

\$ Terraform Runbook

Xử lý sự cố State, Drift, Lock & Refactor thực chiến.



1 State Lock & Emergency

`terraform force-unlock <id>`

Mở khoá khi CI/CD crash giữ lock.

`terraform plan -refresh-only`

Đồng bộ trạng thái thực tế từ cloud.

`terraform state pull > tf.bak`

Sao lưu state trước khi sửa đổi.

2 State Drift & Inspection

`terraform plan -detailed-exitcode`

Exit code 2: phát hiện drift hạ tầng.

`terraform state list | grep <pat>`

Liệt kê resource đang quản lý.

`terraform state show <res>`

Xem chi tiết cấu hình resource.

3 Refactor & Rename

`terraform state mv <old> <new>`

Đổi tên resource mà không bị destroy.

`moved { from=<old> to=<new> }`

Khai báo migrate trong file HCL.

`terraform state rm <res>`

Gỡ khỏi state, giữ nguyên cloud.

4 Import Cloud Resources

`import { to=<res> id="<id>" }`

Khai báo import block mới (1.5+).

`terraform plan -generate-config-out=<f>`

Tự động sinh code HCL từ cloud.

`terraform import <res> <id>`

Import truyền thống vào file state.

5 Targeted Apply & Replace

`terraform apply -target=<res>`

Chỉ deploy 1 resource chỉ định.

`terraform apply -replace=<res>`

Ép buộc huỷ và tạo lại resource.

`terraform untaint <res>`

Huỷ đánh dấu hỏng cho resource cũ.

6 Debug & Performance

`export TF_LOG=DEBUG TF_LOG_PATH=tf.log`

Bật log chi tiết HTTP Provider.

`terraform plan -parallelism=30`

Tăng số luồng chạy song song.

`terraform validate && tflint`

Soát lỗi cú pháp & best practice.

6 Nguyên Tắc Vàng Khi Quản Trị State

1. **S3 + DynamoDB** luôn bật state locking
2. **Exit code 2** báo hiệu phát hiện drift
3. **Block moved** thay thế state mv thủ công
4. **Block import** tự động sinh code chuẩn
5. **State pull** luôn backup trước khi sửa
6. **Hạn chế target** tránh sót dependency ng

<res> = resource.name · <id> = cloud_id · <old>/<new> = res_address

Vibe DevOps · Build · Run · Fix