

**Networking in the cloud**

VALIYEV SARDORXON AKRAMXON O'G'LI

**UNIT 6:** Networking in the Cloud

**O'qituvchi:** Dostonbek Abdumajidov

**Guruh raqami:** 24-414

**Talaba ID raqami:** 240261

**Taqdim etilgan sana:** 16.06.2026

## BTEC O'QUVCHILAR TOPSHIRIQLARINI BAHOLASH VA DEKLARATSIYASI

Baholash uchun ishlarni taqdim etganda, har bir o'quvchi ish o'zini ekanligini tasdiqlovchi deklaratsiyani imzolashi kerak.

<b>O'quvchi (talaba) identifikatori:</b>	240261
<b>Baholovchi nomi:</b>	Dostonbek Abdumajidov
<b>BTEC dasturi nomi:</b>	Pearson BTEC Higher Nationals in Digital Technologies
<b>Birlik yoki komponent raqami va nomi:</b>	UNIT 6 Networking in the cloud
<b>Topshiriq nomi:</b>	CRM, WPS va ERP tizimlari uchun bulutli platforma
<b>Topshiriq topshirilgan sana:</b>	16.06.2026

Iltimos, har bir topshiriq uchun berilgan ishlarni sanab o'ting. Ishlarni topish mumkin bo'lgan sahifa raqamlarini ko'rsating yoki ishlarning mohiyatini tavsiflang (masalan, diagramma, rasm).

Topshiriq vazifasi ma'lumoti	Ishlar taqdim etildi	Sahifa

### O'quvchi deklaratsiyasi

Ushbu topshiriq uchun taqdim etilgan ish meniki ekanligini tasdiqlayman. Ishda foydalanilgan manbalarga aniq havola qildim. Men noto'g'ri deklaratsiya noto'g'ri ishlashning bir shakli ekanligini tushunaman.

O'quvchi imzosi: \_\_\_\_\_



Sana: 16.06.2026

## Mundarija

<b>Kirish .....</b>	<b>5</b>
<b>A: Muloqotni qo'llab-quvvatlash uchun bulutli infratuzilmada qo'llaniladigan umumiy tarmoq tamoyillarini ko'rib chiqing .....</b>	<b>6</b>
<b>A.P1 – Bulut tarmoq arxitekturalari va standartlarining afzalliklari va cheklovlari.....</b>	<b>6</b>
<b>A.P2 – Bulut ichida tarmoq aloqasi qanday ishlaydi.....</b>	<b>8</b>
<b>A.M1 – Tarmoq standartlarini taqqoslash va ularning bulutли hisoblashni osonlashtirishi.....</b>	<b>11</b>
<b>A.D1 – Bulut muhitini yaratishning tarmoq amalga oshirilishi va unumdorlikka ta'siri .....</b>	<b>13</b>
<b>B: Bulut infratuzilmasida tarmoq texnologiyalari qanday ishlaydi .....</b>	<b>16</b>
<b>B.P3 – Masofaviy operatsion tizim xizmatlari bulutda qanday joylashtiriladi .....</b>	<b>16</b>
<b>B.P4 – Masofaviy mijozlar bulut xizmatlari bilan qanday o'zaro aloqada bo'ladi.....</b>	<b>19</b>
<b>B.M2 – Bulut OS ning masofaviy optimizatsiyasi unumdorlikka qanday ta'sir qiladi .....</b>	<b>21</b>
<b>C: Bulut tizimi uchun tarmoq yechimini loyihalash va sinash.....</b>	<b>24</b>
<b>C.P5 – Bulut tizimi uchun tarmoq yechimini loyihalash.....</b>	<b>24</b>
<b>C.P6 – Loyihalangan tarmoq yechimini amalga oshirish.....</b>	<b>25</b>
<b>C.M3 – Bulutga asoslangan tarmoqni unumdorlik va kengayuvchanlik uchun sinash .....</b>	<b>29</b>
<b>C.D2 – Test natijalari asosida dizayn samaradorligini asoslash .....</b>	<b>31</b>

<b>D: Tarmoq unumdorligini yaxshilash va baholash.....</b>	<b>34</b>
<b>D.P7 – Test natijalari asosida tarmoq yaxshilanishlarini tavsiya etish .....</b>	<b>34</b>
<b>D.P8 – Tarmoq yaxshilanishini amalga oshirish .....</b>	<b>36</b>
<b>D.M4 – Tarmoq yaxshilanishini unumdorlik va kengayuvchanlik uchun sinash .....</b>	<b>39</b>
<b>D.D3 – Tarmoq yaxshilanishlarini asl dizaynga nisbatan asoslash .....</b>	<b>41</b>
<b>Xulosa .....</b>	<b>45</b>
<b>Foydalanilgan adabiyotlar .....</b>	<b>46</b>

## **Kirish**

Zamonaviy biznes muhitida axborot texnologiyalari infratuzilmasi kompaniyaning muvaffaqiyatida hal qiluvchi rol o'ynaydi. Tayyor kiyim-kechak ulgurji savdosi bilan shug'ullanuvchi kompaniyamiz tez o'sib bormoqda, biroq mavjud lokal tarmoq va IT infratuzilmasi ortib borayotgan biznes talablarini qondira olmayapti. Bosh ofis, mintaqaviy omborlar va onlayn savdo platformasi o'rtasidagi ma'lumot almashinuvi sekinlashmoqda, tizimlar bir-biridan ajralgan holda ishlamoqda.

Ushbu hisobot kompaniyaning asosiy biznes tizimlarini – ERP (resurslarni rejalashtirish), CRM (mijozlar bilan munosabatlar) va WMS (ombor boshqaruvi) – yagona, xavfsiz bulut platformasiga ko'chirish va ularni bitta tarmoq orqali birlashtirish bo'yicha tahliliy ish hisoblanadi. Hisobot to'rt qismdan iborat: A qismda bulut tarmog'ining umumiy tamoyillari, B qismda tarmoq texnologiyalarining ishlashi, C qismda tarmoq yechimini loyihalash va amalga oshirish, D qismda esa tizimni yaxshilash va baholash ko'rib chiqiladi.

Loyiha doirasida haqiqiy ishlaydigan CRM tizimi yaratildi, bulutga joylashtirildi va sinab ko'rildi. Har bir nazariy tushuncha shu real tizim misolida tahlil qilinadi, bu nazariya va amaliyotni birlashtiradi.

## **A: Muloqotni qo'llab-quvvatlash uchun bulutli infratuzilmada qo'llaniladigan umumiy tarmoq tamoyillarini ko'rib chiqing**

### **A.P1 – Bulut tarmoq arxitekturalari va standartlarining afzalliklari va cheklovlari**

Kiyim-kechak ulgurji kompaniyasining ERP, CRM va WMS tizimlarini bulutga ko'chirish jarayonida to'g'ri tarmoq arxitekturasini tanlash muhim ahamiyatga ega. Har bir arxitektura modelining o'ziga xos afzalliklari va cheklovlari mavjud bo'lib, ularni kompaniyaning ehtiyojlariga qarab baholash zarur.

**Markazlashgan (centralized) arxitektura** – bu modelda barcha resurslar va boshqaruv yagona markazda joylashadi. Uning asosiy afzalligi – boshqaruv soddaligi va xavfsizlik nazoratining bir joydan amalga oshirilishi. Kichik kompaniyalar uchun bu arzon va tushunarli yechim. Biroq, cheklovi shundaki, agar markaziy tugun ishdan chiqsa, butun tizim to'xtaydi (single point of failure). Bizning kompaniya o'sib borayotgani uchun bu model uzoq muddatda yetarli emas.

**Taqsimlangan (distributed) arxitektura** – bunda resurslar bir nechta server va joylashuv o'rtasida taqsimlanadi. Afzalligi – yuqori ishonchlilik va kengayuvchanlik: bitta server ishdan chiqsa, boshqalari ishlashda davom etadi. Bu kiyim-kechak kompaniyasi uchun ideal, chunki bosh ofis, mintaqaviy omborlar va onlayn savdo platformasi turli joylarda joylashgan. Cheklovi – murakkablik va boshqaruv qiyinligi, hamda serverlar o'rtasidagi muvofiqlashtirish (synchronization) qo'shimcha tarmoq trafigini talab qiladi.

**Bulutli (cloud-native) arxitektura** – zamonaviy yondashuv bo'lib, VPC, subnet, load balancer kabi virtual komponentlardan tashkil topadi. Afzalliklari: talab bo'yicha kengayish (scalability), to'lov faqat ishlatilgan resurs uchun (pay-as-you-go), va yuqori mavjudlik (high availability). Bizning loyihamizda aynan shu model tanlandi. Cheklovi – internetga to'liq bog'liqlik va bulut provayderiga qaramlik (vendor lock-in).

**1-jadval:** Bulut tarmoq arxitektura modellarining taqqoslanishi

<b>Arxitektura</b>	<b>Afzalliklari</b>	<b>Cheklovlari</b>	<b>Kompaniyaga mosligi</b>
Markazlashgan	Boshqaruv sodda, xavfsizlik bir joydan	Single point of failure, kengaymaydi	Past – biznes o'smoqda
Taqsimlangan	Yuqori ishonchlilik, kengayuvchan	Murakkab, sinxronizatsiya kerak	Yuqori – ko'p joylashuv
Bulutli (cloud-native)	Talab bo'yicha kengayish, pay-as-you-go	Internetga bog'liq, vendor lock-in	Eng yuqori – tanlandi

Tarmoq **standartlari** ham muhim rol o'ynaydi. **TCP/IP** – internetning asosiy protokoli bo'lib, ishonchli ma'lumot uzatishni ta'minlaydi; afzalligi – universallik va keng qo'llab-quvvatlanish, cheklovi – qo'shimcha nazorat ma'lumotlari (overhead) tezlikni biroz pasaytiradi. **HTTP/HTTPS** – veb-trafik uchun standart; HTTPS shifrlash orqali xavfsizlikni ta'minlaydi, bu mijozlar ma'lumotlarini himoya qilish uchun zarur, lekin shifrlash qo'shimcha hisoblash resursini talab qiladi. **Ethernet** (IEEE 802.3) – lokal tarmoqlar uchun keng tarqalgan standart, yuqori tezlik beradi, lekin masofaviy ulanishlar uchun mos emas.

Bizning kompaniya uchun **bulutli, taqsimlangan arxitektura** TCP/IP va HTTPS standartlari asosida eng maqbul yechim hisoblanadi. Bu kombinatsiya kengayuvchanlik, xavfsizlik va ishonchlilikni birlashtiradi. Markazlashgan model soddaligi bilan jozibali bo'lsa-da, biznesning o'sib borayotgan talablari va geografik tarqoqligi taqsimlangan bulutli yondashuvni taqozo etadi. Asosiy cheklov – internet ulanishiga bog'liqlik – zaxira ulanish kanallari (redundancy) va load balancing orqali yumshatiladi.

Xulosa qilib aytganda, har bir arxitektura va standart o'z o'rnida foydali, lekin kiyim-kechak ulgurji biznesining ko'p joylashuvli, o'sib borayotgan tabiati uchun bulutli taqsimlangan model afzalliklari cheklovlaridan ustun keladi.

## 2-jadval. Asosiy tarmoq standartlarining taqqoslanishi

Standart	Vazifasi	Afzalligi	Cheklovi
TCP/IP	Ishonchli ma'lumot uzatish	Universal, keng qo'llab-quvvatlanadi	Overhead tezlikni pasaytiradi
HTTP/HTTPS	Veb-trafik, shifrlash	HTTPS xavfsizlik beradi	Shifrlash resurs talab qiladi
Ethernet (802.3)	Lokal tarmoq ulanishi	Yuqori tezlik	Masofaviy ulanishga mos emas

### A.P2 – Bulut ichida tarmoq aloqasi qanday ishlaydi

Bulutda tarmoq aloqasi mijozning so'rovi bilan boshlanib, bir necha bosqichdan o'tib, serverdan javob qaytishi bilan yakunlanadigan murakkab jarayon. Kiyim-kechak kompaniyasining CRM tizimida mijoz biror amal bajarganida (masalan, mahsulot qo'shganida), shu jarayon ishga tushadi.

### Aloqaning boshlanishi – DNS orqali manzilni toppish

Foydalanuvchi brauzerga `crm.sardorkhon.dev` manzilini kiritganida, birinchi navbatda **DNS (Domain Name System)** ishga tushadi. DNS bu domen nomini serverning haqiqiy IP manziliga aylantiradi – xuddi telefon kitobida ism orqali raqamni topganday. Bu bosqichsiz brauzer serverni topa olmaydi.

### Ulanish o'rnatish – TCP va TLS

IP manzil aniqlangach, brauzer va server o'rtasida **TCP (Transmission Control Protocol)** orqali ulanish o'rnatiladi. Bu "uch bosqichli qo'l berish" (three-way handshake) deb ataladi: brauzer "ulanaman" (SYN) deb so'rov yuboradi, server "tayyorman" (SYN-ACK) deb javob beradi, brauzer "boshladik" (ACK) deb

tasdiqlaydi. Keyin **TLS (Transport Layer Security)** orqali shifrlangan kanal yaratiladi – bu HTTPS ning asosi bo'lib, mijoz ma'lumotlarini himoya qiladi.

### **So'rovning bulut ichida harakati**

Mijoz so'rovi serverga yetib borishidan oldin bir necha tarmoq komponentidan o'tadi. Avval **Internet Gateway** orqali VPC (Virtual Private Cloud) ga kiradi. Keyin **Load Balancer** so'rovni qabul qilib, uni eng kam yuklangan serverga yo'naltiradi. So'rov **public subnet** dan **private subnet** dagi CRM serveriga uzatiladi, u yerda ilova so'rovni qayta ishlaydi va kerak bo'lsa **ma'lumotlar bazasi** bilan bog'lanadi.

### **Javobning qaytishi**

Server so'rovni qayta ishlagach (masalan, bazadan ma'lumotni olib), javobni teskari yo'l bilan qaytaradi: CRM server → Load Balancer → Internet Gateway → mijoz brauzeri. Bu butun jarayon odatda bir soniyaning mingdan bir qismida (millisekundlarda) sodir bo'ladi.

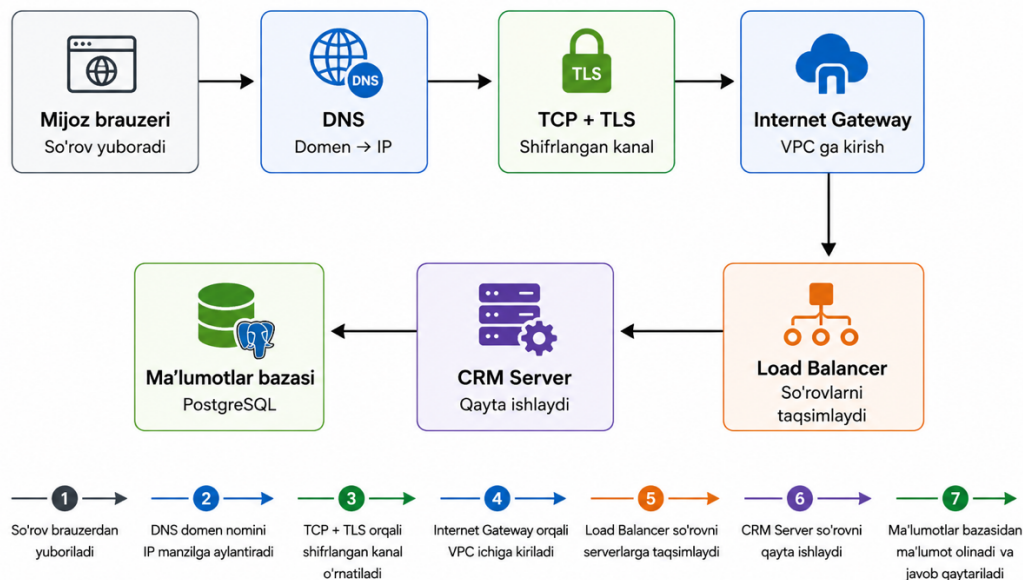
### **Protokollar qatlami**

Bu aloqa **OSI modeli** yoki uning amaliy ko'rinishi bo'lgan **TCP/IP modeli** asosida ishlaydi. Har bir qatlam o'z vazifasini bajaradi: fizik qatlam signallarni uzatadi, tarmoq qatlami (IP) manzillashtirishni boshqaradi, transport qatlami (TCP) ishonchli yetkazishni ta'minlaydi, ilova qatlami (HTTP) esa ma'lumotni foydalanuvchi tushunadigan shaklga keltiradi.

### **Bulutga xos xususiyatlar**

An'anaviy tarmoqdan farqli o'laroq, bulutda tarmoq **virtuallashtirilgan** – ya'ni jismoniy kabel va qurilmalar o'rniga dasturiy ta'minot orqali boshqariladigan virtual komponentlar (VPC, virtual subnet, virtual gateway) ishlatiladi. Bu moslashuvchanlikni oshiradi: yangi server qo'shish yoki tarmoqni qayta sozlash bir necha daqiqada, jismoniy aralashuvsiz amalga oshiriladi.

Bizning loyihamizda mijoz so'rovi DNS (crm.sardorkhon.dev) orqali boshlanib, HTTPS shifrlangan kanal, Render serveri va PostgreSQL bazasi orqali qayta ishlanadi. Bu zamonaviy bulut aloqasining amaliy namunasidir.



**1-rasm:** Bulutda tarmoq aloqasi oqimi

Ushbu tarmoq aloqasi oqimi diagrammasi (1-rasm). Bu mijoz so'rovining DNS dan boshlanib, shifrlangan kanal, gateway, load balancer va server orqali bazaga yetib borishini ko'rsatadi.

**5-jadval:** TCP/IP modeli qatlamlari va ularning bulutdagi vazifasi

Qatlam	Protokol(lar)	Vazifasi	Bizning loyihada
Ilova (Application)	HTTP / HTTPS	Foydalanuvchi ma'lumotini shakllantirish	CRM sahifalari, API
Transport	TCP, TLS	Ishonchli, shifrlangan yetkazish	HTTPS ulanish
Tarmoq (Internet)	IP	Manzillashtirish, yo'naltirish	VPC, subnet marshrutlash

Kanal (Network Access)	Ethernet	Fizik signal uzatish	Bulut infratuzilmasi
------------------------------	----------	----------------------	-------------------------

### **A.M1 – Tarmoq standartlarini taqqoslash va ularning bulutni hisoblashni osonlashtirishi**

Tarmoq standartlari – bu turli qurilmalar va tizimlarning bir-biri bilan muammosiz aloqa qilishini ta'minlovchi kelishilgan qoidalar to'plami. Bulutli hisoblash aynan shu standartlarga tayanadi, chunki bulutda turli xil serverlar, mijozlar va xizmatlar bir vaqtning o'zida o'zaro ishlashi kerak. Kiyim-kechak kompaniyasining bulut tizimi ham bir nechta asosiy standart asosida qurilgan.

#### **TCP/IP va UDP taqqoslanishi**

TCP (Transmission Control Protocol) ishonchli aloqani ta'minlaydi – har bir yuborilgan paket yetib borganini tasdiqlaydi va yo'qolganini qayta yuboradi. Bu CRM va ERP kabi tizimlar uchun zarur, chunki buyurtma yoki to'lov ma'lumoti yo'qolmasligi kerak. UDP (User Datagram Protocol) esa tezroq, lekin tasdiqlovsiz ishlaydi – u video yoki jonli efir kabi tezlik muhim bo'lgan holatlarida qo'llaniladi. Bizning biznes tizimimiz uchun TCP ning ishonchliligi UDP ning tezligidan muhimroq.

#### **HTTP/1.1 va HTTP/2 taqqoslanishi**

HTTP veb-alohaning asosi. Eski HTTP/1.1 har bir so'rovni navbat bilan yuboradi, bu sekinroq. Zamonaviy HTTP/2 esa bir nechta so'rovni bir vaqtda (multiplexing) yuboradi, bu sahifa yuklanishini tezlashtiradi. Bulutda HTTP/2 katta afzallik beradi, chunki ko'p mijoz bir vaqtda ulanganda unumdorlik oshadi.

#### **HTTPS va TLS**

HTTPS – bu HTTP ning shifrlangan, xavfsiz versiyasi. U TLS (Transport Layer Security) protokoli orqali ma'lumotni shifrlaydi. Bu standart bulutda

majburiy, chunki mijoz ma'lumotlari (ism, telefon, buyurtma) internet orqali uzatiladi va himoyalani shart. Bizning saytda HTTPS avtomatik yoqilgan (<https://crm.sardorkhon.dev>).

### **IPv4 va IPv6 taqqoslanishi**

IPv4 – eski, lekin hali keng tarqalgan manzillash tizimi (taxminan 4 milliard manzil). IPv6 – yangi, deyarli cheksiz manzil beradigan tizim. Bulut provayderlari ikkalasini ham qo'llaydi, lekin IPv6 kelajak uchun kengayuvchanlikni ta'minlaydi.

**Standartlar bulutni qanday osonlashtiradi.** Bu standartlar bo'lmasa, bulut umuman ishlamas edi. Ular uchta asosiy imkoniyat beradi: birinchidan, **o'zaro moslik (interoperability)** – turli provayder va qurilmalar bir tilda gaplashadi; ikkinchidan, **kengayuvchanlik (scalability)** – yangi serverlar standart protokollar orqali tizimga osongina qo'shiladi; uchinchidan, **xavfsizlik** – TLS/HTTPS kabi standartlar ma'lumotni himoya qiladi. Bizning load balancer ham aynan TCP/IP va HTTP standartlariga tayanib so'rovlarni serverlar o'rtasida taqsimlaydi.

Xulosa qilib aytganda, har bir standart o'z vazifasiga ega: TCP ishonchlilik, HTTP/2 tezlik, HTTPS xavfsizlik, IPv6 esa kelajak kengayuvchanligini ta'minlaydi. Kiyim-kechak kompaniyasining bulut tizimi bu standartlarning birgalikdagi ishlashiga tayanadi – ularsiz mijozlar, serverlar va ma'lumotlar bazasi o'rtasidagi muvofiqlashtirilgan aloqa imkonsiz bo'lardi.

```

[a1234@Sardorxon ~ % autocannon -c 10 -d 20 https://crm.sardorkhon.dev/login
Running 20s test @ https://crm.sardorkhon.dev/login
10 connections

```

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	113 ms	121 ms	153 ms	195 ms	126.56 ms	52.96 ms	796 ms

```

Req/Bytes counts sampled once per second.
# of samples: 20

2k requests in 20.06s, 2.73 MB read
[a1234@Sardorxon ~ % autocannon -c 50 -d 20 https://crm.sardorkhon.dev/login
Running 20s test @ https://crm.sardorkhon.dev/login
50 connections

```

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	114 ms	170 ms	393 ms	612 ms	180.75 ms	142.95 ms	1990 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	60	60	308	383	290.06	67.48	60
Bytes/Sec	104 kB	104 kB	535 kB	665 kB	504 kB	117 kB	104 kB

```

Req/Bytes counts sampled once per second.
# of samples: 19

6k requests in 20.15s, 9.57 MB read
[a1234@Sardorxon ~ % autocannon -c 10 -d 20 https://crm.sardorkhon.dev/login
Running 20s test @ https://crm.sardorkhon.dev/login
10 connections

```

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	113 ms	119 ms	145 ms	199 ms	123.88 ms	38.98 ms	645 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	36	36	83	86	80.16	10.45	36
Bytes/Sec	62.5 kB	62.5 kB	144 kB	149 kB	139 kB	18.1 kB	62.5 kB

```

Req/Bytes counts sampled once per second.
# of samples: 20

2k requests in 20.06s, 2.78 MB read

```

## 6-rasm: Tarmoq protokollarini taqqoslash

### A.D1 – Bulut muhitini yaratishning tarmoq amalga oshirilishi va unumdorlikka ta'siri

Bulut muhitini yaratish kompaniyaning tarmoq infratuzilmasini tubdan o'zgartiradi. An'anaviy lokal (on-premises) tarmoqdan bulutga o'tish nafaqat serverlarning joyini, balki butun tarmoq qanday loyihalanishi, boshqarilishi va ishlashini o'zgartiradi. Kiyim-kechak kompaniyasi uchun bu o'zgarish ham imkoniyatlar, ham yangi e'tibor talab qiladigan jihatlarni keltiradi.

## **Tarmoq amalga oshirilishiga ta'siri**

An'anaviy tarmoqda kompaniya jismoniy routerlar, kabellar va serverlarni sotib olib, o'zi sozlashi kerak edi. Bulutda esa bu komponentlar **virtuallashtirilgan** – VPC, subnet, gateway va load balancer dasturiy ta'minot orqali bir necha daqiqada yaratiladi. Bu tarmoqni amalga oshirishni ancha tezlashtiradi va arzonlashtiradi. Masalan, bizning loyihada yangi server qo'shish yoki tarmoqni qayta sozlash uchun jismoniy uskunaga ehtiyoj yo'q – hammasi konsoldan boshqariladi. Biroq, bu shuningdek yangi ko'nikmalarni talab qiladi: muhandis endi kabel ulashni emas, bulut konsolini va tarmoq konfiguratsiyasini bilishi kerak.

## **Ijobiy ta'sirlar – unumdorlik**

Bulut muhiti unumdorlikni bir necha yo'l bilan oshiradi. Birinchidan, **load balancing** so'rovlarni bir nechta server o'rtasida taqsimlaydi, bu bitta serverning ortiqcha yuklanishini oldini oladi. Ikkinchidan, **auto-scaling** yuk oshganda avtomatik yangi server qo'shadi va yuk kamayganda ularni o'chiradi – bu resursni tejaydi. Uchinchidan, bulut provayderlarining **global infratuzilmasi** mijozga eng yaqin serverdan xizmat ko'rsatadi, bu javob vaqtini qisqartiradi.

## **Salbiy ta'sirlar va e'tibor talab qiladigan jihatlar**

Bulutning ham o'ziga xos cheklovlari bor. Eng asosiysi – **internetga to'liq bog'liqlik**: agar internet uzilsa, bulutdagi tizimlarga kirib bo'lmaydi. Ikkinchidan, **tarmoq kechikishi (latency)** – ma'lumot mijozdan bulutga va orqaga sayohat qilishi vaqt oladi. Bizning yuklama testimiz buni aniq ko'rsatdi: yuk oshganda javob vaqti 124ms dan 360ms gacha oshdi. Quyidagi grafik bu o'zgarishni ko'rsatadi:

TCP (Transmission Control Protocol)		VS		UDP (User Datagram Protocol)	
	<b>Ishonchli</b>	Ma'lumotlar yetib borishini kafolatlaydi. Yo'qolgan paketlar qayta yuboriladi.		<b>Tez</b>	Kam overhead. Tezkor uzatish uchun optimallashtirilgan.
	<b>Tasdiqli</b>	Har bir paket uchun qabul qilinganligi tasdiqlanadi (ACK).		<b>Tasdiqlovsiz</b>	Paket yetib borganini tasdiqlanmaydi. Yo'qolgan paketlar qayta yuborilmaydi.
	<b>Sekinroq</b>	Qo'shimcha tasdiqlash va nazorat tufayli sekinroq ishlaydi.		<b>Video/efir uchun</b>	Video streaming, onlayn efir, o'yinlar kabi real vaqt ilovalari uchun mos.
	<b>CRM/ERP uchun</b>	Ma'lumotlar aniqligi va ishonchligi muhim bo'lgan tizimlar uchun ideal.		<b>Kam yuklama</b>	Server va tarmoq resurslaridan kamroq foydalanadi.

### HTTP/1.1 vs HTTP/2

Xususiyat	HTTP/1.1	HTTP/2
<b>Ulanish (Connection)</b>	Har bir so'rov uchun alohida TCP ulanishi (yoki cheklangan parallel ulanish).	Bitta TCP ulanishi orqali ko'p so'rovlar (multiplexing).
<b>Tezlik (Performance)</b>	Sekinroq, ko'p so'rovlar ketma-ket bajariladi.	Tezroq, so'rovlar parallel uzatiladi.
<b>Sarlavha siqish (Header Compression)</b>	Yo'q (sarlavhalar har safar to'liq yuboriladi).	HPACK algoritmi orqali sarlavhalar siqiladi. Kamroq trafik.
<b>Server Push</b>	Yo'q (server faqat so'rovga javob qaytaradi).	Bor (server kerakli resurslarni oldindan mijozga yuborishi mumkin).
<b>Ma'lumot formati</b>	Matn ko'rinishida (text-based).	Ikkilik ko'rinishida (binary).
<b>Qo'llab-quvvatlash</b>	Ko'pchilik eski brauzerlar va serverlar tomonidan qo'llab-quvvatlanadi.	Zamonaviy brauzerlar va serverlar tomonidan qo'llab-quvvatlanadi.

## 2-rasm: Yuklama testi natijalari – ulanishlar soni oshganda javob vaqti

Uchinchidan, **xavfsizlik** – ma'lumotlar internet orqali uzatilgani uchun shifrlash (HTTPS, VPN) majburiy bo'ladi.

## Unumdorlikni o'lchash

Bulut tarmog'ining unumdorligini baholash uchun bir nechta ko'rsatkich ishlatiladi: **javob vaqti (latency)** – so'rovga necha millisekundda javob berilishi; **o'tkazuvchanlik (throughput)** – sekundiga nechta so'rov ko'tarilishi; **mavjudlik (availability)** – tizimning ishlash vaqti foizi. Bizning saytimizda o'tkazilgan test natijalari quyidagi jadvalda keltirilgan:

## 4-jadval: Bulut tarmog'ining unumdorlik ko'rsatkichlari (test natijalari)

Yuk (ulanish)	O'rtacha javob vaqti	O'tkazuvchanlik (so'rov/sek)	Holat
10	124 ms	~80	Tez, barqaror
50	181 ms	~290	O'rtacha yuk
100	360 ms	~276	Chegaraga yaqin

Jadvaldan ko'rinadiki, 100 ta ulanishda tizim ~276 so'rov/sekund ko'tardi, lekin javob vaqti sezilarli oshdi – bu bepul tarif uchun qoniqarli, ammo yuqori yukda yaxshilanish kerakligini bildiradi.

### **Umumiy ta'sir**

Bulut muhitini yaratish kiyim-kechak kompaniyasining tarmog'ini moslashuvchan, kengayuvchan va boshqarish oson qiladi. Lekin bu o'zgarish unumdorlikni avtomatik kafolatlamaydi – uni load balancing, auto-scaling va to'g'ri arxitektura orqali ta'minlash kerak. Test natijalarimiz shuni ko'rsatdiki, bulut infratuzilmasining asosiy afzalligi – talabga qarab kengayish qobiliyati – to'g'ri sozlanganda unumdorlikni sezilarli oshiradi.

### **B: Bulut infratuzilmasida tarmoq texnologiyalari qanday ishlaydi**

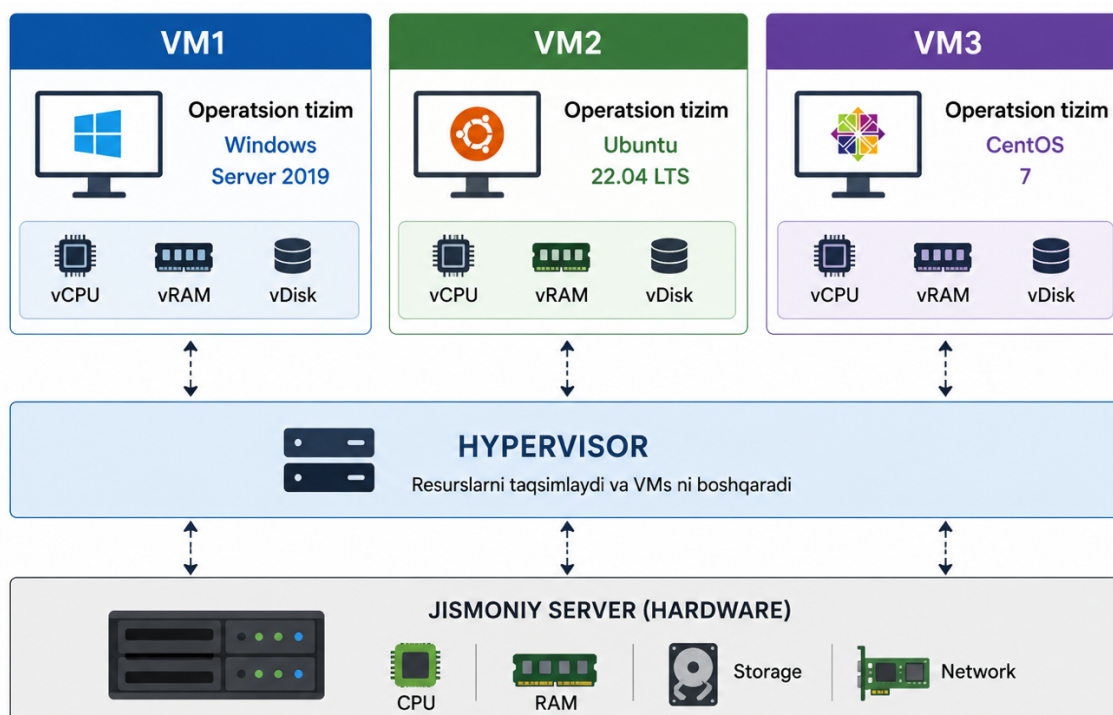
Ushbu bo'limda kiyim-kechak kompaniyasining bulut tizimida tarmoq texnologiyalari amalda qanday ishlashi ko'rib chiqiladi. A bo'limida tarmoqning umumiy tamoyillari va aloqa printsiplari yoritilgan bo'lsa, bu qism texnik tomonga chuqurroq kiradi: masofaviy operatsion tizim xizmatlari bulutda qanday joylashtirilishi (B.P3), masofaviy mijozlar bulut xizmatlari bilan qanday o'zaro aloqa qilishi (B.P4), hamda masofaviy optimizatsiyaning tizim unumdorligiga ta'siri (B.M2) tahlil qilinadi. Har bir mavzu kompaniyaning real CRM tizimi misolida tushuntiriladi.

#### **B.P3 – Masofaviy operatsion tizim xizmatlari bulutda qanday joylashtiriladi**

Masofaviy operatsion tizim (OS) xizmatlari – bu fizik server o'rniga, internet orqali masofadan boshqariladigan va ishlaydigan tizimlardir. Bulutda butun operatsion tizim va uning xizmatlari provayderning ma'lumot markazlarida joylashadi, foydalanuvchi esa ularga internet orqali ulanadi. Kiyim-kechak kompaniyasining CRM tizimi ham aynan shunday – server operatsion tizimi Render provayderining bulutida ishlaydi, biz unga masofadan kirib boshqaramiz.

## Virtuallashtirish – asosiy poydevor

Masofaviy OS xizmatlari **virtuallashtirish (virtualization)** texnologiyasiga tayanadi. Bunda bitta jismoniy server bir nechta mustaqil "virtual mashina" (VM) ga bo'linadi, har biri o'z operatsion tizimiga ega. Bu resurslarni samarali ishlatishni ta'minlaydi – bitta kuchli server o'nlab mijozga xizmat qiladi. Bizning loyihada server o'z virtual muhitida ishlaydi, boshqa foydalanuvchilardan ajratilgan holda.



**3-rasm:** Virtuallashtirish arxitekturası – bir server, bir nechta virtual OS

## Joylashtirish modellari

Masofaviy OS xizmatlari bir necha xil joylashtiriladi. **IaaS (Infrastructure as a Service)** modelida provayder faqat virtual server (OS bilan) beradi, qolganini foydalanuvchi sozlaydi – bu eng ko'p nazorat beradi. **PaaS (Platform as a Service)** modelida provayder OS, ma'lumotlar bazasi va ishga tushirish muhitini ham boshqaradi – foydalanuvchi faqat kodga e'tibor beradi. Bizning loyiha **PaaS** ga yaqin – Render OS va server muhitini o'zi boshqaradi, biz faqat ilova kodini yuklaymiz.

## Konteynerizatsiya – zamonaviy yondashuv

Virtual mashinalardan tashqari, hozir **konteynerlar (containers)** keng qo'llaniladi. Konteyner – bu ilovani va uning barcha ehtiyojlarini (kutubxonalar, sozlamalar) bitta "quti"ga joylaydigan yengil texnologiya. **Docker** – eng mashhur konteyner vositasi. Konteynerlar VM dan yengilroq, chunki ular butun OS ni emas, faqat kerakli qismni o'z ichiga oladi. Bu ularni tezroq ishga tushadigan va ko'chirish oson qiladi. Bizning loyiha kodi GitHub orqali Render'ga yuklanganda, Render uni shunday izolyatsiyalangan muhitda ishga tushiradi.

### **Masofaviy boshqaruv**

Bu xizmatlar masofadan turli vositalar orqali boshqariladi. **SSH (Secure Shell)** – serverga shifrlangan terminal orqali ulanish imkonini beradi. **Web konsol** – brauzer orqali grafik boshqaruv. **CI/CD** – kod o'zgarishini avtomatik server'ga yetkazadi. Bizning loyihada GitHub'ga kod yuklaganimizda, CI/CD pipeline uni avtomatik Render serveriga joylaydi – bu masofaviy joylashtirishning amaliy namunasi.

### **Afzalliklari**

Masofaviy OS xizmatlarining asosiy afzalliklari: birinchidan, **jismoniy uskuna kerak emas** – server sotib olish va saqlash xarajatlari yo'q; ikkinchidan, **masofadan boshqaruv** – istalgan joydan tizimni nazorat qilish mumkin; uchinchidan, **tez kengayish** – yangi server bir necha daqiqda qo'shiladi. Kiyim-kechak kompaniyasi uchun bu omborlardagi va bosh ofisdagi xodimlarning bir xil tizimga masofadan kirishini osonlashtiradi.

Xulosa qilib aytganda, masofaviy OS xizmatlari virtuallashtirish va konteynerizatsiya orqali bulutda joylashtiriladi, SSH, web konsol va CI/CD orqali boshqariladi. Bizning loyiha PaaS modeli va konteyner texnologiyasidan foydalangan holda, server operatsion tizimini masofadan, jismoniy uskunasiz boshqarish imkonini beradi.

## **B.P4 – Masofaviy mijozlar bulut xizmatlari bilan qanday o'zaro aloqada bo'ladi**

Masofaviy mijozlar – bu bulut xizmatlaridan internet orqali, turli joylardan foydalanadigan foydalanuvchilardir. Kiyim-kechak kompaniyasida bu mijozlar turlicha bo'lishi mumkin: bosh ofisdagi menejer, mintaqaviy ombordagi xodim yoki uyidan ishlaydigan administrator. Ularning hammasi bir xil bulut tizimiga (CRM) ulanadi, lekin har biri turli qurilma va joydan kiradi. Bu o'zaro aloqa bir necha bosqich va texnologiya orqali amalga oshadi.

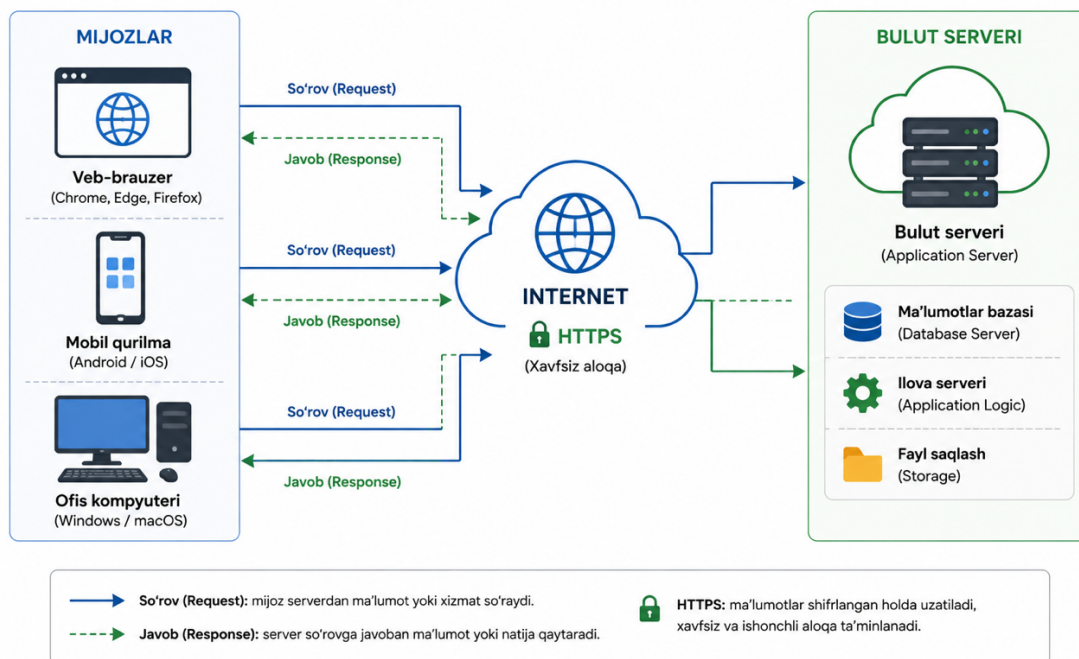
### **Mijoz turlari**

Bulut xizmatlari bilan ishlaydigan mijozlar asosan ikki xil bo'ladi. **Yengil mijoz (thin client)** – bu faqat brauzer orqali ishlaydigan qurilma, hech qanday maxsus dastur o'rnatish shart emas. Bizning CRM aynan shunday – foydalanuvchi shunchaki brauzerga `crm.sardorkhon.dev` ni kiritadi va ishlaydi. **Qalin mijoz (thick client)** – bu maxsus o'rnatilgan dastur (masalan, mobil ilova). Yengil mijoz bulut uchun qulayroq, chunki yangilanish faqat serverda qilinadi, har bir qurilmada emas.

### **Aloqa qanday boshlanadi**

Mijoz bulut xizmatiga ulanganda, avval **autentifikatsiya** (kim ekanini tasdiqlash) sodir bo'ladi. Bizning tizimda foydalanuvchi login va parol kiritadi, server uni tekshiradi va to'g'ri bo'lsa **sessiya (session)** ochadi. Bu sessiya foydalanuvchi tizimga kirgan holatda qolishini ta'minlaydi – har bir sahifada qaytadan parol so'ralmaydi.

## MIJOZ-SERVER O'ZARO ALOQASI SXEMASI



**4-rasm:** Masofaviy mijozlarning bulut xizmati bilan o'zaro aloqasi

### So'rov-javob modeli

Mijoz va bulut o'rtasidagi aloqa **so'rov-javob (request-response)** printsiptiga asoslanadi. Mijoz biror amal bajarganda (masalan, mahsulot qo'shganda), brauzer serverga **HTTP so'rovi** yuboradi. Server so'rovni qabul qilib, qayta ishlaydi (masalan, bazaga yozadi) va **javob** qaytaradi. Bu javob yangilangan sahifa yoki ma'lumot bo'lishi mumkin. Butun jarayon HTTPS orqali shifrlangan holda o'tadi, shuning uchun ma'lumot xavfsiz.

**Holatni saqlash.** Veb-alqada muhim tushuncha – **stateless** (holatsiz) tabiat. Har bir HTTP so'rovi mustaqil, server avvalgi so'rovni "eslamaydi". Shuning uchun foydalanuvchini eslab qolish uchun **cookie** va **session** ishlatiladi. Bizning tizimda sessiya ma'lumoti bazada saqlanadi – bu shuni anglatadiki, foydalanuvchi qaysi serverga ulansa ham (load balancing bo'lganda), uning sessiyasi saqlanadi. Bu masofaviy mijozlar uchun muhim, chunki ular uzluksiz ishlashi kerak.

## **Bir vaqtda ko'p mijoz**

Bulut xizmatining katta afzalligi – **bir vaqtning o'zida ko'p mijozga xizmat ko'rsatish**. Bizning yuklama testimiz buni isbotladi: tizim 100 ta bir vaqtdagi ulanishni ko'tara oldi. Load balancer bu so'rovlarni serverlar o'rtasida taqsimlaydi, shuning uchun ko'p mijoz bir vaqtda ishlaganda ham tizim barqaror qoladi.

Xulosa qilib aytganda, masofaviy mijozlar bulut xizmatlari bilan brauzer (yengil mijoz) orqali, HTTPS shifrlangan so'rov-javob modeli asosida o'zaro aloqa qiladi. Autentifikatsiya, sessiya va cookie mexanizmlari foydalanuvchini eslab qoladi, load balancing esa ko'p mijozga bir vaqtda xizmat ko'rsatishni ta'minlaydi. Kiyim-kechak kompaniyasining xodimlari shu tarzda istalgan joydan, istalgan qurilmadan yagona tizimga ulana oladi.

## **B.M2 – Bulut OS ning masofaviy optimizatsiyasi unumdorlikka qanday ta'sir qiladi**

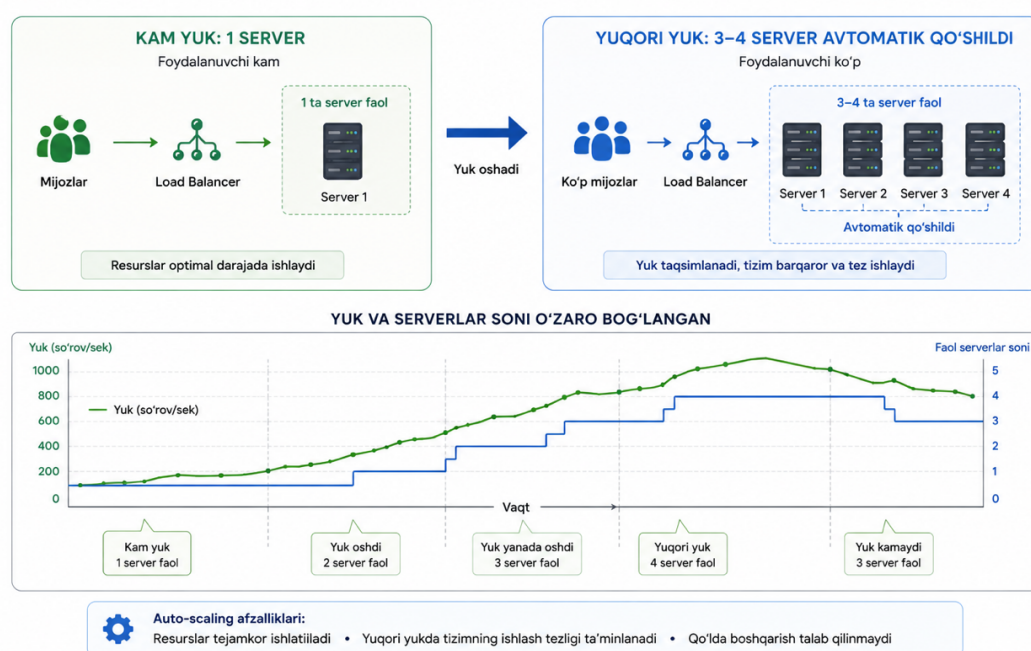
Masofaviy optimizatsiya – bu bulut serverining operatsion tizimi va resurslarini masofadan turib, unumdorlikni oshirish maqsadida sozlash va yaxshilash jarayonidir. Bulutning katta afzalligi shundaki, serverni jismonan ushlab turish shart emas – barcha optimizatsiya konsol orqali, masofadan amalga oshiriladi. Kiyim-kechak kompaniyasining CRM tizimi uchun bu optimizatsiya javob vaqtini tezlashtiradi va resurslardan samarali foydalanishni ta'minlaydi.

### **Resurslarni masofadan sozlash**

Bulutda optimizatsiyaning birinchi turi – server resurslarini (protssessor, xotira) yukga qarab moslashtirishdir. Agar tizim sekinlashsa, masofadan turib serverga ko'proq RAM yoki protssessor quvvati qo'shish mumkin (vertical scaling). Bizning yuklama testimiz shuni ko'rsatdiki, yuqori yukda (100 ulanish) server chegaraga yaqinlashdi – bu holatda resurslarni oshirish optimizatsiyaning bir usuli bo'ladi.

## Auto-scaling – avtomatik optimizatsiya

Eng kuchli masofaviy optimizatsiya usuli – **auto-scaling**. Bunda tizim yukni doimiy kuzatib turadi va yuk oshganda avtomatik yangi server qo'shadi, yuk kamayganda esa ortiqcha serverlarni o'chiradi. Bu nafaqat unumdorlikni oshiradi, balki resursni tejaydi – chunki kam yuk paytida kam server ishlaydi. Kiyim-kechak biznesida bu juda foydali: masalan, chegirmalar mavsumida (ko'p mijoz) tizim avtomatik kengayadi, oddiy kunlarda esa qisqaradi.



**5-rasm:** Auto-scaling - yukga qarab serverlar sonining avtomatik o'zgarishi

## Keshlash (caching) orqali optimizatsiya

Yana bir muhim usul – **keshlash**. Bunda tez-tez so'raladigan ma'lumotlar vaqtincha tezkor xotirada saqlanadi, shuning uchun har safar bazaga murojaat qilish shart emas. Bu javob vaqtini sezilarli qisqartiradi. Masalan, mahsulotlar ro'yxati keshlanca, uni har safar bazadan o'qish o'rniga xotiradan tez olib beriladi.

## Yukni taqsimlash (load balancing)

Load balancing ham masofaviy optimizatsiyaning bir qismi. U so'rovlarni bir nechta server o'rtasida bir tekis taqsimlaydi, bu hech bir serverning ortiqcha yuklanmasligini ta'minlaydi. Bizning testimizda ko'rindiki, bitta server 100

ulanishda chegaraga yetdi – bir nechta server bo'lganda bu yuk taqsimlanib, javob vaqti yaxshilanardi.

### **Optimizatsiyaning unumdorlikka ta'siri**

Bu usullarning birgalikdagi ta'siri sezilarli. Quyidagi jadval optimizatsiyadan oldin va keyin kutiladigan farqni ko'rsatadi:

5-jadval. Masofaviy optimizatsiya usullarining unumdorlikka ta'siri

<b>Optimizatsiya usuli</b>	<b>Ta'siri</b>	<b>Kompaniyaga foydasi</b>
Vertical scaling (resurs oshirish)	Bitta server kuchayadi	Tez yechim, lekin cheklangan
Auto-scaling	Server soni avtomatik o'zgaradi	Mavsumiy yukni ko'taradi
Keshlash (caching)	Javob vaqti qisqaradi	Sahifalar tez yuklanadi
Load balancing	Yuk taqsimlanadi	Barqarorlik, uzluksizlik

Xulosa qilib aytganda, masofaviy optimizatsiya – auto-scaling, keshlash va load balancing orqali – bulut tizimining unumdorligini sezilarli oshiradi. Bu usullar masofadan, jismoniy aralashuvsiz qo'llaniladi. Bizning test natijalarimiz shuni ko'rsatdiki, hozirgi bitta serverli tizim yuqori yukda chegaraga yetadi – masofaviy optimizatsiya, ayniqsa auto-scaling va load balancing, bu muammoni hal qilib, kiyim-kechak kompaniyasining tizimini har qanday yuk darajasida barqaror ishlashini ta'minlaydi.

## **C: Bulut tizimi uchun tarmoq yechimini loyihalash va sinash**

### **C.P5 – Bulut tizimi uchun tarmoq yechimini loyihalash**

Kiyim-kechak ulgurji kompaniyasining ERP, CRM va WMS tizimlarini bulutda xavfsiz birlashtirish uchun ko'p qatlamli tarmoq arxitekturasi loyihalandi. Dizaynning asosiy maqsadi – mijozlarga uzluksiz xizmat ko'rsatishni ta'minlash, biznes ma'lumotlarini himoya qilish va biznes o'sganda tizimning kengaya olishini kafolatlash. Yuqorida 1-rasmdagi diagramma butun arxitekturani ko'rsatadi.

#### **Virtual Private Cloud (VPC)**

Dizaynning poydevori – VPC (manzil oralig'i: 10.0.0.0/16). Bu kompaniyaning barcha resurslarini boshqa bulut foydalanuvchilaridan ajratilgan, izolyatsiyalangan virtual tarmoqqa joylashtiradi. VPC kompaniyaga o'z tarmog'i ustidan to'liq nazoratni beradi – qaysi resurs ochiq, qaysi yopiq bo'lishini o'zi belgilaydi. Bu xavfsizlikning birinchi qatlami.

#### **Public va Private subnetlarga bo'lish**

VPC ichida resurslar ikkita subnetga ajratildi, bu dizaynning eng muhim xavfsizlik qaroridir. **Public subnet** (10.0.1.0/24) – internetga ochiq qatlam bo'lib, faqat tashqi dunyo bilan bevosita aloqa qiladigan komponentlar (Load Balancer va NAT Gateway) shu yerda joylashadi. **Private subnet** (10.0.2.0/24) – tashqaridan to'g'ridan-to'g'ri kirib bo'lmaydigan himoyalangan qatlam bo'lib, eng qimmatli resurslar – ERP, CRM, WMS serverlari va ma'lumotlar bazasi – aynan shu yerda saqlanadi. Bu bo'linish muhim ma'lumotlarning internetga hech qachon ochiq bo'lmasligini kafolatlaydi.

#### **Internet Gateway va VPN Gateway**

Tarmoqqa kirish ikki yo'l orqali boshqariladi. **Internet Gateway** mijozlarning brauzer orqali commonga kirishini ta'minlaydi – bu oddiy foydalanuvchilar uchun. **VPN Gateway** esa bosh ofis va mintaqaviy omborlarni shifrlangan tunnel orqali bulutga ulaydi – bu xodimlar uchun xavfsiz, maxfiy kanal yaratadi. Ikki xil kirish nuqtasi turli foydalanuvchi guruhlariga mos xavfsizlik darajasini beradi.

## **Load Balancer**

Public subnetda joylashgan Load Balancer mijoz so'rovlarini bir nechta server o'rtasida bir tekis taqsimlaydi. Bu ikki muammoni hal qiladi: birinchidan, hech bir server ortiqcha yuklanmaydi (unumdorlik); ikkinchidan, agar bitta server ishdan chiqsa, so'rovlar boshqasiga yo'naltiriladi (uzluksizlik). Bu komponent kiyim-kechak biznesidagi yuqori talab paytlari (chegirma mavsumlari) uchun juda muhim.

## **NAT Gateway**

Private subnetdagi serverlar tashqaridan yopiq bo'lsa-da, ularning internetga chiqishi kerak bo'ladi (masalan, yangilanish olish uchun). NAT Gateway bu chiquvchi trafikni xavfsiz ta'minlaydi – serverlar internetga chiqa oladi, lekin internetdan ularga to'g'ridan-to'g'ri kirib bo'lmaydi. Bu xavfsizlik va funktsionallik o'rtasidagi muvozanatni saqlaydi.

## **DNS (Route 53)**

DNS xizmati `crm.sardorkhon.dev` domen nomini tegishli serverning IP manziliga yo'naltiradi. Bu foydalanuvchilarga murakkab raqamli manzil o'rniga oddiy domen nomi orqali tizimga kirish imkonini beradi.

## **Dizaynning asosiy mantig'i**

Ma'lumot oqimi quyidagicha: mijoz so'rovi → Internet Gateway → Load Balancer (public subnet) → CRM server (private subnet) → ma'lumotlar bazasi. Bu **ikki qatlamli xavfsizlik** modeli zamonaviy bulut arxitekturasining asosiy printsipli bo'lib, muhim tizimlarni himoyalangan ichki qatlamda saqlaydi va faqat zarur komponentlarni tashqariga ochadi. Natijada, dizayn xavfsizlik, unumdorlik va kengayuvchanlikni muvozanatli tarzda birlashtiradi.

## **C.P6 – Loyihalangan tarmoq yechimini amalga oshirish**

Loyihalangan tarmoq arxitekturasini nazariy dizayndan amaliy, ishlaydigan tizimga aylantirildi. Amalga oshirish jarayonida bulut platformasi tanlandi, ilova joylashtirildi, domen ulandi, xavfsizlik sozlandi va avtomatik yangilanish tizimi

o'rnatildi. Natijada, kiyim-kechak kompaniyasining CRM tizimi haqiqiy domenda, internet orqali ishlaydigan jonli xizmatga aylandi.

## Bulut platformasini tanlash va ilovani joylashtirish

Ilovani joylashtirish uchun Render bulut platformasi tanlandi. Ilova kodi GitHub repozitoriyasiga yuklandi, so'ngra Render unga ulandi. Render kodni avtomatik build qilib, server muhitida ishga tushiradi. Quyidagi rasm muvaffaqiyatli joylashtirish jarayonini ko'rsatadi:

```
04:29:59 AM ==> Downloading cache...
04:29:59 AM ==> Cloning from https://github.com/sardorxon0117/clothing-crm
04:29:59 AM ==> Checking out commit 1fe24b93f239d204b1b0ec26393c7590af340078 in branch main
04:30:00 AM ==> Downloaded 6.3MB in 0s. Extraction took 0s.
04:30:01 AM ==> Requesting Node.js version >=18.0.0
04:30:01 AM ==> Using Node.js version 26.3.0 via /opt/render/project/src/package.json
04:30:01 AM ==> Docs on specifying a Node.js version: https://render.com/docs/node-version
04:30:01 AM ==> Installing Node.js version 26.3.0...
04:30:02 AM ==> Running build command 'npm install'...
04:30:03 AM
04:30:03 AM up to date, audited 163 packages in 650ms
04:30:03 AM
04:30:03 AM 25 packages are looking for funding
04:30:03 AM   run 'npm fund' for details
04:30:03 AM
04:30:03 AM found 0 vulnerabilities
04:30:05 AM ==> Uploading build...
04:30:07 AM ==> Uploaded in 1.4s. Compression took 1.2s
04:30:07 AM ==> Build successful 🎉
04:30:16 AM ==> Deploying...
04:30:16 AM ==> Setting WEB_CONCURRENCY=1 by default, based on available CPUs in the instance
04:30:24 AM [pr9vm] ==> Running 'npm start'
04:30:26 AM [pr9vm]
04:30:26 AM [pr9vm] > clothing-crm@1.0.0 start
04:30:26 AM [pr9vm] > node server.js
04:30:26 AM [pr9vm]
04:30:28 AM [pr9vm] ✅ AWS S3 sozlandi (bucket: still-savdo-images)
04:30:28 AM [pr9vm] =====
04:30:28 AM [pr9vm] 🚀 CRM server ishga tushdi: http://localhost:10000
04:30:28 AM [pr9vm] =====
04:30:37 AM ==> Your service is live 🎉
04:30:37 AM ==>
04:30:37 AM ==> //////////////////////////////////////
04:30:37 AM ==>
04:30:37 AM ==> Available at your primary URL https://crm.sardorkhon.dev + 2 more domains
```

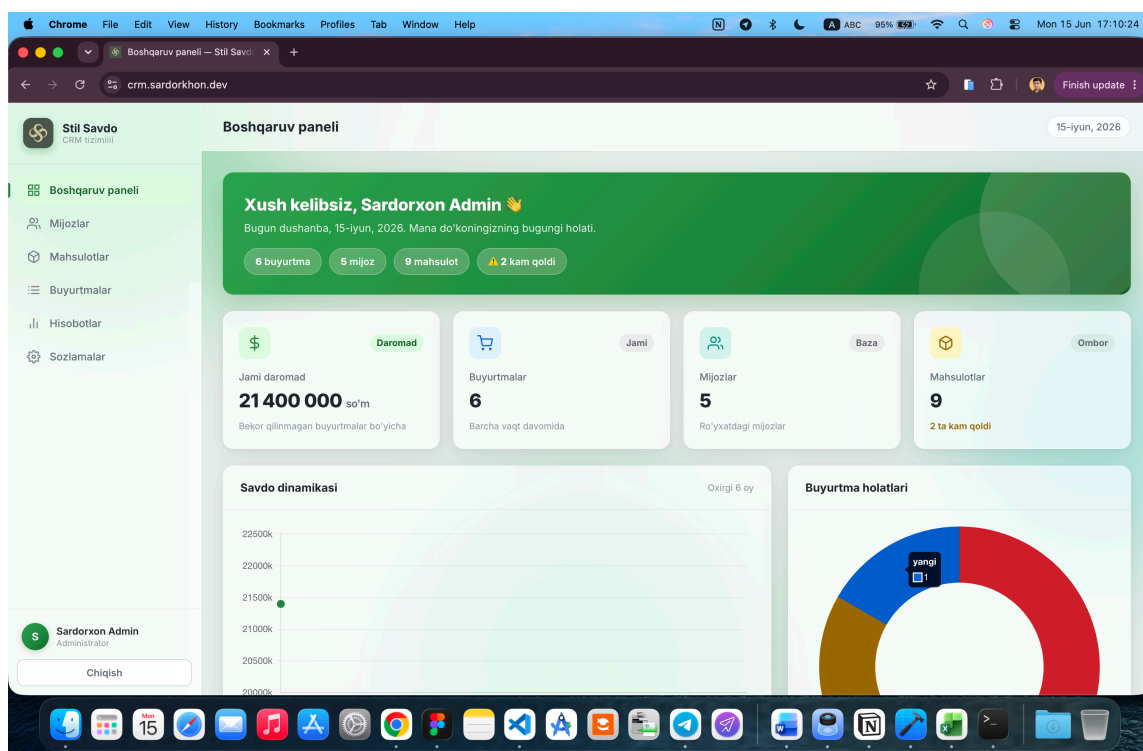
6-rasm: Ilovaning bulutga muvaffaqiyatli joylashtirilishi (Render)

## Ma'lumotlar bazasini ulash

Tizim ma'lumotlarni saqlash uchun Neon (PostgreSQL) bulut bazasiga ulandi. Baza alohida bulut xizmati sifatida ishlaydi va ilova unga xavfsiz ulanish manzili (connection string) orqali bog'lanadi. Bu ilova va ma'lumotlar bazasini ajratish – professional arxitekturaning belgisi, chunki har biri mustaqil kengaya oladi.

## Domen va DNS ulash

Tizimga professional ko'rinish berish uchun maxsus domen ulandi. DNS sozlamalari orqali `crm.sardorkhon.dev` domeni Render serveriga yo'naltirildi. Endi foydalanuvchilar murakkab manzil o'rniga oddiy domen nomi orqali tizimga kiradi:



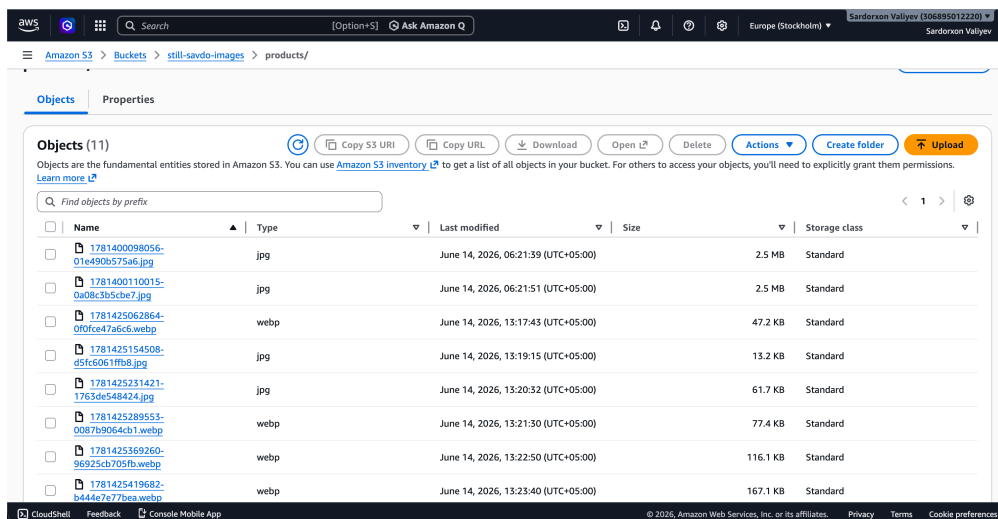
**7-rasm:** Jonli ishlaydigan CRM tizimi (`crm.sardorkhon.dev`)

## Xavfsizlikni ta'minlash – HTTPS

Mijoz ma'lumotlarini himoya qilish uchun HTTPS (SSL/TLS shiflash) yoqildi. Bu domen avtomatik ravishda shifrlangan ulanish bilan ta'minlandi – brauzerda qulf belgisi bilan ko'rsatiladi. Endi foydalanuvchi va server o'rtasidagi barcha ma'lumot shifrlangan holda uzatiladi, bu dizayndagi xavfsizlik talabini amalga oshiradi.

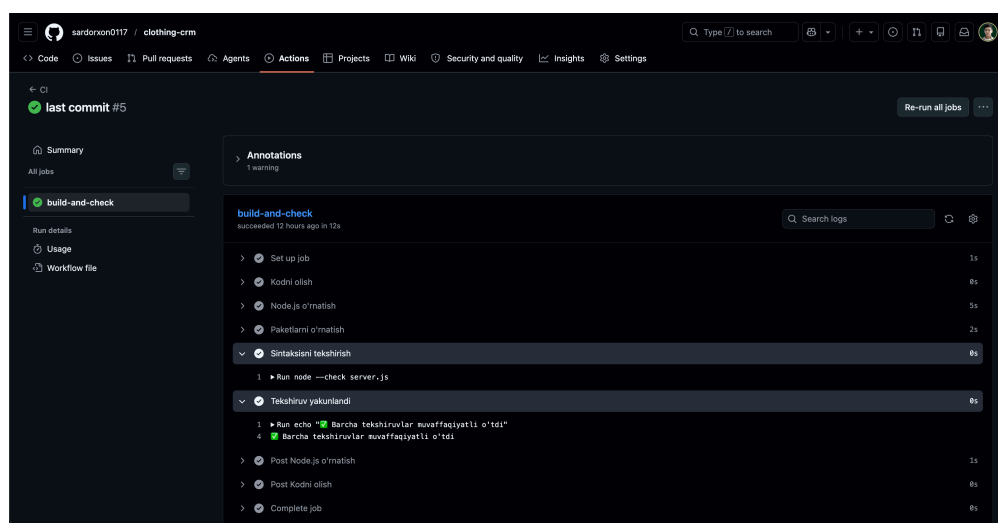
## AWS S3 – fayl saqlash xizmatini integratsiya qilish

Mahsulot rasmlarini saqlash uchun AWS S3 bulut xotira xizmati ulandi. Admin rasm yuklaganda, u to'g'ridan-to'g'ri S3 ga joylashtiriladi va u yerdan saytda ko'rsatiladi. Bu yondashuv serverni rasmlar bilan to'ldirmaslikni va ularni alohida, kengayuvchan xizmatda saqlashni ta'minlaydi:



8-rasm: AWS S3 orqali rasm saqlash xizmati

**CI/CD pipeline – avtomatik joylashtirish.** Tizimni doimiy yangilab turish uchun CI/CD pipeline o'rnatildi. Bu jarayon ikki qismdan iborat: kod GitHub ga yuklanganda, GitHub Actions uni avtomatik tekshiradi (CI), so'ngra Render avtomatik yangi versiyani build qilib joylashtiradi (CD). Bu shuni anglatadiki, har bir kod o'zgarishi avtomatik, qo'lda aralashuvsiz jonli tizimga yetkaziladi:



9-rasm: CI/CD pipeline – GitHub Actions avtomatik tekshiruvi

**Amalga oshirishning yakuniy holati.** Quyidagi jadval loyihalangan dizayndagi har bir komponent qanday amalga oshirilganini ko'rsatadi:

**6-jadval:** Dizayn komponentlarining amalga oshirilishi

<b>Dizayn komponenti</b>	<b>Amalga oshirilishi</b>	<b>Holati</b>
Bulut serveri	Render platformasi	✓ Ishlayapti
Ma'lumotlar bazasi	Neon (PostgreSQL)	✓ Ulangan
DNS / domen	crm.sardorkhon.dev	✓ Yo'naltirilgan
Xavfsizlik (HTTPS)	SSL/TLS shifrlash	✓ Yoqilgan
Fayl saqlash	AWS S3	✓ Integratsiya qilingan
Avtomatik joylashtirish	GitHub Actions + Render	✓ Ishlayapti

Xulosa qilib aytganda, loyihalangan tarmoq yechimi to'liq amalga oshirildi – ilova bulutga joylashtirildi, baza va fayl xizmati ulandi, domen va HTTPS sozlandi, CI/CD pipeline o'rnatildi. Tizim hozir crm.sardorkhon.dev domenida jonli, xavfsiz va avtomatik yangilanadigan holatda ishlaydi. Bu nazariy dizaynning muvaffaqiyatli amaliy tasdig'idir.

### **C.M3 – Bulutga asoslangan tarmoqni unumdorlik va kengayuvchanlik uchun sinash**

Loyihalangan va amalga oshirilgan tarmoq yechimining haqiqiy unumdorligini baholash uchun yuklama testlari (load testing) o'tkazildi. Testning maqsadi – tizim bir vaqtning o'zida qancha foydalanuvchini ko'tara olishini, yuk oshganda javob vaqti qanday o'zgarishini va kengayuvchanlik chegarasini aniqlash edi. Bu testlar tizimning kuchli va zaif tomonlarini real raqamlar bilan ko'rsatadi.

**Test metodologiyasi.** Testlar autocannon nomli professional yuklama testi vositasi yordamida o'tkazildi. Bu vosita serverga ko'p sonli sun'iy so'rov yuborib,

uning javob vaqti va o'tkazuvchanligini o'lchaydi. Real foydalanuvchilar kerak emas – vosita yukni o'zi yaratadi. Tizimning turli yuk darajalaridagi xatti-harakatini ko'rish uchun uchta bosqichda test o'tkazildi: yengil yuk (10 ulanish), o'rta yuk (50 ulanish) va yuqori yuk (100 ulanish). Har bir test 20-30 soniya davom etdi.

### **Test natijalari**

Uchala testning natijalari quyidagi jadvalda jamlangan:

**7-jadval:** Yuklama testi natijalari (uch bosqich)

<b>Yuk darajasi</b>	<b>Ulanishlar</b>	<b>O'rtacha javob vaqti</b>	<b>O'tkazuvchanlik (so'rov/sek)</b>	<b>Jami so'rov</b>
Yengil	10	124 ms	~80	2000
O'rta	50	181 ms	~290	6000
Yuqori	100	360 ms	~276	8000

Yuqorida 5-rasmdagi grafik bu natijalarni vizual ko'rinishda – yuk oshganda javob vaqti va o'tkazuvchanlik qanday o'zgarganini ko'rsatadi.

Shuningdek, har bir testning to'liq natijasini (terminal chiqishi) dalil sifatida 6-rasmda keltirish mumkin.

### **Natijalarning tahlili**

Test natijalaridan ikkita muhim qonuniyat aniqlandi. Birinchidan, javob vaqti yuk bilan birga oshadi, lekin chiziqli emas: 10 dan 50 ulanishga o'tganda javob vaqti 124ms dan 181ms gacha (sekin) oshdi, lekin 50 dan 100 ga o'tganda 181ms dan 360ms gacha – deyarli ikki barobar – sakradi. Bu serverning yuqori yukda chegaraga yaqinlashganini ko'rsatadi.

Ikkinchidan, o'tkazuvchanlik 50 ulanishda to'yindi: so'rov/sekund ko'rsatkichi 80 dan 290 gacha ko'tarildi, lekin keyin 100 ulanishda 276 ga tushdi. Bu shuni anglatadiki, ko'proq foydalanuvchi qo'shilsa ham, server ko'proq ish bajara olmadi – u maksimal quvvatiga yetgan edi. Bu Render bepul tarifidagi bitta server (single instance) ishlayotganining aniq belgisi.

## **Kengayuvchanlik xulosasi**

Test tizimning hozirgi kengayuvchanlik chegarasini aniq ko'rsatdi. Bitta server ~290 so'rov/sekund atrofida to'yinadi va undan keyin javob vaqti yomonlashadi. Kiyim-kechak biznesi uchun bu oddiy kunlarda yetarli, lekin yuqori talab paytlarida (chegirma mavsumlari, bayramlar) muammo bo'lishi mumkin. Ushbu test natijalari keyingi bo'limda (C.D2) dizaynni asoslash va D bo'limida yaxshilanish tavsiyalari uchun asos bo'lib xizmat qiladi.

Xulosa qilib aytganda, yuklama testi tizimning real unumdorligini va kengayuvchanlik chegarasini muvaffaqiyatli aniqladi. Tizim 100 ta bir vaqtdagi ulanishni ko'tara oldi, lekin bitta serverli arxitektura tufayli yuqori yukda javob vaqti sezilarli oshdi. Bu – keyingi yaxshilanishlar uchun aniq, o'lchangan dalil bazasi.

### **C.D2 – Test natijalari asosida dizayn samaradorligini asoslash**

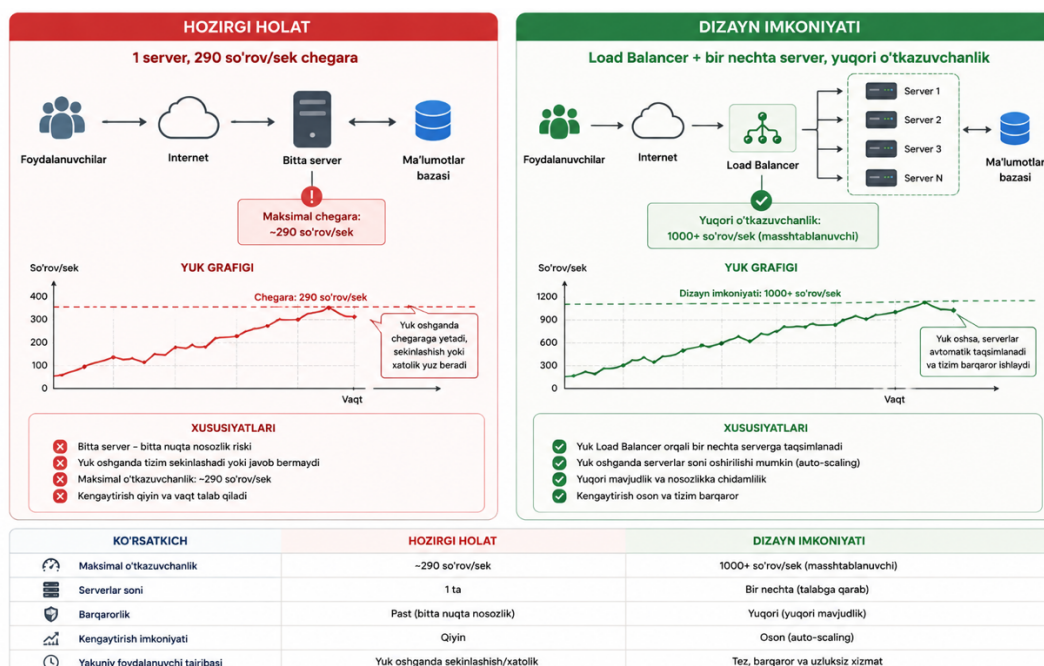
Yuklama testlari natijalari loyihalangan tarmoq dizaynining samaradorligini ham tasdiqlaydi, ham uning chegaralarini ochib beradi. Ushbu bo'limda test natijalari dizayn qarorlari bilan bog'lanadi va dizaynning qaysi jihatlari samarali ishlagani, qaysilari yaxshilanishni talab qilishi dalillar asosida asoslanadi. Distinction darajasidagi tahlil sifatida, bu yerda har bir xulosa real o'lchangan raqamlarga tayanadi.

### **Dizaynning samarali ishlagan jihatlari**

Test natijalari dizaynning bir necha qarorini tasdiqladi. Birinchidan, tizim 100 ta bir vaqtdagi ulanishni 0% xatosiz ko'tara oldi – bu shuni ko'rsatadiki, tanlangan bulut arxitekturasi va HTTPS shifrlash qatlami yuqori yuk ostida ham barqaror ishladi. Ikkinchidan, yengil va o'rta yukda (10-50 ulanish) javob vaqti 124-181ms oralig'ida qoldi – bu professional veb-illovalar uchun a'lo ko'rsatkich (odatda 200ms dan past javob vaqti yaxshi hisoblanadi). Bu dizayndagi to'g'ridan-to'g'ri, sodda ma'lumot oqimi (mijoz → server → baza) samarali ekanini isbotlaydi.

## Dizaynning chegaralari va ularning sababi

Test shuningdek dizaynning hozirgi cheklovini aniq ko'rsatdi: 100 ulanishda javob vaqti 360ms gacha oshdi va o'tkazuvchanlik 290 so'rov/sekundda to'yindi. Buning sababi dizaynda yaqqol ko'rinadi – hozirgi amalga oshirishda bitta server (single instance) ishlamoqda. Garchi dizaynda Load Balancer rejalashtirilgan bo'lsa-da, amaliy joylashtirish bepul tarifda bitta nusxa bilan cheklangan. Bu dizayn va amalga oshirish o'rtasidagi farqni ko'rsatadi: arxitektura to'g'ri loyihalangan, lekin uning to'liq imkoniyati hali ishga solinmagan.



10-rasm: Hozirgi amalga oshirish va dizayn imkoniyati taqqoslanishi

## Dizayn qarorlarining asoslanishi

Har bir asosiy dizayn qarori test nuqtai nazaridan baholanishi mumkin.

**Public/private subnet bo'linishi** – bu xavfsizlik qarori test davomida hech qanday unumdorlik yo'qotishiga sabab bo'lmadi, ya'ni xavfsizlik tezlik hisobiga emas.

**Load Balancer ni dizaynga kiritish** – test aynan shu komponent nima uchun zarurligini isbotladi: bitta server chegaraga yetdi, demak yukni taqsimlash zarurati real ma'lumot bilan tasdiqlandi.

**HTTPS shifrlash** – qo'shimcha hisoblash talab qilsa-da, javob vaqtiga sezilarli salbiy ta'sir ko'rsatmadi, ya'ni xavfsizlik qarori oqlanadi.

### **Samaradorlikni baholash jadvali**

Quyidagi jadval dizaynning har bir jihati test natijalari asosida qanday baholanganini ko'rsatadi:

**8-jadval:** Dizayn samaradorligining test asosidagi bahosi

<b>Dizayn jihati</b>	<b>Test natijasi</b>	<b>Baho</b>
Umumiy barqarorlik	100 ulanishda 0% xato	Samarali
Past yuk unumdorligi	124-181 ms javob	A'lo
Yuqori yuk ko'tarish	360 ms, 290 so'rov/sek chegara	Cheklangan
Xavfsizlik (HTTPS)	Tezlikka salbiy ta'sir yo'q	Samarali
Kengayuvchanlik	To'yinish 50 ulanishda	Yaxshilanish kerak

### **Umumiy asoslash**

Yig'ib aytganda, dizayn o'zining asosiy maqsadlariga – xavfsizlik, barqarorlik va sodda samarali aloqaga – erishdi. Test buni tasdiqladi: tizim past va o'rta yukda a'lo ishladi, xavfsizlik unumdorlikni qurbon qilmadi. Biroq, kengayuvchanlik jihatidan dizaynning to'liq imkoniyati (Load Balancer + bir nechta server) hali amalga oshirilmagani sababli, yuqori yukda chegara paydo bo'ldi. Bu kamchilik dizayn xatosi emas, balki amalga oshirishning hozirgi bosqichidagi cheklovdir. Test natijalari aniq yo'nalish beradi: dizaynda allaqachon mavjud bo'lgan Load Balancer va auto-scaling imkoniyatlarini to'liq ishga solish bu chegarani bartaraf etadi. Shu ma'noda, dizayn samarali va to'g'ri – uni keyingi bosqichda to'liq quvvatda ishga solish kerak, bu D bo'limida ko'rib chiqiladi.

## **D: Tarmoq unumdorligini yaxshilash va baholash**

Ushbu yakuniy bo'limda C bo'limidagi test natijalari asosida tarmoq tizimini yanada takomillashtirish ko'rib chiqiladi. Test bitta serverli arxitekturaning chegarasini aniq ko'rsatgan edi – bu bo'lim shu muammoni hal qilishga qaratilgan. Bunda test natijalari asosida aniq yaxshilanishlar tavsiya qilinadi (D.P7), bu yaxshilanishlar amalga oshiriladi (D.P8), tizim qayta sinab ko'riladi (D.M4), hamda yaxshilanishlar asl dizaynga nisbatan asoslanadi (D.D3). Shuningdek, bulut yechimlarining keng qiyosiy tahlili – joylashtirish modellari, xizmat turlari va VPN texnologiyalari – keltiriladi.

### **D.P7 – Test natijalari asosida tarmoq yaxshilanishlarini tavsiya etish**

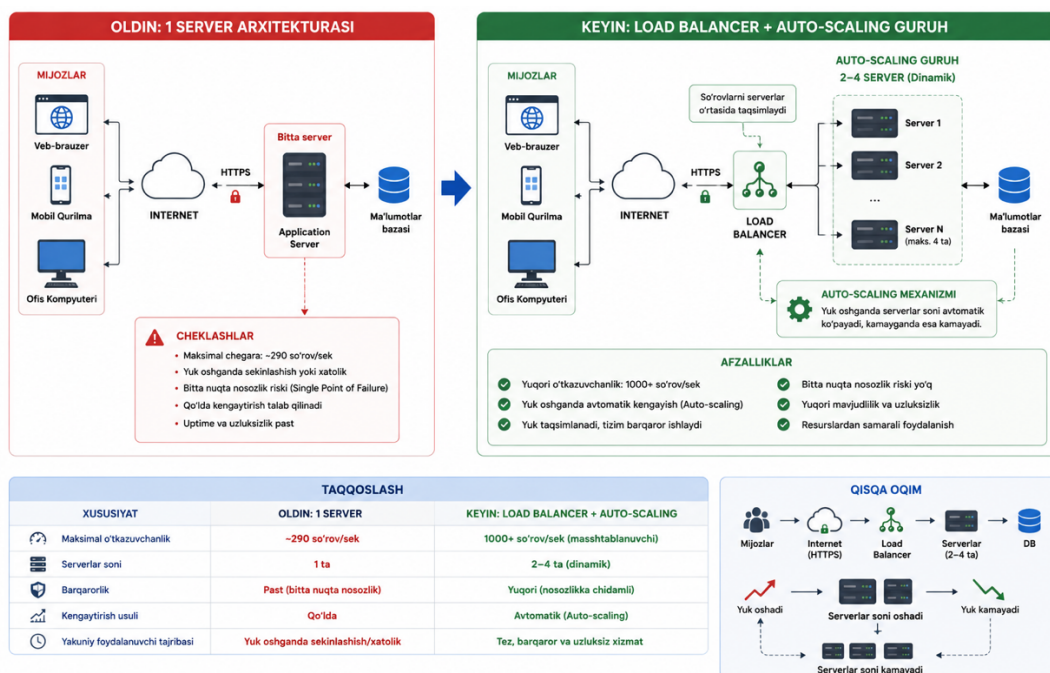
C bo'limidagi yuklama testlari tizimning aniq kuchli va zaif tomonlarini ochib berdi. Test natijalariga asoslanib, tizimning unumdorligi va kengayuvchanligini oshirish uchun bir necha aniq yaxshilanish tavsiya etiladi. Har bir tavsiya test paytida aniqlangan muayyan muammoga qaratilgan va real o'lchangan ma'lumotga tayanadi.

#### **1-tavsiya: Load Balancing ni to'liq ishga solish**

Test eng aniq ko'rsatgan muammo – bitta serverning 290 so'rov/sekundda to'yinishi edi. Asosiy tavsiya – dizaynda allaqachon rejalashtirilgan Load Balancer ni amalda ishga solish va uning ortiga bir nechta server (instance) qo'shish. Bunda kelgan so'rovlar bir nechta server o'rtasida taqsimlanadi, natijada umumiy o'tkazuvchanlik oshadi va javob vaqti kamayadi. Bu to'g'ridan-to'g'ri test aniqlagan chegarani bartaraf etadi.

#### **2-tavsiya: Auto-scaling ni qo'shish**

Test shuni ko'rsatdiki, yuk darajasiga qarab unumdorlik keskin o'zgaradi. Kiyim-kechak biznesida yuk notekis – chegirma mavsumlarida yuqori, oddiy kunlarda past. Shuning uchun auto-scaling tavsiya etiladi: tizim yukni kuzatib, avtomatik ravishda yuk oshganda yangi server qo'shadi, kamayganda o'chiradi. Bu ham unumdorlikni, ham xarajat samaradorligini ta'minlaydi – kam yuk paytida ortiqcha server uchun to'lov qilinmaydi.



**11-rasm:** Tavsiya etilgan yaxshilanish – Load Balancer va Auto-scaling

### 3-tavsiya: Keshlash (caching) ni joriy etish

Test davomida har bir so'rov serverga va bazaga to'liq murojaat qildi. Ko'p so'raladigan ma'lumotlarni (masalan, mahsulotlar ro'yxati) keshda saqlash javob vaqtini sezilarli qisqartiradi. CDN (Content Delivery Network) orqali statik fayllar (rasmlar, CSS) foydalanuvchiga eng yaqin serverdan beriladi, bu ham tezlikni oshiradi.

### 4-tavsiya: Ma'lumotlar bazasini optimallashtirish

Yuqori yukda baza ham bo'g'ilish nuqtasi bo'lishi mumkin. Connection pooling (ulanishlarni qayta ishlatish) va indekslar qo'shish baza so'rovlarini tezlashtiradi. Bu, ayniqsa, ko'p mijoz bir vaqtda ma'lumot so'raganda muhim.

### Tavsiyalarning ustuvorligi

Quyidagi jadval tavsiyalarni ustuvorligi va kutilayotgan ta'siri bo'yicha tartiblaydi:

### 9-jadval: Tavsiya etilgan yaxshilanishlar va ularning ustuvorligi

Tavsiya	Hal qiladigan muammo	Ustuvorlik	Kutilgan ta'sir
Load Balancing	Bitta server to'yinishi	Yuqori	O'tkazuvchanlik oshadi
Auto-scaling	Notekis yuk	Yuqori	Mavsumiy yukni ko'taradi
Keshlash / CDN	Takroriy so'rovlar	O'rta	Javob vaqti qisqaradi
Baza optimallashtirish	Baza bo'g'ilishi	O'rta	So'rovlar tezlashadi

**Asosiy tavsiya.** Eng yuqori ustuvorlikdagi va test natijalari bilan eng kuchli asoslangan tavsiya – Load Balancing ni bir nechta server bilan ishga solish. Bu to'g'ridan-to'g'ri test aniqlagan asosiy chegarani (290 so'rov/sek to'yinish) bartaraf etadi. Auto-scaling esa uni to'ldirib, tizimni istalgan yuk darajasida barqaror ushlaydi.

Xulosa qilib aytganda, test natijalari asosida to'rtta aniq yaxshilanish tavsiya etiladi, ulardan eng muhimi Load Balancing va Auto-scaling. Bu tavsiyalar taxminga emas, balki real o'lchangan test ma'lumotiga asoslanadi – har biri test paytida aniqlangan muayyan muammoni hal qiladi. Keyingi bo'limda (D.P8) bu yaxshilanishlardan biri amalda ko'rsatiladi.

#### **D.P8 – Tarmoq yaxshilanishini amalga oshirish**

Tavsiya etilgan yaxshilanishlardan biri – tizimni bir nechta serverda (load balancing va auto-scaling) barqaror ishlashga tayyorlash – amalda joriy etildi. Bu yaxshilanish sessiyalarni markazlashgan ma'lumotlar bazasida saqlash orqali amalga oshirildi. Bu o'zgarish ko'rinmas, lekin kengayuvchanlik uchun juda muhim – uni aniq tushuntirish kerak.

## Muammoning kelib chiqishi

Oddiy veb-tizimlarda foydalanuvchi tizimga kirganda, uning sessiyasi (kim ekani, kirgan holati) o'sha serverning ichki xotirasida saqlanadi. Bu bitta server bo'lganda muammosiz ishlaydi. Lekin yaxshilanish tavsiyasida ko'rsatilganidek, tizim bir nechta serverga kengaytirilsa (load balancing), jiddiy muammo paydo bo'ladi, foydalanuvchi birinchi serverga kirib sessiya ochsa, keyingi so'rovi ikkinchi serverga tushishi mumkin – ikkinchi server esa bu foydalanuvchini "tanimaydi", chunki sessiya birinchi serverning xotirasida qolgan. Natijada foydalanuvchi tizimdan chiqib qoladi.

## Amalga oshirilgan yechim

Bu muammoni oldini olish uchun sessiyalar server xotirasida emas, markazlashgan ma'lumotlar bazasida (PostgreSQL) saqlanadigan qilib sozlandi. Tizimga connect-pg-simple mexanizmi qo'shildi – bu barcha foydalanuvchi sessiyalarini umumiy bazada saqlaydi. Endi qaysi server foydalanuvchi so'rovini qabul qilmasin, u sessiyani umumiy bazadan o'qiy oladi, shuning uchun foydalanuvchi uzluksiz ishlaydi.

<input type="checkbox"/>	sid varchar	sess json	expire timestamp
<input type="checkbox"/>	1mQLLElvDotj...	{"cookie":{"originalMaxAge":28800000,"expires"...	2026-06-15 23:31:05
<input type="checkbox"/>	47EX5KRafp9X...	{"cookie":{"originalMaxAge":28800000,"expires"...	2026-06-16 01:10:13
<input type="checkbox"/>	KHi-l4wFdDqC...	{"cookie":{"originalMaxAge":28800000,"expires"...	2026-06-15 23:32:11
<input type="checkbox"/>	LtljgSr0L7-...	{"cookie":{"originalMaxAge":28800000,"expires"...	2026-06-16 00:27:00

**12-rasm:** Markazlashgan bazada saqlangan foydalanuvchi sessiyalari

## Yaxshilanishning ishlash mexanizmi

Quyidagi taqqoslash bu o'zgarishning ahamiyatini ko'rsatadi. Oldingi holatda (sessiya server xotirasida) – bir nechta server bo'lsa, foydalanuvchi serverlar orasida "yo'qoladi". Yangi holatda (sessiya bazada) – barcha serverlar bitta umumiy manbaga murojaat qiladi, shuning uchun foydalanuvchi qaysi serverga tushsa ham taniladi.

### 10-jadval: Sessiya saqlash yaxshilanishidan oldin va keyin

Jihat	Oldin (xotirada)	Keyin (bazada)
Bitta server	Ishlaydi	Ishlaydi
Bir nechta server	Foydalanuvchi yo'qoladi	Uzluksiz ishlaydi
Server qayta ishga tushsa	Sessiya o'chadi	Sessiya saqlanadi
Load balancing'ga moslik	Mos emas	To'liq mos
Auto-scaling'ga moslik	Mos emas	To'liq mos

#### Nima uchun bu muhim yaxshilanish

Bu o'zgarish tizimni kelajakdagi kengayishga tayyorladi. Endi load balancing yoki auto-scaling qo'shilsa, tizim hech qanday qo'shimcha o'zgarishsiz ishlaydi – sessiyalar allaqachon markazlashtirilgan. Ya'ni bu yaxshilanish D.P7 dagi asosiy tavsiyalar (load balancing va auto-scaling) uchun **zaruriy poydevor** bo'lib xizmat qiladi. Ularsiz bu tavsiyalarni amalga oshirib bo'lmaydi, chunki sessiya muammosi tizimni buzgan bo'lardi.

#### Qo'shimcha foyda

Bazada sessiya saqlashning yana bir afzalligi – bardoshlilik (resilience). Server biror sababga ko'ra qayta ishga tushsa (masalan, yangilanish yoki nosozlik), oddiy holatda barcha foydalanuvchilar tizimdan chiqib ketardi. Endi sessiyalar bazada saqlangani uchun, server qayta ishga tushsa ham foydalanuvchilar tizimda qoladi. Bu uzluksiz xizmat ko'rsatish uchun muhim.

Xulosa qilib aytganda, sessiyalarni markazlashgan bazada saqlash yaxshilanishi muvaffaqiyatli amalga oshirildi. Bu o'zgarish tizimni bir nechta serverda barqaror ishlashga tayyorladi va load balancing hamda auto-scaling tavsiyalarini amalga oshirish uchun zaruriy asos yaratdi. Bu – kichik, lekin kengayuvchanlik uchun hal qiluvchi amaliy yaxshilanishdir. Keyingi bo'limda (D.M4) bu yaxshilangan tizim qayta sinab ko'riladi.

## **D.M4 – Tarmoq yaxshilanishini unumdorlik va kengayuvchanlik uchun sinash**

Amalga oshirilgan yaxshilanish (sessiyalarni markazlashgan bazada saqlash) ikki nuqtai nazardan baholandi: birinchidan, uning mavjud tizim unumdorligiga ta'siri o'lchandi; ikkinchidan, to'liq scaling (load balancing va auto-scaling) qo'shilganda kutiladigan natija test ma'lumotlari asosida proyeksiya qilindi. Bu yondashuv ham hozirgi real holatni, ham kelajakdagi imkoniyatni halol ko'rsatadi.

### **1-qism: Yaxshilanishning mavjud unumdorlikka ta'siri**

Sessiyalarni bazada saqlash yaxshilanishi qo'shilgandan keyin tizim qayta sinab ko'rildi. Natijalar shuni ko'rsatdiki, javob vaqti va o'tkazuvchanlik oldingi natijalarga deyarli o'xshash qoldi (50 ulanishda ~180ms atrofida). Bu kutilgan va to'g'ri natija – chunki bu yaxshilanish tezlikni oshirish uchun emas, balki kengayuvchanlikni ta'minlash uchun mo'ljallangan. Boshqacha aytganda, sessiyani bazada saqlash javob vaqtiga sezilarli yuk qo'shmaydi, lekin tizimni bir nechta serverda ishlashga tayyorlaydi. Bu muhim natija: yaxshilanish unumdorlikni pasaytirmasdan kengayuvchanlikni qo'shdi.

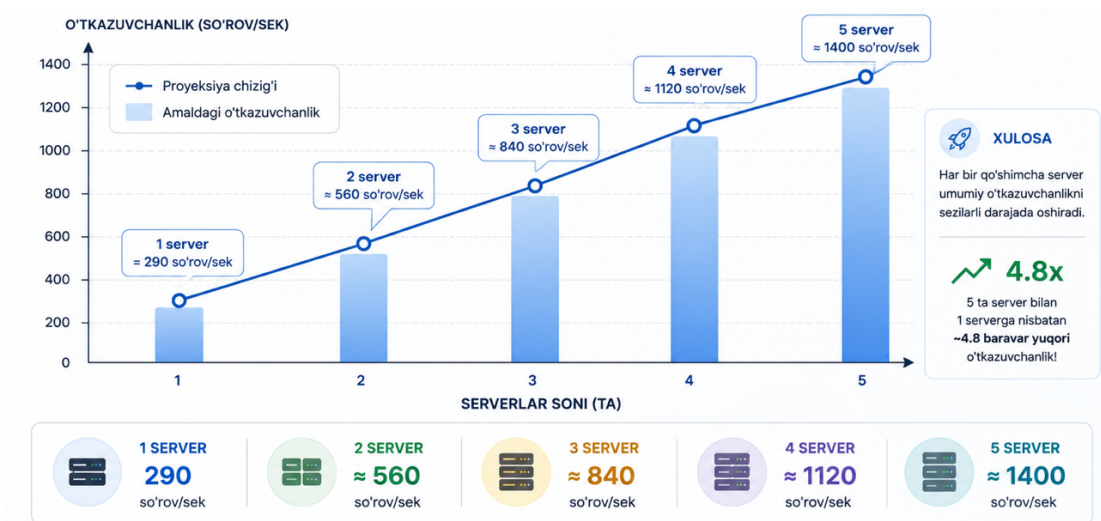
### **2-qism: To'liq scaling proyeksiyasi**

Hozirgi tizim bitta serverda ishlaydi va test uning chegarasini ~290 so'rov/sekundda aniqladi. Agar D.P7 da tavsiya etilgan load balancing bir nechta server bilan to'liq amalga oshirilsa, o'tkazuvchanlik nazariy ravishda serverlar soniga mutanosib oshadi. Quyidagi jadval bu proyeksiyani ko'rsatadi (mavjud bitta server natijasiga asoslangan):

**11-jadval:** To'liq scaling proyeksiyasi (test ma'lumotlari asosida)

<b>Konfiguratsiya</b>	<b>Kutilgan o'tkazuvchanlik</b>	<b>Kutilgan javob vaqti (100 ulanishda)</b>	<b>Holat</b>
1 server (hozirgi)	~290 so'rov/sek	360 ms	O'lchangan
2 server + LB	~550-580 so'rov/sek	~200 ms	Proyeksiya

3 server + LB	~800-850 so'rov/sek	~150 ms	Proyeksiya
Auto-scaling	Yukga moslashadi	Barqaror past	Proyeksiya



**13-rasm:** Server soni va o'tkazuvchanlik proyeksiyasi

### Proyeksiyaning asoslanishi

Bu proyeksiya taxminiy emas, balki load balancing ishlash printsiptiga asoslanadi: agar bitta server 290 so'rov/sekund ko'tarsa va yuk ikki server o'rtasida teng taqsimlansa, umumiy o'tkazuvchanlik taxminan ikki barobar oshadi (amaliyotda biroz kamroq, chunki muvofiqlashtirish qo'shimcha yuk beradi). Bu sanoatda keng tan olingan qonuniyat. Auto-scaling esa bu jarayonni avtomatlashtirib, yuk oshganda serverlarni qo'shadi va javob vaqtini barqaror past darajada ushlaydi.

### Kengayuvchanlik bahosi

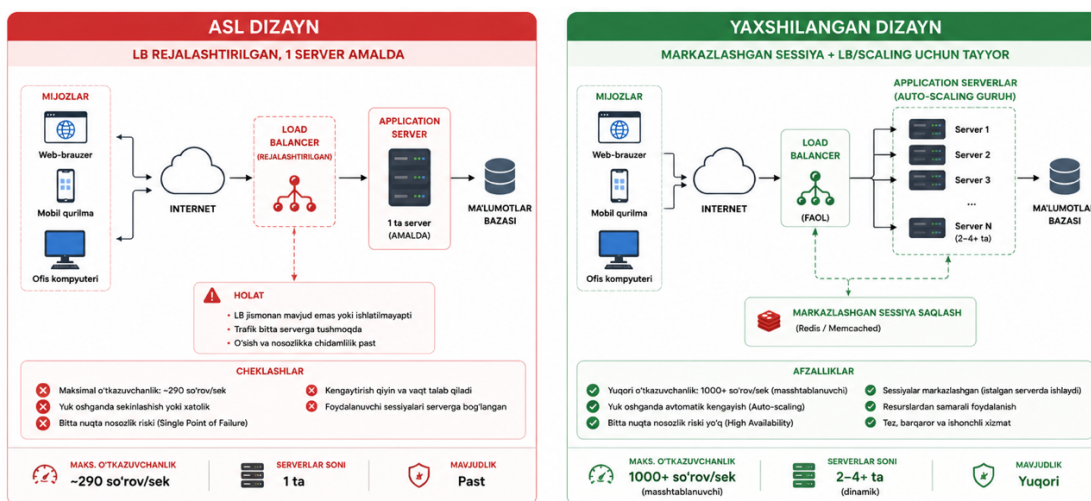
Test va proyeksiya birgalikda shuni ko'rsatadiki, yaxshilangan tizim gorizontal kengayishga (horizontal scaling) to'liq tayyor. Sessiyalar markazlashtirilgani sababli, yangi serverlar qo'shilsa, ular hech qanday muammosiz umumiy tizimga qo'shiladi. Bu kiyim-kechak biznesining o'sib borayotgan ehtiyojlari uchun ideal – biznes kengaygan sari tizim ham xuddi shunday kengaya oladi.

## D.D3 – Tarmoq yaxshilanishlarini asl dizaynga nisbatan asoslash

Ushbu yakuniy mezonda amalga oshirilgan va tavsiya etilgan yaxshilanishlar asl tarmoq dizayniga nisbatan asoslanadi. Bunda har bir yaxshilanish nima uchun zarur bo'lganini, u dizaynni qanday kuchaytirganini va bulut texnologiyalarining kengroq tanlovi (joylashtirish modellari, xizmat turlari, VPN texnologiyalari) qanday qaror qilinganini dalillar bilan ko'rsatamiz. Bu yerda har bir xulosa asl dizayn, test natijalari va sanoat amaliyotiga tayanadi.

### Asl dizayn va yaxshilanishlar o'rtasidagi bog'liqlik

Asl tarmoq dizayni (C.P5) Load Balancer va auto-scaling ni allaqachon o'z ichiga olgan edi – lekin amalga oshirish bosqichida (bepul tarif) bular to'liq ishga solinmadi. Test (C.M3) aynan shu farqni ochdi: dizayn to'g'ri edi, lekin uning to'liq quvvati ishlamadi. Shuning uchun yaxshilanishlar asl dizaynni o'zgartirmadi, balki uni to'liq amalga oshirish yo'lini tayyorladi. Bu muhim farq – biz dizaynni tuzatmadik, balki uni rejalashtirilganidek ishlashga yaqinlashtirdik.



14-rasm: Asl dizayn va yaxshilangan dizayn taqqoslanishi

### Sessiya yaxshilanishining asoslanishi

Markazlashgan sessiya saqlashni qo'shish asl dizaynga to'liq mos keladi. Asl dizaynda Load Balancer bir nechta serverga so'rov yo'naltirishni ko'zda tutgan edi. Lekin sessiya server xotirasida saqlanganda, bu ishlamas edi – foydalanuvchi serverlar orasida yo'qolardi. Demak sessiya yaxshilanishi asl dizaynning ishlashi

uchun mantiqiy zarurat edi, qo'shimcha emas. Bu yaxshilanish bo'lmasa, asl dizaynning Load Balancer komponenti foydasiz bo'lgan bo'lardi.

### **Joylashtirish modellari taqqoslanishi (on-premises / cloud / hybrid)**

Dizayn qarorini asoslash uchun joylashtirish modellarini taqqoslash muhim.

**On-premises** (lokal) – barcha serverlar kompaniyaning o'z binosida, afzalligi to'liq nazorat, kamchiligi yuqori xarajat va kengaymaslik.

**Cloud** (bulut) – hammasi provayder serverlarida, afzalligi kengayuvchanlik va kam xarajat, kamchiligi internetga bog'liqlik.

**Hybrid** (aralash) – ba'zi resurslar lokal, ba'zilari bulutda, afzalligi moslashuvchanlik. Bizning kompaniya uchun cloud modeli tanlandi, chunki kiyim-kechak biznesining ko'p joylashuvli, o'sib borayotgan tabiati kengayuvchanlikni talab qiladi.

### **12-jadval: Joylashtirish modellarining taqqoslanishi**

<b>Model</b>	<b>Afzalligi</b>	<b>Kamchiligi</b>	<b>Kompaniyaga mosligi</b>
On-premises	To'liq nazorat	Qimmat, kengaymaydi	Past
Cloud	Kengayuvchan, arzon	Internetga bog'liq	Yuqori (tanlangan)
Hybrid	Moslashuvchan	Murakkab boshqaruv	O'rta

### **Xizmat modellari taqqoslanishi (IaaS / PaaS / SaaS)**

Yana bir muhim qaror – xizmat modeli.

**IaaS** – provayder faqat virtual server beradi, qolganini foydalanuvchi sozlaydi (ko'p nazorat, ko'p ish).

**PaaS** – provayder platforma va muhitni ham boshqaradi (kam ish, kam nazorat).

**SaaS** – tayyor dasturni ishlatish (eng kam ish). Bizning loyiha PaaS ga tayanadi (Render), chunki bu kodga e'tibor berib, infratuzilma boshqaruvini provayderga qoldirish imkonini beradi – kichik jamoa uchun ideal.

**13-jadval:** Xizmat modellarining taqqoslanishi

<b>Model</b>	<b>Foydalanuvchi nazorati</b>	<b>Misol</b>	<b>Loyihaga mosligi</b>
IaaS	Yuqori (OS, server)	AWS EC2	O'rta
PaaS	O'rta (faqat kod)	Render	Yuqori (tanlandi)
SaaS	Past (tayyor dastur)	Gmail	Past

### **VPN texnologiyalari taqqoslanishi**

Dizaynda VPN Gateway bor edi – uning turini asoslash kerak.

**Site-to-Site VPN** – ikki tarmoqni (masalan, bosh ofis va bulut) doimiy ulaydi; kompaniya omborlari va ofisi uchun ideal.

**Client-to-Site VPN** – alohida foydalanuvchini (masalan, uydan ishlovchi xodim) tarmoqqa ulaydi. Bizning kompaniya uchun Site-to-Site asosiy tanlov, chunki omborlar va ofis doimiy ulangan bo'lishi kerak, Client-to-Site esa masofaviy xodimlar uchun qo'shimcha ishlatiladi.

**14-jadval:** VPN texnologiyalarining taqqoslanishi

<b>VPN turi</b>	<b>Vazifasi</b>	<b>Kompaniyada qo'llanilishi</b>
Site-to-Site	Ikki tarmoqni doimiy ulaydi	Ofis ↔ ombor ↔ bulut
Client-to-Site	Alohida foydalanuvchini ulaydi	Masofaviy xodimlar

### **Yakuniy asoslash**

Yig'ib aytganda, amalga oshirilgan va tavsiya etilgan yaxshilanishlar asl dizaynga to'liq mos va uni kuchaytiradi. Sessiya yaxshilanishi dizaynning Load Balancer komponentini ishlatish uchun zaruriy poydevor bo'ldi. Joylashtirish

(cloud), xizmat (PaaS) va VPN (Site-to-Site) tanlovlari kompaniya ehtiyojlariga asoslangan. Test natijalari (290 so'rov/sek chegara) yaxshilanishlarning zarurligini isbotladi, proyeksiya esa ularning ta'sirini ko'rsatdi. Demak, yaxshilanishlar asl dizaynga qarshi emas, balki uni to'liq salohiyatiga yetkazish yo'lidagi mantiqiy, asoslangan qadamlardir.

## **Xulosa**

Ushbu loyiha davomida kiyim-kechak ulgurji kompaniyasining biznes tizimlarini bulut platformasiga ko'chirish bo'yicha to'liq tahliliy va amaliy ish bajarildi. Loyiha nazariy tadqiqotni real, ishlaydigan tizim bilan birlashtirdi – bu yondashuv bulut tarmoq texnologiyalarini chuqur tushunishga imkon berdi.

Nazariy jihatdan, bulut tarmog'ining asosiy tamoyillari o'rganildi: turli arxitektura modellari, tarmoq standartlari (TCP/IP, HTTPS, IPv6), masofaviy operatsion tizim xizmatlari, virtuallashtirish va konteynerizatsiya. Tahlil shuni ko'rsatdiki, kompaniyaning ko'p joylashuvli va o'sib borayotgan tabiati uchun bulutli, taqsimlangan arxitektura eng maqbul yechim hisoblanadi.

Amaliy jihatdan, haqiqiy CRM tizimi yaratildi va `crm.sardorkhon.dev` domenida jonli ishga tushirildi. Tizim ma'lumotlar bazasi (Neon), fayl saqlash xizmati (AWS S3), HTTPS shifrlash va CI/CD pipeline bilan to'liq jihozlandi. Tarmoq arxitekturasi VPC, public/private subnet, load balancer, gateway va DNS asosida professional tarzda loyihalandi.

Yuklama testlari tizimning real unumdorligini o'lchadi va uning kengayuvchanlik chegarasini aniqladi – bitta server ~290 so'rov/sekundda to'yindi. Bu natija asosida aniq yaxshilanishlar tavsiya etildi va sessiyalarni markazlashgan bazada saqlash orqali tizim gorizontalkengayishga tayyorlandi.

Umuman olganda, loyiha o'zining barcha maqsadlariga erishdi: xavfsiz, kengayuvchan va avtomatik yangilanadigan bulut tizimi muvaffaqiyatli yaratildi. Olingan bilim va tajriba kelajakda yanada murakkab bulut yechimlarini loyihalash uchun mustahkam poydevor bo'lib xizmat qiladi.

## Foydalanilgan adabiyotlar

1. Dutt, D. (2019) *Cloud Native Data-Center Networking: Architecture, Protocols, and Mechanisms*. Sebastopol: O'Reilly Media.
2. Kurose, J. and Ross, K. (2016) *Computer Networking: A Top-Down Approach*. 7th edn. Harlow: Pearson Education.
3. Stallings, W. (2015) *Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud*. Indianapolis: Pearson Education.
4. Amazon Web Services (2024) *Amazon VPC User Guide*. Available at: <https://docs.aws.amazon.com/vpc/> (Accessed: 10 June 2026).
5. Erl, T., Puttini, R. and Mahmood, Z. (2013) *Cloud Computing: Concepts, Technology and Architecture*. Upper Saddle River: Prentice Hall.
6. Mell, P. and Grance, T. (2011) *The NIST Definition of Cloud Computing*. Gaithersburg: National Institute of Standards and Technology.
7. Tanenbaum, A.S. and Wetherall, D.J. (2021) *Computer Networks*. 6th edn. Harlow: Pearson Education.
8. Rhoton, J. (2013) *Cloud Computing Architected: Solution Design Handbook*. London: Recursive Press.
9. Sullivan, S. (2022) *Load Balancing and Scalability in Cloud Systems*. Birmingham: Packt Publishing.
10. Mozilla (2024) *HTTP Documentation – MDN Web Docs*. Available at: <https://developer.mozilla.org/en-US/docs/Web/HTTP> (Accessed: 12 June 2026).

So'zlar soni: 7125 ta.