

Give Your Application Auto-Deploy Superpowers

Deploying Working, Trustworthy Software	2
Public Git Repository	2
Recorded Project Progress	2
CircleCI Project Creation	2
Implement build Workflow Jobs	2
[SCREENSHOT01]	3
Implement test Workflow Jobs	4
[SCREENSHOT02]	5
Implement Scan Workflow Jobs	6
[SCREENSHOT03]	7
Build Alerts	8
[SCREENSHOT04]	8
Create the Infrastructure	9
[SCREENSHOT05]	10
Configure Infrastructure	10
[URL01]	11
Deploy Phase	11
[URL02]	12
Smoke Test	12
[SCREENSHOT06]	12
Rollback after Failed Smoke Test	13
[SCREENSHOT07]	13
Promotion Phase	13
[SCREENSHOT08]	14
[URL03_SCREENSHOT]	14
[URL04_SCREENSHOT]	15
Cleanup Phase	15
[SCREENSHOT09]	16
Filter Non-Master Branches	16
[SCREENSHOT10]	16
Run the Application	16
Set-up Prometheus Server	17
Set-up Back End Monitoring	17
[SCREENSHOT11]	18
Node Free Memory Back End Instance Graph	18
Node CPU Usage Back End Instance	18
Node Disk Usage Back End Instance Graph	19
[URL05_SCREENSHOT]	20
Setup Alerts	20

Deploying Working, Trustworthy Software

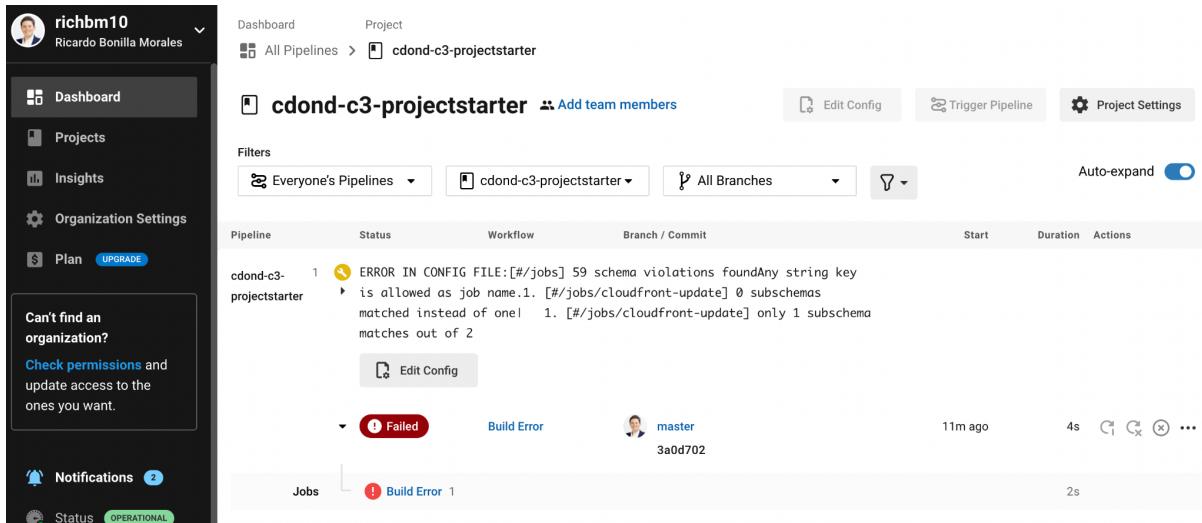
Public Git Repository

[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 has one job named 'cdond-c3-projectstarter' which 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.

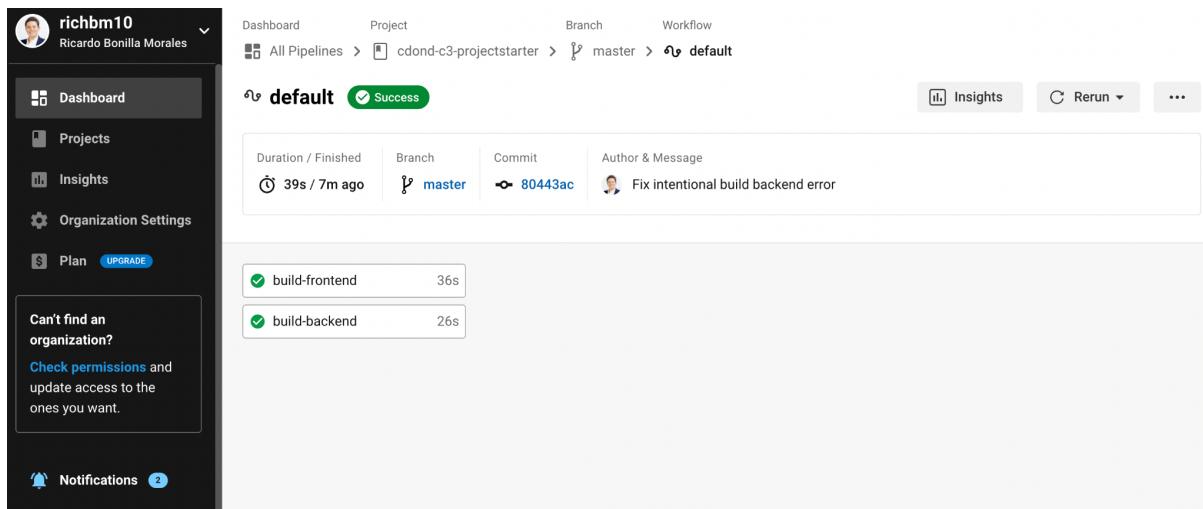
The screenshot shows the CircleCI dashboard for a user named 'richbm10'. The main navigation bar includes 'Dashboard', 'Project', 'Branch', and 'Workflow'. Below this, a breadcrumb trail shows 'All Pipelines > cdond-c3-projectstarter > master > default'. The 'default' pipeline is listed as 'Failed'. The pipeline details card shows a duration of '38s / 15m ago', a branch of 'master', and a commit hash of 'fdbd0c9'. An author message from 'Ricardo Bonilla Morales' states: 'Rename boilerplate circle ci config and write a new config with build-frontend and build-backend jobs'. Below this, two job cards are shown: '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 displays the detailed view of the 'build-backend' step within the 'default' pipeline. The top header shows the pipeline name and status as 'Failed'. The pipeline summary card includes 'Duration / Finished' (27s / 13m ago), 'Queued' (0s), 'Executor / Resource Class' (Docker / Large), 'Branch' (master), and 'Commit' (fdbd0c9). The author's message is identical to the main pipeline. Below the summary, tabs for 'STEPS', 'TESTS', 'TIMING (BETA)', 'ARTIFACTS', and 'RESOURCES (NEW)' are visible, with 'STEPS' being the active tab. The 'Parallel runs' section indicates 1/16 parallel runs. The step details show four steps: 'Spin up environment' (green, 1s), 'Preparing environment variables' (green, 0s), 'Checkout code' (green, 0s), and 'Build back-end' (red, 25s). The 'Build back-end' step is expanded to show the terminal output:

```
19 packages are looking for funding
  run `npm fund` for details
20
21 found 550 vulnerabilities (2 low, 175 moderate, 316 high, 57 critical)
22   run `npm audit fix` to fix them, or `npm audit` for details
23
24 > glee2@1.0.0 build /home/circleci/project/backend
25 > tsc
26
27 src/main.ts:31:21 - error TS1005: ',' expected.
28
29   .addBearerAuth()x // here is an intentional compile error. Remove the "x" and the backend should compile.
30   ~
31
32 src/main.ts:32:5 - error TS1128: Declaration or statement expected.
33
34 32 .build();
35
36
37 Found 2 errors.
38
39 npm ERR! code ELIFECYCLE
40 npm ERR! errno 2
41 npm ERR! glee2@1.0.0 build: `tsc`
42 npm ERR! Exit status 2
43 npm ERR!
44 npm ERR! Failed at the glee2@1.0.0 build script.
45 npm ERR! This is probably not a problem with npm. There is likely additional logging output above.
```

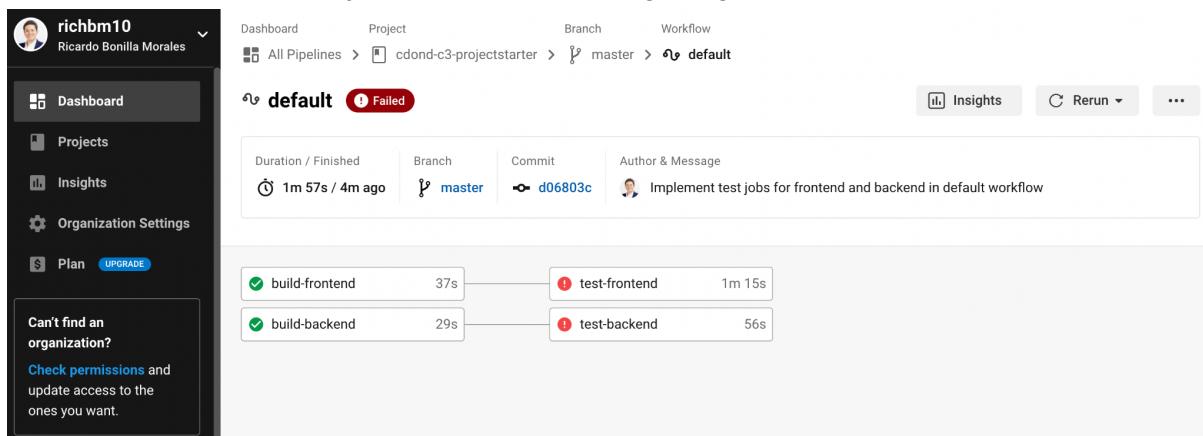
Then, I fixed the intentional error in the backend source code, and pushed the [git commit](#) change. So now both workflow jobs were successful.



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 workflow consists of two steps: 'build-frontend' (36s) and 'build-backend' (26s), both of which are marked as green (successful). The sidebar on the left includes links for Dashboard, Projects, Insights, Organization Settings, Plan (with an 'UPGRADE' button), and Notifications (with 2 notifications).

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 workflow 'default' 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 workflow structure is identical to the previous screenshot, but the test jobs ('test-frontend' and 'test-backend') are now marked as red (failed). The sidebar on the left remains the same.

[SCREENSHOT02]

The screenshot displays two separate test sessions within the CircleCI interface.

Test front-end:

```
116 containers/Employee/components/Employees | 0 | 0 | 0 | 0 | 0 | ... 44,247,257,262 |
117 index.tsx | 0 | 0 | 0 | 0 | 0 | ...
118 containers/Employee/components/ViewEmployee | 0 | 0 | 0 | 0 | 0 | ...
119 index.tsx | 0 | 0 | 0 | 0 | 0 | ...
120 containers/Employee/models | 0 | 0 | 0 | 0 | 0 | ...
121 EmployeeModel.ts | 0 | 0 | 0 | 0 | 0 | ...
122 index.ts | 0 | 100 | 100 | 0 | 0 | 1,2,3,6,7,8,66,67 |
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 | 0 | 1,4,7,32 |
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
```

Test back-end:

```
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
```

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 a CircleCI pipeline named 'default' for a project 'cdond-c3-projectstarter' on the 'master' branch. The pipeline status is 'Success'. The pipeline duration was 1m 49s, completed 3m ago. The commit hash is 3c80cd1, and the author's message is 'Fix intentional errors to succeed unit tests on frontend and backend'. The pipeline consists of two stages: 'build-frontend' (35s) and 'test-frontend' (1m 10s), followed by 'build-backend' (28s) and 'test-backend' (53s). All stages are marked as green (successful).

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 a CircleCI pipeline named 'default' for a project 'cdond-c3-projectstarter' on the 'master' branch. The pipeline status is 'Failed'. The pipeline duration was 1m 35s, completed 4m ago. The commit hash is da5bcc2, and the author's message is 'Change scan-backend job dependency'. The pipeline consists of four stages: 'build-frontend' (27s), 'test-frontend' (53s), 'scan-frontend' (7s), 'build-backend' (21s), 'test-backend' (50s), and 'scan-backend' (9s). The 'test-*' and 'scan-*' stages are marked as red (failed), while the 'build-*' stages are green (successful).

[SCREENSHOT03]

The screenshot displays two parallel run logs from CircleCI:

- Parallel runs:** Shows a summary of the environment setup and code checkout steps, followed by the "Scan front-end" step.
- Scan front-end log:**

```

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 log:**

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
```

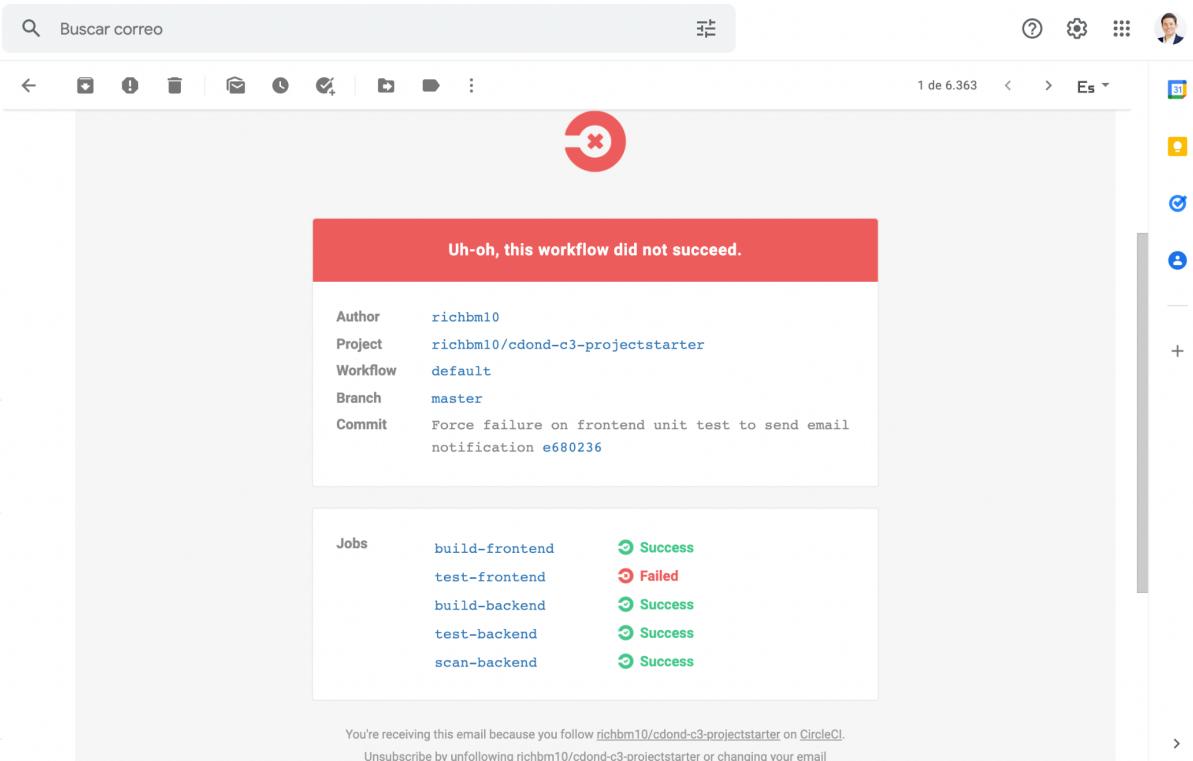
The screenshot shows the CircleCI dashboard. On the left is a sidebar with a user profile for 'richbm10' (Ricardo Bonilla Morales), navigation links for Dashboard, Projects, Insights, Organization Settings, Plan (with an 'UPGRADE' button), Notifications (2 notifications), Status (OPERATIONAL), and Docs. The main area displays a workflow named 'default' which has just succeeded. It shows the duration (2m 16s / 3m ago), branch (master), commit (31bdb64), and author's message: '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]

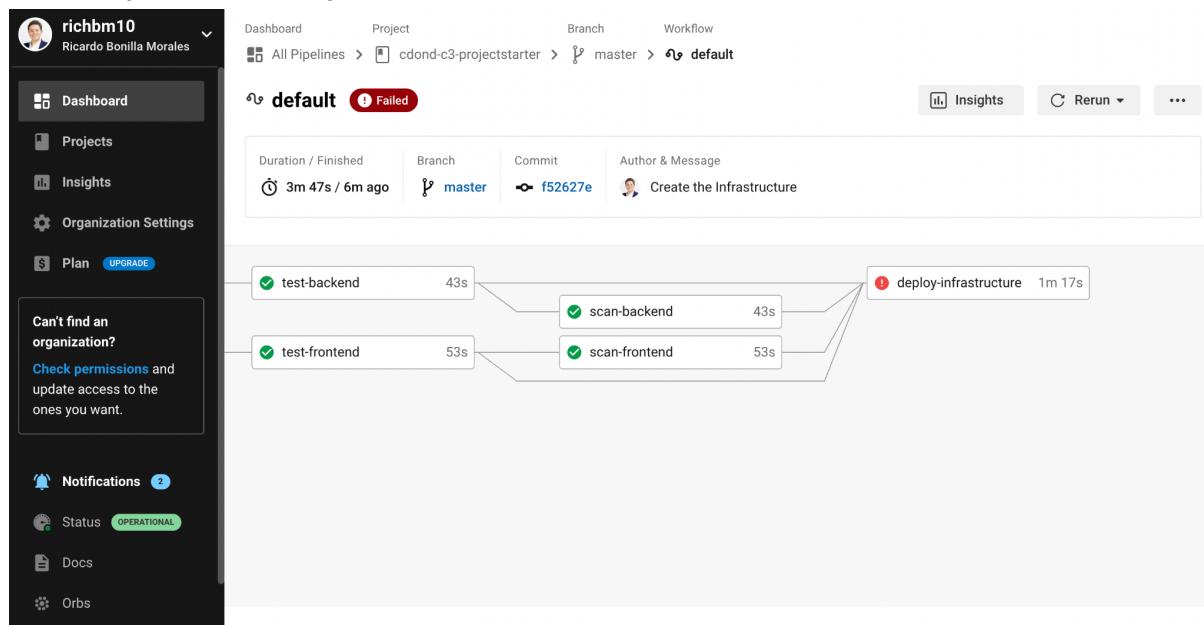
The screenshot shows an email inbox interface. At the top, there is a search bar with 'Buscar correo' and various filter and search icons. The main list shows one email from 'CircleCI Builds <builds@circleci.com>' with the subject '[Workflow] Failed: richbm10/cdond-c3-projectstarter on master / default (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 reads '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.



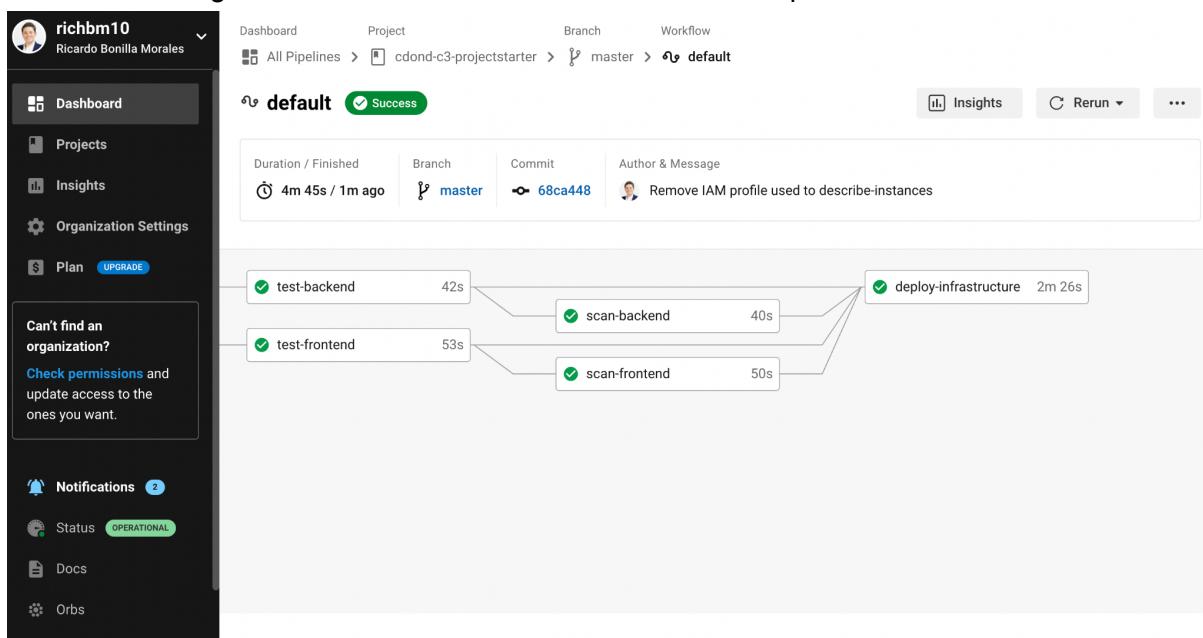
[SCREENSHOT05]

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

```

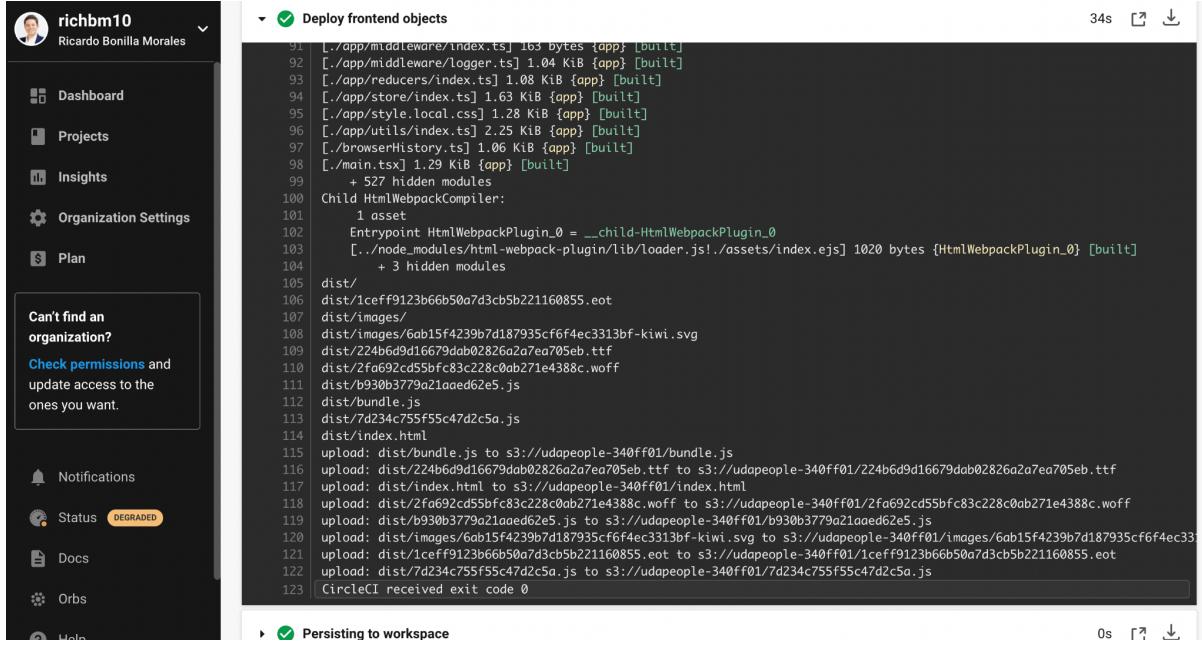


```

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 web 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/224b6d9d16679dab02826a27ea705eb.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/224b6d9d16679dab02826a27ea705eb.ttf to s3://udapeople-340ff01/224b6d9d16679dab02826a27ea705eb.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/iceff9123b66b50a7d3cb5b221160855.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:

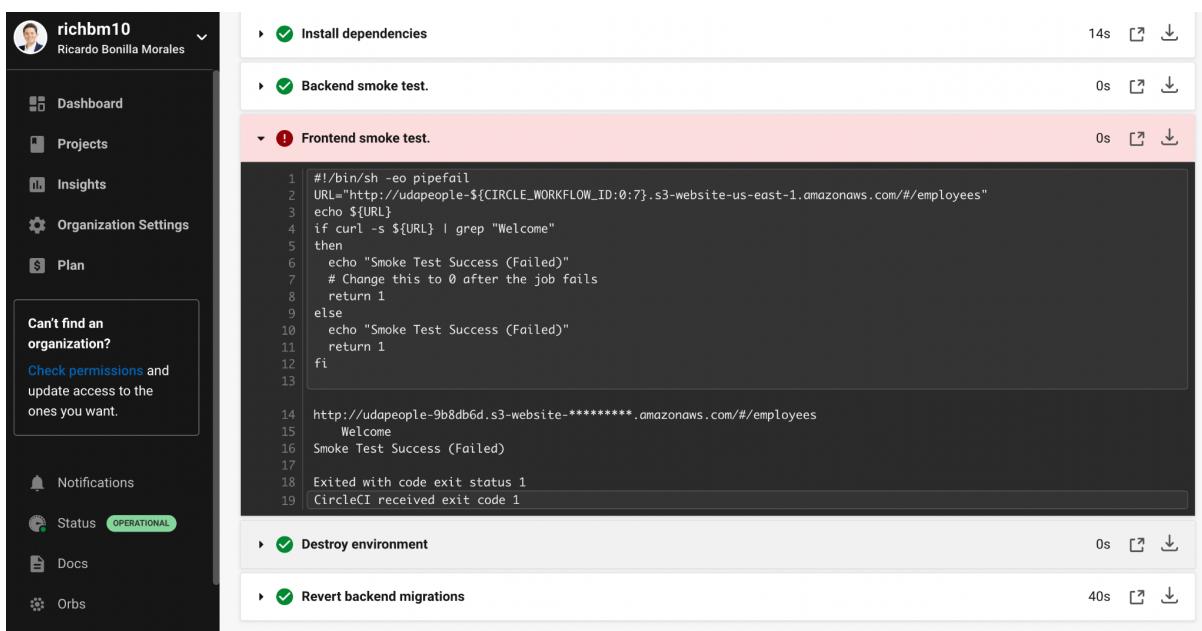
<http://udapeople-6703a02.s3-website-us-east-1.amazonaws.com>.

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 web 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. On the left, there's a sidebar with user information (richbm10, Ricardo Bonilla Morales), navigation links (Dashboard, Projects, Insights, Organization Settings, Plan), and a message about organization access. Below that are Notifications, Status (OPERATIONAL), Docs, and Orbs. The main area is titled "Parallel runs" and shows a single job named "Update cloudfront distribution". The job status is "Success" with a duration of 3m 42s. It includes a terminal log window displaying the command used to deploy CloudFormation and the resulting stack creation logs.

```
#!/bin/bash -eo pipefail
aws cloudformation deploy \
--template-file .circleci/files/cloudfront.yml \
--stack-name InitialStack \
--parameter-overrides WorkflowID="${CIRCLE_WORKFLOW_ID:0:7}" \
--tags project=udapeople

Waiting for changeset to be created..
Waiting for stack create/update to complete
Successfully created/updated stack - InitialStack
CircleCI received exit code 0
```

[URL03_SCREENSHOT]

The screenshot shows the AWS CloudFront console. The left sidebar has sections for Distributions, Policies, Functions, What's new (NEW), Telemetry (Monitoring, Alarms, Logs), Reports & analytics (Cache statistics, Popular objects, Top referers, Usage, Viewers), Security (Origin access identities, Field-level encryption), and Key management (Public keys). The main content area shows the "Distributions" page with one item listed:

ID	Description	Domain name	Alternate ...	Origins
E6NR1YHKEHCLE	-	d3gpr0odqpg7st.cloudfront.net	-	udapeople-6703a02.s3.amazonaws.com

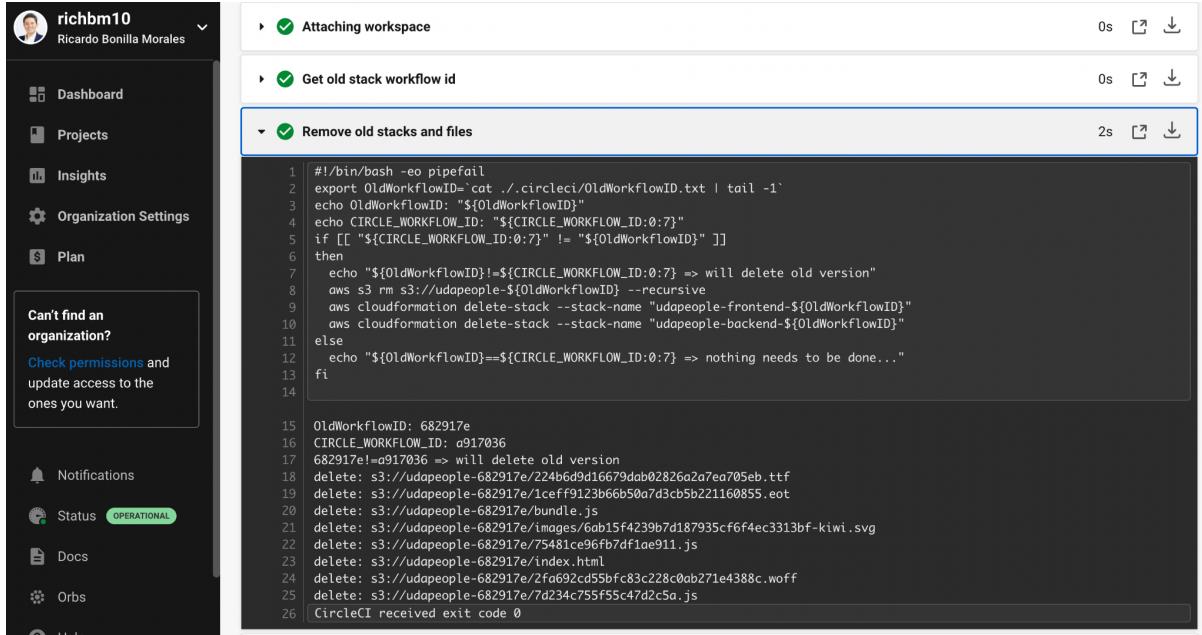
[URL04_SCREENSHOT]

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0cbb9c6dd65dc5c6c (backend-01899fb)	54.158.244.16 open address	172.31.23.177
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-54-158-244-16.compute-1.amazonaws.com open address
Hostname type	Private IP DNS name (IPv4 only)	Answer private resource DNS name
IP name: ip-172-31-23-177.ec2.internal	ip-172-31-23-177.ec2.internal	-
Instance type	Elastic IP addresses	VPC ID
t2.micro	-	vpc-2280e85f
AWS Compute Optimizer finding	IAM Role	Subnet ID
Opt-in to AWS Compute Optimizer for recommendations. Learn more	-	subnet-44e20d08

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 workspace interface. On the left is a sidebar with user information (richbm10, Ricardo Bonilla Morales) and navigation links (Dashboard, Projects, Insights, Organization Settings, Plan). A message box says "Can't find an organization? Check permissions and update access to the ones you want." Below the sidebar are notifications, status (OPERATIONAL), Docs, Orbs, and Help buttons. The main area displays a workflow history:

- Attaching workspace (0s)
- Get old stack workflow id (0s)
- Remove old stacks and files (2s)

The "Remove old stacks and files" step contains the following shell script:

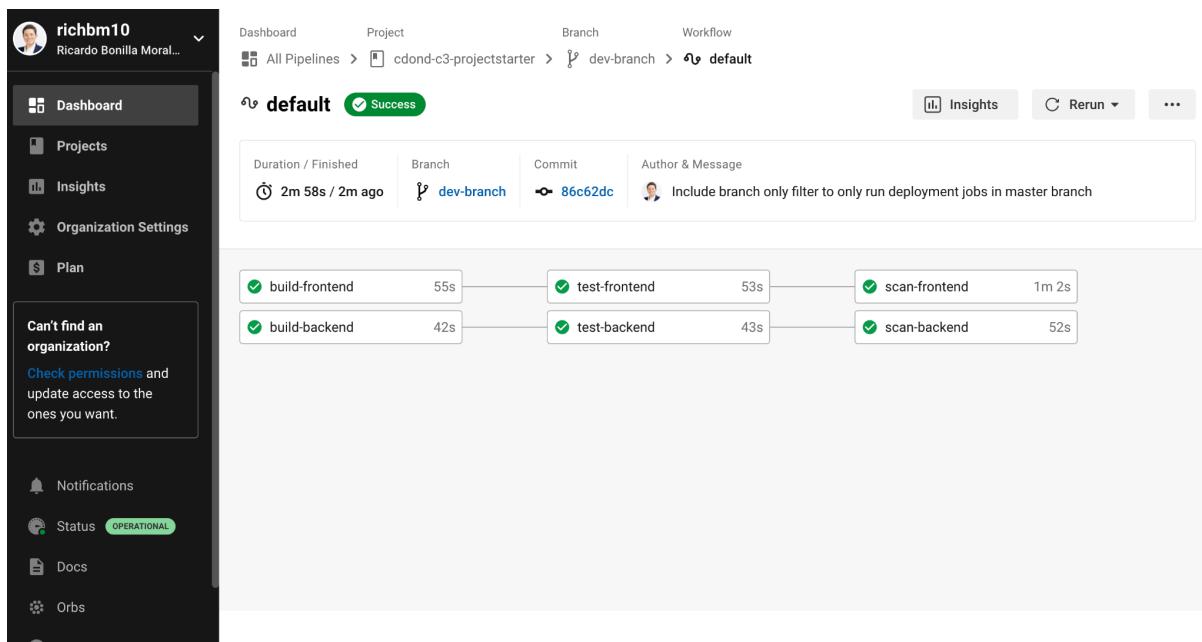
```
#!/bin/bash -eo pipefail
export OldWorkflowID=$(cat ./circleci/OldWorkflowID.txt | tail -1)
echo OldWorkflowID: "$OldWorkflowID"
echo CIRCLE_WORKFLOW_ID: "${CIRCLE_WORKFLOW_ID:0:7}"
if [[ "${CIRCLE_WORKFLOW_ID:0:7}" != "$OldWorkflowID" ]]
then
    echo "${OldWorkflowID}!=${CIRCLE_WORKFLOW_ID:0:7} => will delete old version"
    aws s3 rm s3://udapeople-$OldWorkflowID --recursive
    aws cloudformation delete-stack --stack-name "udapeople-frontend-$OldWorkflowID"
    aws cloudformation delete-stack --stack-name "udapeople-backend-$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/lcef9123b66b50a7d3cb5b221160055.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	
Ricardo	Morales	ricardo.bonilla.moral...		2/10/2022	2/10/2022	

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`:

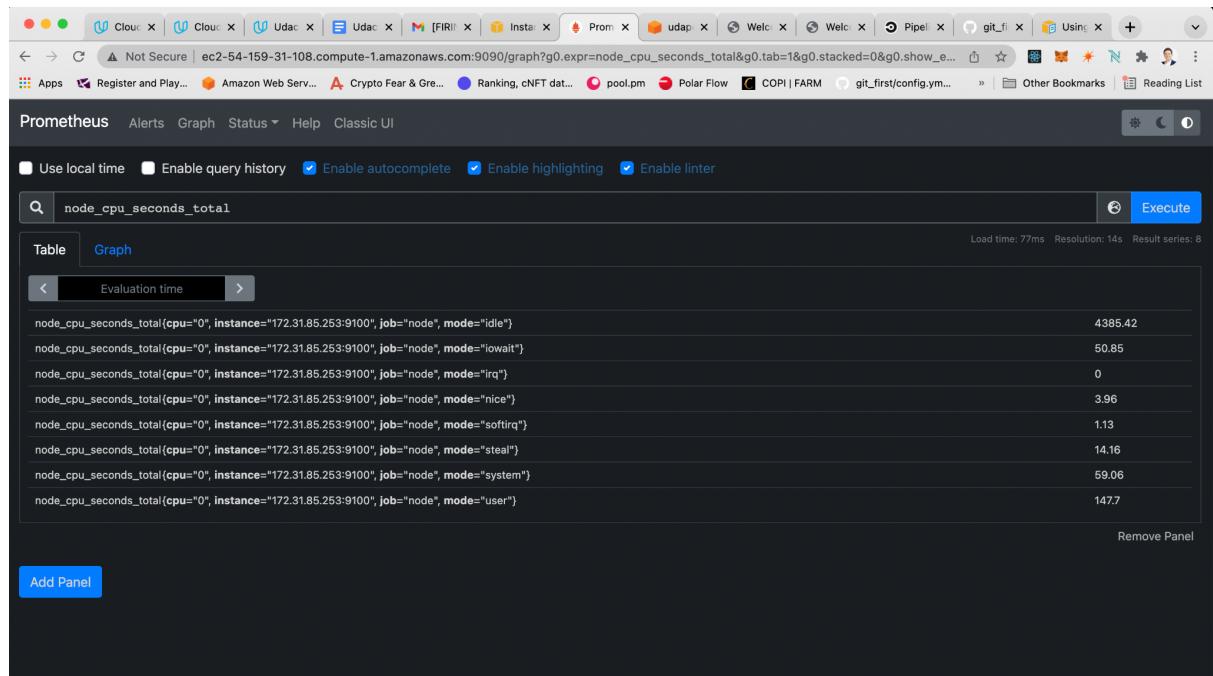
```
- job_name: 'node'  
  ec2_sd_configs:  
    - region: us-east-1  
      access_key: "IAM_KEY"  
      secret_key: "IAM_KEY"  
      port: 9100
```

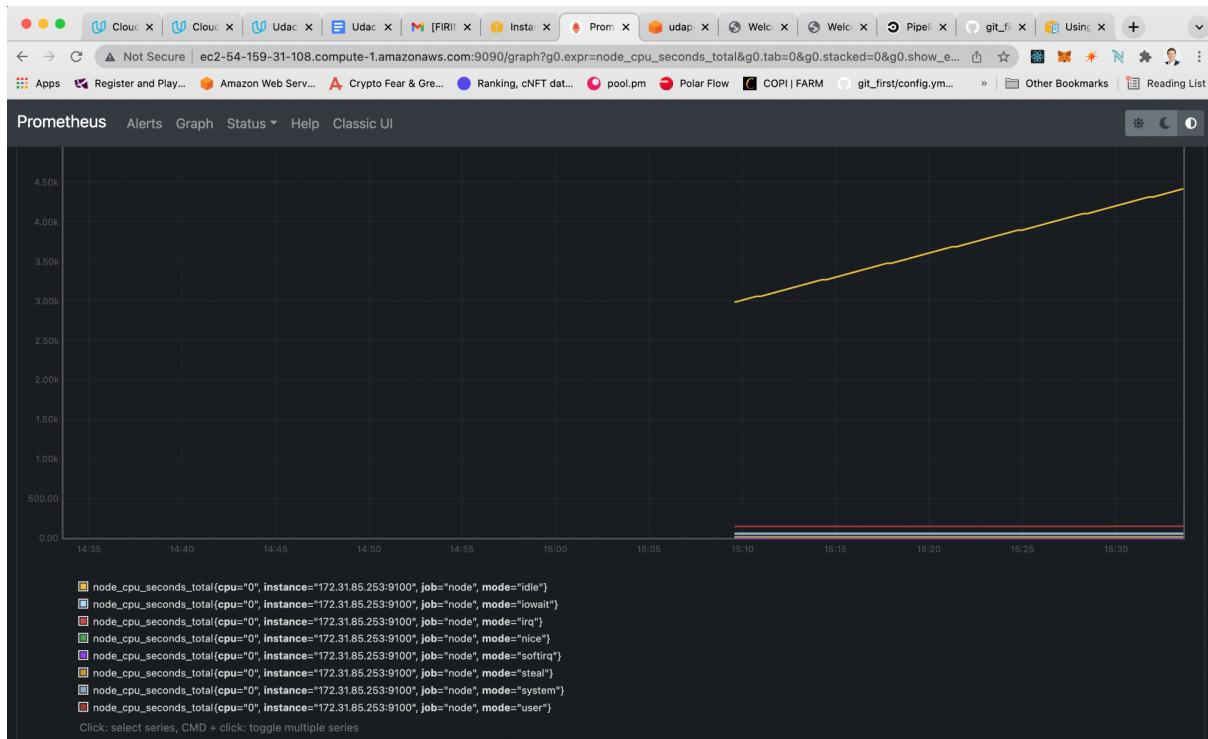
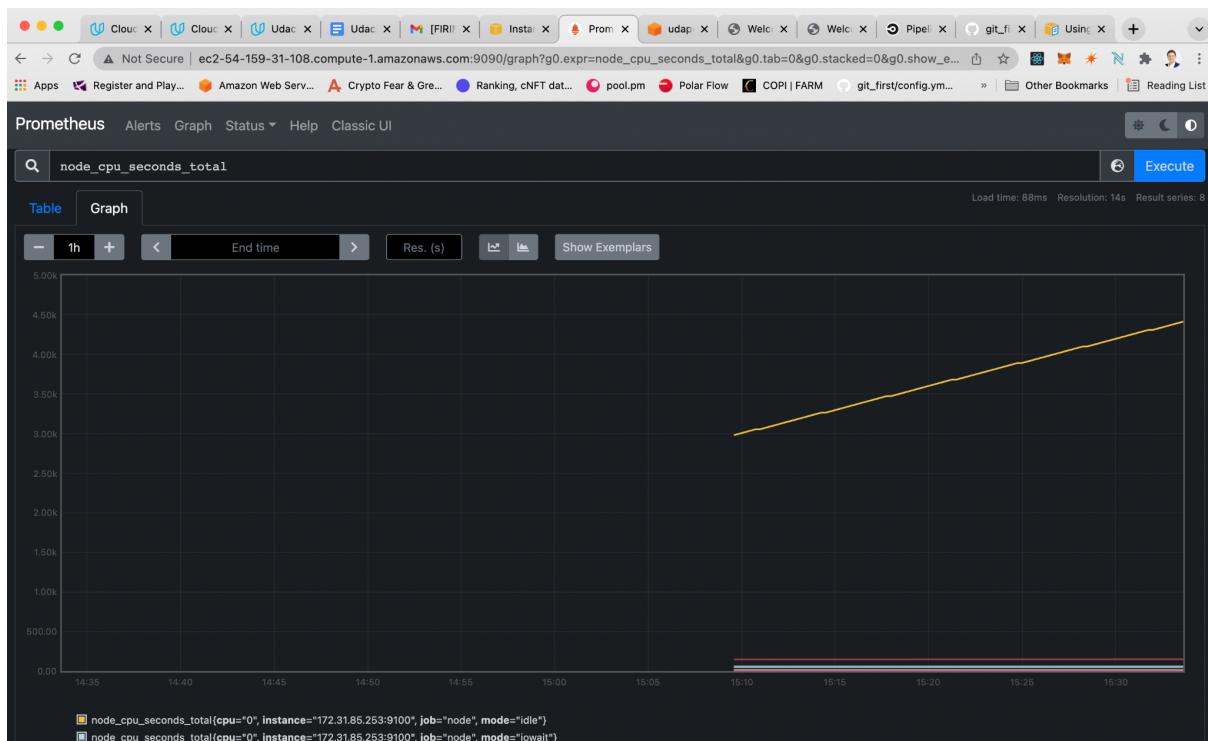
[SCREENSHOT11]

Node Free Memory Back End Instance Graph

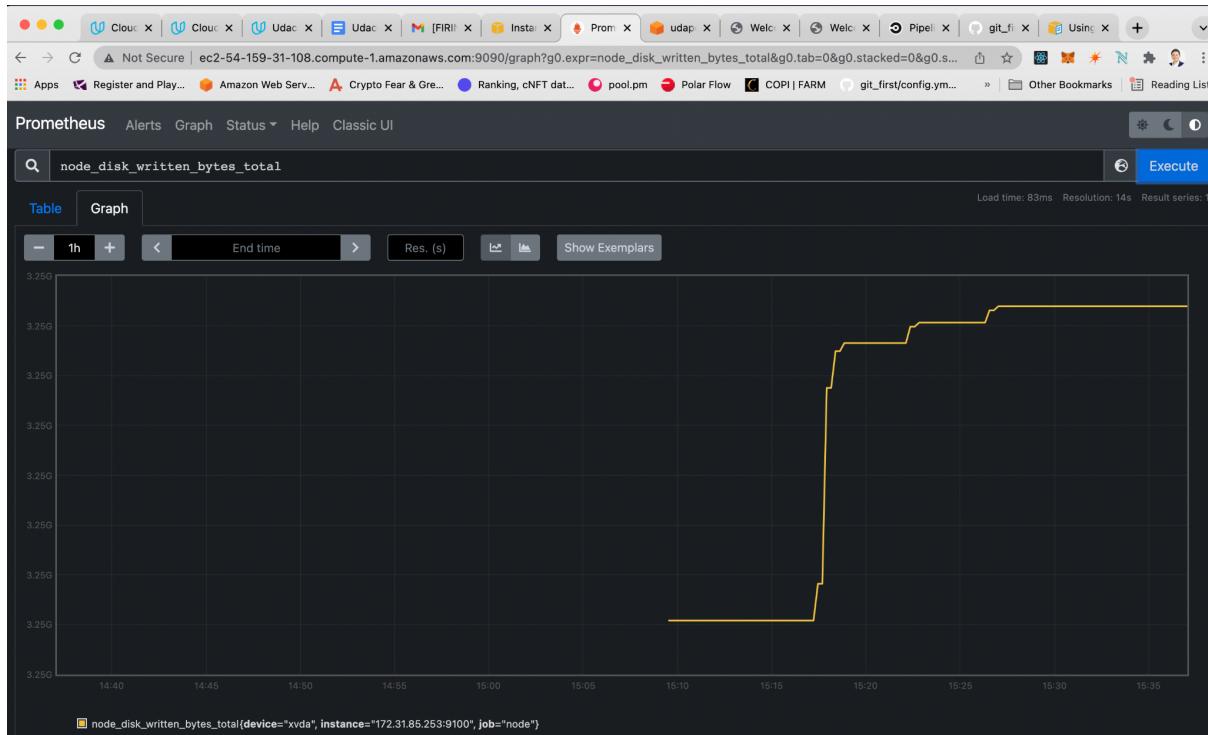


Node CPU Usage Back End Instance

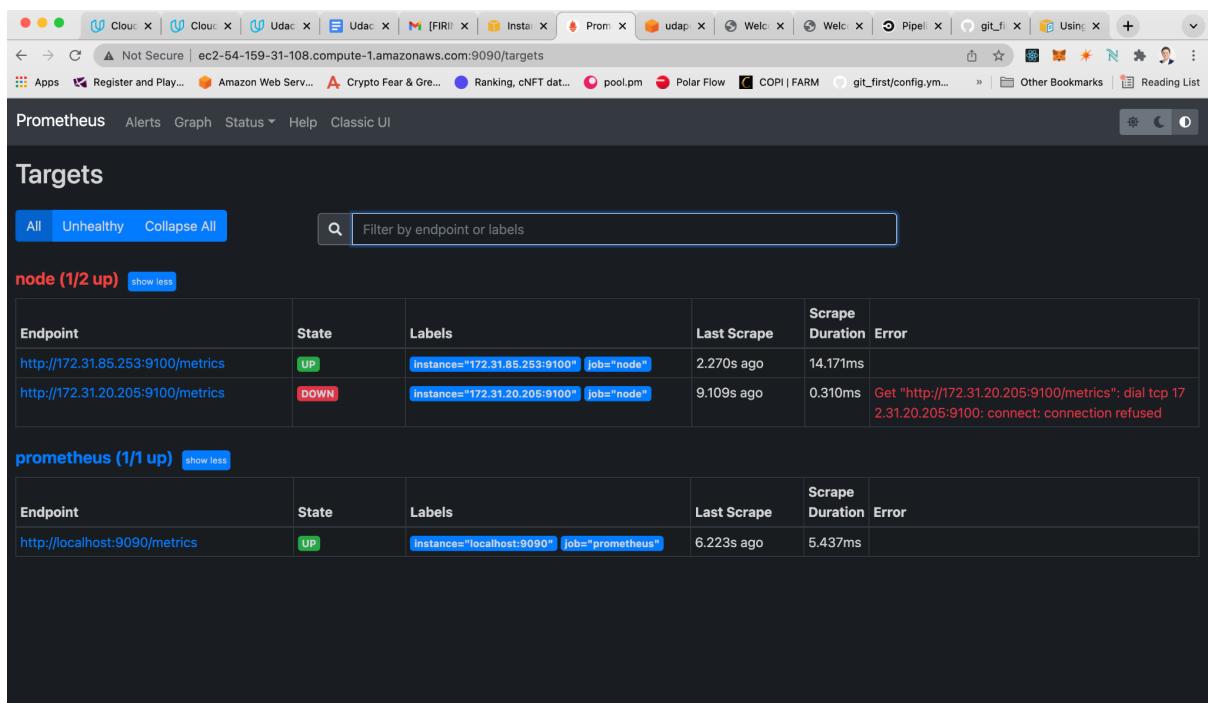




Node Disk Usage Back End Instance Graph



[URL05_SCREENSHOT]



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 a Gmail inbox with the URL mail.google.com/mail/u/0/?tab=rm&ogbl#inbox/1MfcgzbMv8tHNRdNNvxPdVPCDbkqNfq. The inbox contains 3,092 messages in the 'Recibidos' folder. Two alerts from 'Alertmanager' are visible in the inbox:

- [2] Firing**
 - Labels**
alertname = 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](#)
- Labels**
alertname = InstanceDown
instance = [172.31.85.253:9100](#)
job = node
severity = critical
- Annotations**
description = Instance [172.31.85.253:9100](#) of job node is down
title = Instance [172.31.85.253:9100](#) is down
[Source](#)

The second alert's 'instance' value is circled in red.