

# Kamronbek Allanazarov

EN · ENGLISH

## Junior Cloud Engineer | AWS & Terraform

Samarkand, Uzbekistan | kamronb3k03@gmail.com | +998 94 423 03 40 | [github.com/A-Kamronb3k](https://github.com/A-Kamronb3k) | [portfolio.kamronb3k03.workers.dev](https://portfolio.kamronb3k03.workers.dev)

### SUMMARY

Junior cloud engineer with a self-built portfolio of four production-style AWS projects — every resource provisioned as code and running at \$0 on the Free Tier. Hands-on across serverless (Lambda, API Gateway, DynamoDB), infrastructure-as-code (Terraform), containers (Docker), and keyless CI/CD (GitHub Actions + OIDC). A background in economics and frontend development brings cost-awareness and a shipping mindset to infrastructure. I document everything I build — including what breaks and how I fix it.

### TECHNICAL SKILLS

**AWS:** Lambda, API Gateway, DynamoDB, S3, CloudFront, Cognito, CloudWatch, SNS, SQS, IAM, X-Ray

**Infrastructure as Code:** Terraform (modules, remote state, OIDC federation)

**CI/CD & Security:** GitHub Actions (OIDC keyless deploys), checkov, Infracost, Trivy

**Containers:** Docker (multi-stage builds, non-root images, GHCR)

**Languages:** Python, JavaScript, Bash, HTML/CSS

**Practices:** Least-privilege IAM, structured logging, observability (EMF, dashboards, alarms), Git, Linux CLI

### PROJECTS

#### Serverless Notes App — full-stack serverless (flagship)

[GitHub](#) · [Live](#)

[Terraform](#) · [Lambda](#) · [API Gateway](#) · [DynamoDB](#) · [Cognito](#) · [GitHub Actions](#)

- ▶ Built a full CRUD notes app where every AWS resource is defined in Terraform: five single-purpose Python Lambdas, an HTTP API with Cognito JWT auth, and a vanilla-JS SPA on private S3 + CloudFront (OAC).
- ▶ Implemented fully keyless CI/CD — pytest + moto on every push, terraform plan posted to pull requests via a read-only OIDC role, and apply-on-merge via a main-branch-only role.

#### Observability Pipeline — event-driven monitoring

[GitHub](#)

[CloudWatch](#) · [EMF](#) · [X-Ray](#) · [SNS](#) · [SQS](#) · [Terraform](#)

- ▶ Built an S3 → Lambda → DynamoDB pipeline instrumented like production: custom metrics via EMF, X-Ray tracing, a dashboard-as-code, and three SNS email alarms with an operations runbook.
- ▶ Added checkov (IaC security) and Infracost (cost estimation) gates to CI; verified the entire stack runs at \$0.00/month.

#### AWS Static Site + CI/CD — static site with CDN

[GitHub](#) · [Live](#)

[S3](#) · [CloudFront](#) · [GitHub Actions](#) · [OIDC](#)

- ▶ Served a static site from a private S3 bucket through CloudFront (OAC) over HTTPS, with custom error pages.
- ▶ Automated deploys via GitHub Actions using OIDC — no long-lived AWS keys stored anywhere.

#### Dockerized Feedback API — production-style container pipeline

[GitHub](#) · [Live](#)

[Docker](#) · [Trivy](#) · [GHCR](#) · [Node.js](#)

- ▶ Packaged an Express API in a multi-stage, non-root container with a healthcheck and no npm in the final image.
- ▶ CI lints, tests, Trivy-scans, and publishes to GHCR; the security gate caught and blocked two real CVEs.

### EDUCATION

#### Cloud Engineering Program — IT Park Uzbekistan, Samarkand

Apr – Aug 2026

State-funded program (Ministry of Digital Technologies + World Bank), AIUB-based curriculum, taught in English. Certification exam: August 2026.

#### B.Sc. in Digital Economy — Tashkent State University of Economics

2021 – 2025

#### Frontend Development (React) — Najot Ta'lim

Dec 2023 – Sep 2024

Hands-on bootcamp: JavaScript, React, responsive UI. Diploma and reference available.

### CERTIFICATIONS

IT Park Cloud Engineering Program certification — **in progress** (exam Aug 2026). AWS Certified Cloud Practitioner — **planned**.

О СЕБЕ

Junior cloud engineer с самостоятельно построенным портфолио из четырёх AWS-проектов уровня production — каждый ресурс развёрнут как код и работает на Free Tier за \$0. Практический опыт в serverless (Lambda, API Gateway, DynamoDB), инфраструктуре как коде (Terraform), контейнерах (Docker) и бесключевом CI/CD (GitHub Actions + OIDC). Опыт в экономике и frontend-разработке даёт понимание стоимости и нацеленность на результат в инфраструктурных задачах. Документирую всё, что создаю, — включая то, что ломается, и как я это исправляю.

ТЕХНИЧЕСКИЕ НАВЫКИ

**AWS:** Lambda, API Gateway, DynamoDB, S3, CloudFront, Cognito, CloudWatch, SNS, SQS, IAM, X-Ray  
**Инфраструктура как код:** Terraform (модули, remote state, OIDC federation)  
**CI/CD и безопасность:** GitHub Actions (keyless-деплой через OIDC), checkov, Infracost, Trivy  
**Контейнеры:** Docker (multi-stage сборки, non-root образы, GHCR)  
**Языки:** Python, JavaScript, Bash, HTML/CSS  
**Практики:** IAM по принципу минимальных привилегий, структурированное логирование, observability (EMF, дашборды, алармы), Git, Linux CLI

ПРОЕКТЫ

**Serverless Notes App** — full-stack serverless (флагманский проект) [GitHub](#) · [Live](#)  
[Terraform](#) · [Lambda](#) · [API Gateway](#) · [DynamoDB](#) · [Cognito](#) · [GitHub Actions](#)

- ▶ Разработал полноценное CRUD-приложение для заметок, где каждый AWS-ресурс описан в Terraform: пять узкоспециализированных Python Lambda-функций, HTTP API с JWT-аутентификацией через Cognito и SPA на vanilla JS в приватном S3 + CloudFront (OAC).
- ▶ Реализовал полностью бесключевой CI/CD: pytest + moto при каждом push, terraform plan публикуется в pull request через read-only OIDC-роль, apply выполняется при мерже через роль, ограниченную веткой main.

**Observability Pipeline** — событийно-ориентированный мониторинг [GitHub](#)  
[CloudWatch](#) · [EMF](#) · [X-Ray](#) · [SNS](#) · [SQS](#) · [Terraform](#)

- ▶ Построил pipeline S3 → Lambda → DynamoDB, инструментированный по стандартам production: кастомные метрики через EMF, трассировка X-Ray, дашборд как код и три email-алармы через SNS с операционным runbook.
- ▶ Добавил в CI проверки checkov (безопасность IaC) и Infracost (оценка стоимости); подтвердил, что весь стек работает за \$0.00/месяц.

**AWS Static Site + CI/CD** — статический сайт с CDN [GitHub](#) · [Live](#)  
[S3](#) · [CloudFront](#) · [GitHub Actions](#) · [OIDC](#)

- ▶ Развернул статический сайт из приватного S3-бакета через CloudFront (OAC) по HTTPS, с кастомными страницами ошибок.
- ▶ Автоматизировал деплой через GitHub Actions с использованием OIDC — без хранения долгоживущих AWS-ключей.

**Dockerized Feedback API** — контейнерный pipeline уровня production [GitHub](#) · [Live](#)  
[Docker](#) · [Trivy](#) · [GHCR](#) · [Node.js](#)

- ▶ Упаковал Express API в multi-stage non-root контейнер с healthcheck и без prpm в финальном образе.
- ▶ CI выполняет lint, тесты, Trivy-сканирование и публикацию в GHCR; security gate обнаружил и заблокировал две реальные CVE.

ОБРАЗОВАНИЕ

**Программа Cloud Engineering** — IT Park Uzbekistan, Самарканд [Апр — Авг 2026](#)  
Государственная программа (Министерство цифровых технологий + Всемирный банк), учебный план на основе AIUB, обучение на английском языке. Сертификационный экзамен: август 2026.

**Бакалавр по направлению «Цифровая экономика»** — Ташкентский государственный экономический университет [2021 — 2025](#)

**Frontend-разработка (React)** — Najot Ta'lim [Дек 2023 — Сен 2024](#)  
Практический буткемп: JavaScript, React, адаптивный UI. Диплом и рекомендация доступны по запросу.

СЕРТИФИКАТЫ

Сертификат программы IT Park Cloud Engineering — **в процессе** (экзамен август 2026). AWS Certified Cloud Practitioner — **запланирован**.

# Kamronbek Allanazarov

UZ · O'ZBEKCHA

## Junior Cloud Engineer | AWS va Terraform

Samarqand, O'zbekiston | kamronb3k03@gmail.com | +998 94 423 03 40 | [github.com/A-Kamronb3k](#) | [portfolio.kamronb3k03.workers.dev](#)

### MEN HAQIMDA

O'zim qurgan, to'rtta production darajasidagi AWS loyihasidan iborat portfolioga ega junior cloud engineer — har bir resurs kod sifatida yaratilgan va Free Tier'da \$0 xarajat bilan ishlaydi. Serverless (Lambda, API Gateway, DynamoDB), infrastructure-as-code (Terraform), konteynerlar (Docker) va kalitsiz CI/CD (GitHub Actions + OIDC) bo'yicha amaliy tajribaga egaman. Iqtisodiyot va frontend yo'nalishidagi tajribam infrastrukturaga xarajatlarni hisobga olgan holda va natijaga yo'naltirilgan tarzda yondashuv beradi. Men yaratgan har bir narsani hujjatlashtiraman — nima buzilgani va uni qanday tuzatganimni ham.

### TEXNIK KO'NIKMALAR

**AWS:** Lambda, API Gateway, DynamoDB, S3, CloudFront, Cognito, CloudWatch, SNS, SQS, IAM, X-Ray

**Infrastructure as Code:** Terraform (modullar, remote state, OIDC federatsiya)

**CI/CD va xavfsizlik:** GitHub Actions (OIDC orqali kalitsiz deploy), checkov, Infracost, Trivy

**Konteynerlar:** Docker (multi-stage build, non-root image, GHCR)

**Tillar:** Python, JavaScript, Bash, HTML/CSS

**Amaliyotlar:** Minimal huquqlar asosidagi IAM, strukturaviy logging, observability (EMF, dashboardlar, alarmlar), Git, Linux CLI

### LOYIHALAR

**Serverless Notes App** — to'liq serverless ilova (flagman loyiha)

[GitHub](#) · [Live](#)

[Terraform](#) · [Lambda](#) · [API Gateway](#) · [DynamoDB](#) · [Cognito](#) · [GitHub Actions](#)

- Har bir AWS resursi Terraform orqali yozilgan to'liq CRUD notes ilovasini yaratdim: beshta yagona-vazifali Python Lambda, Cognito JWT autentifikatsiyali HTTP API va S3 (private) + CloudFront (OAC) ustida ishlaydigan vanilla-JS SPA.
- To'liq kalitsiz CI/CD joriy qildim — har bir push'da pytest + moto, terraform plan natijasi read-only OIDC role orqali pull request'ga chiqadi, apply esa faqat main branch'ga mos role orqali merge vaqtida ishlaydi.

**Observability Pipeline** — hodisaga asoslangan monitoring

[GitHub](#)

[CloudWatch](#) · [EMF](#) · [X-Ray](#) · [SNS](#) · [SQS](#) · [Terraform](#)

- Production standartlariga mos S3 → Lambda → DynamoDB pipeline qurdim: EMF orqali maxsus metrikalar, X-Ray tracing, kod sifatidagi dashboard va operatsion runbook bilan uchta SNS email alarm.
- CI jarayoniga checkov (IaC xavfsizligi) va Infracost (xarajatni baholash) tekshiruvlarini qo'shdim; butun stek oyiga \$0.00 bilan ishlashini tasdiqladim.

**AWS Static Site + CI/CD** — CDN bilan statik sayt

[GitHub](#) · [Live](#)

[S3](#) · [CloudFront](#) · [GitHub Actions](#) · [OIDC](#)

- Xususiy S3 bucket'dan CloudFront (OAC) orqali HTTPS ostida statik sayt xizmat ko'rsatdim, maxsus xato sahifalari bilan.
- GitHub Actions va OIDC yordamida deploy jarayonini avtomatlashtirdim — hech qayerda uzoq muddatli AWS kalitlari saqlanmaydi.

**Dockerized Feedback API** — production darajasidagi konteyner pipeline

[GitHub](#) · [Live](#)

[Docker](#) · [Trivy](#) · [GHCR](#) · [Node.js](#)

- Express API'ni multi-stage, non-root konteynerga joyladim — healthcheck bilan va yakuniy image'da npm'siz.
- CI jarayoni lint, test, Trivy skanerlash va GHCR'ga publish qiladi; xavfsizlik gate'i ikkita haqiqiy CVE'ni aniqlab, bloklagan.

### TA'LIM

**Cloud Engineering dasturi** — IT Park Uzbekistan, Samarqand

Apr – Avg 2026

Davlat tomonidan moliyalashtiriladigan dastur (Raqamli texnologiyalar vazirligi + Jahon banki), AIUB asosidagi o'quv dasturi, ingliz tilida o'qitiladi. Sertifikatsiya imtihoni: 2026-yil avgust.

**Raqamli iqtisodiyot yo'nalishi bo'yicha bakalavr** — Toshkent davlat iqtisodiyot universiteti

2021 – 2025

**Frontend dasturlash (React)** — Najot Ta'lim

Dek 2023 – Sen 2024

Amaliy bootcamp: JavaScript, React, responsive UI. Diplom va tavsiyanoma mavjud.

### SERTIFIKATLAR

IT Park Cloud Engineering dasturi sertifikati — **jarayonda** (imtihon 2026-yil avgust). AWS Certified Cloud Practitioner — **rejalashtirilgan**.