

Eventim DevOps Challenge — Terraform + AWS CDK + Kubernetes (Rancher)

Objective

The goal of this challenge is to evaluate your ability to design and deploy cloud infrastructure using **Terraform** and **AWS CDK**, integrating Kubernetes and serverless services.

You'll build:

1. **A Kubernetes cluster (via Terraform)** — running **Rancher** to manage workloads and apps.
2. **A Lambda function (via CDK)** — that inserts random click metrics into MongoDB every 2 minutes.

We want to see your ability to:

- Structure infrastructure-as-code (IaC) projects.
- Automate deployments reproducibly.
- Apply best practices for modularity, naming, security, and automation.

Part 1 — Terraform (Kubernetes + Rancher)

Requirements

- Create a **Terraform project** (in a separate Git repository) that:
 - Provisions a **Kubernetes cluster** (EKS preferred).
 - Installs **Rancher** via Helm or manifests.
 - Ensures the cluster is accessible via Rancher UI.
 - Optionally: expose Rancher securely using a load balancer or ingress controller (bonus points).

Deliverables

- A GitHub repository named terraform-k8s-rancher.
- A README.md describing:

- How to deploy the infrastructure (terraform init/plan/apply).
- Prerequisites (e.g., AWS credentials, Terraform version).
- Any post-deployment steps to access Rancher.

Part 2 — AWS CDK (Lambda + MongoDB)

Requirements

- Create another **CDK project** (TypeScript) in a separate repo.
- It should deploy:
 - An **AWS Lambda function** that runs **every 2 minutes** (using EventBridge schedule).
 - The function connects to **MongoDB Atlas** (or a local MongoDB URI provided via environment variable).
 - The function inserts a new document into a collection named clicks with:

```
{  
  
  "timestamp": "2025-10-08T14:02:00Z",  
  
  "clicks": 57  
}
```

Where clicks is a random integer between 0 and 100.

- The MongoDB URI should **not** be hardcoded — use **AWS Secrets Manager** or an **environment variable**.

Deliverables

- A GitHub repository named cdk-lambda-clicks.
- Include:
 - bin/ and lib/ CDK structure.
 - The Lambda code (provided below).
 - A clear README.md with deployment instructions (cdk synth, cdk deploy).
 - Bonus: include CDK integration tests or validation via assertions.

Provided Code — Lambda Function (TypeScript)

<https://gist.github.com/joselfonseca/1edc7b02b68afb3750b8e72302ffe782>

Notes for the Candidate

- Use best practices for:
 - Terraform module organization (e.g., VPC, cluster, Helm releases).
 - CDK app structure (bin and lib separation).
 - Reusable and configurable variables.
- Add a short explanation of your architecture and decisions in each repo's README.
- Deployment scripts or GitHub Actions for automation are a plus.

Submission

Submit:

1. Two GitHub repositories:
 - a. terraform-k8s-rancher
 - b. cdk-lambda-clicks
2. Each with clear instructions for deployment.
3. Include any diagrams or screenshots (optional but nice).