot és skálázhatóságot biztosít. A felhő biztosít használat alapú elszámolást ezzel csökkentve a futtatási költségeket.

9. feladat 0 / 3 pont

Melyik telepítési minta (release pattern) nem segíti a DevOps kód alacsony kockázattal történő telepítését az éles rendszerbe?

- ☐ Kanári telepítési minta (Canary deployment pattern) – Sötét indítás (Dark Launch)
- ☐ Kék-zöld telepítési minta (Blue-green deployment pattern)
- ☐ Cluster immunrendszer telepítési minta (Cluster immune system release pattern)
- ☒ Nagy durranás telepítések (Big bang code deployments), teljesen tesztelt és validált verziók

Magyarázat a megoldáshoz

Az első három opció célja, hogy minél kisebb rizikóval juttassunk kódot az éles rendszerbe.

Az utolsó telepítési minta sok változtatást juttat élesbe mely több helyen elvégezhet, nagyobb kockázattal jár.



## Ismertető a feladathoz

### KERETTÖRTÉNET:

Az Imagination Wizard Kft. egy közepes méretű egészségmegőrzési kellékeket gyártó és forgalmazó cég amely szignifikáns üzleti potenciállal rendelkezik a Kelet Közép Európai régióban. Értékesítése az elmúlt 3 évben meghaladta az évi 600 millió eurót. Dolgozói 5 országban, 3 gyár és 4 háttértámogató szervezet egységében tevékenykedik az üzletkötők kiterjedt nemzetközi hálózata mellett.

A cég vezetése új termékek piacra bocsátását tervezi, amelytől az árbevétel és a piaci részesedés megduplázódását várja."

Elemzések egybehangzó iránymutatása alapján a siker egyik kulcsa a megfelelő informatikai támogatás, rendszerfejlesztés megvalósítása, ami rugalmas kiszolgálást, skálázható szolgáltatásokat, megbízható rendszereket, önálló szoftverfejlesztési folyamatirányítást nyújt. Ezen célkitűzések súlyozott prioritást kapnak korunk virushelyzetre felkészült, rugalmas működéstést igénylő időszakában.

A te feladatod az lesz, hogy feltérképezd a jelenlegi IT környezetet, felhőalapú infrastruktúrát és kiszolgáló rendszert tervezz és implementálj, valamint a DevOps vívmányait megfontolva modern és minden igényt kielégítő szoftverfejlesztési folyamatot alakíts ki a cég számára.

Ebben a fordulóban folytatódik az igények összegyűjtése és a koncepció kialakítása.

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 12 perc került rögzítésre mint megoldáshoz felhasznált idő.

## 1. feladat 0 / 2 pont

Több alkalmazásunk is szenzitív személyes adatot kezel és tárol, amelyet PaaS felhőbe fogunk mozgatni. Mit használjunk hogy a lehető legnagyobb biztonságban legyenek az adatok és más folyamat nem tud hozzáférni vagy lemásolni?

- ☐ Tároljuk az adatot publikus felhő alapú szolgáltatáson
- ☒ Tároljuk az adatot magán felhő alapú szolgáltatáson
- ☐ Használjunk disk szintű titkosítást
- ☐ Használjunk PCI DSS követelményeknek megfelelő adatbázis motort

### Magyarázat a megoldáshoz

Felhő szolgáltatás esetén az adatok tárolásának (data at rest) legbiztonságosabb módja ha egy dedikált (private cloud) szerveren tároljuk azt. Ez esetben egyetlen konténer vagy middleware hiba sem eredményezheti hogy más hozzáférjen az adatunkhoz. Az adatokat a felhasználás (data in motion) közben titkosítva kell tárolni és használni.

## 2. feladat 0 / 4 pont

Milyen folyamatokat vagy megoldásokat alkalmazzunk, hogy a céget elhagyó munkatársak ne tudjanak a továbbiakban hozzáférni a publikus felhőben tárolt adatokhoz? (Több helyes válasz van.)

- ☒ Kétlépcsős azonosítás (2FA) saját mobiltelefonnal
- ☒ Kétlépcsős azonosítás (2FA) céges mobiltelefonnal
- ☒ Jelszavak és kulcsok (API, secret) megváltoztatása
- ☐ Cloud endpoint URL-ek megváltoztatása

### Magyarázat a megoldáshoz

A felhőszolgáltatás egyik ismérve, hogy nincsenek VPN mögött, az URL-t, felhasználói nevet és jelszót ismerő kollégák a világon bárholnan be tudnak jelentkezni. Ez potenciális veszélyforrás hogy a bejelentkezési adatokat ismerő, a céget elhagyó kollégák továbbra is hozzáférhetnek az adatokhoz akár otthonról is. Ez könnyen kiküszöbölhető 2 lépcsős belépéssel ami céges mobiltelefonra küld kódot pl valamint minden felhasználó / API kulcs megváltoztatásával amikor egy kollega elhagyja a céget vagy akár csak a csapatot is.

## 3. feladat 0 / 2 pont

Ha az alkalmazás architektúra helyes, melyik terhelés elosztás képes végtelen terhelést is kiszolgálni?

- ☒ Horizontális
- ☐ Vertikális

### Magyarázat a megoldáshoz

A horizontális elosztás újabb és újabb szerverek hozzáadásával virtuálisan végtelen terhelést is képes kiszolgálni.

## 4. feladat 0 / 4 pont

Milyen faktorok alapján válasszunk adatközpont lokációt a felhőszolgáltatónkból?

- ☒ Ország szintű adat lokalizációs megkorlátások
- ☒ Végfelhasználók helye (ország / régió)
- ☒ A felhő szolgáltatás ára az adott régióban
- ☒ A végfelhasználók által tapasztalható átlagos válaszidő az adott hosztíng régióból

### Magyarázat a megoldáshoz

A döntésnek sok faktora lehet, ezeket kell a menedzsmenttel karöltve megfeloően súlyozni: Ár, felhasználók helye, elvárt válaszidő, valamint adat lokalizációs megkötések.

## 5. feladat 0 / 3 pont

Melyik alkalmazásokat és azok adatbázisait lássuk el Disaster Recovery szolgáltatással?

- ☒ Azokat, amik közvetlenül gyűjtenek adatot
- ☐ Azokat, amik adatot gyűjtenek, átalakítanak, majd továbbküldenek (ETL)
- ☒ Azokat, amiknek katasztrófa esetén is maximum 1 órán belül újra elérhetőnek kell lennie
- ☐ Azokat, amelyek riportolási funkciókat látnak el.

### Magyarázat a megoldáshoz

A biztonsági másolat készítésénél legfontosabb a forrás adatok védelme, azok az alkalmazásoké amik közvetlenül gyűjtenek adatot. Ezekből az adatfolyam újraépíthető.

Védelmet igényelnek továbbá azok az alkalmazások, ahol az adatfolyam újraépítésére nem áll elegendő idő rendelkezésre.

## 6. feladat 0 / 3 pont

DevOps fázisai: TERVEZÉS (PLAN) - ÖSSZEÉPÍTÉS (BUILD) - TELEPÍTÉS (DEPLOY) - ÜZEMELTETÉS (OPERATE) - MONITOROZÁS (MONITOR)

Mely elemek hiányznak a DevOps fázisai közül?

- ☐ TESZTELÉS (TEST) és KONFIGURÁLÁS (CONFIG)
- ☒ TESZTELÉS (TEST) és KÓDOLÁS (CODE)
- ☐ AUTOMATIZÁLÁS (AUTOMATE) és KONFIGURÁLÁS (CONFIG)
- ☐ AUTOMATIZÁLÁS (AUTOMATE) és BEMUTATÁS (SHOWCASE)

### Magyarázat a megoldáshoz

A fejlesztést tervezéssel kezdjük, majd a kódolás és a kód összeépítés következik. Ezután a tesztelési fázis, majd ennek sikeressége után az alkalmazásunk telepíthető. A sort az üzemeltetés és a működés folyamatos monitorozása zárja. Monitorozás során változási igények jelenhetnek meg, minek hatására kódjavítást, fejlesztést kezdeményezhetünk és a folyamat kezdődik újra. A legsikeresebb DevOps megvalósítás ezt a kört gyorsan végig futtatja és a fázisok egymást automatikusan meghívják.

## 7. feladat 0 / 2 pont

Igaz vagy hamis? A DevSecOps és a Felhő elválaszthatatlanul csak együtt használható.

- ☐ Igaz
- ☒ Hamis

### Magyarázat a megoldáshoz

A felhő környezet skálázhatóságot és igény szerinti használatot biztosít ami hasznos a DevSecOps rendszernek, azonban a DevSecOps felhő nélkül is lehetséges.

## 8. feladat 0 / 2 pont

Melyik az a migrációs stratégia, amely a meglévő alkalmazások felhőbe való mozgatására, módosítására és átírására szolgál?

- ☐ Replatforming
- ☒ Refactoring
- ☐ Rehosting
- ☐ Repurchasing

### Magyarázat a megoldáshoz

A refactoring teljes újrairást jelent, hogy az alkalmazás az új felhő alapú infrastruktúrának megfeleljen. Sokszor ez magába foglalja az alkalmazás és a kód nagy részének módosítását, hogy jobban kihasználja a felhő biztosította előnyöket és rugalmasságot.

## 9. feladat 0 / 3 pont

Melyek lehetnek a sikeres DevSecOps bevezetésének előnyei?

- ☒ A szoftverfejlesztési folyamatok javulása
- ☒ Magasabb automatizációs szint
- ☒ Hatékonyabb, gyorsabb kommunikáció a fejlesztő és támogató csapatok között

### Magyarázat a megoldáshoz

Mind a három a DevSecOps bevezetéséből származható előny.

## 10. feladat 0 / 4 pont

Melyek lehetnek az elemei a folyamatos szállítás (Continuous Delivery) biztonságossá tételének?

- ☒ Levédni (pl. digitálisan aláírni) a binary-kat és más build artifact-okat manipulálás, szabotálás, stb. elől
- ☒ Az egymással kapcsolatban lévő rendszerkomponensek adatfolyamának integritását biztosítani
- ☒ Levédni a jelszavakat, kulcsokat és más titkokat
- ☒ Erős hozzáférés felügyelet az egész folyamatban és hozzáférés auditok

### Magyarázat a megoldáshoz

Mindre szükség van ahhoz, hogy biztonságos legyen a DevOps folyamat.

## 11. feladat 0 / 2 pont

Azonosítsd azt az eszközt, amelynek aza lényege, hogy kis csapatok folyamatosan változó, jó minőségű kódot tudjanak szállítani?

- ☐ Vizeses metodológia (Waterfall Methodology)
- ☐ Szolgáltatás orientált architektúra (Service-Oriented Architecture)
- ☐ Struktúrált fejlesztési metodológia (Structured Development Methodology)
- ☒ Agilis metodológia (Agile Methodology)

### Magyarázat a megoldáshoz

Az agilis szoftverfejlesztés és olyan metodológia, ahol a szoftver követelmények és a megoldások együttműködésen keresztül együtt fejlődnek az önszerveződő és multifunkcionális csapatok között. Ez elősegíti az alkalmazkodó tervezést, az evolúciós fejlesztést, korai szállítást, folytonos továbbfejlesztést és bátorít a változásokra adható gyors és rugalmas válaszokra.



Cloud Engineering (IBM)  
4. forduló

## Ismertető a feladathoz

### KERETTÖRTÉNET:

Az Imagination Wizard Kft. egy közepes méretű egészségmegőrzési kellékeket gyártó és forgalmazó cég amely szignifikáns üzleti potenciállal rendelkezik a Kelet Közép Európai régióban. Értékesítése az elmúlt 3 évben meghaladta az évi 600 millió eurót. Dolgozói 5 országban, 3 gyár és 4 háttértámogató szervezet egységében tevékenykedik az üzletkötők kiterjedt nemzetközi hálózata mellett.

A cég vezetése új termékek piacra bocsájtását tervezi, amelytől az árbevétel és a piaci részesedés megduplázódását várja."

Elemzések egybehangzó iránymutatása alapján a siker egyik kulcsa a megfelelő informatikai támogatás, rendszerfejlesztés megvalósítása, ami rugalmas kiszolgálást, skálázható szolgáltatásokat, megbízható rendszereket, önjáró szoftverfejlesztési folyamatirányítást nyújt. Ezen célkitűzések súlyozott prioritást kapnak korunk vírushelyzetre felkészült, rugalmas működtetést igénylő időszakában.

A te feladatod az lesz, hogy feltérképezd a jelenlegi IT környezetet, felhőalapú infrastruktúrát és kiszolgáló rendszert tervezz és implementálj, valamint a DevOps vívmányait megfontolva modern és minden igényt kielégítő szoftverfejlesztési folyamatot alakíts ki a cég számára.

Ebben a fordulóban a koncepció kialakítására fókuszálunk. Most a migráció tesztelésének megtervezése, a jövőbeni alkalmazás tesztelési standardok kialakítása valamint a felhő alapú szolgáltatás konfigurációjának lefektetése a feladat.

**Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 12 perc került rögzítésre mint megoldáshoz felhasznált idő.**

## 1. feladat 0 / 3 pont

Meglévő alkalmazások felhőbe migrálásánál milyen alkalmazás karakterisztikákat kell tesztelni?

- ☒ Komponens teszt
- ☒ Integrációs interface teszt
- ☐ Rendelkezésre állás teszt
- ☒ End-to-end system teszt
- ☒ Teljesítmény terhelés teszt

### Magyarázat a megoldáshoz

Bármilyen migráció esetén mind a funkcionális (amit az alkalmazás tesz) és a nem funkcionális (válaszidő, teljesítmény, rendelkezésre állás) tesztelni kell. Felhő alapú migrációnál a rendelkezésre állás a kivétel mert ezt a szolgáltató garantálja, arra a szolgáltatás használójának befolyása nincs.

## 2. feladat 0 / 2 pont

A jövőben ahhoz, hogy a legjobb minőségű alkalmazás kódot fejlesszük le, mikor érdemes megtervezni a tesztelést?

- ☐ Az igényfelmérési fázisban mielőtt összegyűjtjük a felhasználói igényeket
- ☒ A kód tervezés fázisban a kódolás megkezdése előtt
- ☐ A teszt fázis elején a kódolás befejezése után

### Magyarázat a megoldáshoz

A teszt vezérelt fejlesztés (Test Driven Development) módszer alkalmazása nagy mértékben növeli az alkalmazás funkcionális minőségét.

## 3. feladat 0 / 2 pont

Melyik felhő alapú szolgáltatás jön garantált, beépített terhelés elosztóval (Workload balancer)?

- ☐ Iaas (Infrastruncture as a Service)
- ☐ PaaS (Platform as a Service)
- ☐ SaaS (Software as a Service)
- ☒ FaaS (Function as a Service)

### Magyarázat a megoldáshoz

FaaS esetében az szolgáltató gondoskodik a kód lefordításáról, annak konténerizációjáról és elérhetővé tételéről. Ez magában foglalja az terhelés elosztását is.

## 4. feladat 0 / 4 pont

Válaszd ki a DevOps automatikus tesztelésben domináns szoftver megoldásait az alábbi listából!

- ☒ Selenium
- ☒ Qualibrate
- ☒ Testcomplete
- ☐ Testimony
- ☐ TestAutoQ
- ☐ Ansible
- ☐ Gradle

### Magyarázat a megoldáshoz

Az automatikus tesztelés kulcsfontosságú a kód minőségének, integrációjának megfelelőségéért. Ezáltal azonnali visszajelzés nyerhető a hibák elkerüléséért, a felhasználói elégedettség megőrzéséért.

## 5. feladat 0 / 6 pont

Az alábbiak közül milyen esetekre lehet automatikus tesztet írni?

- ☐ Környezet verziókövetésének paraméterezése
- ☐ A kód modularitásának megfelelősege
- ☐ A kód nem fordul le
- ☒ Email értesítés
- ☒ API elérhetőség
- ☒ Adatbázis elérhetőség
- ☐ Nem megfelelő felhasználói funkcionálisok
- ☒ Operációs rendszer kompatibilitás

### Magyarázat a megoldáshoz

A rohamos léptekben történő alkalmazásfejlesztés során szinte biztos, hogy nem tudjuk a tesztjeinket olyan mértékben és ütemben végrehajtani, mint ahogy azt szeretnénk.

Automatizált tesztekkel különböző célokat próbálunk meg teljesíteni, amelyek egy adott cégnél, vagy vállalatnál nagyon eltérőek, de a kérdés válaszai a legtipikusabb előfordulásai az automatikus teszteknek.

## 6. feladat 0 / 3 pont

Mi az A/B tesztelés lényege?

- ☐ A tesztelés folyamán két hipotézist vizsgálunk, amelyiket hatékonyabbnak mérjük azzal számolunk. A méréshez számost eszköz elérhető.
- ☒ Két verzió elérhető élesen, amelyek nagyon hasonlóak egy (a tesztelendő) elem kivételével. Az egyik verziót a felhasználók egy csoportjának, míg a másikat egy attól különböző csoportnak biztosítjuk. Amelyik verzió sikeresebb az kerül bevezetésre mindenkinél.
- ☐ Az Alfa és Béta teszt környezetben is tesztelünk egyszerre különböző paraméterekkel a teljesítmény hangolásáért. A tesztelés végén az egyiket megbízhatóbbnak ítéljük SWOT analízis alapján, így ezzel a megvalósítással számolunk.

### Magyarázat a megoldáshoz

Az A/B tesztelés (vagy másnéven split testing, ritkábban bucket testing) egy konverziónövelő módszer, mely során valaminek két verzióját hasonlítjuk össze azért, hogy egyértelműen megkapjuk melyik teljesít jobban. Ezzel a módszerrel nem csak weblapokat, de landing page-eket, Facebook vagy AdWords hirdetéseket, appokat és e-mail üzeneteket is tesztelhetesz. Sőt, lényegében bármit letesztelhetsz így, amiből legalább két verzió készíthető.

## 7. feladat 0 / 4 pont

Melyik az a 4 az alábbiak közül, mely a szoftver tesztelés alapelvei közé tartozik?

- ☒ A tesztelés olyan korán kell kezdődjön, amilyen korán a fejlesztés fázisában lehet.
- ☐ Tesztelni mindenki tud, nem kell hozzá különösebb tudás.
- ☒ A tesztelés a megtalált defekteket mutatja, sosem az alkalmazás összes hibáját.
- ☒ Tökéletes tesztelés nem létezik.
- ☒ Tesztelni fontosabb, mint jó program kódot írni.
- ☐ A tesztelést az élesítéssel nem lehet abbahagyni.
- ☐ A program funkcióinak minden szereplőjét be kell vonni a tesztelésbe.

### Magyarázat a megoldáshoz

Tesztelésre azért van szükség, hogy a szoftver termékben meglévő hibákat még az üzembe helyezés előtt megtaláljuk, ezzel növeljük a termék minőségét, megbízhatóságát.

Érdemes a tesztelést az életciklus minél korábbi szakaszában elkezdeni, mert minél hamarabb találunk meg egy hibát (mondjuk a specifikációban), annál olcsóbb javítani.A tesztelés képes felfedni a hibákat, de az alkalmazás összes hibáját nem mutatja meg.

Minden bemeneti kombinációt nem lehet letesztelni és nem is érdemes. Általában a magas kockázatú és magas prioritású részeket teszteljük.

## 8. feladat 0 / 3 pont

Melyik nem a jó teszteléshez tartozó DevOps elv?

- ☐ A tesztautomatizálás gyors és korai visszajelzést ad a munkád minőségéről.
- ☒ Soha ne keverd a tesztvezérelt fejlesztést a teszt automatizációs módszerrel!
- ☐ A teszt futtatások folyamatos, determinisztikus és ismétlődő eredményeket kell hogy hozzanak amennyiben a körülmények ugyanazok.
- ☐ A tesztautomatizációval el lehet kerülni a lassú és alkalmi visszajelzést. Amire szükség van az a gyors visszajelzés minden esetben mikor új kód készül.

### Magyarázat a megoldáshoz

Mind a tesztvezérelt fejlesztés és a teszt automatizáció hasznos eszköz és semmi nem zárja ki, hogy mindkettőt alkalmazzuk egyszerre.



## Ismertető a feladathoz

### KERETTÖRTÉNET:

Az Imagination Wizard Kft. egy közepes méretű egészségmegőrzési kellékeket gyártó és forgalmazó cég amely szignifikáns üzleti potenciállal rendelkezik a Kelet Közép Európai régióban. Értékesítése az elmúlt 3 évben meghaladta az évi 600 millió eurót. Dolgozói 5 országban, 3 gyár és 4 háttértámogató szervezet egységében tevékenykedik az üzletkötők kiterjedt nemzetközi hálózata mellett.

A cég vezetése új termékek piacra bocsájtását tervezi, amelytől az árbevétel és a piaci részesedés megduplázódását várja."

Elemzések egybehangzó iránymutatása alapján a siker egyik kulcsa a megfelelő informatikai támogatás, rendszerfejlesztés megvalósítása, ami rugalmas kiszolgálást, skálázható szolgáltatásokat, megbízható rendszereket, önjáró szoftverfejlesztési folyamatirányítást nyújt. Ezen célkitűzések súlyozott prioritást kapnak korunk vírushelyzetre felkészült, rugalmas működtetést igénylő időszakában.

A te feladatod az lesz, hogy feltérképezd a jelenlegi IT környezetet, felhőalapú infrastruktúrát és kiszolgáló rendszert tervezz és implementálj, valamint a DevOps vívmányait megfontolva modern és minden igényt kielégítő szoftverfejlesztési folyamatot alakíts ki a cég számára.

Ez idáig megtörtént a tesztelés koncepciójának felépítése valamint a felhő környezet konfigurálása. Most a beüzemelés és a felhő környezet elindítása a feladat.

**Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 12 perc került rögzítésre mint megoldáshoz felhasznált idő.**

## 1. feladat 0 / 3 pont

Amíg a transzformáció fut és hibrid felhő szolgáltatás struktúrát használunk, a felhőben futó alkalmazások hogyan cserélhetnek adatot biztonságosan a saját szerverteremben futó alkalmazásokkal?

- ☐ REST API hívásokkal.
- ☒ A 2 adatközpont között létrehozott titkosított, direkt hálózati kapcsolaton keresztül.
- ☐ XML file-okon keresztül.
- ☐ Automatizált, éjszaka futó file átadással.

### Magyarázat a megoldáshoz

Hibrid felhő állapot esetén a biztonságos kommunikációt egy titkosított direct connect kapcsolaton keresztül lehet legjobban biztosítani.

## 2. feladat 0 / 3 pont

Az elemzés után arra a döntésre jutottunk, hogy a JAVA alapú web szervereket fogjuk a felhőbe mozgatni először. Mi a legjobb megoldás a gyors mozgásra?

- ☐ Írjuk újra az alkalmazást, hogy használja a PaaS szolgáltatásokat.
- ☐ Vezessük ki az alkalmazásokat, és használjunk helyette SaaS megoldást.
- ☒ Tegyük a kódot egy konténerbe (Kubernetes, stb.), és futtassuk úgy a változatlan kódot.
- ☐ Mozgassuk át az adatbázisokat a felhőbe és futtassuk a JAVA kódot a régi, saját szervereken.

### Magyarázat a megoldáshoz

A meglévő alkalmazások konténerizációja és PaaS platformon való futtatása a leggyorsabb módja a felhő migrációnak.

## 3. feladat 0 / 3 pont

Mit csinál a felhő szolgáltató, ha a publikus felhőben hosztolt konténerünk nem futtat semmilyen folyamatot?

- ☒ Lementi az állapotot és log-okat lemezre, és felszabadítja a memóriát, hogy más futó folyamatokra allokálja az erőforrást.
- ☐ Futtatja a konténeret változatlanul amíg újabb feladat nem érkezik.
- ☐ Rendszerűenetet küld, hogy optimalizáljuk a folyamatot.
- ☐ Felszabadítja a memóriát hogy más futó folyamatokra allokálja az erőforrást.

### Magyarázat a megoldáshoz

Publikus felhő szolgáltatásnál az alkalmazásunk minden esetben megosztott erőforrásokat használ, olyan szerverek amiket a felhő szolgáltató más vevői is használnak. Hogy az erőforrás felhasználást a felhő szolgáltató optimalizálja, a nem futó alkalmazásokat eltávolítja a memóriából hogy azt más futó alkalmazások tudják használni.

## 4. feladat 0 / 5 pont

Nagyméretű, éles adatokat tartalmazó strukturált adatbázisokat kell a felhőbe mozgatnunk amely minimum 2 napot fog igénybe venni a hálózati átvitel limitációja miatt. Ez idő alatt az eredeti alkalmazásnak éles üzemben működnie kell tovább. Hogyan oldható meg az adat konzisztencia biztosítása?

- ☐ Egylépcsős migráció használata, az éles rendszer lekapcsolása a teljes migráció alatt.
- ☒ Kétlépcsős migráció használata, ahol egy adott időbélyegig vagy elsődleges kulcsig átmásolunk mindent. Második lépésben leállítjuk egy rövid időre az alkalmazást, átmásoljuk az azóta keletkezett (delta) adatokat, majd újraindítjuk az alkalmazást már az új adatbázis használatával.
- ☐ Háromlépcsős, ETL (Extract, Transform, Load) migráció használata.

### Magyarázat a megoldáshoz

Az ipari standard megoldás ebben az esetben egy kétlépcsős migráció alkalmazása. Első lépésben egy adott időbélyegig vagy elsődleges kulcsig átmásolunk mindent. Második lépésben leállítjuk egy rövid időre az alkalmazást, átmásoljuk az azóta keletkezett (delta) adatokat, majd újraindítjuk az alkalmazást már az új adatbázis használatával.

## 5. feladat 0 / 5 pont

A GitHub verziókövetés gyakorlatilag ipari szabvánnyá nőtte ki magát. Amikor első alkalommal publikálunk kódot a távoli branch-be (commit to remote branch), melyik opcionális paraméterrel tudjuk a követést létrehozni a helyi és a távoli branch között?

- ☐ -t
- ☒ -u
- ☐ -us
- ☐ -track

### Magyarázat a megoldáshoz

A "-u" paraméter hozza létre az upstream kapcsolatot, ami az első publikálásnál összekapcsolja a helyi és a távoli branch-et

## 6. feladat 0 / 7 pont

Az alábbiak közül mely objektumokra, termékekre terjedhet ki az automatikus bevezetés ("automatic deployment")?

- ☒ Tranzakciós adatok
- ☐ Szolgáltatás egyezmények
- ☒ Alapadatok
- ☒ Adatbázis objektumok
- ☒ Konfigurációk
- ☒ Tesztesetek
- ☒ Programkód
- ☐ Kulcs metrikák
- ☐ Infrastruktúra redundancia
- ☐ Adatmentési médiák

### Magyarázat a megoldáshoz

Az automatikus bevezetés lehetővé teszi, hogy az alkalmazások a különböző környezetekben legyenek bevezetve, ugyanúgy, mint a végső prod környezetben. Így a funkciók megbízhatóbbá válnak, és maga a fejlesztési folyamat is felgyorsul.

## 7. feladat 0 / 7 pont

Mely elemekre kell csinálnunk verziókövetést az alábbiak közül?

- ☒ Programozási kód
- ☒ Tranzakciós adatok
- ☒ Alapadatok
- ☐ Külső adatbázisok
- ☒ Adatbázis objektumok
- ☒ Konfigurációs fájlok
- ☐ Kód összeépítés (compile) eredménye
- ☐ Az alkalmazás elérésének portjai
- ☒ Tesztesetek
- ☒ Infrastruktúra komponensek leíró fájljai

### Magyarázat a megoldáshoz

A DevOps sikerességének egyik fő tényezője a verziókövetés. A verziókövetés rendszer lehetővé teszi a fejlesztők számára, hogy a kódban működjenek együtt, és nyomon kövessék a módosításokat. Az ágak használata lehetővé teszi a fejlesztők számára a kód különböző verzióinak közötti váltást, és a csapat többi tagjától függetlenül működnek.

## 8. feladat 0 / 3 pont

Mi a kék/zöld telepítés (blue/green deployment)?

- ☐ Módszer az alkalmazás két verziójának összehasonlítására, hogy kiderüljön, hogy melyik teljesít jobban.
- ☐ Verziók telepítési mintája alrendszerekhez.
- ☒ Telepítési modell, mely egyszerre két megegyező éles rendszert futtat, de csak az egyiken van aktív forgalom.
- ☐ Telepítési metódus, mely erőforrást használ fel, telepíti az új változásokat, majd visszaállítja az erőforrásokat.

### Magyarázat a megoldáshoz

A kék-zöld telepítés egy technika amely lecsökkenti az állásidőt (downtime) és a kockázatot azzal, hogy egyszerre két megegyező éles rendszert futtat (kékét és zöldet).

Aktívan mindig csak egyikük szolgálja ki a forgalmat.



## Ismertető a feladathoz

### KERETTÖRTÉNET:

Az Imagination Wizard Kft. egy közepes méretű egészségmegőrzési kellékeket gyártó és informatikai támogatás, rendszerfejlesztés megvalósítása, ami rugalmas kiszolgálást, skálázható szolgáltatásokat, megbízható rendszereket, önjáró szoftverfejlesztési folyamatirányítást nyújt. Ezen célkitűzések súlyozott prioritást kapnak korunk vírushelyzetre felkészült, rugalmas működtetést igénylő időszakában.

A cég vezetése új termékek piacra bocsájtását tervezi, amelytől az árbevétel és a piaci részesedés megduplázódását várja."

Elemzések egybehangzó iránymutatása alapján a siker egyik kulcsa a megfelelő informatikai támogatás, rendszerfejlesztés megvalósítása, ami rugalmas kiszolgálást, skálázható szolgáltatásokat, megbízható rendszereket, önjáró szoftverfejlesztési folyamatirányítást nyújt. Ezen célkitűzések súlyozott prioritást kapnak korunk vírushelyzetre felkészült, rugalmas működtetést igénylő időszakában.

A te feladatod az lesz, hogy feltérképezd a jelenlegi IT környezetet, felhőalapú infrastruktúrárt és kiszolgáló rendszert tervezz és implementálj, valamint a DevOps vívmányait megfontolva modern és minden igényt kielégítő szoftverfejlesztési folyamatot alakíts ki a cég számára.

Az előző lépésekben beüzemeltük az első alkalmazásainkat a felhő környezetben és elkezdődött a teljes portfólio migrálása. Most a támogatás fejlesztésére és a finomhangolásra helyezük a hangsúlyt.

**Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 15 perc került rögzítésre mint megoldáshoz felhasználni idő.**

## 1. feladat 0 / 4 pont

Melyik felhő szolgáltatás típus fut és érhető el akkor is, ha nincs rajta futó alkalmazás (idle)?

- ☒ Iaas (Infrasturcture as a Service)
- ☐ PaaS (Platform as a Service)
- ☐ SaaS (Software as a Service)
- ☐ FaaS (Function as a Service)

### Magyarázat a megoldáshoz

Az IaaS platform esetében a virtuális gép mindig fut és elérhető. A memóriában és/vagy lemezen tárolt adatok mindig lementésre és megtartásra kerülnek.

## 2. feladat 0 / 12 pont

Melyek azok a metrikák, amik a DevOps környezetre vonatkoznak?

- ☒ Alap metrika, például a kódolás soraira, a kommentelés számosságára.
- ☒ Függőségi mérés arra vonatkozólag, hogy az objektumok és osztályok a kódban miként vannak összekapcsolva.
- ☐ Rizikó mérés, ami megmondja, mekkora az esélye annak, hogy a jelenlegi programkód a kívánt eredményt szolgálja.
- ☐ Kohéziós metrika, azért, hogy a kód az infrastruktúra komponensekkel mennyire működik összhangban.
- ☐ Nyitottsági metrika, amely a kód általánosításának jelenlegi helyzetét, fokát adja meg.
- ☐ Kompatibilitási metrika, mely kimutatja, hogy az előző verziójú kód mennyire eltérő módon lett írva.
- ☒ Öröklődési mérés, mely megmutatja, hogy milyen mélységig történik meg az osztályok közötti öröklődés.

### Magyarázat a megoldáshoz

A DevOps legfőbb célkitűzése a sebesség, a minőség és az alkalmazás teljesítményének fokozása. A DevOps metrikái abban segítenek, hogy ezeket a célkitűzéseket mérjük és elérhessük.

## 3. feladat 0 / 6 pont

Miért érdemes az adatainkról független másolatot készíteni még akkor is, ha az adatbázisaink valós időben másolva vannak egy disaster recovery szerverre?

- ☒ Hogy az emberi katasztrófákat (véletlen adattörlés, tábla eldobás file törlés) meg tudjuk oldani.
- ☐ Hogy az emberi katasztrófákat (véletlen adattörlés, tábla eldobás file törlés) meg tudjuk akadályozni.
- ☐ Hogy megfeleljünk a belső szabályzatnak.

### Magyarázat a megoldáshoz

Harmadik, nem valós idejű biztonsági másolat készítése lehetővé teszi egy adott állapot visszaállítását. Ez megoldhat emberek által előidézett katasztrófákat pl tábla vagy adatbázis eldobás, véletlen adat törlés, stb.

## 4. feladat 0 / 4 pont

Az alkalmazások hogyan kezeljék a futás idejű logokat?

- ☐ Ne készítsen logot.
- ☒ Tölts be be egy központosított log kezelő alkalmazásba.
- ☒ Írja ki helyi lemeze.
- ☐ Tartsa memóriában az utolsó 24 óra logját.

### Magyarázat a megoldáshoz

Az optimális megoldás egy központosított log kezelő alkalmazása. Ekkor egy közös helyen érhető el az összes alkalmazás logjait és egy jobb rálátásunk lesz a teljes informatikai rendszer állapotára.

## 5. feladat 0 / 5 pont

Milyen alkalmazásokat futtassunk több adatközpontban párhuzamosan, hogy biztosítsuk a 99.99%+ rendelkezésre állást?

- ☒ 24/7-ben használt alkalmazások
- ☐ Háttérfolyamatokért (back-office) felelős alkalmazások
- ☒ Üzleti kritikus alkalmazások
- ☐ Ade telemzésért felelős alkalmazások

### Magyarázat a megoldáshoz

A folyamatos elérés (high availability) általában az üzletileg kritikus és a folyamatosan, 24/7-ben használt alkalmazásoknál merül fel igényként. Ekkor High Availability megoldásokkal védekezhetünk egy adatközpont kiesésre (hálózati hiba, természeti katasztrófa, stb)

## 6. feladat 0 / 5 pont

Új végfelhasználói igényként az egyik noSQL adatbázist használó alkalmazásra riportozási funkciót kell készítenünk az üzleti felhasználóknak. Milyen megoldás(ok) a preferáltak?

- ☐ Adjunk közvetlen hozzáférést az adatbázishoz a felhasználóknak, akik megírhatják a saját lekérdezéseiket.
- ☐ Hozzunk létre egy csapatot, akik a riportokat elkészítik és elküldik a végfelhasználóknak.
- ☒ Adjunk egy riportolási / BI felületet, amin keresztül az adatbázis elérhető.
- ☒ Előre definiáljunk lekérdezéseket, amiket a felhasználók futtathatnak.

### Magyarázat a megoldáshoz

A noSQL adatbázis riportozása nehézkes mivel a lekérdezési felülete és típusa változó, nem egységesített mint a hagyományos adatbázisok esetében (SQL). A jó megoldás ebben az esetben egy kezelőfelület biztosítása amin keresztül előre megírt és optimalizált lekérdezések futtathatnak. Így biztosítható hogy az adatbázist nem terheli le a riportozás olyannyira hogy az az általános válaszidő rovására menjen.

## 7. feladat 0 / 3 pont

A docker alap konténerek használata jelenleg a PaaS (Platform as a Service) szolgáltatás gerincét adja. Melyik paranccsal lehet egy konténert eltávolítani?

- ☒ docker container rm
- ☐ docker remove
- ☐ docker delete container
- ☐ docker rmi

### Magyarázat a megoldáshoz

Mind a „docker container rm” vagy a „docker rm” paranccsok elváltolítják a paraméterben megkapott konténer(e)ket).

## 8. feladat 0 / 5 pont

Milyen megoldásokat választhatunk a futásidejű hibák kezelésére?

- ☒ Az alkalmazás megpróbálja önmagától megoldani a hibát egy előre definiált készlet alapján.
- ☐ Költség / Fontosság összehasonlítása.
- ☒ Üzenetet küld egy meghatározott chat vagy email csoportnak a hibáról.
- ☒ Automatikusan létrehoz egy hibajegyet az erre létrehozott kezelő rendszerben.

### Magyarázat a megoldáshoz

A futásidejű hibák kezelésének számos módja ismert. Alkalmazásba épített hibakezelés (adathibák tipikus javítása), L1 support értesítése és hibajegy készítése mind jó módszer.

## 9. feladat 0 / 9 pont

Mely esetekben végezz egy programozó refaktorálást?

- ☐ Amikor van ideje.
- ☐ Amikor új funkciót kell a programhoz adni.
- ☒ Amikor kódot archivál.
- ☒ Amikor hibát javít.
- ☒ Amikor programteljesítményt optimalizál.
- ☒ Amikor duplikációt keres.
- ☒ Amikor programkódokat integrál.
- ☒ Amikor a kódot átvizsgálja.
- ☐ Amikor kommenteli a kódját.

### Magyarázat a megoldáshoz

A refaktorálás egyszerűen fogalmazva a kód optimalizálását vagy valamilyen szintű manipulációját jelenti. Alacsony szintű refaktorálásról beszélünk, ha a kódban egy metódust szeretnénk átnevezni. Ez a metódusra való hivatkozás sok osztály rengeteg helyén fordulhat elő és a fejlesztői környezetnek az összes hivatkozást érzékelnie és az átnevezés során javítania kell. A magas szintű refaktorálás során a fejlesztői környezetnek bizonyos mértékig már a kód szemantikai, tartalmi elemeit is tudnia kell kielemeznie. Ennek természetesen vannak korlátai, hiszen azért mégsem veheti át a programozó analitikus feladatát, de bizonyos esetekben azt jelentős mértékben megkönnyítheti. Az intelligens fejlesztői környezetnek többek között képesnek kell lennie a kódban többször szereplő, de ugyanazon funkciójú kód felfedezésére, a túl bonyolult (ezek legtöbbször egymásba ágyazott) ciklusok optimalizálására, stb.

## 10. feladat 0 / 12 pont

Melyek tipikus refaktorálást segítő eszközök?

- ☒ ReSharper
- ☐ Sneet
- ☒ Checkstyle
- ☐ Refacto
- ☒ Project Analyzer
- ☒ PMD
- ☒ Findbugs
- ☐ Smellycode
- ☒ VS Code

### Magyarázat a megoldáshoz

A refaktoráló eszközök összegyűjtik a szükséges információkat, amelyek alapján el tudják végezni az összes kompenzációt a programozó helyett, s ezzel időt és munkát spórolnak meg. A modern, divatos imperatív nyelvekhez készült fejlesztői környezetek természetesen tartalmaznak refaktoráló funkciókat, ám refaktorálásra régebbi, vagy esetlegesen kevésbé elterjedt nyelvekben is szükség lehet.



Cloud Engineering (IBM)  
7. forduló

Ismertető a feladathoz

KERETTÖRTÉNET

Az Imagination Wizard Kft. egy közepes méretű egészségmegőrzési kellékeket gyártó és forgalmazó cég amely szignifikáns üzleti potenciállal rendelkezik a Kelet Közép Európai régióban. Értékesítése az elmúlt 3 évben meghaladta az évi 600 millió eurót. Dolgozói 5 országban, 3 gyár és 4 háttértámogató szervezet egységében tevékenykedik az üzletkötők kiterjedt nemzetközi hálózata mellett.

A cég vezetése új termékek piacra bocsájtását tervezi, amelytől az árbevétel és a piaci részesedés megduplázódását várja."

Elemzések egybehangzó iránymutatása alapján a siker egyik kulcsa a megfelelő informatikai támogatás, rendszerfejlesztés megvalósítása, ami rugalmas kiszolgálást, skálázható szolgáltatásokat, megbízható rendszereket, önjáró szoftverfejlesztési folyamatirányítást nyújt. Ezen célkitűzések súlyozott prioritást kapnak korunk vírushelyzetre felkészült, rugalmas működtetést igénylő időszakában.

A te feladatod az lesz, hogy feltérképezd a jelenlegi IT környezetet, felhőalapú infrastruktúrát és kiszolgáló rendszert tervezz és implementálj, valamint a DevOps vívmányait megfontolva modern és minden igényt kielégítő szoftverfejlesztési folyamatot alakíts ki a cég számára.

Az előző lépésben a már felhőbe mozzgatott alkalmazásainkat hibakezelésére fókuszáltunk. Most (végezetül) az optimalizálás, biztonság, fenntartás és automatizálás következik.

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 12 perc került rögzítésre mint megoldáshoz felhasznált idő.

1. feladat 0 / 5 pont

Mi a preferált mód sorrendje két alkalmazás közti adatcsereének?

- ☐ Direkt adatbázis olvasás > API > File
- ☐ File > API > Direkt adatbázis olvasás
- ☐ File > Message Queue > API
- ☒ API > Direkt adatbázis olvasás > File

Magyarázat a megoldáshoz

Az iparági standard jelenleg az API hívások alkalmazása. Ezt követi a direct adazbázis kapcsolat létrehozása majd legvégül a file alapú adatcsere.

2. feladat 0 / 10 pont

Melyek azok az eszközök, szoftverek melyek támogatják a funkcionális tesztelés automatizálását?

- ☐ Redit
- ☐ Rabbit
- ☐ Jinga
- ☒ Cucumber
- ☒ Selenium
- ☒ Junit

Magyarázat a megoldáshoz

A funkcionális tesztek célja a rendszer működésének logikái, tartalmi szempontú feltérképezése, a programok követelmény specifikációja és a tényleges működés közötti esetleges eltérések, illetve működési hibák felderítése. A funkcionális tesztelés folyamán a rendszerek funkcionális megfeleléségét kell ellenőrizni, tehát hogy a rendszer funkciói (menüpontjai) a bemenő adatokra a megfelelőképpen reagálnak-e. Ezen folyamat automatizálásához számtalan eszköz létezik, a legismertebbeket a helyes választban találod.

3. feladat 0 / 5 pont

Amikor a fejlesztők új kódot vezetnek be mindig fenn áll a veszélye, hogy nem engedélyezett hozzáférést tesznek elérhetővé. Melyik eszköz nem csökkenti ezt a rizikót?

- ☐ Kód felülvizsgálat (code review)
- ☐ Kód tesztelés
- ☒ Kód hibajavítás (bugfix)
- ☐ Penetrációs vizsgálat

Magyarázat a megoldáshoz

A patchelés nem segít a fejlesztő kódjának ellenőrzésében csak bugokat javít.

4. feladat 0 / 5 pont

Az alábbiak közül melyik az, amelyik csökkenti az operációs és biztonsági veszélyeket és egyben támogatja a szabályozást?

- ☐ Változáskezelési folyamat (Change management process).
- ☐ Konfigurációs menedzsment folyamat.
- ☒ Verzió és telepítési menedzsment folyamat (Release and deployment management process).
- ☐ Szolgáltatási szint menedzsment folyamat (Service level management process).

Magyarázat a megoldáshoz

Verzió és telepítési menedzsment folyamat (Release and deployment management process) az ami csökkenti az operációs és biztonsági veszélyeket és egyben támogatja a szabályozást is.

5. feladat 0 / 5 pont

Mit jelent: “A biztonság balra tolása?”/”Shifting security to the left?”

- ☐ Keretrendszerek és könyvtárak használata.
- ☒ Biztonsági eszközök minél korábbi bevezetése.
- ☐ A konzervatív biztonsági intézkedések lecserélése liberálisabbakra.
- ☐ Fejlesztés későbbi szakaszában bevezetni a biztonsági gyakorlatokat.

Magyarázat a megoldáshoz

A DevSecOpsban próbáljuk “balra tolni a biztonságot”. Ez azt jelenti, hogy a fejlesztés korai szakaszában már bevezetjük a biztonsági szabályozásokat és eszközöket.

6. feladat 0 / 5 pont

Mi a hátránya, ha a feladatköröket szétválasztását bevezetitek DevOpsban (separation of duties control)?

- ☒ Ez a szabályozás gyakran hátráltatja a fejlesztési munkát. Ennek az oka, hogy a munka lelassulhat és csökkenhet a visszajelzések mennyisége, amit a fejlesztő közvetlen kap a munkájára.
- ☐ A fejlesztő egy kódkönyvtárosnak (code librarian) kell, hogy a változásokat elküldje, aki elsőnek átnézi a kódot majd jóváhagyja.
- ☐ Ez a szabályozás felesleges munkát generál, mivel bevezeti a kód felülvizsgálatokat melyek szükségesek a minőség biztosításához.

Magyarázat a megoldáshoz

Ez a szabályozás lelassíthatja a munkát és csökkenti a fejlesztők feladatkörét ezáltal kevesebb visszajelzést kapva munkájukra. Így a fejlesztők nem tudnak teljes felelősséget vállalni munkájuk minőségéről és a organizáció szintű tanulás is nehezebbé válik.

7. feladat 0 / 6 pont

A DevSecOpsban szükség van monitoringra és analízisre. A következők közül melyik eszköz szolgál erre?

- ☐ Kubernetes
- ☐ Hadoop
- ☒ Splunk
- ☐ Docker

Magyarázat a megoldáshoz

A Splunk eszközök használhatóak a logok összegyűjtésére és analizálására.