

JEL Classification Codes: M15, O33

DOCKER VƏ KONTEYNERLƏŞDİRİLMƏ TEXNOLOGİYASININ MÜASİR PROQRAM İNKİŞAFINDA ROLU

Məhəmməd Əliyev

Azərbaycan Dövlət İqtisad Universiteti (UNEC)

Beynəlxalq Magistratura və Doktorantura Mərkəzi (BMDM)

Xülasə

Bu məqalədə proqramların inkişafı prosesində konteyner texnologiyalarının, xüsusilə də Docker platformasının əsas prinsipləri və tətbiq sahələri araşdırılır. Konteyner texnologiyaları, ənənəvi virtual maşinlardan fərqli olaraq, tətbiqlərin daha asan daşınmasını və istənilən mühitdə fasiləsiz işləməsini təmin edir. Docker texnologiyası proqram təminatının hazırlanması və yayılması mərhələsində mühüm və inqilabi yeniliklərə səbəb olmuşdur. Bu texnologiya istifadəçilərə sadə struktur və idarəetmə üstünlükləri təqdim edərək, iş proseslərinin daha sürətli və səmərəli aparılmasına şərait yaradır. Məqalədə konteyner və Docker texnologiyalarının istifadəsinin gətirdiyi üstünlüklər və bu texnologiyaların effektivliyi geniş şəkildə təhlil edilmişdir. Eyni zamanda, Docker platformasının əsas komponentləri haqqında da ətraflı məlumat verilmişdir. Araşdırmanın nəticələri göstərir ki, konteyner texnologiyaları müasir proqramlaşdırmada resurs istifadəsini optimallaşdırır və tətbiqlərin mühitə olan asılılığını azaldır.

Açar sözlər: *Docker, Konteyner, Docker Hub, Docker Compose, Kubernetes*

Abstract

This article examines the basic principles and application areas of container technologies, especially the Docker platform, in the process of software development. Container technologies, unlike traditional virtual machines, provide easier portability of applications and uninterrupted operation in any environment. Docker technology has led to important and revolutionary innovations in the development and distribution of software. This technology provides users with simple structural

and management advantages, allowing for faster and more efficient work processes. The article extensively analyzes the advantages of using container and Docker technologies and the effectiveness of these technologies. At the same time, detailed information is provided about the main components of the Docker platform. The results of the study show that container technologies optimize resource usage in modern programming and reduce the dependence of applications on the environment.

Keywords: *Docker, Container, Docker Hub, Docker Compose, Kubernetes*

Giriş

Konteynerlər, proqramların və tətbiqlərin müxtəlif hesablama mühitlərində problemsiz işləyə bilməsi üçün gerekli bütün resursları daxilində cəmləşdirən və onları müstəqil şəkildə işlədən bir texnologiyadır. Bu yanaşma proqramları, onların bütün asılılıqları və işləmə mühiti ilə birlikdə vahid bir icra edilə bilən paket formasında təqdim etməklə, virtuallaşdırma üsulu kimi fəaliyyət göstərir. Konteyner texnologiyası proqramın istifadə olunduğu mühitdən – istər inkişaf mərhələsində olan noutbuk, istər test sistemi, istərsə də istehsalat serveri – asılı olmadan sabit və eyni şəkildə işləməsini təmin edir. Bu texnologiya tətbiqləri əsas əməliyyat sistemindən ayırdığı üçün, host sistemində mövcud olan hər hansı bir parametrlə və ya proqram təminatı ilə birbaşa əlaqəyə ehtiyac duymur. Proqramın düzgün çalışması üçün zəruri olan bütün komponentlər konteynerin daxilində mövcud olur. Bu cür izolyasiya konteynerləri ənənəvi virtual maşınlarla nisbətən daha qənaətcil edir, çünki virtual maşınlar bütöv bir əməliyyat sistemi ilə birlikdə fəaliyyət göstərdiyi halda, konteynerlər yalnız tətbiq üçün zəruri olan resursları özündə birləşdirir.

Konteyner texnologiyası, ənənəvi virtuallaşdırma üsulları ilə müqayisədə bir sıra üstünlüklərə malikdir. Onların sadə quruluşu, sürətli başladılması və az miqdarda resurs tələb etməsi, eyni sistemdə bir neçə konteynerin effektiv şəkildə idarə olunmasına şərait yaradır. Bundan əlavə, konteynerlər sabit və təkrarlana bilən bir iş mühiti təqdim edərək proqramların hazırlanması, test prosesi və yerləşdirmə mərhələlərini daha sadə və öncədən proqnozlaşdırıla bilən hala gətirir.

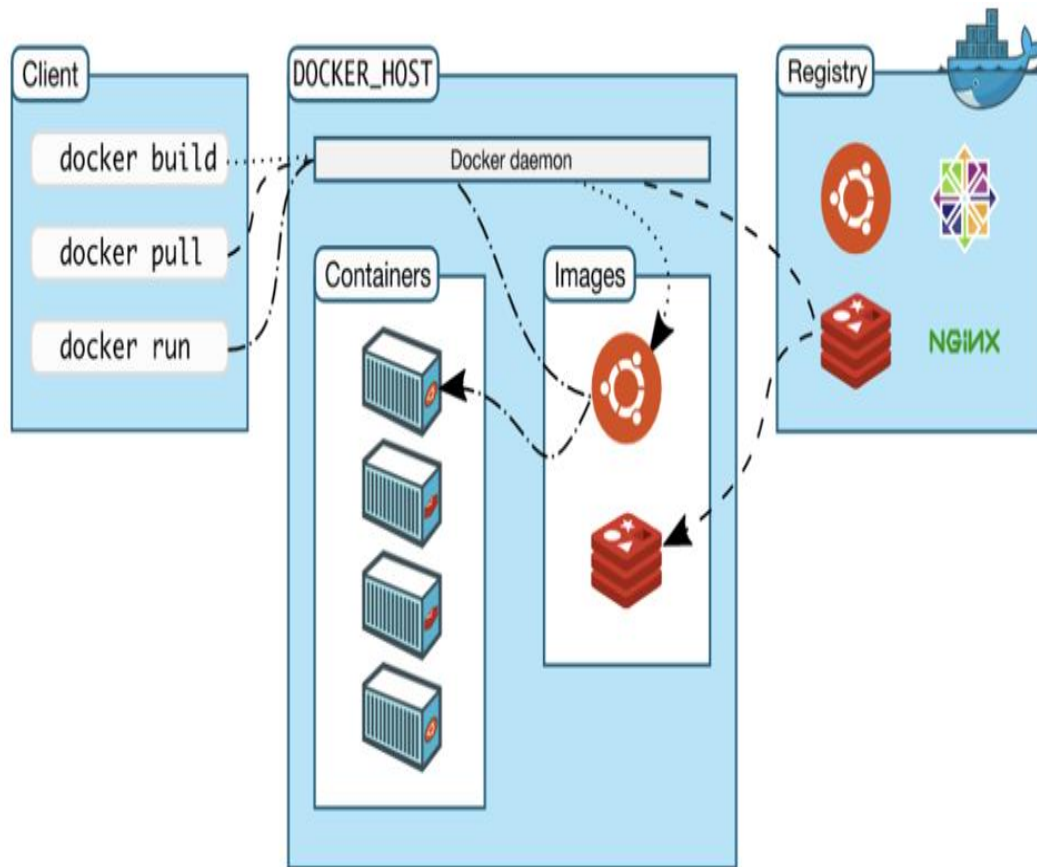
Docker

Docker, proqramların hazırlanması, daşınması və icrası üçün açıq mənbəli və geniş şəkildə istifadə edilən bir platformadır. Bu texnologiya tətbiqlərin daha sürətli, daha asan və etibarlı şəkildə inkişaf etdirilməsini təmin etməyə yönəlib. Docker konteynerləri, tətbiqin işləməsi üçün gerekli olan bütün kodu, işləmə mühitini, sistem alətlərini, kitabxanaları və konfigurasiya fayllarını bir paket halında

birleştirərək, müxtəlif sistemlərdə eyni şəkildə fəaliyyət göstərməsini mümkün edir. Nəticə etibarilə, Docker proqram təminatının inkişafında sadəlik, sabitlik və yüksək məhsuldarlıq təmin edir.

Docker, tərtibatçılara konteyner daxilində tam izolyasiya olunmuş bir mühit yaratmağa imkan verir. Bu konteynerlər, istifadə olunan əməliyyat sistemindən asılı olmayaraq, fərqli cihazlarda eyni şəkildə işləyir və bununla da proqramın hər bir mühitdə fasiləsiz çalışmasını təmin edir. Məsələn, bir proqramçı tətbiqi öz şəxsi kompüterində hazırlayıb test etdikdə, həmin tətbiqin başqa bir cihazda da eyni şəkildə işləməsi lazım gəlsə, Docker bu uyğunluğu qoruyaraq tətbiqin əvvəlki mühitdə olduğu kimi problemsiz çalışmasını təmin edir. Bu imkan, “mənim kompüterimdə işləyir” şəklində tanınan problemi aradan qaldırır və inkişaf prosesində qarşılaşılan çətinlikləri xeyli azaldır. Ənənəvi proqramlaşdırma yanaşmasında bu problem, tətbiqlərin müxtəlif mühitlər və əməliyyat sistemləri ilə uyğunluq problemləri səbəbilə tez-tez səhvlərlə qarşılaşmasına gətirib çıxarırdı. Docker bu kimi uyğunsuzluqları aradan qaldıraraq, proqram təminatının inkişafını daha rahat və güvənilir edir. (Ibryam, B., & Huß, R. (2023) s. 34-36)

Docker, tətbiqlərin həyat dövrünü idarə etmək üçün geniş funksionallı və səmərəli vasitələr təqdim edir. Bu vasitələr sayəsində tərtibatçılar və əməliyyat heyətləri yalnız tətbiqləri yaratmaqla kifayətlənmir, həmçinin onların test olunması, yerləşdirilməsi, miqyaslandırılması və idarə olunması kimi vacib mərhələləri də effektiv şəkildə həyata keçirə bilirlər. Docker konteynerləri, bu proseslərin hər birinin asan və nəzarətli şəkildə icrasına şərait yaradır və müxtəlif mühitlərdə tətbiqlərin və xidmətlərin etibarlı işləməsini təmin edir. Bu cür imkanlar, xüsusilə geniş miqyaslı tətbiqlər və mikroservis arxitekturasını mənimsəmiş layihələr üçün əvəzolunmazdır. Docker-ın təqdim etdiyi bu üstünlüklər onu yalnız proqramçılar üçün deyil, həmçinin sistem administratorları və əməliyyat mütəxəssisləri üçün də olduqca populyar və geniş istifadə olunan bir vasitəyə çevirib. Docker müxtəlif sahələrdə tətbiq oluna biləcək qədər çevikdir və bu da onun funksional gücünü daha da artırır. Veb proqramlaşdırmadan tutmuş mikroservis əsaslı sistemlərə, bulud texnologiyalarından IoT yönümlü layihələrə qədər geniş istifadə sahəsinə malikdir. Veb inkişafı və mikroservislər sahəsində Docker tətbiqlərin daha sürətli və səmərəli hazırlanmasını və yerləşdirilməsini təmin edir. Bulud mühitlərində Docker, tətbiqlərin istənilən infrastruktura asanlıqla yerləşdirilməsinə imkan verir. IoT sahəsində isə, Docker konteynerləri məhdud resurslara malik kiçik cihazlarda belə mürəkkəb tətbiqlərin problemsiz işləməsinə şərait yaradır. (Hightower, K., Burns, B., & Metcalfe, J. (2022), s. 120-125)



Şəkil 1. Docker arxitekturası

Docker, tətbiqlərin paketlənməsi və yayılması üçün sadə və səmərəli bir həll təqdim etməklə yanaşı, konteynerlərin bütün həyat dövrü – miqyaslandırma, idarəetmə və yerləşdirmə mərhələləri də daxil olmaqla – rahat şəkildə idarə olunmasına imkan yaradır. Bu xüsusiyyətləri sayəsində Docker həm proqramçıların, həm də əməliyyat komandalarının geniş şəkildə istifadə etdiyi və populyarlıq qazanmış bir vasitəyə çevrilmişdir. O, veb proqramlaşdırmadan tutmuş mikroservis memarlığına, bulud əsaslı həllərdən IoT yönümlü layihələrə qədər bir çox müxtəlif sahədə geniş şəkildə tətbiq olunur.

Konteynerləşmə

Konteynerləşdirmə – proqram təminatının bütün asılılıqları ilə birlikdə bir konteyner daxilində toplanaraq bir hesablama mühitindən digərinə problemsiz şəkildə işləməsini təmin edən bir prosesdir. Başqa sözlə, bu yanaşma proqramların eyni host əməliyyat sisteminin nüvəsindən istifadə edərək, bir neçə təcrid olunmuş konteyner şəklində eyni cihazda paralel şəkildə işləməsinə əsaslanır. Bu üsul tətbiqlərin daha yüngül olması ilə yanaşı, onların yerləşdirilməsini də əhəmiyyətli dərəcədə səmərəli edir. (Ibryam, B., & Huß, R. (2019) s. 58)

Konteynerlərin bir sıra üstünlükləri var. Konteynerlərin əsas üstünlükləri:

Daşınma qabiliyyəti: Konteynerlər bütün asılılıqları özündə cəmlədiyi üçün, proqramların müxtəlif mühitlərdə ardıcıl və problemsiz işləməsinə təmin edir.

Effektivlik: Konteynerlər, virtual maşınlardan (VM-lərdən) daha az resurs istifadə edir, bu da eyni cihazda daha çox tətbiqin işləməsinə imkan verir.

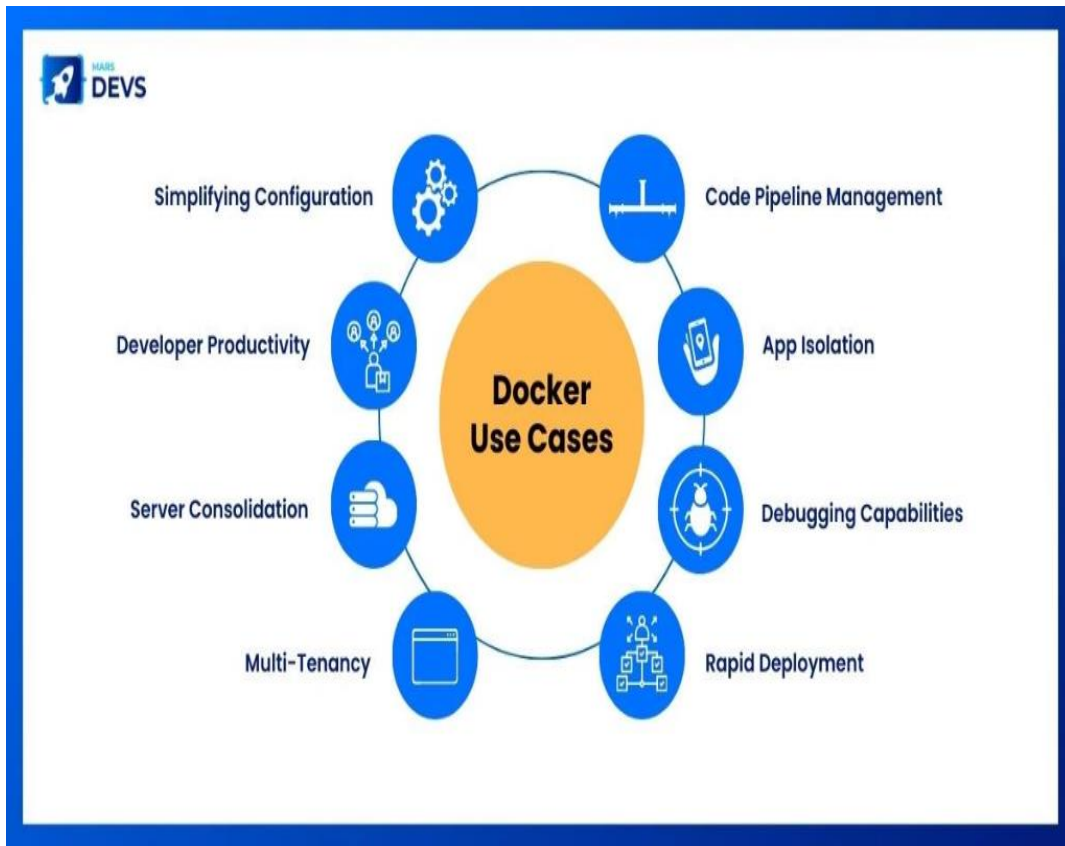
Miqyaslanma bilmə: Konteynerlər asanlıqla genişləndirilə və ya kiçildilə bilər, bu da dinamik iş yüklərinin səmərəli şəkildə idarə olunmasına şərait yaradır.

Daha sürətli yerləşdirmə: Yüngül strukturları sayəsində konteynerlər inkişaf və CI/CD (davamlı inteqrasiya və davamlı yerləşdirmə) proseslərini sürətləndirərək çox az bir zamanda yerləşdirilə bilər.

Ardıcılıq: Konteynerlər tətbiqin inkişaf, sınaq və istehsal mühitlərində eyni şəkildə işləməsinə təmin edir. (Poulton, N. (2024) s. 45-47)

Docker konteynerləri, proqram təminatlarını və onların asılılıq mühitlərini bir araya gətirərək işə salmaq üçün yüngül və daşınan bir yanaşma təklif edir. Bu konteynerlər, izolyasiya olunmuş və sabit bir iş mühiti formalaşdıraraq, tətbiqlərin ardıcıl şəkildə hazırlanması, testdən keçirilməsi və yerləşdirilməsini daha proqnozlaşdırıla bilən və rahat edir. Docker konteynerləri, tətbiqin işləməsi üçün gərəkli olan kodu, müvafiq mühiti, sistem kitabxanalarını və konfiqurasiya parametrlərini özündə cəmləşdirən əvvəlcədən qurulmuş Docker təsvirləri əsasında yaradılır. Bu konteynerlər, host əməliyyat sisteminin nüvəsindən istifadə etməklə, istifadəçi sahəsində fəaliyyət göstərir və eyni zamanda onların fayl sistemi və şəbəkə resursları digər proseslərdən ayrılmış olur. Belə təcridolma mühiti, qarşılıqlı təsirlərdən doğan problemlərin qarşısını alır və sistemin sabitliyi ilə təhlükəsizliyini artırır. Docker konteynerlərini istənilən vaxt asanlıqla işə salmaq, dayandırmaq və silmək mümkündür və onlar Docker mühitinin mövcud olduğu istənilən cihazda əlavə uyğunlaşdırmaya ehtiyac qalmadan işləyə bilər. Bu, konteynerlərin proqram inkişafından başlayaraq test və yerləşdirmə mərhələlərinə qədər olan bütün dövrünü effektiv şəkildə idarə etməyə şərait yaradır. Bundan başqa, Docker konteynerləri Docker Compose və Kubernetes kimi orkestrləşdirmə vasitələri ilə qarşılıqlı əlaqəyə girərək birlikdə idarə edilə bilər ki, bu da kompleks və çoxsəviyyəli tətbiqlərin qurulması və yerləşdirilməsi prosesini əhəmiyyətli dərəcədə sadələşdirir. (Matthias, K., & Kane, S. P. (2015) s. 120)

Nəticə etibarilə, Docker konteynerləri proqramların paketlənməsi və işə salınması üçün sürətli, çevik və miqyaslanma bilən bir həll təqdim edir və müasir proqram inkişafı və yerləşdirmə proseslərinin ayrılmaz hissəsinə çevrilmişdir.



Şəkil 2. Docker texnologiyasının istifadə sahələri

Docker Konteynerlərinin Üstünlükləri:

Docker-in sənayedəki geniş yayılması, onun təmin etdiyi bir sıra üstünlüklərə bağlıdır. Aşağıda Docker istifadəsinin bəzi əsas faydaları təqdim olunur:

1. Xərc Effektivliyi

Docker konteynerlərinin əsas üstünlüklərindən biri onların təmin etdiyi yüksək qənaət potensialıdır. Bu texnologiyanın tətbiqi investisiyaların geri dönüşünü artıraraq, tərtibatçılara xərclərin minimuma endirilməsi və gəlirlərin yüksəldilməsi baxımından mühüm üstünlüklər verir. Xüsusilə davamlı və uzunmüddətli gəlir əldə etmək niyyətində olan iri təşkilatlar üçün Docker-in istifadəsi son dərəcə əlverişlidir. Bu texnologiya proqram inkişafında tələb olunan infrastruktur və resurs ehtiyaclarını azaltdığına görə, zamanla xeyli qənaət əldə etməyə imkan verir. Ən əhəmiyyətli qənaət sahələrindən biri isə server resurslarına olan ehtiyacın azalmasıdır. Docker tətbiqlərinin səmərəli işləməsi nəticəsində onları yaratmaq və idarə etmək üçün daha az sayda mühəndis heyəti tələb olunur ki, bu da insan resursları ilə bağlı xərclərin azalmasına gətirib çıxarır. (Poulton, N. (2024), s. 48-50)

2. Məhsuldarlıq

Docker konteynerləri, müxtəlif inkişaf və buraxılış mərhələlərini eyni platformada ardıcıl şəkildə idarə etməyə imkan verir. Standartlaşdırılmış mühit istehsal proseslərinin daha səmərəli olmasını təmin edir. Komanda üzvləri və mühəndislər, avtomatik olaraq səhvləri aradan qaldırmaq və tətbiq performansını analiz etmək üçün lazımi vasitələrlə təmin olunur. Bu, səhvlərin düzəldilməsi və yeni xüsusiyyətlərin təkmilləşdirilməsi üçün sərf olunan vaxtı azaldır və funksionallığın inkişafına daha çox zaman ayırmağa imkan verir. Docker-in modul əsaslı infrastrukturu sayəsində versiya yenilikləri asanlıqla tətbiq oluna bilər. Problemi olan komponentin təkmilləşdirilməsi geri alınabilir və sınaq məqsədilə əvvəlki versiyaya keçid etmək mümkündür.

3. Uyğunluq

Docker ilə inkişaf etdirilən tətbiqlər, müxtəlif əməliyyat sistemlərində və parametrlərdə eyni şəkildə fəaliyyət göstərir. Bu xüsusiyyət, tərtibatçılara yeni mühitlər qurmaq və bu mühitlərlə əlaqəli spesifik problemləri həll etmək üçün sərf olunan vaxtı əhəmiyyətli dərəcədə azaldır. Kod bazasının yaradılması daha sadə və digər platformalar üçün daha da portativ olur. Ənənəvi inkişaf mühitlərinə nisbətən, istehsal infrastrukturu daha asan idarə olunur və daha yüksək etibarlılığa malikdir. Eyni konteyner şəkildən tətbiqin bütün inkişaf mərhələlərində fasiləsiz istifadə etmək mümkündür.

4. Asan Yerləşdirmə

Docker-də tətbiqin yerləşdirilməsi prosesi yalnız bir neçə saniyə çəkir. Platforma hər bir fərdi proses üçün konteyner yaradır, bu da tərtibatçılara yükləmə vaxtını azaltmağa imkan verir. Proqramlar birbaşa əməliyyat sisteminə yüklənmir, əksinə əməliyyat sisteminin nüvəsindən istifadə edərək məlumatları effektiv şəkildə oxumaq, yazmaq və yenidən yazmaq imkanı qazanır.

5. Davamlı Test

Docker, proqramların ilkin inkişaf mərhələsindən istehsalın sonuna qədər eyni mühitlərdə test edilməsini təmin edir. Docker konteynerləri əvvəlcədən konfigurasiya edilə bilər və tərtibatçılar heç bir əlavə müdaxilə olmadan istehsal mərhələlərinə qədər eyni konteynerlərdən istifadə edə bilərlər. Bununla yanaşı, tərtibatçılar buraxılış dövrü boyunca yeni funksiyaları test edə və yalnız mövcud konteynerlərdə dəyişikliklər edərək bu yenilikləri tətbiq edə bilərlər. Docker-in təqdim etdiyi çeviklik, platformanın ümumi səmərəliliyini artırır.

6. Təhlükəsizlik

Docker-in geniş miqyaslılıq və çevikliyinə əlavə olaraq, təhlükəsizlik də onun mühüm üstünlüklərindən biridir. Docker, konteynerlərdə işləyən tətbiqləri bir-birindən tamamilə təcrid

edərək, tətbiq trafikinə tam nəzarət etməyə imkan yaradır. Konteynerlər bir-birinin məlumatlarına daxil ola bilməz və yalnız onlara təyin olunmuş resursları istifadə edir, bu da tətbiqlərin təhlükəsizliyini təmin edir. (Turnbull, J. (2014) s. 101-105)

7. İzolyasiya Edilmiş Mühit

Docker, proqramları və onların resurslarını bir-birindən tam təcrid edərək, hər bir konteynerin yalnız özünə lazım olan resurslara çıxışını təmin edir. Tərtibatçılar, hər bir proqram üçün fərqli konteynerlər istifadə edə bilər və bu konteynerlər müstəqil şəkildə işləyə bilər. Hər tətbiq yalnız ona təyin olunmuş resurslardan istifadə edir, bu da proqramların təcridini və təhlükəsizlik risklərinin azaldılmasını təmin edir.

8. Bulud Texnologiyası

Docker-in əsas üstünlüklərindən biri də onun bulud texnologiyaları vasitəsi ilə daşıma bilinməsidir. Bu xüsusiyyət müəssisələrə bulud əsaslı hesablama sistemlərinə keçmə imkanını verir. Məşhur bulud platformaları, məsələn, AWS və GCP, artıq Docker-i geniş şəkildə dəstəkləyir. Tərtibatçılar konteynerləri bu bulud əsaslı platformalarda işlədə bilər və tətbiqlər asanlıqla bir bulud xidmətindən digərinə köçürülə bilər.

Aşağıdakı cədvəldə Docker-in əsas üstünlükləri və onların proqram inkişafına təsiri haqqında qısa məlumat verilir.

Üstünlük	Təsviri
Xərc Effektivliyi	Server və insan resursları xərclərini azaldır, investisiyanın geri dönüşünü artırır.
Məhsuldarlıq	Standartlaşdırılmış mühit sayəsində səhvlərin düzəldilməsi və inkişaf prosesi sürətlənir.
Uyğunluq	Fərqli sistemlərdə eyni kod bazasını istifadə etməyə imkan verir.
Asan Yerləşdirmə	Proqramların sürətli yerləşdirilməsini təmin edir, OS nüvəsindən istifadə edərək resursları səmərəli idarə edir.
Davamlı Test	İnkişafdan istehsal qədər eyni konteyner mühitində avtomatik testləri yerinə yetirir.
Təhlükəsizlik	Konteynerlər arasında tam izolyasiya təmin edir, məlumat təhlükəsizliyini artırır.
İzolyasiya Edilmiş Mühit	Hər bir konteyner müstəqil işləyir, yalnız təyin olunmuş resurslardan istifadə edir.
Bulud Texnologiyası	Docker bulud əsaslı sistemlərdə asanlıqla tətbiq edilir və platformalar arasında daşına bilir.

Cədvəl 1. Docker-in əsas üstünlükləri və onların proqram inkişafına təsiri

Docker Texnologiyasının Əsas Komponentləri

1. Docker Engine

Docker Engine (DE), konteynerlərin yaradılması və idarə edilməsi ilə bağlı əsas müştəri-server texnologiyasıdır. Docker Engine üç əsas hissədən ibarətdir:

- Server: Docker demonu (dockerd) konteynerləri yaratmaq və idarə etmək üçün əsas funksionu yerinə yetirir.
- Rest API: Docker ilə digər proqramlar arasında əlaqə quraraq, dockerd-ə əmrlər göndərir.

- Command Line Interface (CLI): Docker əməllərini icra etmək üçün istifadəçi tərəfindən çağırılan interfeysdən ibarətdir.

2. Docker Müştərisi

Docker istifadəçiləri, müştəri vasitəsilə Docker ilə əlaqə qura bilirlər. Docker müştərisi, verilən əməlləri dockerd demonuna göndərir. Docker API, müştəri tərəfindən istifadə olunur və bir neçə demonla əlaqə qurulmasını mümkün edir.

3. Docker Reyestrləri

Docker reyestri, konteyner şəkillərinin saxlanıldığı yerdir. Docker Hub, Docker şəkillərinin standart olaraq saxlandığı ictimai reyestrdir, amma şəxsi reyestrlər də qurula bilər. Docker əməlləri (docker pull, docker run) vasitəsilə konteyner şəkilləri reyestrdən yüklənir, docker push əmri ilə isə reyestrə yüklənir.

4. Docker Şəkilləri

Docker şəkilləri, konteynerləri yaratmaq üçün istifadə olunan şablonlardır. Hər şəkil dəyişilməz və yalnız oxunabilən fayllardan ibarətdir, həmçinin mənbə kodu, kitabxanalar, asılılıqlar və digər proqram fayllarını özündə cəmləyir. Şəkil, Dockerfile-da olan təlimatlarla yaradılır.

5. Dockerfile

Dockerfile, Docker şəkilini necə yaratmağı müəyyənləşdirən bir skriptdir. O, əməliyyat sistemi, proqramlaşdırma dilləri, mühit dəyişənləri, şəbəkə portları və digər zəruri komponentləri özündə saxlayır. Dockerfile-da olan bütün əməllər avtomatik olaraq icra olunur.

6. Docker Hub

Docker Hub, Docker tərəfindən təqdim olunan bulud əsaslı depozitordur və burada çoxsaylı açıq mənbəli konteyner şəkilləri yerləşir. Bu platforma tərtibatçılara konteynerləri sürətlə yükləmək, komanda üzvləri ilə əməkdaşlıq etmək və inkişaf boru kəmərinə daha tez integrasiya etmək imkanı verir.

7. Docker Həcmələri

Docker həcmələri, konteynerlər arasında məlumatların saxlanması və paylaşılması məqsədilə istifadə edilir. Bu həcmələr, konteynerin fəaliyyət müddətindən asılı olmayaraq, host sistemində saxlanılır və konteynerlərin işləyişi dayandırıldıqda belə məlumatların itirilməsinin qarşısını alır.

8. Docker Compose

Docker Compose, bir neçə konteynerin eyni vaxtda idarə olunmasına imkan verən bir alətdir. YAML formatında yazılmış konfigurasiya faylı vasitəsilə tətbiqin müxtəlif xidmətləri təyin edilir və bu, konteynerləri sadəcə bir komanda ilə işə salmaq, icra etmək, qarşılıqlı əlaqə qurmaq və ya dayandırmaq üçün rahatlıq təmin edir.

9. Docker Masaüstü

Docker Desktop, Windows və Mac əməliyyat sistemlərində Docker mühitinin qurulması və konteynerlərin işə salınması üçün istifadə olunan sadə və istifadəsi rahat bir tətbiqdır. Bu proqram, Docker Engine, Docker Compose, Docker CLI, Kubernetes və digər əlaqəli alətləri ehtiva edir və proqramçıların inkişaf mühitlərini qurmasını və mikroservis arxitekturası yaratmalarını olduqca asanlaşdırır.

10. Kubernetes

Kubernetes, açıq mənbəli bir konteyner orkestrasiya platformasıdır və ilkin olaraq Google tərəfindən daxili istifadə məqsədilə hazırlanmışdır. Bu alət, konteynerlərin yerləşdirilməsi, yaddaş resurslarının təchizatı, yük balanslaşdırılması, sistem yeniləmələrinin həyata keçirilməsi, xidmətlərin aşkar edilməsi, sağlamlıq monitorinqi və digər konteyner əsaslı arxitekturaların idarə olunmasına dair mühüm tapşırıqları avtomatlaşdırmaq və planlaşdırmaq üçün istifadə olunur. (Poulton, N. (2024), s. 45-47)

Docker və konteynerləşdirmə texnologiyasının xarici təcrübəsi

Docker və konteynerləşdirmə texnologiyası müasir proqram inkişafında geniş istifadə edilən qabaqcıl yanaşmadır. Xüsusilə, yüksək miqyaslı və sürətlə dəyişən mühitlərə uyğunlaşma qabiliyyəti tələb edən iri şirkətlər və startaplar tərəfindən bu texnologiya geniş tətbiq olunur. Məsələn, Netflix, mikroservis arxitekturasını Docker konteynerləri vasitəsilə idarə edir və hər gün yüzlərlə konteyneri avtomatik olaraq işə salır. Bu yanaşma, onlara xidmətlərin miqyasını artırmağa, tətbiqləri qısa müddət ərzində yeniləməyə və müştərilərə dayanıqlı xidmət göstərməyə imkan verir. Digər tərəfdən, Google konteynerləşdirmə sahəsində lider şirkətlərdən biridir. Google-un daxili infrastrukturunu Kubernetes platforması üzərində qurub, bu da Docker konteynerlərinin idarə edilməsini daha da optimallaşdırır. Kubernetes ilə birlikdə, Google inkişafçılara konteynerlərin avtomatik yerləşdirilməsi, balanslaşdırılması və miqyaslandırılması sahəsində geniş imkanlar təqdim edir. Amazon Web Services (AWS) də konteynerləşdirmə texnologiyasını bulud əsaslı xidmətlərində intensiv şəkildə istifadə edir. AWS Elastic Container Service (ECS) və Elastic Kubernetes Service (EKS) kimi platformalar inkişafçılara konteynerləşdirilmiş tətbiqləri sürətli və təhlükəsiz şəkildə idarə etməyə imkan verir. Bu xidmətlər, xüsusilə startaplar və kiçik müəssisələr üçün resursların daha səmərəli

idarə edilməsini təmin edir. Xarici təcrübə göstərir ki, Docker texnologiyası yalnız iri şirkətlər üçün deyil, həm də kiçik və orta ölçülü müəssisələr üçün çox faydalıdır. Docker istifadəsi vasitəsilə müəssisələr resurslardan daha effektiv istifadə edir, proqram təminatını sürətlə işə salır və miqyaslı dəyişikliklərə uyğunlaşmada asanlıqla müvəffəqiyyət əldə edirlər. Bu təcrübələr Azərbaycan mühitində də konteynerləşdirmənin tətbiqi üçün geniş potensial imkanlar yaradır və yeni layihələr üçün nümunə ola bilər.

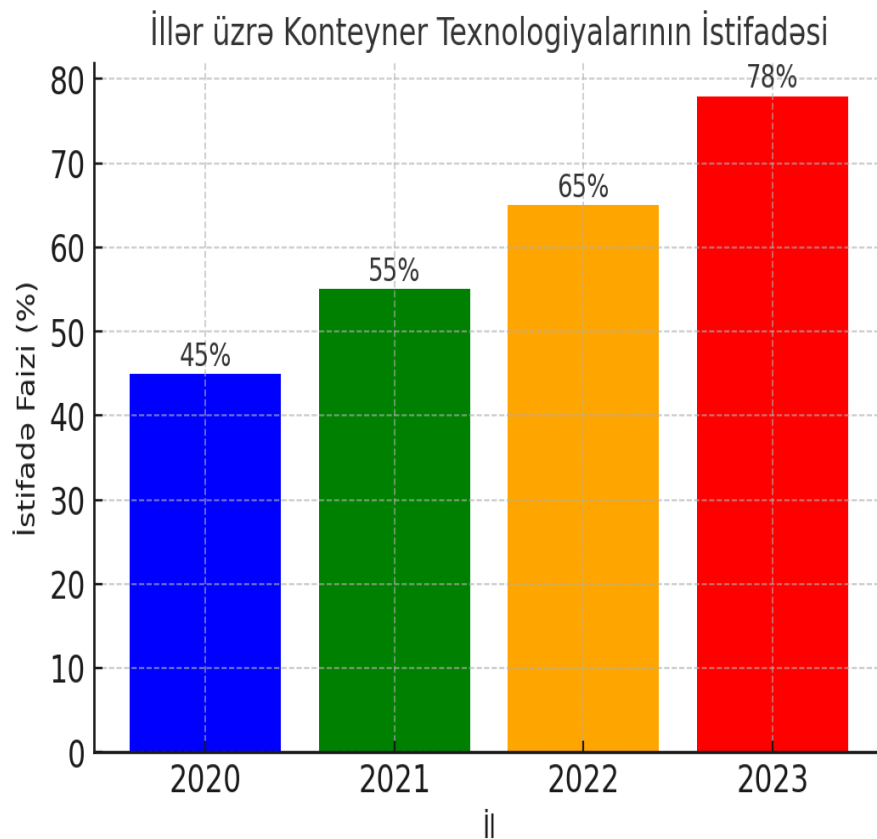
Konteyner və Docker Texnologiyalarının İstifadəsinin Artımı

Son illərdə konteyner və Docker texnologiyalarının istifadəsi sürətlə artmışdır. Bu texnologiyaların populyarlığının artmasının əsas səbəbləri arasında proqram inkişafında çevikliyin artırılması, resursların daha effektiv istifadəsi və tətbiqlərin sürətli şəkildə yerləşdirilməsi kimi amillər önə çıxır. Riyazi təhlillər və statistik göstəricilər, konteyner texnologiyalarının yayılmasının eksponensial artım modelinə uyğun inkişaf etdiyini və yaxın gələcəkdə bazarın daha da genişləncəyini göstərir.

Statistik Göstəricilər 2023-cü ilin statistik məlumatlarına əsaslanaraq, Docker və konteyner texnologiyalarının istifadəsində əhəmiyyətli artım müşahidə olunub. Sysdig-in 2023 Cloud-Native Security and Usage Report hesabatına görə:

- 2020-ci ildə konteyner əsaslı sistemlərdən istifadə 45%-i təşkil edirdi.
- 2021-ci ildə bu göstərici 55%-ə çatdı,
- 2022-ci ildə 65%-ə yüksəldi,
- 2023-cü ildə isə 78%-ə qədər artdı.

Bu statistik göstəricilər konteyner texnologiyalarının IT sahəsində dominant bir mövqeyə sahib olmağa başladığını və yaxın gələcəkdə daha geniş yayılacağını təsdiqləyir.



Şəkil 3. Konteyner texnologiyasının illər üzrə artımı

Nəticə

Docker və konteyner texnologiyalarının istifadəyə verilməsi proqram təminatının inkişafında əsaslı dəyişikliklərə səbəb olmuşdur. Ənənəvi yanaşmalarla müqayisədə, bu texnologiyalar tərtibatçılara tətbiqləri daha sürətli, asan və effektiv şəkildə yerləşdirməyə imkan verir. Docker konteynerləri tətbiqlərin mühitdən asılı olmayaraq eyni şəkildə işləməsinə təmin edir. Bu xüsusiyyət həm proqram tərtibatçıları, həm də əməliyyat komandaları üçün mühüm üstünlüklər təqdim edir. Xüsusilə, resurslardan daha optimal istifadə edilərək müasir texnoloji proseslərin optimallaşdırılması təmin olunur. Docker texnologiyası tətbiqlərin sürətli hazırlanması, buraxılması və daimi testlərin keçirilməsini sadələşdirərək proqram inkişafını daha çevik edir. Bundan əlavə, Docker alətləri, məsələn, Docker Hub və Docker Compose, müxtəlif konteynerlərin idarə edilməsini asanlaşdırır və onların qarşılıqlı əlaqəsini təşkil edir. Bu vasitələr konteynerlərin yaradılması, saxlanması və müxtəlif mühitlərdə işləməsi üçün rahat interfeyslər təqdim edir. Eyni zamanda, bu texnologiyalar təhlükəsizlik, uyğunluq və bulud əsaslı həllərlə tam inteqrasiyanı dəstəkləyir. Docker həm proqram inkişafı, həm də əməliyyat komandalarının işlərini daha səmərəli şəkildə həyata keçirməsinə kömək edir və həm böyük şirkətlərdə, həm də kiçik startaplarda geniş istifadə olunur. Bu texnologiyanın çevikliyi və yüksək effektivliyi onu müasir proqram təminatının ayrılmaz bir hissəsinə çevirir.

Gələcəkdə Docker texnologiyasının davamlı inkişafı nəticəsində proqram təminatının daha da avtomatlaşdırılması və optimallaşdırılması gözlənilir.

Docker və konteyner texnologiyalarının proqram təminatının inkişafına olan təsiri yalnız funksional üstünlüklərlə deyil, həm də konkret statistik göstəricilərlə dəstəklənir. 2023-cü ilin məlumatlarına əsasən, konteyner əsaslı sistemlərin istifadəsi 78%-ə çatmış və bu göstəricinin 2026-cı ilə qədər 90%-ə yüksələcəyi gözlənilir. Docker Hub platformasında yüklənən konteyner imiclərinin sayı 15 milyarddan çox olub və bu göstərici illik 35%-lik artım tempi nümayiş etdirir. Şirkətlərin konteyner texnologiyalarına keçidi nəticəsində server xərclərində 30%-lik azalma, proqram təminatının yerləşdirilmə müddətində 60%-lik azalma və İT resurslarının səmərəli istifadəsində isə 40%-lik artım müşahidə edilmişdir. Mikroservis əsaslı sistemlərin 92%-i konteyner texnologiyalarına əsaslandığından, konteynerlər müasir proqram təminatında mühüm yer tutmaqdadır.

Ümumilikdə, Docker texnologiyası təkcə bu gün üçün deyil, gələcəkdə də proqram təminatının əsas elementlərindən biri kimi vacib rol oynamaqda davam edəcəkdir. Eyni zamanda, bu texnologiya müasir texnoloji trendlərə, xüsusən mikroservis arxitekturasına tam şəkildə uyğunlaşır. Bu baxımdan, proqram təminatının inkişafında sürət, təhlükəsizlik və iqtisadi səmərəlilik kimi əsas göstəriciləri təmin edərək, tərtibatçılara innovativ həllər təqdim edir.

Ədəbiyyat

1. Hightower, K., Burns, B., & Metcalfe, J. (2022). Kubernetes Up & Running: Dive into the Future of Infrastructure. O'Reilly Media, s. 120-125.
2. Ibryam, B., & Huß, R. (2019). Kubernetes Patterns: Reusable Elements for Designing Cloud-Native Applications. O'Reilly Media. s. 34-58.
3. Matthias, K., & Kane, S. P. (2015). Docker: Up & Running: Shipping Reliable Containers in Production. O'Reilly Media. s. 110-120.
4. Poulton, N. (2024). Docker Deep Dive: 2024 Edition. Docker, s. 45-50.
5. Poulton, N. (2024). Kubernetes Patterns: Reusable Elements for Designing Cloud-Native Applications. O'Reilly Media, s. 45-51.
6. Turnbull, J. (2014). The Docker Book: Containerization is the New Virtualization. James Turnbull. s. 75-80.
7. <https://www.linkedin.com/pulse/heres-a-brief-history-docker-ashadozzaman-shvou-hdimc>
8. <https://www.geeksforgeeks.org/introduction-to-docker/>
9. <https://www.geeksforgeeks.org/what-is-a-container/>
10. <https://www.channelfutures.com/connectivity/why-is-docker-so-popular-explaining-the-rise-of-containers-and-docker>
11. <https://www.siddhatech.com/what-is-docker-and-why-is-it-so-popular/>