UDACITY

‹ Return to Classroom

Give Your Application Auto-Deploy Superpowers

REVIEW

CODE REVIEW 3

HISTORY

Meets Specifications

⭐ Congratulations on successfully completing this project! ⭐

You have done an amazing work in this challenging project using CI/CD automation tools and submitting excellent screenshots from your CircleCI Jobs, including the whole screen with your **CircleCI profile** in the top left and also the **CircleCI URL** with the pipeline number.

You have certainly acquired all the important concepts and skills from this project with great value on the market. 👍

Good luck with the next project! 👏
Happy learning and stay safe!

💡 **Here are a few resources you might find useful for more insight and further learning:**

- AWS Website endpoints
- Why am I getting a syntax error in this script?
- Key-Value Store API
  Get a key-value database and web API up and running in seconds.

Section 1: Selling CI/CD to your Team/Organization

The CI/CD benefits proposal contains essential benefits of CI/CD, and describes the business context that will benefit from the automation tools. Explanation should include benefits that translate to revenue and cost for the business.

PS
A previous reviewer already passed the student on this section, showing the **technical** benefits of CI/CD automation tools to business!👏

Section 2: Deploying Working, Trustworthy Software

A public git repository with your project code. [URL01]

Evidence of code-based CI/CD configuration in the form of yaml files in your git repository.

Console output of various pre-deploy job failure scenarios:

- **Build Jobs that failed because of compile errors. [SCREENSHOT01]**
- **Failed unit tests. [SCREENSHOT02]**
- **Failure because of vulnerable packages. [SCREENSHOT03]**
- **An alert from one of your failed builds. [SCREENSHOT04]**

Evidence in your code that:

- **Compile errors have been fixed.**
- **Unit tests have been fixed.**
- **All critical security vulnerabilities caught by the “Analyze” job have been fixed.**

✔ URL01: A public git repository with your project code. 👍

- https://github.com/hesham98/cdond-c3-projectstarter

✔ Your CircleCI Pipelines related to your submitted GitHub repository.

- https://app.circleci.com/pipelines/github/hesham98/cdond-c3-projectstarter

---

PS
A previous reviewer already passed the student on this rubric item: Deploying Working, Trustworthy Software. 👏

Console output of appropriate failure for infrastructure creation job (using CloudFormation). [SCREENSHOT05]

Console output of a smoke test job that is failing appropriately. [SCREENSHOT06]

Console output of a successful rollback after a failed smoke test. [SCREENSHOT07]

Console output of successful promotion of new version to production in CloudFront. [SCREENSHOT08]

Console output of successful cleanup job that removes old S3 bucket and EC2 instance. [SCREENSHOT09]

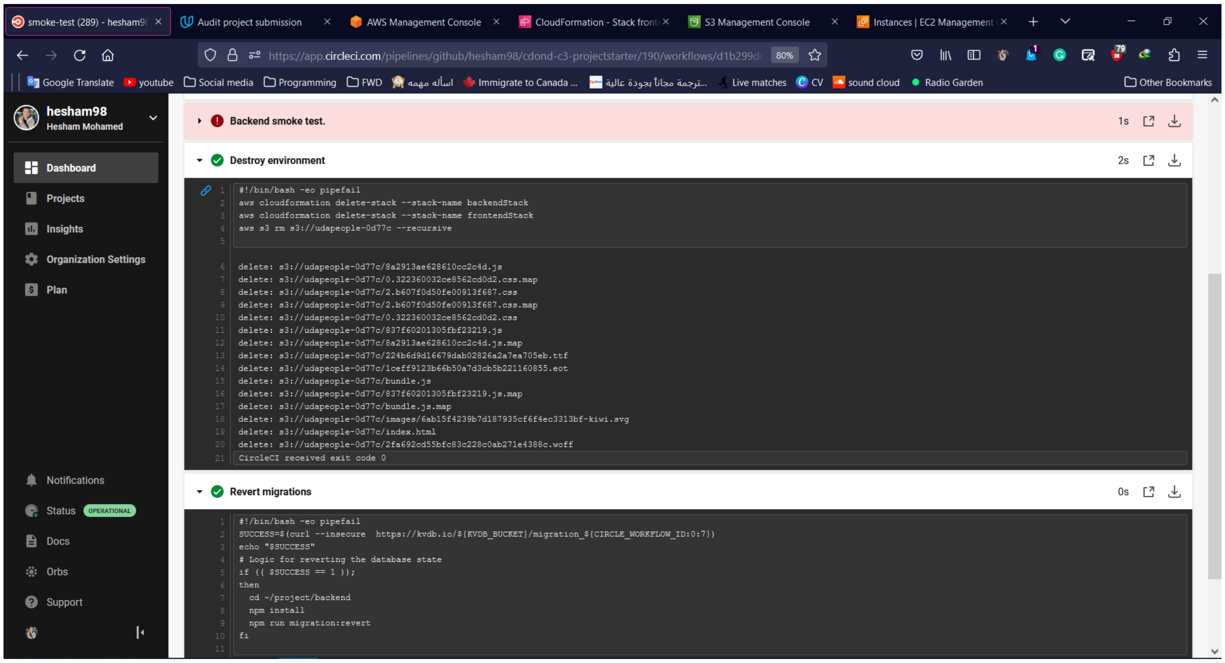
Evidence that the deploy jobs only happen on the `master` branch. [SCREENSHOT10]

Evidence of deployed and functioning front-end application in an S3 bucket [URL02].

Evidence of deployed and functioning front-end application in CloudFront. [URL03_SCREENSHOT]

Evidence of healthy back-end application. [URL04_SCREENSHOT]

✅ SCREENSHOT07: **Console output** of a successful rollback **after** a failed **smoke test**.



Backend smoke test.

Destroy environment

```
#!/bin/bash -eo pipefail
aws cloudformation delete-stack --stack-name backendStack
aws cloudformation delete-stack --stack-name frontendStack
aws s3 rm s3://udapeople-0d77c --recursive


delete: s3://udapeople-0d77c/8a2913ae628610cc2c4d.js
delete: s3://udapeople-0d77c/0.322360032ce8562cd0d2.css.map
delete: s3://udapeople-0d77c/2.b607f0d50fe00913f687.css
delete: s3://udapeople-0d77c/2.b607f0d50fe00913f687.css.map
delete: s3://udapeople-0d77c/0.322360032ce8562cd0d2.css
delete: s3://udapeople-0d77c/837f60201305fbf23219.js
delete: s3://udapeople-0d77c/8a2913ae628610cc2c4d.js.map
delete: s3://udapeople-0d77c/224b6d9d16679dab02826a2a7ea705eb.ttf
delete: s3://udapeople-0d77c/1ceff9123b66b50a7d3cb5b221160855.eot
delete: s3://udapeople-0d77c/bundle.js
delete: s3://udapeople-0d77c/837f60201305fbf23219.js.map
delete: s3://udapeople-0d77c/bundle.js.map
delete: s3://udapeople-0d77c/images/6ab15f4239b7d187935cf6f4ec3313bf-kiwi.svg
delete: s3://udapeople-0d77c/index.html
delete: s3://udapeople-0d77c/2fa692cd55bfc83c228c0ab271e4388c.woff
CircleCI received exit code 0
```

Revert migrations

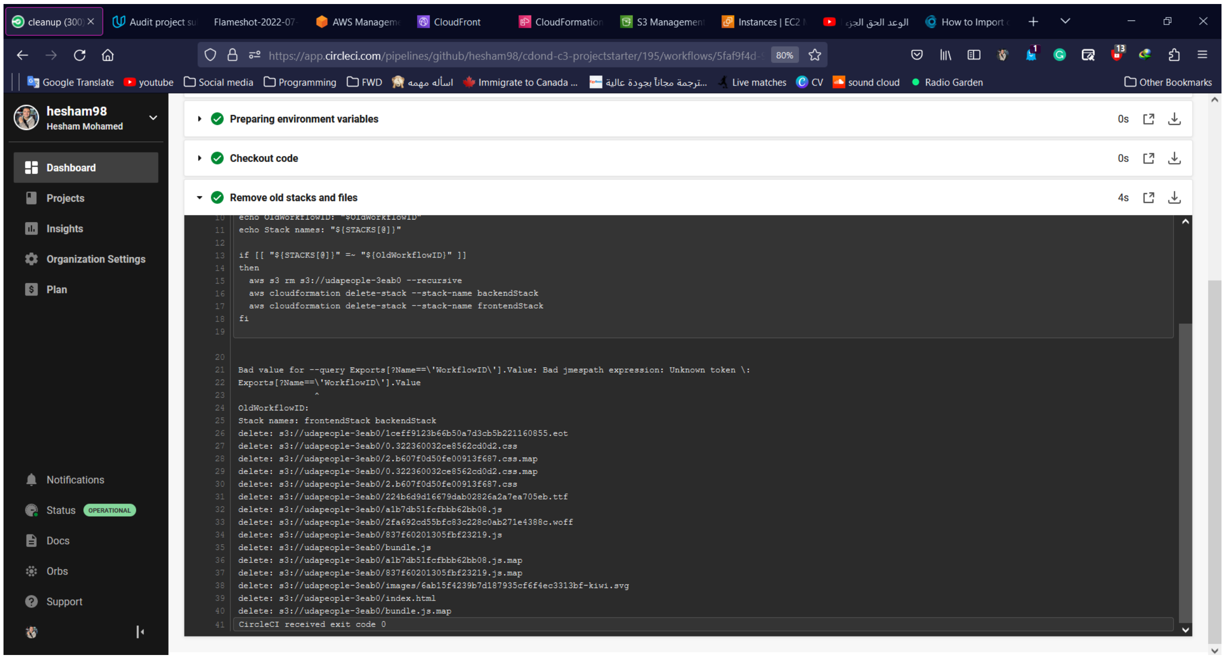
```
#!/bin/bash -eo pipefail
SUCCESS=$(curl --insecure  https://kvdb.io/${KVDB_BUCKET}/migration_${CIRCLE_WORKFLOW_ID:0:7})
echo "$SUCCESS"
# Logic for reverting the database state
if (( $SUCCESS == 1 ));
then
  cd ~/project/backend
  npm install
  npm run migration:revert
fi
```

👏 The screenshot includes the whole screen with the required steps, your **CircleCI profile** in the top left, and also the **CircleCI URL** with the pipeline number of your job.

Excellent and very detailed work! 👏

💡 Automating the process of *rolling back and cleaning* any infrastructure left, triggered by a **job failure**, will help to **avoid cost** and **protect revenue**.

✅ SCREENSHOT09: **Console output** of successful cleanup job that removes old **S3 bucket** and **EC2 instance**.



Preparing environment variables

Checkout code

Remove old stacks and files

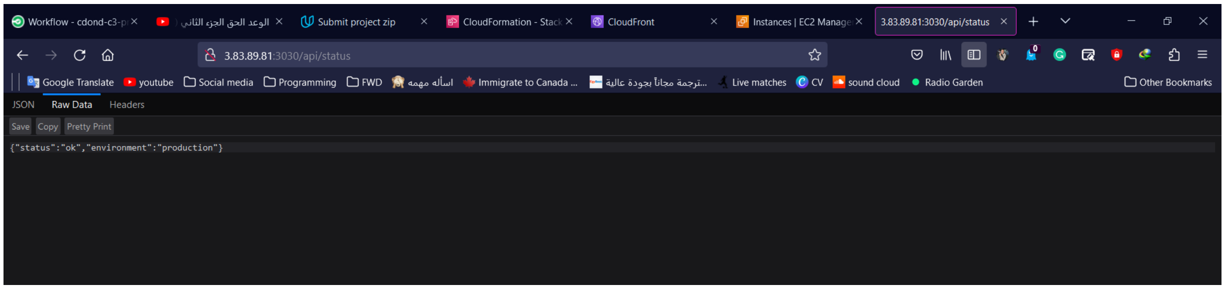
```
echo OldWorkflowID: "$OldWorkflowID"
echo Stack names: "${STACKS[@]}"

if [[ "${STACKS[@]}" =~ "${OldWorkflowID}" ]]
then
  aws s3 rm s3://udapeople-3eab0 --recursive
  aws cloudformation delete-stack --stack-name backendStack
  aws cloudformation delete-stack --stack-name frontendStack
fi



Bad value for --query Exports[?Name==\'WorkflowID\'].Value: Bad jmespath expression: Unknown token \:
Exports[?Name==\'WorkflowID\'].Value
               ^
OldWorkflowID:
Stack names: frontendStack backendStack
delete: s3://udapeople-3eab0/1ceff9123b66b50a7d3cb5b221160855.eot
delete: s3://udapeople-3eab0/0.322360032ce8562cd0d2.css
delete: s3://udapeople-3eab0/2.b607f0d50fe00913f687.css.map
delete: s3://udapeople-3eab0/0.322360032ce8562cd0d2.css.map
delete: s3://udapeople-3eab0/2.b607f0d50fe00913f687.css
delete: s3://udapeople-3eab0/224b6d9d16679dab02826a2a7ea705eb.ttf
delete: s3://udapeople-3eab0/a1b7db51fcfbbb62bb08.js
delete: s3://udapeople-3eab0/2fa692cd55bfc83c228c0ab271e4388c.woff
delete: s3://udapeople-3eab0/837f60201305fbf23219.js
delete: s3://udapeople-3eab0/bundle.js
delete: s3://udapeople-3eab0/a1b7db51fcfbbb62bb08.js.map
delete: s3://udapeople-3eab0/837f60201305fbf23219.js.map
delete: s3://udapeople-3eab0/images/6ab15f4239b7d187935cf6f4ec3313bf-kiwi.svg
delete: s3://udapeople-3eab0/index.html
delete: s3://udapeople-3eab0/bundle.js.map
CircleCI received exit code 0
```

- https://app.circleci.com/pipelines/github/hesham98/cdond-c3-projectstarter/195/workflows/5faf9f4d-9e5b-4345-9695-343414932323/jobs/300

Well done! 👏

💡 The purpose of this job is to show **automated infrastructure clean-up** to eliminate human errors and avoid unnecessary cost of unused or invalid infrastructure.
So, it is quite important to **see** the old files being deleted by the **automated infrastructure clean-up**. 👍

✅ URL04: Evidence of healthy back-end application. [`URL04_SCREENSHOT`]



```
{"status":"ok","environment":"production"}
```

The screenshot shows a full **backend URL04** open in a web browser! 👏
And, the response is perfect!
Good job! 👏

Great job on this section! 👏
Each point is achieved correctly! 👍

Glad to see that you have fulfilled all the requirements and specifications! 👏

Section 3: Turn Errors into Sirens

Evidence of Prometheus Server. [URL05_SCREENSHOT].

Evidence that Prometheus is monitoring memory, cpu and disk usage of EC2 instances. [SCREENSHOT11]

Evidence that Prometheus and AlertManager send alerts when certain conditions exist in the EC2 instance. [SCREENSHOT12]

PS
This section on Prometheus server has already been approved! 👏

DOWNLOAD PROJECT

3 CODE REVIEW COMMENTS ›

RETURN TO PATH

Rate this review

START