

[Про проект](#)

Репозиторій Online Boutique (microservices-demo)

<https://github.com/GoogleCloudPlatform/microservices-demo> — це demo e-commerce застосунок, який Google використовує для демонстрації сучасних практик і продуктів у клауді. Застосунок працює на будь-якому Kubernetes-кластері (локально або в інших cloud-провайдерах — AWS, Azure).

Склад системи: 11 мікросервісів, які взаємодіють між собою: frontend, cartservice, productcatalogservice, currencyservice, paymentservice, shippingservice, emailservice, checkoutservice, recommendationservice, adservice, loadgenerator (Locust).

У проєкті вже реалізовано багато DevOps-підходів та інструментів. Учасникам потрібно їх опанувати та доповнити.

[Завдання](#)

1. Інфраструктура

Розгорнути систему з двома середовищами (**Staging** та **Production**) у різних namespaces K8S кластера на будь-якому публічному cloud-провайдері (GCP, Azure або AWS) за допомогою **Terraform (IaC)**.

У репозиторії надано Terraform-код для GCP — ви можете його перевикористати або, за потреби, створити власний Terraform-код для розгортання в іншому cloud-провайдері (AWS або Azure) якщо вам буде так зручніше.

Для виконання завдання:

- Створіть окремий акаунт у GCP, Azure або AWS (де за можливості, можна використати безкоштовні кредити).
- Після завершення роботи не забудьте видалити всю створену інфраструктуру.

2. CI/CD

Підняти та налаштувати систему автоматичної збірки й розгортання (CI/CD) для хочаб двох мікросервісів на вибір, для двох середовищ — **Staging** та **Production**.

Для CI/CD можна використовувати будь-які сервіси, наприклад: GitHub (GitHub Actions), GitLab, Bitbucket, Jenkins, або інший SCM + CI-інструмент.

3. Моніторинг та Observability

Побудувати систему моніторингу сервісів і інфраструктурних компонентів. Розробити декілька дашбордів, які допомагатимуть вимірювати масштабованість та надійність застосунку.

Для моніторингу рекомендується використовувати:

- класичний стек **Prometheus + Grafana**,
або
- managed-сервіси відповідного cloud-провайдера.

Критерії оцінювання

1. Must-Have functionality:

- Terraform піднімає/зносить інфраструктуру.
- Є staging+prod (на базі K8S namespaces) з доступним frontend.
- CI/CD для 2 сервісів: build+push+deploy у staging і prod.
- Є 2–3 корисні dashboards (не просто “CPU usage”, але і метрики про reliability/scalability).

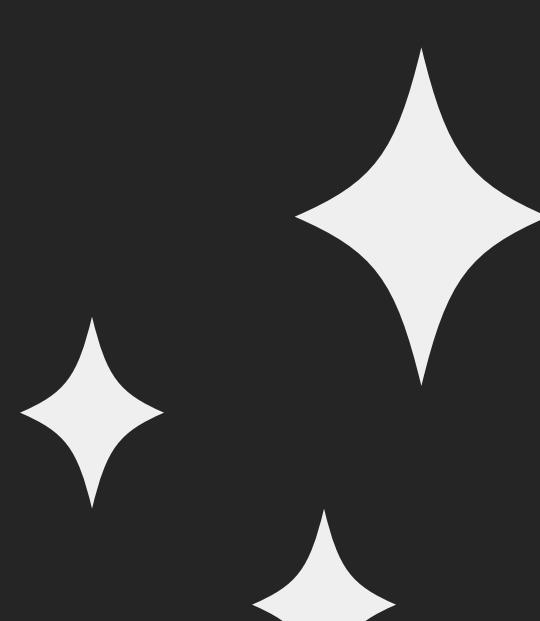
2. Наданий репозиторій із внесеними змінами та описаний README-файл із скріншотами:

- задеплоєної системи в клауді,
- налаштованого CI/CD,
- створених дашбордів.

3. Високорівневий опис cloud-архітектури.

4. **Обов'язково:** записане відео із голосовим супроводом з end-to-end workflow & demo:

- що саме було реалізовано,
- як це працює,
- обґрунтування чому були обрані ті чи інші інструменти або прийняті саме такі архітектурні рішення.



Використання AI-інструментів під час реалізації завдання вітається