

Ajánlások a modern, felhőalapú alkalmazások tervezéséhez

Mi az a Cloud Native alkalmazás?

A Cloud Native Computing Foundation hivatalos definíciója szerint:

„A felhőalapú technológiák lehetővé teszik a vállalatok számára, hogy skálázható alkalmazásokat építsenek és futtassanak modern, dinamikus környezetekben, mint például a nyilvános, a privát és a hibrid felhőkben. A megközelítés alkalmazásának tipikus példái a konténerek, a Service Mesh, a microservice-ek, a megváltoztathatatlan infrastruktúra és a deklaratív API-k.

Ezek a technikák lazán csatolt rendszerek építését teszik lehetővé, amelyek rugalmasak, könnyen kezelhetők és megfigyelhetők. Robusztus automatizálással kombinálva lehetővé teszik a mérnökök számára, hogy minimális erőforrás ráfordításával gyakran és kiszámítható módon jelentős változtatásokat hajtsanak végre a rendszeren.”

A fenti definíció alapján megállapítható, hogy a felhőalapú alkalmazások megvalósításának érdekében már a tervezési szakaszban érdemes figyelembe venni **a konténerizációt és a skálázhatóságot**, továbbá az architektúra tervezését is jelentősen befolyásolja a Cloud Native szemlélet, ugyanis a legkönnyebben **microservice**-ek segítségével lehet felépíteni egy ilyen alkalmazást. Egy Cloud Native alkalmazásnál a legnagyobb agilitás és költséghatékonyság érdekében **érdemes kihasználni a felhőszolgáltatók által nyújtott szolgáltatásokat**, és érdemes úgy tervezni őket, hogy ezekkel a szolgáltatásokkal hatékonyan működjenek együtt.

A Cloud Native alkalmazások – szemben a korábbi szemlélettel – masszív CI/CD pipeline-ok segítségével **automatizált módon kerülnek kitelepítésre**. A „Platform as a Service” infrastruktúra modell által nyújtott előnyök kihasználásával **a meghibásodott komponenseket képesek eldobni és teljesen új komponens példányra cserélni**, köszönhetően az állapotmentes konténereknek, amelyek **háttérszolgáltatások segítségével** tárolják az adatokon szükséges állapotváltoztatásokat.

A Cloud Native architektúra alappillérei

A fenti definíció alapján levont következtetésekből már látható, hogy egy Cloud Native architektúrával tervezett alkalmazás több alappilléren nyugszik, amelyek szükségesek az architektúra előnyeinek kiaknázásához. Ezek az alappillérek a következők:

- **Felhő infrastruktúra:** A felhőalapú architektúra teljes mértékben képes kihasználni a felhőszolgáltatók infrastruktúráját és az általuk menedzselt szolgáltatásokat annak érdekében, hogy minél hatékonyabban működjenek egy dinamikus, virtualizált környezetben. Az automatizáció segítségével **eldobható komponenseket használ, amelyek példányai felmerülő igény esetén percekben belül kitelepíthetők, skálázhatók vagy eldobhatók.**
- **Microservices:** A microservice architektúra a modern, felhőalapú alkalmazások közkedvelt architektúra modellje, amelyben **a kicsi és egymástól független szolgáltatások alkotják a rendszer egészét.** A microservice-ek jellemzően
 - egy üzleti folyamatot szolgálnak ki a teljes rendszeren belül;
 - fejlesztésük és kitelepítésük jól elkülönül a többi szolgáltatástól;
 - önhordozók, azaz tartalmazzák a saját adattároló technológiájukat, függőségeiket és a programozási platformjukat;
 - a saját folyamataikat futtatják, a többi szolgáltatással történő kommunikációt pedig általános protokollok – például http/https – segítségével végzik; és

- közösen alkotnak egy alkalmazást.
- **Modern tervezés:** A „The 12 Factor Application” nevű megközelítés széles körben elfogadott, modern tervezési módszer, amelynek szabályait a fejlesztők követik annak érdekében, hogy modern és felhő környezetre optimalizált alkalmazásokat készítsenek. Ezen szabályok betartásával **a rendszerkomponensek könnyen skálázható és hatékonyan kitelepíthető egészet alkotnak**, amelyekkel könnyen követhető a piaci igények változása. Fontos továbbá szem előtt tartani a különböző platformok közötti hordozhatóságot és az automatizációt is.
- **Konténerek:** a konténerek teremtik meg a következetes hordozhatóságot a platformok között. Egy konténer-képfájl egyesíti a kódot, a függőséget és a futtató környezetet egy csomagban, amelyek terjesztése vállalaton belül – és akár kívül is – konténer tároló adattárak segítségével jelentősen leegyszerűsödik. **A microservice-ek jellemzően könnyen konténerizálható alkalmazások.**
- **Háttér szolgáltatások:** Egy felhőalapú rendszer hatékony használata függ sok különböző kiegészítő szolgáltatástól is, amelyek lehetnek adattárolásra, biztonságra, monitorozásra vagy éppen üzenetküldésre specializált szolgáltatások. A megfelelő teljesítmény elérésének érdekében ezekből a szolgáltatásokból érdemes a felhőszolgáltató által kezelt szolgáltatást használni, így **a megfelelő teljesítmény és rendelkezésre állás is a felhőszolgáltató felelőssége.** A háttér szolgáltatások kezelését érdemes csatolt erőforrásként kezelni, ami egy **URL cím és esetleges bejelentkezési adatok konfigurációs állományokba történő megadása után könnyen elérhető vagy akár le is cserélhető.**
- **Automatizáció:** A platform kitelepítése az „Infrastructure as Code” eszközökkel automatizálható, ami könnyen **újratelepíthető, konzisztens** környezetet eredményez. Ezeknek a leíró szkripteknek köszönhetően **a kitelepítés folyamata újra és újra megismételhető teljesen automatizált módon**, mindenféle mellékhatás nélkül. A leírófájlokat is ajánlott verziózáva kezelni. A platformon felül az alkalmazás komponenseinek buildje, tesztelése és kitelepítése is teljesen automatizálható a különböző DevOps eszközök használatával. Ideális esetben az alkalmazáskomponensek forráskódjának feltöltése automatizáltan elindítja a build folyamatot, ami előállítja a bináris, futtatható állományokat (CI pipeline). Release folyamán a bináris állományok megkapják a környezetspecifikus konfigurációkat, amelyekkel együtt **egy módosíthatatlan egységet alkotnak, és így kerülnek kihelyezésre a kívánt környezetbe** (CD pipeline). Minden módosítás egy új release verziót jelent.