

DEVOPS JARAYONLARIDA KONTEYNERLASHTIRISH TEKNOLOGIYALARI: DOCKER VA KUBERNETES TAHLILI

Ergasheva Farida Yunus qizi, TATU, talaba
ergashevafarida06303@gmail.com

Kalit so'zlar: DevOps, Docker, Kubernetes, konteynerlashtirish, CI/CD, orkestratsiya, mikroservis arxitekturası.

Kirish. Zamonaviy dasturiy ta'minot ishlab chiqish sohasida DevOps metodologiyasi ishlab chiqish (Development) va operatsion (Operations) jarayonlarni birlashtirib, mahsulotni tezroq va sifatliroq yetkazib berishni ta'minlaydi. Dasturiy ta'minot ishlab chiqish jarayanonida barcha jamoalarni birlashtira olgan DevOps metodologiyasi nafaqat to'g'ri infrastruktura, balki dastur mahsulotining foydalanuvchanligini ham ta'minlashga qodir. Bu esa konteynerlashtirish jarayoni orqali amalga oshiriladi. Ushbu metodologiyaning muvaffaqiyatli amalga oshirilishida konteynerlashtirish texnologiyalari, xususan Docker va Kubernetes, markaziy o'rin tutadi. Dasturiy ta'minot ishlab chiqish metodologiyalari o'z rivojlanishida bir necha bosqichni bosib o'tdi: qattiq va chiziqli bosqichlardan iborat Waterfall (1970-y.), moslashuvchan qisqa sikllarga asoslangan Agile (2000-y.), ishlab chiqish va operatsion jarayonlarni avtomatlashtirish hamda integratsiyalash orqali birlashtirgan DevOps (2009-y.) va, nihoyat, xavfsizlikni butun jamoaning umumiy mas'uliyatiga aylantirgan DevSecOps. Waterfall modelida xavfsizlik odatda loyihaning yakuniy bosqichida (va ko'pincha umuman) tekshirilardi. Agile ishlab chiqish tezligini oshirdi, biroq xavfsizlik masalasini orqada qoldirdi. DevOps esa avtomatlashtirish va jarayonlarning ko'rinishchanligini keltirib, aynan shu ikki xususiyat tufayli xavfsizlikni ishlab chiqishning eng erta bosqichlariga integratsiya qilish imkoniyatini yaratdi — bu yerda DevSecOps tushunchasi tug'iladi. Konteynerlashtirish dasturiy ilovani uning barcha bog'liqliklari bilan birga izolyatsiyalangan muhitga joylashtirish imkonini

beradi, bu esa “mening kompyuterimda ishlaydi” muammosini bartaraf etadi va turli muhitlarda (ishlab chiqish, testlash, ishlab chiqarish) bir xil natijaga erishishni kafolatlaydi.

Tadqiqotning dolzarbligi. An’anaviy virtual mashinalarga nisbatan konteynerlar yengilroq bo‘lib, resurslarni tejamkorroq sarflaydi va ilovalarni bir necha soniyada ishga tushirish imkonini beradi. Docker ushbu konteynerlarni yaratish, saqlash va tarqatish uchun standart vosita sifatida sanoat standartiga aylandi. Docker orqali har bir jarayonni tashqi izolatsiyalab himoya qilish, boshqa konteynerlarga zarar yetganda kamchiliklarsiz ishlashda davom etishni ta’minlab bersa, Docker Compose vositasi orqali bir tarmoqqa birlashtirilgan konteynerlarni ishga tushirish mumkin. Docker Swarmning qulayligi esa hajmni gorizontal kengaytira olish va masshtabni boshqarish imkoniyatida. Kubernetes esa ko‘plab konteynerlarni avtomatik boshqarish, masshtablash va nosozliklardan tiklash vazifalarini bajaradigan orkestratsiya platformasi hisoblanadi. DevOps metodologiyasining oltita asosiy ustuni — CI/CD, infratuzilmani kod sifatida boshqarish (IaC), konfiguratsiyani boshqarish, orkestratsiya, monitoring va mikroservis arxitekturasini — har biri o‘ziga xos xavfsizlik xavflarini keltirib chiqaradi. Shu sababli xavfsizlik choralarini rivojlanish siklining barcha bosqichlariga, ayniqsa eng erta bosqichlariga, joylashtirish g‘oyasi ("shifting left") dolzarblik kasb etadi.

Tadqiqot maqsadi va vazifalari. Ushbu tezisning maqsadi DevOps jarayonlarida Docker va Kubernetes texnologiyalarining o‘rni va afzalliklarini tahlil qilish, ularning CI/CD (Continuous Integration/Continuous Deployment) yo‘llariga integratsiyasini ko‘rib chiqishdir. Vazifalar sifatida quyidagilar belgilandi: konteynerlashtirishning nazariy asoslarini o‘rganish; Docker yordamida ilovani konteynerlashtirish bosqichlarini tahlil qilish; Kubernetes klasteri orqali konteynerlarni orkestratsiya qilish mexanizmlarini yoritish; konteynerlashtirilgan muhitning an’anaviy joylashtirish usullariga nisbatan afzalliklarini asoslash. Zamonaviy CI/CD quvuri kod repozitoriyaga yuklangandan (push) so‘ng bir necha

ketma-ket xavfsizlik bosqichlaridan o'tadi: sirlarni skanerlash (GitLeaks), statik kod tahlili — SAST (Sengrep, CodeQL), bog'liqliklarni tahlil qilish — SCA (Trivy, Snyk), infratuzilma kodini skanerlash — IaC scan (Checkov, tfsec), konteyner tasvirini skanerlash va raqamli imzolash (Trivy, Grype, Cosign), sinov muhitida dinamik tahlil — DAST (OWASP ZAP) hamda ishlab chiqarishga joylashtirilgandan keyin ishga tushirish vaqtidagi monitoring (Falco). Tahlil uchta o'zaro to'ldiruvchi texnik domenni ajratib ko'rsatadi. IaC xavfsizligi Terraform yoki Kubernetes YAML konfiguratsiyalaridagi xatolarni oldini oladi — masalan, ma'lum bir yirik moliyaviy tashkilotga nisbatan qo'llanilgan jarima jamoat uchun ochiq qolib ketgan S3 saqlash joyi bilan bog'liq bo'lgan. Konteyner xavfsizligi image yaratish, orkestratsiya va ishga tushirish bosqichlarini qamrab oladi; noto'g'ri sozlangan docker.sock ulanishi konteyner ichidan butun host tizimiga administrator huquqlari bilan kirish xavfini keltirib chiqarishi mumkin. Ta'minot zanjiri xavfsizligi esa kodning kelib chiqishi, qurilish jarayoni va dasturiy komponentlar ro'yxatini (SBOM) nazorat qilishni anglatadi — 2024-yilda aniqlangan xz-utils dasturidagi maxfiy zaiflik bunga yorqin misol bo'la oladi.

Tahlil va natijalar. Docker texnologiyasi Dockerfile deb ataluvchi konfiguratsiya fayli orqali ilova muhitini tavsiflash imkonini beradi, bu fayl asosida yaratilgan image barcha zarur kutubxonalar va sozlamalarni o'z ichiga oladi. Bunday image'lar Docker Hub yoki xususiy registrlar orqali jamoa a'zolari o'rtasida osongina almashinadi. Kubernetes esa Pod, Deployment, Service kabi abstraktsiyalar orqali konteynerlarning avtomatik masshtablanishini (Horizontal Pod Autoscaler), o'z-o'zini tiklashini (self-healing) va yuklamani muvozanatlashtirishni (load balancing) ta'minlaydi. CI/CD quvurlarida ushbu texnologiyalarning integratsiyasi natijasida kod o'zgarishlaridan tortib ishlab chiqarish muhitiga joylashtirilgunga qadar bo'lgan jarayon avtomatlashtiriladi.

Xulosa. Docker va Kubernetes texnologiyalari zamonaviy DevOps madaniyatining ajralmas qismiga aylanib, dasturiy mahsulotlarni ishlab chiqish,

testlash va joylashtirish jarayonlarini sezilarli darajada soddalashtirmoqda. Ularning qo'llanilishi resurslardan samarali foydalanish, moslashuvchan masshtablash va yuqori mavjudlikni ta'minlash imkonini beradi.

Foydalanilgan adabiyotlar:

1. Kane S.P. DevOps Culture and Practice with OpenShift. — Packt Publishing, 2021.
2. Burns B., Beda J., Hightower K. Kubernetes: Up and Running. — O'Reilly Media, 2022.
3. Docker rasmiy hujjatlari. — <https://docs.docker.com>
4. Kubernetes rasmiy hujjatlari. — <https://kubernetes.io/docs>