

# Give Your Application Auto-Deploy Superpowers

<b>Deploying Working, Trustworthy Software</b>	<b>1</b>
Public Git Repository	1
Recorded Project Progress	1
CircleCI Project Creation	1
Implement build Workflow Jobs	1
[SCREENSHOT01]	2
Implement test Workflow Jobs	3
[SCREENSHOT02]	4
Implement Scan Workflow Jobs	5
[SCREENSHOT03]	6
Build Alerts	7
[SCREENSHOT04]	7
Create the Infrastructure	8
[SCREENSHOT05]	9
Configure Infrastructure	9
[URL01]	10
Deploy Phase	10
[URL02]	11
Smoke Test	11
[SCREENSHOT06]	11
Rollback after Failed Smoke Test	12
[SCREENSHOT07]	12
Promotion Phase	12
[SCREENSHOT08]	13
[URL03_SCREENSHOT]	13
[URL04_SCREENSHOT]	14
Cleanup Phase	14
[SCREENSHOT09]	15
Filter Non-Master Branches	15
[SCREENSHOT10]	15
Run the Application	15
Set-up Prometheus Server	16
Set-up Back End Monitoring	16
[SCREENSHOT11]	17
Node Free Memory Back End Instance Graph	17
Node CPU Usage Back End Instance	17
Node Disk Usage Back End Instance Graph	18
[URL05_SCREENSHOT]	19
Setup Alerts	19



# Deploying Working, Trustworthy Software

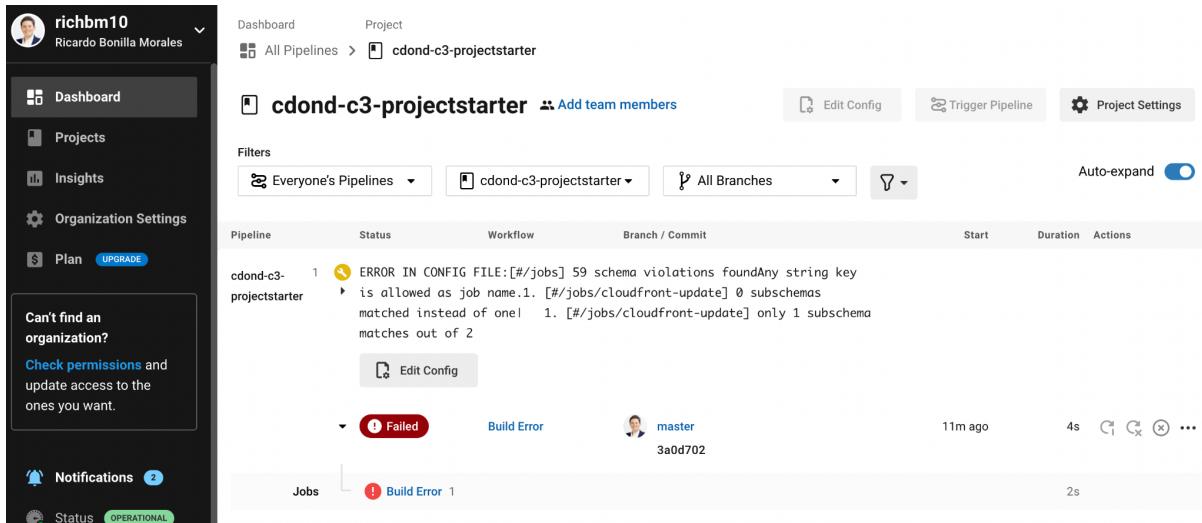
## Public Git Repository

[Forked richbm10 project repository.](#)

## Recorded Project Progress

### CircleCI Project Creation

I created a new CircleCI project from the initial forked repository. These are the first failures that were displayed after CircleCI created the project.



The screenshot shows the CircleCI dashboard for the project 'cdond-c3-projectstarter'. The pipeline 'cdond-c3-projectstarter' has one job that failed. The error message is: "ERROR IN CONFIG FILE: [#/jobs] 59 schema violations foundAny string key is allowed as job name.1. [#/jobs/cloudfront-update] 0 subschemas matched instead of one 1. [#/jobs/cloudfront-update] only 1 subschema matches out of 2". The job was started 11m ago and completed 4s ago. It was run on branch master with commit 3a0d702. The status bar at the bottom indicates 'OPERATIONAL'.

As we see, there is an initial build error in the circleci config file.

### Implement build Workflow Jobs

I renamed the provided circleci config.yml file into config-sample.yml, to be able to start from scratch my own circleci config.

Following the project instructions, I wrote both build jobs: build-frontend and build-backend. Each of them use the circleci/node:13.8.0 docker image and run the package.json build script.

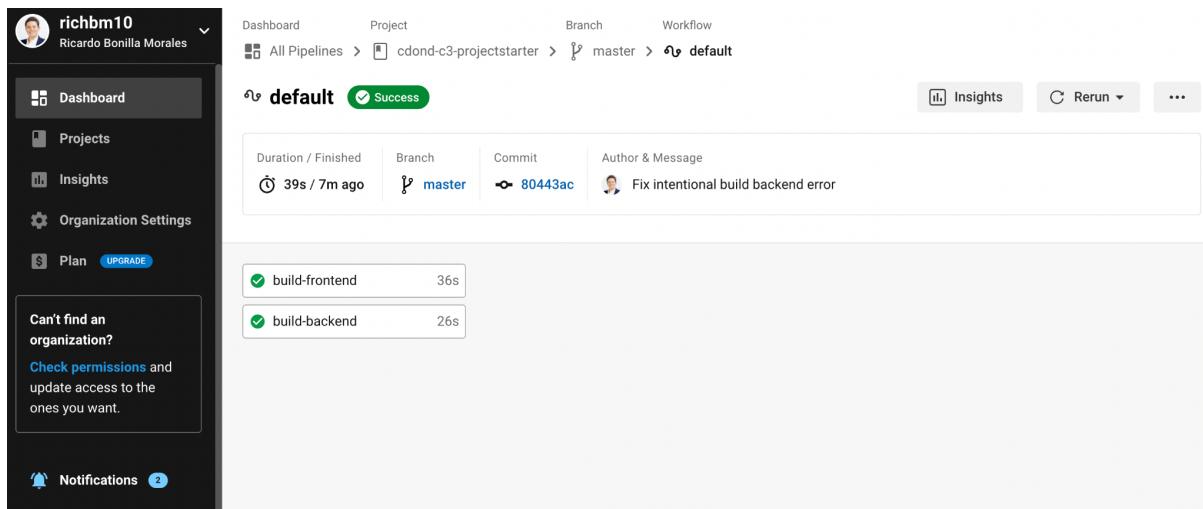
When I pushed the [git commit](#) including the changes, the result was a failed build on the build-backend job.

The screenshot shows the CircleCI dashboard for a user named 'richbm10'. The main navigation bar includes 'Dashboard', 'Project', 'Branch', and 'Workflow'. Below this, the pipeline path is shown: 'All Pipelines > cdond-c3-projectstarter > master > default'. The 'default' branch is highlighted with a red 'Failed' status. On the right, there are 'Insights', 'Rerun', and a three-dot menu. The main content area displays the build details for the 'default' branch. It shows a duration of '38s / 15m ago', a commit hash 'fd8d0c9', and a message from the author: 'Rename boilerplate circle ci config and write a new config with build-frontend and build-backend jobs'. Below this, two jobs are listed: 'build-frontend' (green, 34s) and 'build-backend' (red, 27s). A sidebar on the left provides links for 'Dashboard', 'Projects', 'Insights', 'Organization Settings', and 'Plan'.

[SCREENSHOT01]

This screenshot shows a detailed view of a failed build for the 'build-backend' job within the 'default' branch. The top navigation bar and pipeline path are identical to the previous screenshot. The 'build-backend' job is shown with a red 'Failed' status. The build details include a duration of '27s / 13m ago', an executor/resource class of 'Docker / Large', and a commit hash 'fd8d0c9'. The message from the author is the same. Below the build details, a 'STEPS' tab is selected, showing a list of parallel runs. One run is visible, starting with 'Spin up environment' (green, 1s), followed by 'Preparing environment variables' (green, 0s), 'Checkout code' (green, 0s), and finally 'Build back-end' (red, 25s). The 'Build back-end' step shows a log output with several errors, including npm audit findings and TypeScript compilation issues. The bottom section of the screenshot shows the standard CircleCI sidebar with links for 'Dashboard', 'Projects', 'Insights', 'Organization Settings', and 'Plan'.

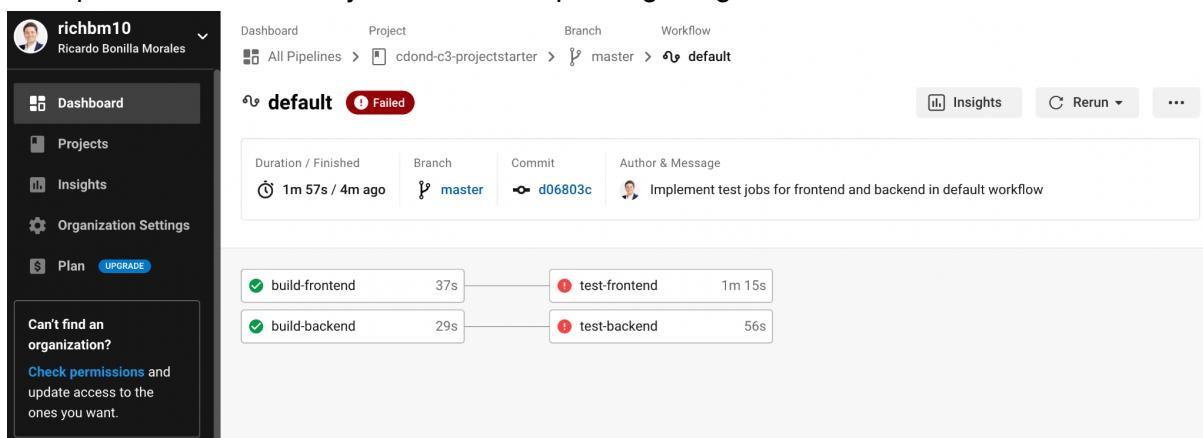
Then, I fixed the intentional error in the backend source code, and pushed the [git commit](#) change. So now both workflow jobs were successful.



The screenshot shows the CircleCI dashboard for a project named 'cdond-c3-projectstarter' on the 'master' branch. The workflow named 'default' is shown as 'Success'. The run was completed 39s ago. The commit hash is 80443ac, and the author message is 'Fix intentional build backend error'. The pipeline consists of two steps: 'build-frontend' (36s) and 'build-backend' (26s), both of which are marked as green (successful).

## Implement test Workflow Jobs

I wrote both testing jobs for frontend and backend source code in this [git commit](#). Each of them (test-frontend and test-backend) have a dependency on their respective build job. As expected both unit test jobs failed after pushing the git commit.



The screenshot shows the CircleCI dashboard for the same project and branch. The 'default' workflow is now 'Failed'. The run was completed 1m 57s ago. The commit hash is d06803c, and the author message is 'Implement test jobs for frontend and backend in default workflow'. The pipeline now includes four steps: 'build-frontend' (37s), 'test-frontend' (1m 15s), 'build-backend' (29s), and 'test-backend' (56s). The 'test-frontend' and 'test-backend' steps are marked as red (failed).

## [SCREENSHOT02]

The screenshot shows two separate test logs from CircleCI:

- Test front-end:** This log shows a table of test results and a detailed command-line output. The table includes columns for file path, status (0 or 100), and various metrics like coverage and execution time. The command-line output shows Jest running tests, failing at the typescript-react-redux-boilerplate@1.0.0 test script, and providing a log file.
- Test back-end:** This log shows a table of test results and a detailed command-line output. The table includes columns for file path, status (0 or 1), and various metrics. The command-line output shows Jest running tests, failing at the glee2@1.0.0 test script, and providing a log file.

```

[Frontend Test Log]
116 containers/Employee/components/Employees | 0 | 0 | 0 | 0 | ... 44,247,257,262
117 index.tsx | 0 | 0 | 0 | 0 | ...
118 containers/Employee/components/ViewEmployee | 0 | 0 | 0 | 0 | ...
119 index.tsx | 0 | 0 | 0 | 0 | ... 95,498,508,513
120 containers/Employee/models | 0 | 0 | 0 | 0 | ...
121 EmployeeModel.ts | 0 | 0 | 0 | 0 | ...
122 index.ts | 0 | 100 | 100 | 0 | ...
123 containers/Employee/reducer | 38.89 | 33.33 | 37.5 | 38.89 | ...
124 employees.ts | 50 | 33.33 | 42.86 | 50 | ... 28,29,30,36,62
125 sampleData.ts | 0 | 100 | 0 | 0 | ...
126 containers/Employee/services | 30 | 0 | 10 | 30.77 | ...
127 employees.ts | 30 | 0 | 10 | 30.77 | ... 50,54,57,61,62
128
129
130 Test Suites: 1 failed, 4 passed, 5 total
131 Tests: 1 failed, 11 passed, 12 total
132 Snapshots: 0 total
133 Time: 48.673s
134 Ran all test suites.
135 npm ERR! code ELIFECYCLE
136 npm ERR! errno 1
137 npm ERR! typescript-react-redux-boilerplate@1.0.0 test: `jest`
138 npm ERR! Exit status 1
139 npm ERR!
140 npm ERR! Failed at the typescript-react-redux-boilerplate@1.0.0 test script.
141 npm ERR! This is probably not a problem with npm. There is likely additional logging output above.
142
143 npm ERR! A complete log of this run can be found in:
144 npm ERR!   /home/circleci/.npm/_logs/2022-01-22T18_37_45_425Z-debug.log

[Backend Test Log]
282 Number of calls: 1
283
284 33 |
285 34 | // Assert
286 > 35 | expect(employeeRepository.findById).toBeCalledWith(100);
287 | ^
288 36 | expect(employeeRepository.save).toBeCalled();
289 37 | });
290 38 | });
291
292 at src/modules/domain/employees/commands/handlers/employee-activator.handler.spec.ts:35:43
293 at fulfilled (src/modules/domain/employees/commands/handlers/employee-activator.handler.spec.ts:5:58)
294
295
296 Test Suites: 1 failed, 50 passed, 51 total
297 Tests: 1 failed, 78 passed, 79 total
298 Snapshots: 0 total
299 Time: 34.62s
300 Ran all test suites.
301 npm ERR! code ELIFECYCLE
302 npm ERR! errno 1
303 npm ERR! glee2@1.0.0 test: `jest --runInBand`
304 npm ERR! Exit status 1
305 npm ERR!
306 npm ERR! Failed at the glee2@1.0.0 test script.
307 npm ERR! This is probably not a problem with npm. There is likely additional logging output above.
308
309 npm ERR! A complete log of this run can be found in:
310 npm ERR!   /home/circleci/.npm/_logs/2022-01-22T18_37_19_766Z-debug.log
311

```

I then fixed the intentional errors on the backend and frontend unit tests, and pushed my [git commit](#). Both unit testing jobs succeed in the workflow.

The screenshot shows the CircleCI web interface for a pipeline named 'default'. The pipeline status is 'Success'. Key details include:

- Duration / Finished:** 1m 49s / 3m ago
- Branch:** master
- Commit:** 3c80cd1
- Author & Message:** Fix intentional errors to succeed unit tests on frontend and backend

The pipeline graph shows two stages: 'build-frontend' and 'build-backend', each with a dependency on its corresponding 'test-' stage. All stages are marked as green (success).

```
graph LR; buildFrontend[build-frontend] --> testFrontend[test-frontend]; buildBackend[build-backend] --> testBackend[test-backend];
```

## Implement Scan Workflow Jobs

I wrote the scan-frontend and scan-backend jobs to check for any vulnerability on each source code. Each job has a dependency on its respective previous test job.

As expected, the result after pushing the git commit was a failure on each job due to vulnerabilities.

The screenshot shows the CircleCI web interface for the same 'default' pipeline, but now it is 'Failed'. The key details are:

- Duration / Finished:** 1m 35s / 4m ago
- Branch:** master
- Commit:** da5bcc2
- Author & Message:** Change scan-backend job dependency

The pipeline graph shows the same structure as before, but the final 'scan-' stages are now marked as red (failure). The 'scan-frontend' job took 7s and the 'scan-backend' job took 9s.

```
graph LR; buildFrontend[build-frontend] --> testFrontend[test-frontend]; buildBackend[build-backend] --> testBackend[test-backend]; testFrontend --> scanFrontend[scan-frontend]; testBackend --> scanBackend[scan-backend];
```

## [SCREENSHOT03]

The screenshot displays two parallel runs in the CircleCI interface. The top run, titled 'Scan front-end', shows a command-line log with several npm ERR! errors related to missing dependencies like 'oauth-sign@^0.9.0'. The bottom run, titled 'Scan back-end', shows a large log of vulnerabilities found in scanned packages, with a note that the output is too large to display in the browser.

**Parallel runs**

1 / 16 parallel runs

**Scan front-end**

```

1 #!/bin/bash -eo pipefail
2 cd frontend
3 npm audit --audit-level=critical
4
5 npm ERR! code ELOCKVERIFY
6 npm ERR! Errors were found in your package-lock.json, run npm install to fix them.
7 npm ERR!     Missing: oauth-sign@^0.9.0
8
9 npm ERR! A complete log of this run can be found in:
10 npm ERR!     /home/circleci/.npm/_logs/2022-01-22T19_12_49_190Z-debug.log
11
12
13 Exited with code exit status 1
14 CircleCI received exit code 1

```

**Scan back-end**

Your output is too large to display in the browser.  
Only the last 400000 characters are displayed.

Download the full output as a file

```

5565 |   | in node-fetch
5566 |   | Package | node-fetch
5567 |   | Patched in | >=2.6.1
5568 |   | Dependency of | @nestjs/core
5569 |   | Path | @nestjs/core > @nuxtjs/opencollective > node-fetch
5570 |   | More info | https://github.com/advisories/GHSA-w7rc-rwvf-8q5r
5571 |
5572 |
5573 |
5574 |
5575 |
5576 |
5577 |
5578 | High | node-fetch is vulnerable to Exposure of Sensitive
5579 |     | Information to an Unauthorized Actor
5580 |
5581 | Package | node-fetch
5582 |
5583 | Patched in | >=3.1.1
5584 |
5585 | Dependency of | @nestjs/core
5586 |
5587 | Path | @nestjs/core > @nuxtjs/opencollective > node-fetch
5588 |
5589 |
5590 |
5591 |
5592 found 550 vulnerabilities (2 low, 175 moderate, 316 high, 57 critical) in 1389 scanned packages
run `npm audit fix` to fix 442 of them.

```

To fix the vulnerabilities I had to add the following audit fix command on each scan job:

```
npm audit fix --audit-level=critical --force
```

Also, for the case of the scan-frontend job, it required the installation of a previous node dependency:

```
npm install oauth-sign@^0.9.0
```

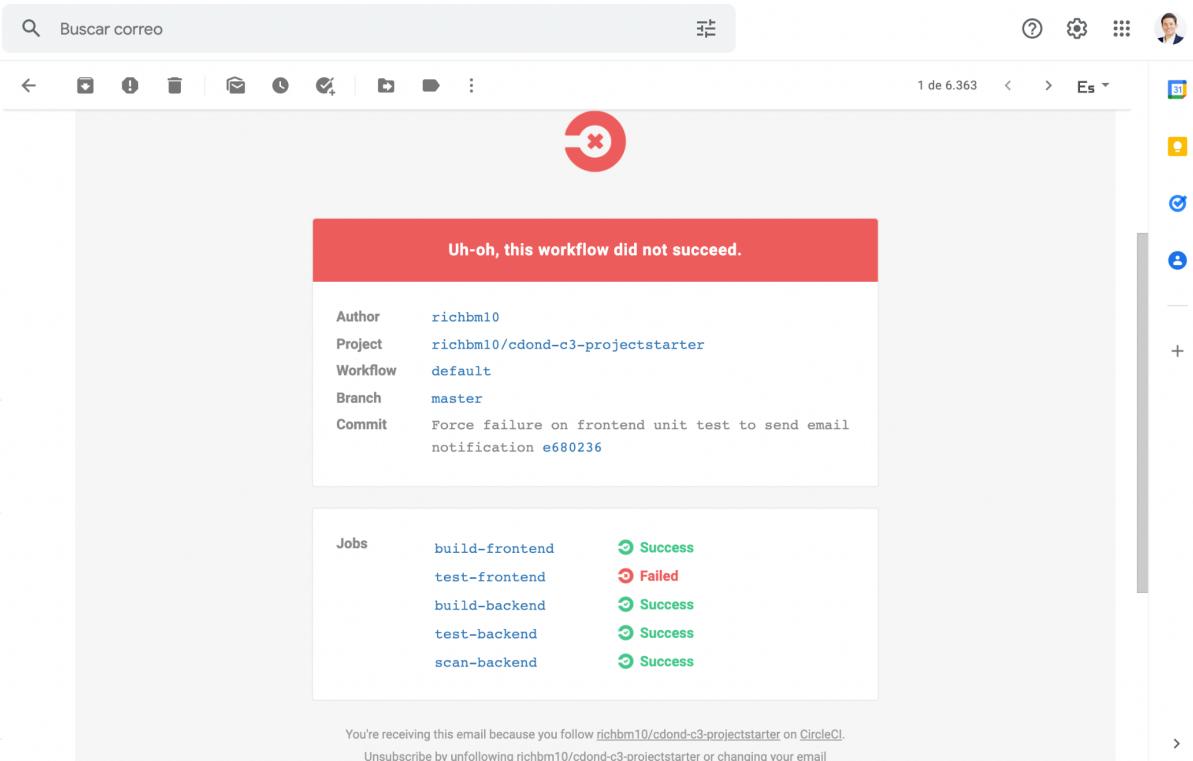
The screenshot shows the CircleCI dashboard. On the left is a sidebar with a user profile for 'richbm10' (Ricardo Bonilla Morales), navigation links for Dashboard, Projects, Insights, Organization Settings, Plan (with an 'UPGRADE' button), Notifications (2 notifications), Status (OPERATIONAL), and Docs. The main area displays a workflow named 'default' which has just succeeded. It shows the duration as 2m 16s / 3m ago, the branch as master, and the commit hash as 31bdb64. The commit message is 'Add missing dependency to pass audit fix command in scan-frontend job'. Below this is a horizontal timeline of tasks: build-frontend (27s), test-frontend (50s), scan-frontend (52s), build-backend (21s), test-backend (42s), and scan-backend (37s). All tasks are marked with green checkmarks.

## Build Alerts

As CircleCI has auto enabled email notifications, my email account received email notifications for each failure on the workflow.

[SCREENSHOT04]

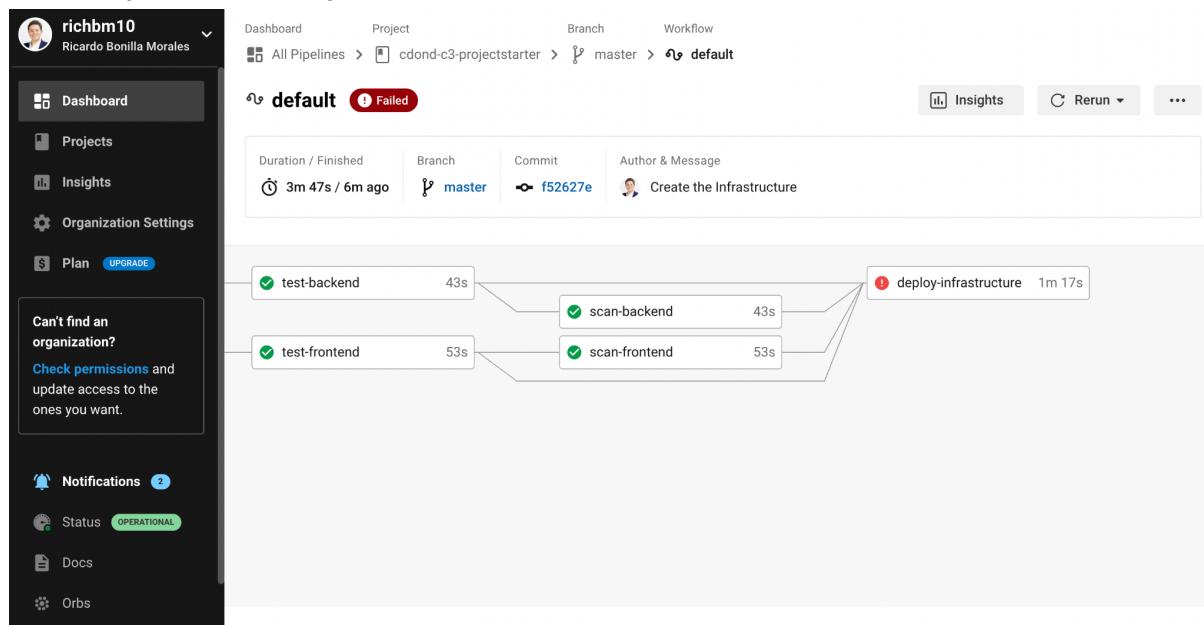
The screenshot shows an email inbox interface. At the top, there's a search bar with 'Buscar correo' and various toolbar icons. The main list shows one email from 'CircleCI Builds <builds@circleci.com>' with the subject '[Workflow] Failed: richbm10/cdond-c3-projectstarter on master / default (e680236)'. The email was received 4 minutes ago at 14:33. The message body starts with a large red circle containing a white 'X'. Below it, a red banner says 'Uh-oh, this workflow did not succeed.' The message details the failure: 'Author: richbm10', 'Project: richbm10/cdond-c3-projectstarter', 'Workflow: default', 'Branch: master', and 'Commit: Force failure on frontend unit test to send email notification e680236'.



## Create the Infrastructure

Following the project guidelines I did the AWS infrastructure and CircleCI setup. After that, I wrote the circleci job to deploy-infrastructure, which is responsible for running the cloudformation templates for backend and frontend infrastructure.

As expected, after pushing the git commit with the new workflow job, a failure was thrown on the deploy-infrastructure job.



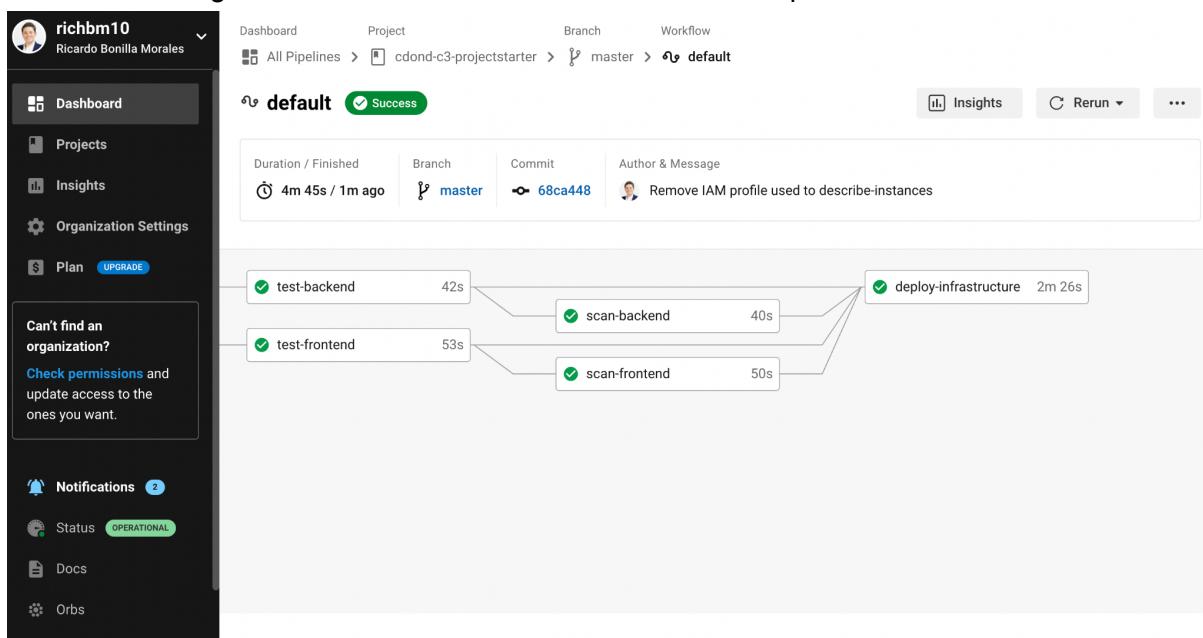
## [SCREENSHOT05]

The screenshot shows a CircleCI pipeline log for a workflow named 'Ensure backend infrastructure exist'. The log includes the following steps:

- Preparing environment variables (0s)
- Checkout code (0s)
- yum install -y tar gzip (6s)
- Ensure backend infrastructure exist (1m 6s):

```
1 #!/bin/bash -eo pipefail
2 aws cloudformation deploy \
3   --template-file .circleci/files/backend.yml \
4   --stack-name "udapeople-backend-${CIRCLE_WORKFLOW_ID:0:7}" \
5   --parameter-overrides ID="${CIRCLE_WORKFLOW_ID:0:7}" \
6   --tags udacity-project=udapeople
7
8 Waiting for changeset to be created..
9 Waiting for stack create/update to complete
10
11 Failed to create/update the stack. Run the following command
12 to fetch the list of events leading up to the failure
13 aws cloudformation describe-stack-events --stack-name udapeople-backend-bc1649a
14
15 Exited with code exit status 255
16 CircleCI received exit code 255
```
- Destroy environment (0s)
- Destroy environment (0s)

After adding my KeyName to the EC2 instance in the backend template file, and assigning an AMI of the region us-east-1. The error was resolved as expected.



## Configure Infrastructure

Then I implemented the circledci job responsible for installing the required dependencies and ran the ansible playbook to configure the EC2 instance.

```

9 54.147.142.203
10 PLAY [configuration play.] ****
11
12 TASK [check for python] ****
13 [WARNING]: raw module does not support the environment keyword
14 ok: [54.147.142.203]
15
16 TASK [install python for Ansible.] ****
17 skipping: [54.147.142.203]
18
19 TASK [configure-server : update apt packages.] ****
20 changed: [54.147.142.203]
21
22 TASK [configure-server : upgrade packages] ****
23 [WARNING]: The value True (type bool) in a string field was converted to 'True'
24 (type string). If this does not look like what you expect, quote the entire
25 value to ensure it does not change.
26 changed: [54.147.142.203]
27
28 TASK [configure-server : remove dependencies that are no longer required] ****
29 ok: [54.147.142.203]
30
31 TASK [configure-server : install dependencies.] ****
32 changed: [54.147.142.203]
33
34 TASK [configure-server : install pm2] ****
35 changed: [54.147.142.203]
36
37 PLAY RECAP ****
38 54.147.142.203 : ok=6    changed=4   unreachable=0   failed=0    skipped=1   rescued=0   ignored=0
39
40 CircleCI received exit code 0

```

[URL01]

URL to my public GitHub [repository](#).

## Deploy Phase

I started the deploy phase implementation by writing the run-migrations job and including it as part of the workflow. After a success in the workflow I sent the migration status to a 3rd party key-value store, in this case I chose [kvdb.io](#).

```

1 #!/bin/bash -eo pipefail
2 cd backend
3 npm install
4 # Run and save the migration output
5 npm run migrations > migrations_dump.txt
6
7 npm WARN @nestjs/cqrs@6.1.0 requires a peer of reflect-metadata@0.1.12 but none is installed. You must install peer dependency
8 npm WARN ajv-keywords@3.4.1 requires a peer of ajv@^6.9.1 but none is installed. You must install peer dependencies yourself
9 npm WARN winston-slack-webhook@0.1.6 requires a peer of winston@2.x.x but none is installed. You must install peer dependency
10 npm WARN glee@2.1.0 No repository field.
11 npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node_modules/fsevents):
12 npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.12: wanted {"os":"darwin","arch":"any"}
13 audited 1389 packages in 7.396s
14
15 24 packages are looking for funding
16   run `npm fund` for details
17
18 found 550 vulnerabilities (2 low, 175 moderate, 316 high, 57 critical)
19   run `npm audit fix` to fix them, or `npm audit` for details
20
21 CircleCI received exit code 0

```

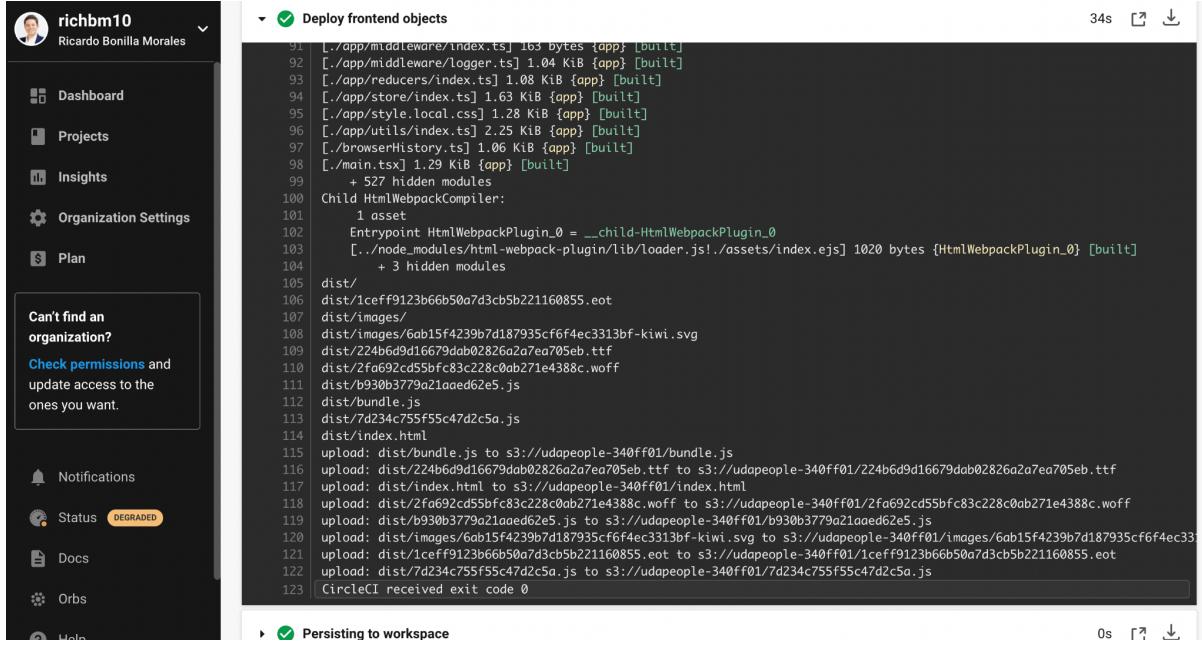
  

```

1 #!/bin/bash -eo pipefail
2 if grep -q "has been executed successfully." ~/project/backend/migrations_dump.txt
3 then
4   curl https://kvdb.io/7NZ6WGXV6a5TB9Qk9gkDpA/migration_${CIRCLE_WORKFLOW_ID:0:7} -d '1'
5 fi
6
7 CircleCI received exit code 0

```

Then I wrote down the job for the front end deployment, which was in charge of persisting the distribution folder on the workspace and copying the folder in the created S3 bucket for the front end static hosting.



The screenshot shows the CircleCI interface. On the left, there's a sidebar with user information (richbm10, Ricardo Bonilla Morales) and navigation links: Dashboard, Projects, Insights, Organization Settings, Plan, Notifications (1), Status (DEGRADED), Docs, Orbs, and Help. A message box says "Can't find an organization? Check permissions and update access to the ones you want." Below the sidebar are buttons for Notifications, Status (DEGRADED), Docs, Orbs, and Help.

The main area displays deployment logs:

```

    Deploy frontend objects
    34s ✓ ⌂ ⏪ ⏴
    91 ['./app/middleware/index.ts'] 1.63 bytes {app} [built]
    92 ['./app/middleware/logger.ts'] 1.04 KiB {app} [built]
    93 ['./app/reducers/index.ts'] 1.08 KiB {app} [built]
    94 ['./app/store/index.ts'] 1.63 KiB {app} [built]
    95 ['./app/style.local.css'] 1.28 KiB {app} [built]
    96 ['./app/utils/index.ts'] 2.25 KiB {app} [built]
    97 ['./browserHistory.ts'] 1.06 KiB {app} [built]
    98 ['./main.tsx'] 1.29 KiB {app} [built]
    + 527 hidden modules
  Child HtmlWebpackPlugin:
  100   1 asset
  101   Entrypoint HtmlWebpackPlugin_0 = __child__HtmlWebpackPlugin_0
  102     ['./node_modules/html-webpack-plugin/lib/loader.js!./assets/index.ejs'] 1020 bytes {HtmlWebpackPlugin_0} [built]
  103       + 3 hidden modules
  104 dist/
  105 dist/iceff9123b66b50a7d3cb5b221160855.eot
  106 dist/images/
  107 dist/images/6ab15f4239b7d187935cf6f4ec3313bf-kiwi.svg
  108 dist/224b6d9d16679db02826a27ea705eb.ttf
  109 dist/2fa692cd55bfc83c228c0ab271e4388c.woff
  110 dist/b930b3779a21aaed62e5.js
  111 dist/bundle.js
  112 dist/7d234c755f5c47d2c5a.js
  113 dist/index.html
  114 upload: dist/bundle.js to s3://udapeople-340ff01/bundle.js
  115 upload: dist/224b6d9d16679db02826a27ea705eb.ttf to s3://udapeople-340ff01/224b6d9d16679db02826a27ea705eb.ttf
  116 upload: dist/index.html to s3://udapeople-340ff01/index.html
  117 upload: dist/2fa692cd55bfc83c228c0ab271e4388c.woff to s3://udapeople-340ff01/2fa692cd55bfc83c228c0ab271e4388c.woff
  118 upload: dist/b930b3779a21aaed62e5.js to s3://udapeople-340ff01/b930b3779a21aaed62e5.js
  119 upload: dist/images/6ab15f4239b7d187935cf6f4ec3313bf-kiwi.svg to s3://udapeople-340ff01/images/6ab15f4239b7d187935cf6f4ec3313bf-kiwi.svg
  120 upload: dist/iceff9123b66b50a7d3cb5b221160855.eot to s3://udapeople-340ff01/images/6ab15f4239b7d187935cf6f4ec3313bf-kiwi.eot
  121 upload: dist/7d234c755f5c47d2c5a.js to s3://udapeople-340ff01/7d234c755f5c47d2c5a.js
  122 upload: dist/7d234c755f5c47d2c5a.js to s3://udapeople-340ff01/7d234c755f5c47d2c5a.js
  123 CircleCI received exit code 0

```

Persisting to workspace 0s ⌂ ⏪ ⏴

## [URL02]

URL to the S3 Bucket public access for the static hosting:

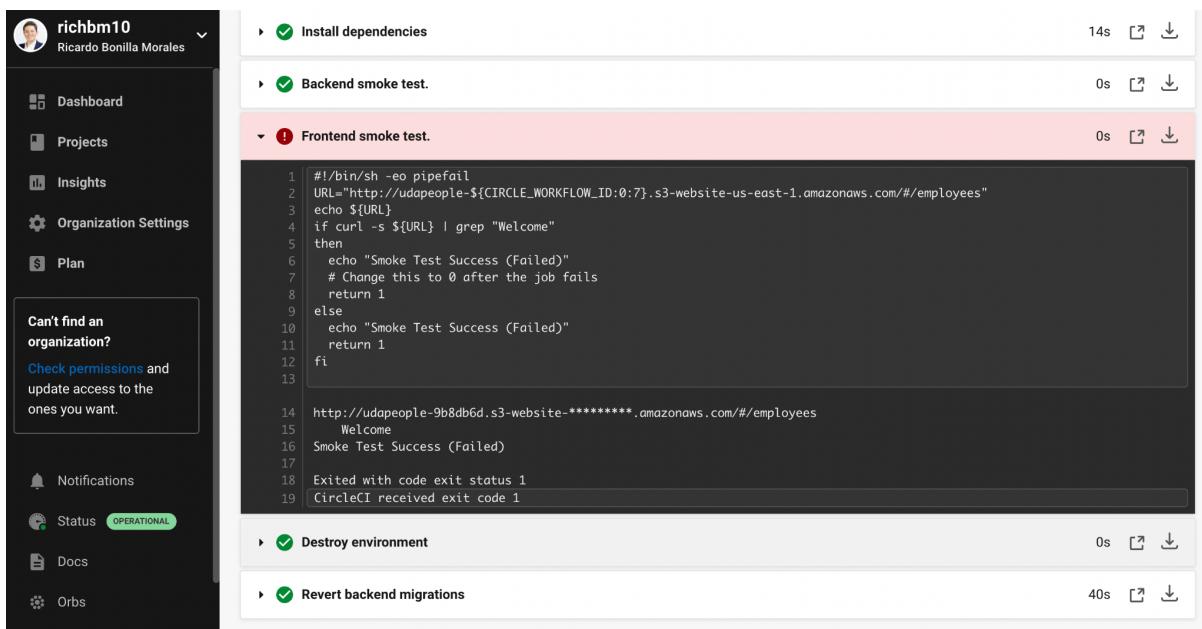
<http://udapeople-340ff01.s3-website-us-east-1.amazonaws.com>.

## Smoke Test

I wrote a job in charge of the frontend and backend smoke tests. On each test in case of failure I have the respective on\_fail handler to destroy frontend and backend infrastructure, as well as rollback the migrations of the RDS.

First as expected, following the project guidelines, the frontend smoke test returned 1, thus triggered a failure in the job.

## [SCREENSHOT06]



The screenshot shows the CircleCI interface. On the left, there's a sidebar with user information (richbm10, Ricardo Bonilla Morales) and navigation links: Dashboard, Projects, Insights, Organization Settings, Plan, Notifications (1), Status (OPERATIONAL), Docs, and Orbs. A message box says "Can't find an organization? Check permissions and update access to the ones you want."

The main area displays smoke test logs:

```

    Install dependencies
    14s ✓ ⌂ ⏪ ⏴
    Backend smoke test.
    0s ✓ ⌂ ⏪ ⏴
    Frontend smoke test.
    0s ✓ ⌂ ⏪ ⏴
    1 #!/bin/sh -eo pipefail
    2 URL="http://udapeople-$CIRCLE_WORKFLOW_ID:0.7.s3-website-us-east-1.amazonaws.com/#/employees"
    3 echo $URL
    4 if curl -s $URL | grep "Welcome"
    5 then
    6   echo "Smoke Test Success (Failed)"
    7   # Change this to 0 after the job fails
    8   return 1
    9 else
    10   echo "Smoke Test Success (Failed)"
    11   return 1
    12 fi
    13
    14 http://udapeople-9b8db6d.s3-website-*****.amazonaws.com/#/employees
    15 Welcome
    16 Smoke Test Success (Failed)
    17
    18 Exited with code exit status 1
    19 CircleCI received exit code 1

```

Destroy environment 0s ⌂ ⏪ ⏴

Revert backend migrations 40s ⌂ ⏪ ⏴

## Rollback after Failed Smoke Test

To handle a failure during the workflow or in the smoke tests jobs I then implemented the destroy environments and revert migrations commands.

### [SCREENSHOT07]

#### Successful rollback after failed smoke test

The screenshot shows the CircleCI web interface. On the left, there's a sidebar with user information (richbm10, Ricardo Bonilla Morales) and navigation links for Dashboard, Projects, Insights, Organization Settings, Plan, Notifications, Status (OPERATIONAL), Docs, and Orbs. A message box says "Can't find an organization? Check permissions and update access to the ones you want." In the main area, there are two logs displayed:

**Frontend smoke test.** (0s)

```
#!/bin/sh -eo pipefail
URL="http://udapeople-${CIRCLE_WORKFLOW_ID:0:7}.s3-website-us-east-1.amazonaws.com/#/employees"
echo ${URL}
if curl -s ${URL} | grep "Welcome"
then
  echo "Smoke Test Success (Failed)"
  # Change this to 0 after the job fails
  return 1
else
  echo "Smoke Test Success (Failed)"
  return 1
fi

http://udapeople-a21f19b.s3-website-*****.amazonaws.com/#/employees
  Welcome
  Smoke Test Success (Failed)
  Exited with code exit status 1
CircleCI received exit code 1
```

**Delete s3 static hosting bucket** (0s)

**Destroy environments** (1s)

```
#!/bin/sh -eo pipefail
echo "Destroying environment: CIRCLE_WORKFLOW_ID:0:7"
aws cloudformation delete-stack --stack-name "udapeople-frontend-${CIRCLE_WORKFLOW_ID:0:7}"
aws cloudformation delete-stack --stack-name "udapeople-backend-${CIRCLE_WORKFLOW_ID:0:7}"

Destroying environment: CIRCLE_WORKFLOW_ID:0:7
```

## Promotion Phase

Following the Blue Green Deployment strategy I implemented the cloudfont-update job in charge of changing the origin domain of the CDN distribution to each new S3 bucket created for the newest deployed front end version.

## [SCREENSHOT08]

The screenshot shows the CircleCI web interface. At the top, there's a navigation bar with tabs: STEPS (selected), TESTS, TIMING (BETA), ARTIFACTS, and RESOURCES (NEW). Below the navigation is a sidebar with links like Dashboard, Projects, Insights, Organization Settings, Plan, and Notifications. A message box says "Can't find an organization? Check permissions and update access to the ones you want." The main area is titled "Parallel runs" and shows a list of steps: "Spin up environment" (1s), "Preparing environment variables" (0s), "Checkout code" (0s), and "Update cloudfront distribution" (3m 42s). The "Update cloudfront distribution" step contains a terminal log:

```

1 #!/bin/bash -eo pipefail
2 aws cloudformation deploy \
3   --template-file .circleci/files/cloudfront.yml \
4   --stack-name InitialStack \
5   --parameter-overrides WorkflowID="${CIRCLE_WORKFLOW_ID:0:7}" \
6   --tags project=udapeople
7
8
9 Waiting for changeset to be created..
10 Waiting for stack create/update to complete
11 Successfully created/updated stack - InitialStack
12 CircleCI received exit code 0

```

## [URL03\_SCREENSHOT]

The screenshot shows the AWS CloudFront console. The left sidebar includes sections for Distributions, Policies, Functions, Telemetry (Monitoring, Alarms, Logs), Reports & analytics (Cache statistics, Popular objects, Top referers, Usage, Viewers), and Security (Origin access identities, Field-level encryption). The main content area is titled "CloudFront > Distributions". It shows a table for "Distributions (1)":

ID	Description	Domain name	Alternate ...	Origins
EGNR1YHKEHCLE	-	d3gpr0odqpg7st.cloudfront.net	-	udapeople-01899fb.s3.amazonaws.com

The screenshot shows a web application interface for managing employees. The title bar includes tabs for 'Cloud DevOps Eng...', 'Instance details | EC...', 'Udacity Project 3 -', 'cloudfront-update | x', 'Commits - richbml...', 'Welcome | x', and '54.158.244.16:303 | x'. Below the tabs, the URL is d3gr0odqpg7st.cloudfront.net/#/employees. The main content area has a header 'View & Manage Employees' with a 'Add New Employee' button. On the left, a sidebar titled 'ACKLEN\AV' shows a navigation menu with 'Employees' selected, along with 'View & Manage', 'Add New', and other options. The main table lists employees with columns: Display Name, Last Name, Email, Tags, Birthdate, Start Date, and Action. A search bar and filter are at the top of the table.

## [URL04\_SCREENSHOT]

The screenshot shows the AWS CloudFront console. The URL is 54.158.244.16:3030/api/status. The response body is {"status": "ok", "environment": "local"}. Below this, the AWS Lambda function configuration is shown, with the function name 'backend-01899fb'. The configuration details include the function ARN, runtime (Node.js 14.x), handler (index.handler), and code size (1.46 MB). The configuration is set to 'Local' environment. The Lambda function is triggered by CloudFront and has a timeout of 300 seconds. The Lambda function is associated with an IAM role 'lambda-role' and has a memory limit of 128 MB. The Lambda function is deployed to a VPC with a subnet of 'subnet-44e20d08' and a security group of 'sg-0f1a23456789012345'. The Lambda function is also associated with an S3 bucket 'cloudfront-update' and a CloudWatch Metrics stream.

## Cleanup Phase

After a successful Blue Green deployment, I implemented the cleanup job to delete the previous environment to reduce infrastructure costs.

## [SCREENSHOT09]

The screenshot shows a CircleCI pipeline interface. On the left is a sidebar with user information (richbm10, Ricardo Bonilla Morales) and navigation links (Dashboard, Projects, Insights, Organization Settings, Plan, Notifications, Status, Docs, Orbs, Help). A message box says "Can't find an organization? Check permissions and update access to the ones you want." Below the sidebar are status indicators for Notifications (green), Status (OPERATIONAL), and Docs.

The main area displays a pipeline step titled "Remove old stacks and files". The step has three sub-tasks: "Attaching workspace" (0s), "Get old stack workflow id" (0s), and "Remove old stacks and files" (2s). The "Remove old stacks and files" step is highlighted with a blue border. It contains a shell script with the following content:

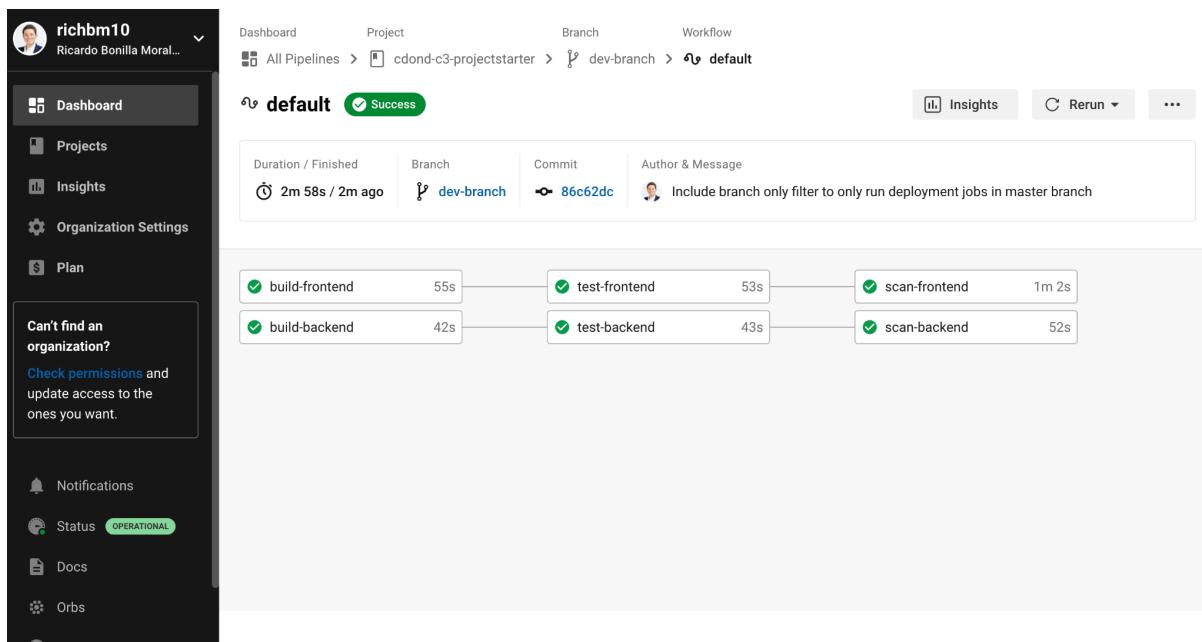
```
#!/bin/bash -eo pipefail
export OldWorkflowID=$(cat ./circleci/OldWorkflowID.txt | tail -1)
echo OldWorkflowID: "${OldWorkflowID}"
echo CIRCLE_WORKFLOW_ID: "${CIRCLE_WORKFLOW_ID:0:7}"
if [[ "${CIRCLE_WORKFLOW_ID:0:7}" != "${OldWorkflowID}" ]]
then
    echo "${OldWorkflowID}!=${CIRCLE_WORKFLOW_ID:0:7} => will delete old version"
    aws s3 rm s3://udapeople-${OldWorkflowID} --recursive
    aws cloudformation delete-stack --stack-name "udapeople-frontend-${OldWorkflowID}"
else
    echo "${OldWorkflowID}==${CIRCLE_WORKFLOW_ID:0:7} => nothing needs to be done..."
fi

OldWorkflowID: 682917e
CIRCLE_WORKFLOW_ID: a917036
682917e!=a917036 => will delete old version
delete: s3://udapeople-682917e/224b6d9d16679dab02826a2a7ea705eb.ttf
delete: s3://udapeople-682917e/1ceff9123b66b50a7d3cb5b221160855.eot
delete: s3://udapeople-682917e/bundle.js
delete: s3://udapeople-682917e/images/6ab15f4239b7d187935cf6f4ec3313bf-kiwi.svg
delete: s3://udapeople-682917e/75481ce96fb7df1ae911.js
delete: s3://udapeople-682917e/index.html
delete: s3://udapeople-682917e/2fa692cd55bfc83c228c0ab271e4388c.woff
delete: s3://udapeople-682917e/7d234c755f55c47d2c5a.js
CircleCI received exit code 0
```

## Filter Non-Master Branches

Following the other considerations guideline of the project, I implemented the filter in the deploy-infrastructure workflow job. This filter blocks this and further job dependencies to be run in a branch different than master.

## [SCREENSHOT10]



## Run the Application

Accessing the application on [URL02](#) and adding new employees.

Display Name	Last Name	Email	Tags	Birthdate	Start Date	Action
iaodsoaisd	asdasd	ricardo.bonilla.moral...		2/10/2022	2/10/2022	<span style="color: green;">More</span>
Ricardo	Morales	ricardo.bonilla.moral...		2/10/2022	2/10/2022	<span style="color: green;">More</span>

## Set-up Prometheus Server

To be able to monitor our back end services we need to configure our AWS Service Discovery with an EC2 Prometheus instance. Then, I had to SSH log into the manually created AWS Ubuntu 20 EC2 instance, download and install prometheus, and start the prometheus server. This instance should have port 9090 and 9091 accessible as inbound rules in its security group.

This AWS EC2 Prometheus server can be reached then by this public DNS:

<http://ec2-54-196-227-233.compute-1.amazonaws.com:9090/>

## Set-up Back End Monitoring

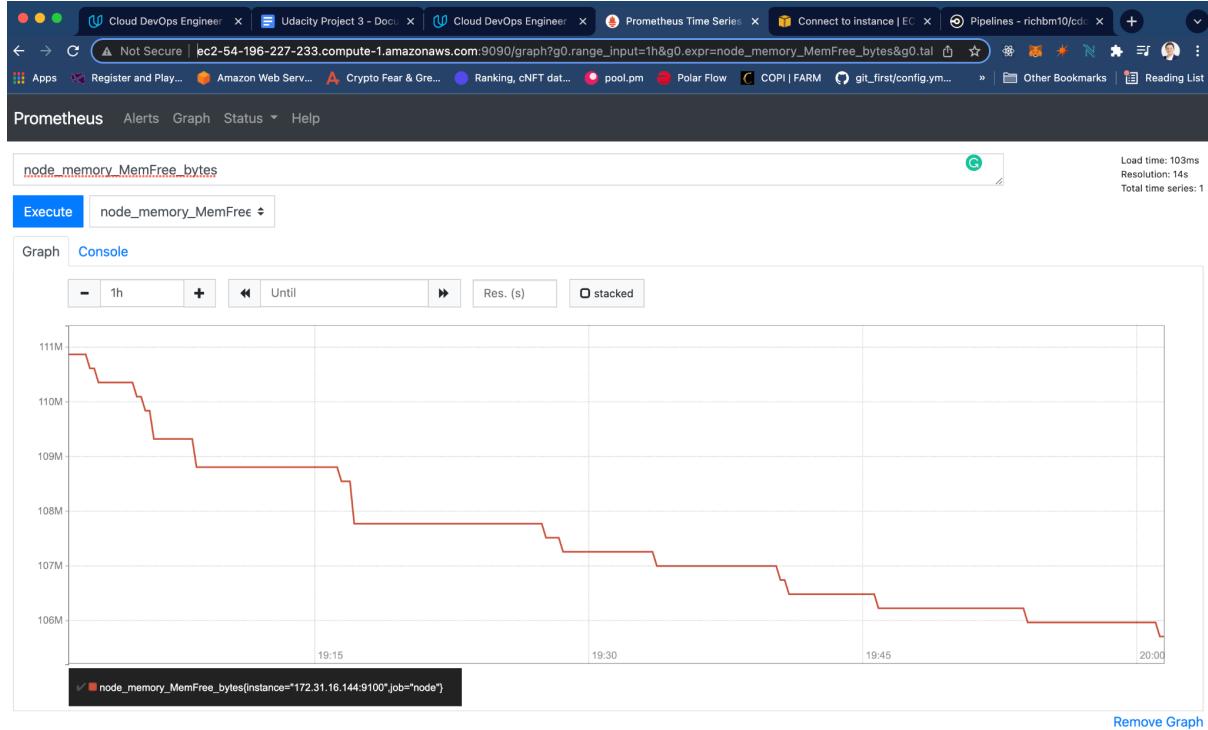
The forked repository includes a `configure-prometheus-node-exporter` ansible role. Then I included this role to the `configure-server.yml` ansible playbook, which installs and starts the prometheus node exporter in our back end service instance.

After including the new ansible role, I had to also update the prometheus server configuration file, so that it was able to pull data from our back end service instance. So I added the following job\_name in the `scrape_configs`:

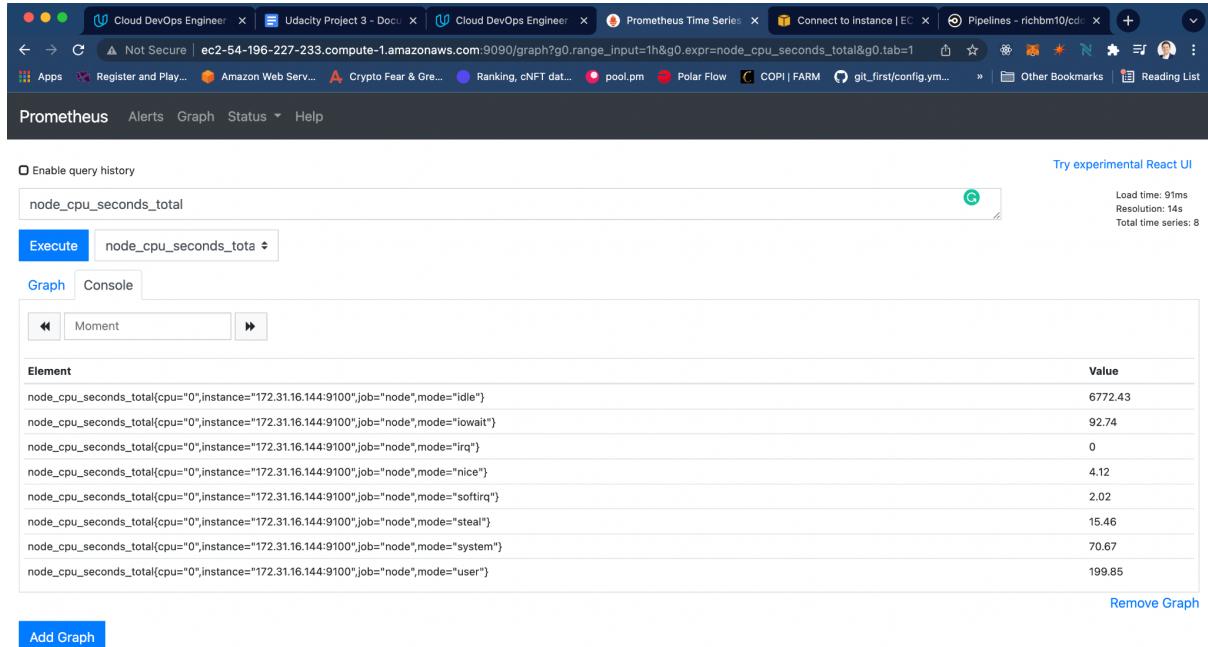
```
- job_name: 'node'
  ec2_sd_configs:
  - region: us-east-1
    access_key: "IAM_KEY"
    secret_key: "IAM_KEY"
    port: 9100
```

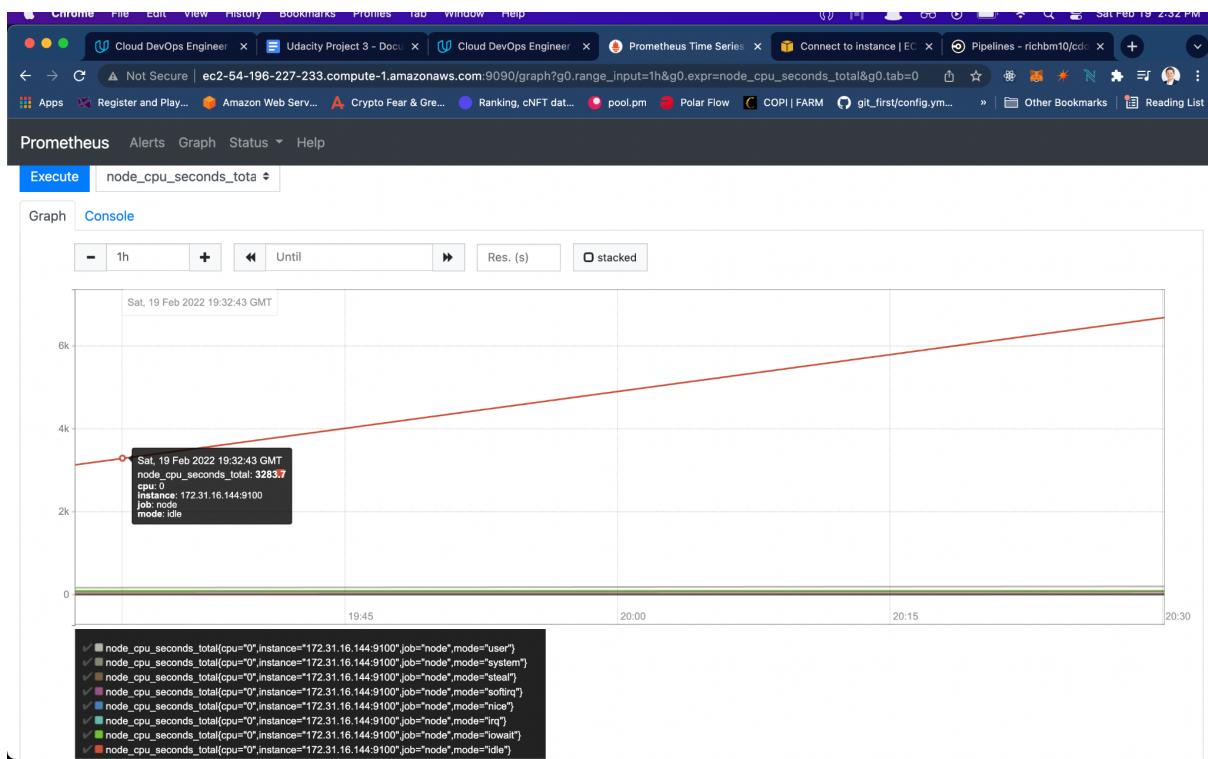
## [SCREENSHOT11]

### Node Free Memory Back End Instance Graph

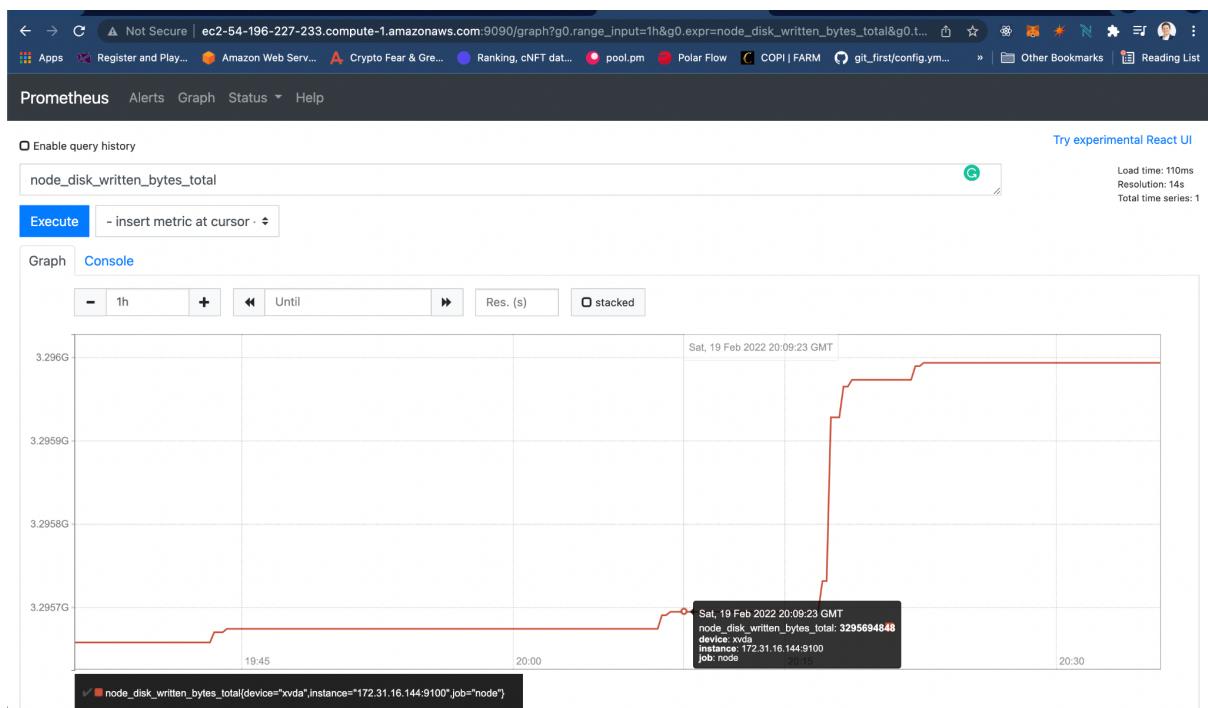


### Node CPU Usage Back End Instance





## Node Disk Usage Back End Instance Graph



## [URL05\_SCREENSHOT]

The screenshot shows the Prometheus Targets page in a browser. The top navigation bar includes tabs for Cloud DevOps Engineer, Udacity Project 3 - Doc..., Cloud DevOps Engineer, Prometheus Time Series, Instances | EC2 Manager, Pipelines - richbm10/cdc, and several other items like Apps, Register and Play..., Amazon Web Serv..., Crypto Fear & Gre..., Ranking, cNFT dat..., pool.pm, Polar Flow, COPI | FARM, git\_first/config.yml, Other Bookmarks, and Reading List.

The main content area is titled "Targets". It has two sections: "node (1/2 up)" and "prometheus (1/1 up)".

**node (1/2 up) show less**

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://172.31.16.144:9100/metrics	UP	instance="172.31.16.144:9100" job="node"	13ms ago	11.4ms	
http://172.31.31.212:9100/metrics	DOWN	instance="172.31.31.212:9100" job="node"	339ms ago	347.3us	Get "http://172.31.31.212:9100/metrics": dial tcp 172.31.31.212:9100: connect: connection refused

**prometheus (1/1 up) show less**

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	822ms ago	5.324ms	

## Setup Alerts

In order to react when any of the conditions in the rules.yml config file is met I had to configure alertmanager. So I downloaded the tar file and updated the alertmanager.yml config file to route alert notifications to my personal email.

Then, to test the alert notifications, I stopped the back end server instance. Then after 3m of having the instance down, the alert was sent to my email.

## [SCREENSHOT12]

The screenshot shows an email interface with the following details:

- From:** ricardo.bonilla.morales@gmail.com
- To:** para mí
- Date:** sábado, 19 feb, 20:55 (hace 11 horas)
- Subject:** 2 alerts for
- View In Alertmanager:** A blue button at the top left of the message content.
- [2] Firing:** The first alert is described as follows:
  - Labels:** alername = InstanceDown, instance = [172.31.20.205:9100](#), job = node, severity = critical
  - Annotations:** description = Instance [172.31.20.205:9100](#) of job node is down, title = Instance [172.31.20.205:9100](#) is down
  - Source:** [Source](#)
- [2] Firing:** The second alert is described as follows:
  - Labels:** alername = InstanceDown, instance = [172.31.81.3:9100](#), job = node, severity = critical
  - Annotations:** description = Instance [172.31.81.3:9100](#) of job node is down, title = Instance [172.31.81.3:9100](#) is down
  - Source:** [Source](#)