

# Give Your Application Auto-Deploy Superpowers

<b>Deploying Working, Trustworthy Software</b>	<b>2</b>
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

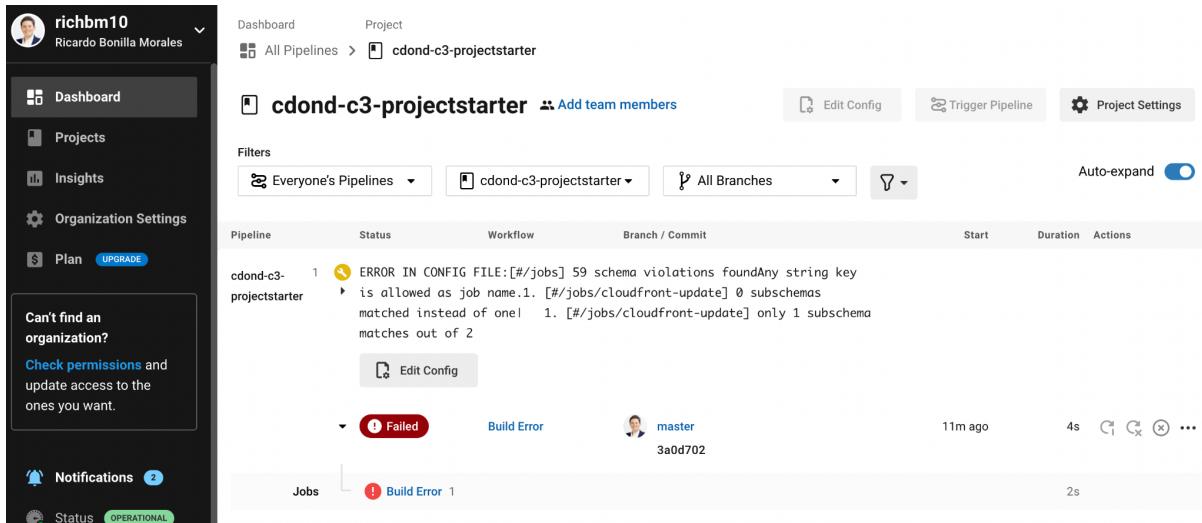
## Public Git Repository

[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.

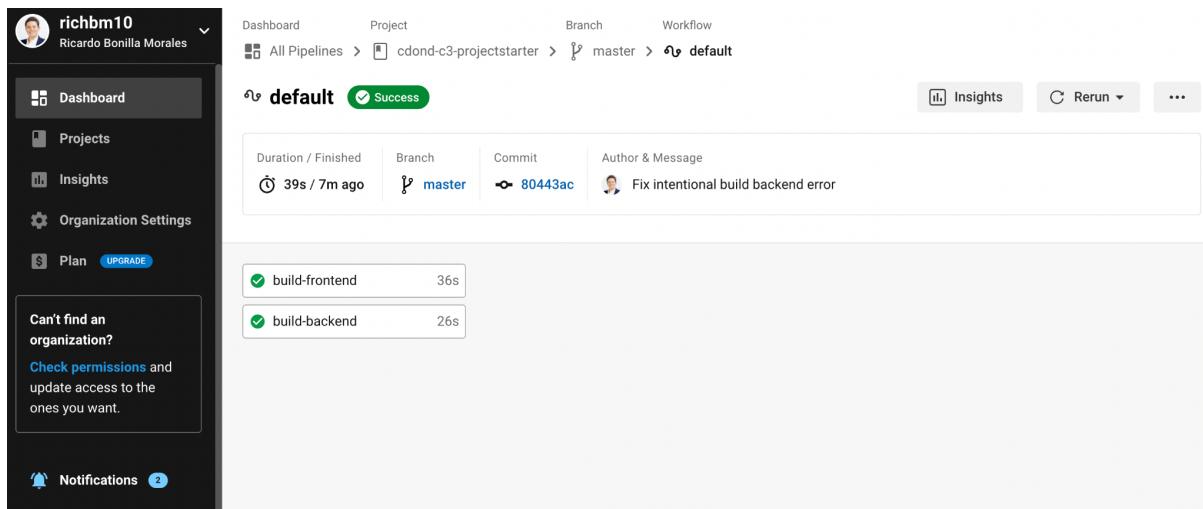
When I pushed the [git commit](#) including the changes, the result was a failed build on the build-backend job.

The screenshot shows the CircleCI dashboard for a user named richbm10. The main navigation bar includes links for Dashboard, Project, Branch, Workflow, and Insights. Below this, a breadcrumb trail shows the pipeline path: All Pipelines > cdond-c3-projectstarter > master > default. A prominent red 'Failed' button indicates the current status of the 'default' branch. On the left sidebar, there are sections for Dashboard, Projects, Insights, Organization Settings, and Plan, along with a note about finding an organization and checking permissions. The main content area displays the build details for the 'default' branch, including duration (38s / 15m ago), branch (master), commit (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 job cards are shown: 'build-frontend' (green, 34s) and 'build-backend' (red, 27s).

[SCREENSHOT01]

This screenshot provides a detailed view of the 'build-backend' step within the 'default' branch pipeline. The top section shows the step's status as 'Failed' with a duration of 27s / 13m ago, using the Docker executor. The commit is fd8d0c9. The message from the author is the same as in the previous screenshot. Below this, a 'Parallel runs' section shows 1/16 parallel runs. The steps listed are: Spin up environment (1s), Preparing environment variables (0s), Checkout code (0s), and Build back-end (25s). The 'Build back-end' step is expanded to show the terminal output, which includes npm audit findings, TypeScript errors, and a final error message: 'npm ERR! Failed at the glee2@1.0.0 build script. npm ERR! This is probably not a problem with npm. There is likely additional logging output above.'

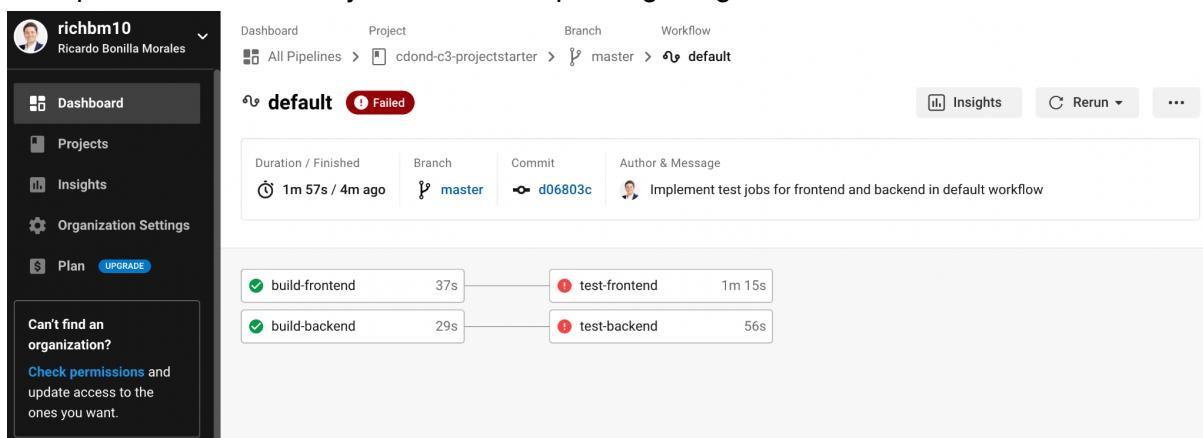
Then, I fixed the intentional error in the backend source code, and pushed the [git commit](#) change. So now both workflow jobs were successful.



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 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.

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 (successful). The total duration for the entire pipeline is 1m 10s.

## 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'. Key details include:

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

The pipeline graph shows the same stages as before, but the final 'scan-' stages have turned red (failed). The 'scan-frontend' job took 7s and the 'scan-backend' job took 9s. The overall pipeline duration is now 1m 10s.

## [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.

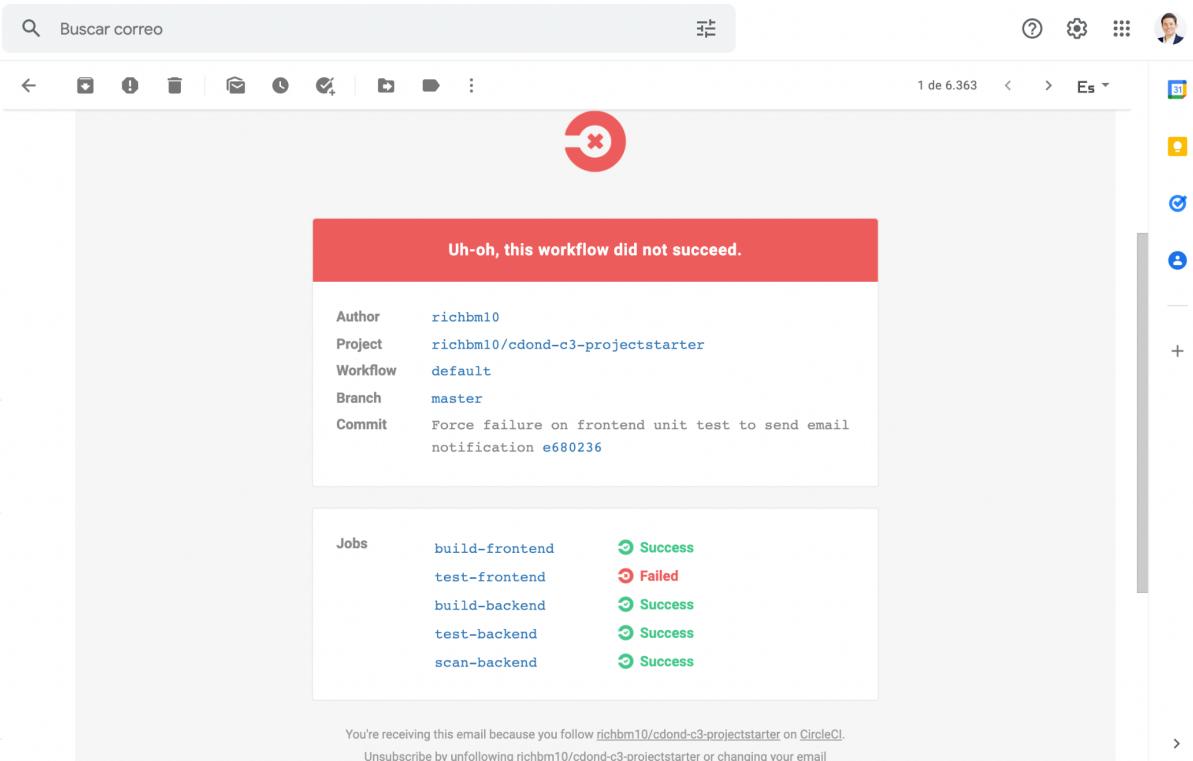
[SCREENSHOT04]

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 content is displayed below the header:

CircleCI Builds <builds@circleci.com>  
para ▾  
14:33 (hace 4 minutos)  
Traducir mensaje Desactivar para: inglés

The message body contains a large red circle with a white 'X' icon. Below it, a red banner reads 'Uh-oh, this workflow did not succeed.' The detailed log information follows:

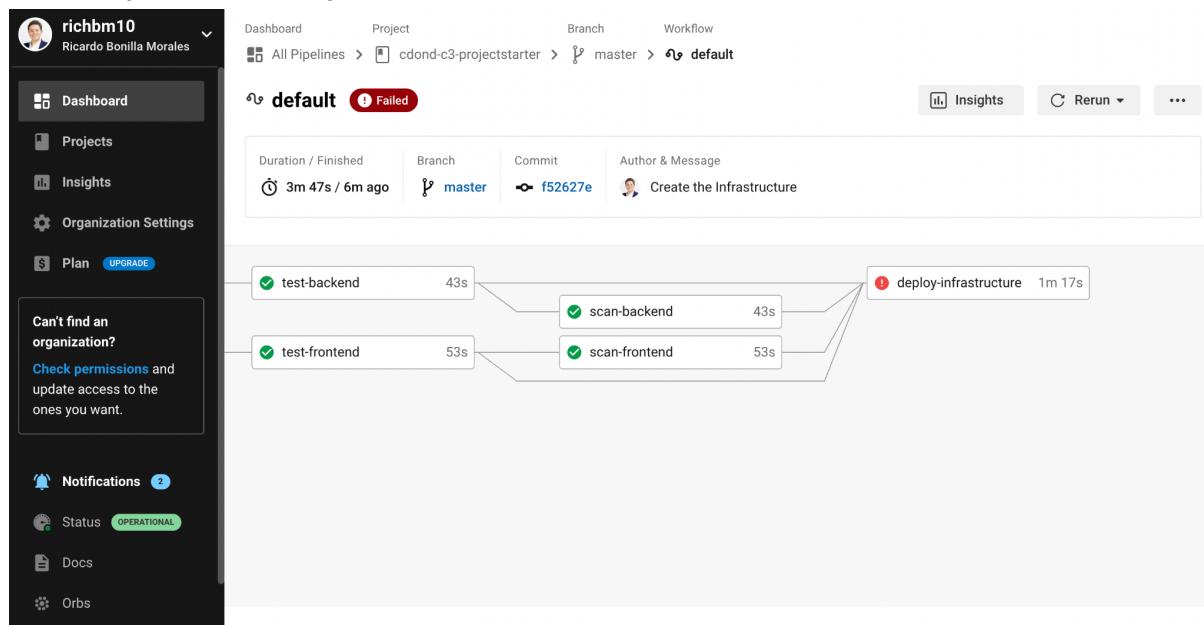
Author	richbm10
Project	richbm10/cdond-c3-projectstarter
Workflow	default
Branch	master
Commit	Force failure on frontend unit test to send email notification e680236



## 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.



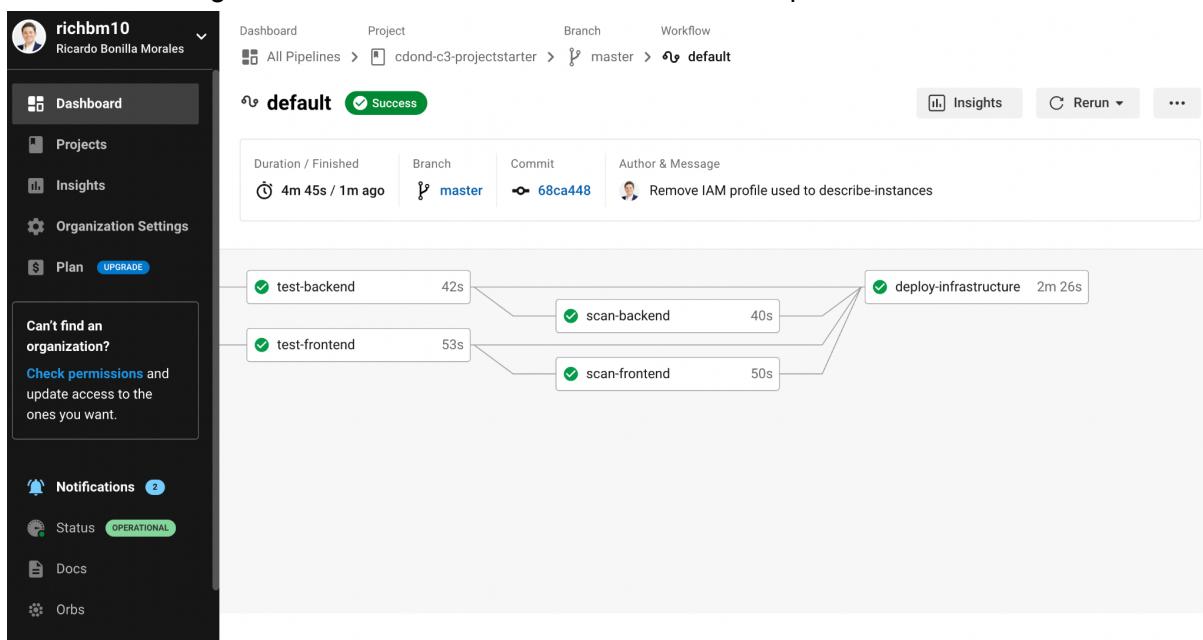
## [SCREENSHOT05]

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 `circleci` 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 (with 1 notification), 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 status icons: Notifications (1), Status (DEGRADED), Docs, Orbs, and Help.

The main area displays deployment logs for "Deploy frontend objects". The logs show the build process, including file sizes and paths, such as index.ts, logger.ts, reducers/index.ts, store/index.ts, style/local.css, utils/index.ts, and main.ts. It also includes entries for HtmlWebpackPlugin and various CSS and font files. The log ends with "CircleCI received exit code 0".

Below the logs, another section titled "Persisting to workspace" is shown, with a status of "0s".

## [URL02]

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

<http://udapeople-c95c33d.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 with the same sidebar as the previous screenshot. The main area displays a multi-step smoke test job:

- Step 1: "Install dependencies" (status 14s, green checkmark)
- Step 2: "Backend smoke test." (status 0s, green checkmark)
- Step 3: "Frontend smoke test." (status 0s, red exclamation mark)
  - Log output:
 

```
#!/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-9b8db6d.s3-website-*****.amazonaws.com/#/employees
Welcome
Smoke Test Success (Failed)
17
18 Exited with code exit status 1
CircleCI received exit code 1
```
- Step 4: "Destroy environment" (status 0s, green checkmark)
- Step 5: "Revert backend migrations" (status 40s, green checkmark)

## 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), navigation links (Dashboard, Projects, Insights, Organization Settings, Plan), and a message about permissions. The main area displays two build logs:

- Frontend smoke test:** A red header indicates a failure. The log shows a shell script attempting to curl a URL and grep for "Welcome". It handles both success and failure cases. The output shows the script failed with status 1, indicating a smoke test failure.
- Destroy environments:** A green header indicates success. The log shows commands to delete an S3 static hosting bucket and then destroy environments using CloudFormation. The output shows the environment was successfully destroyed.

## 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 log pane with the following content:

```

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 contains navigation links for Distributions, Policies, Functions, 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 is titled "CloudFront > Distributions" and shows a table of distributions. There is one distribution listed:

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

ACKLEN

Employees / View & Manage

Add New Employee

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	

## [URL04\_SCREENSHOT]

Cloud DevOps Eng | CloudFront | Udacity Project 3 - | cloudfront-update | Commits - richbm11 | Welcome | 54.158.244.16:3030 | + | -

Not Secure | 54.158.244.16:3030/api/status

aws Services Search for services, features, blogs, docs, and more [Option+S] N. Virginia vocabs/user1538616=759e8fde-6536-11ea-a85f-fb24ea123fe2 @ 28... ▾

New EC Experience Learn more

EC2 Dashboard

EC2 Global View

Events

Tags

Limits

Instances

- Instances
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts
- Scheduled Instances
- Capacity Reservations

Images

- AMIs
- AMI Catalog

Elastic Block Store

Instance summary for i-0cbb9c6dd65dc5c6c (backend-01899fb)

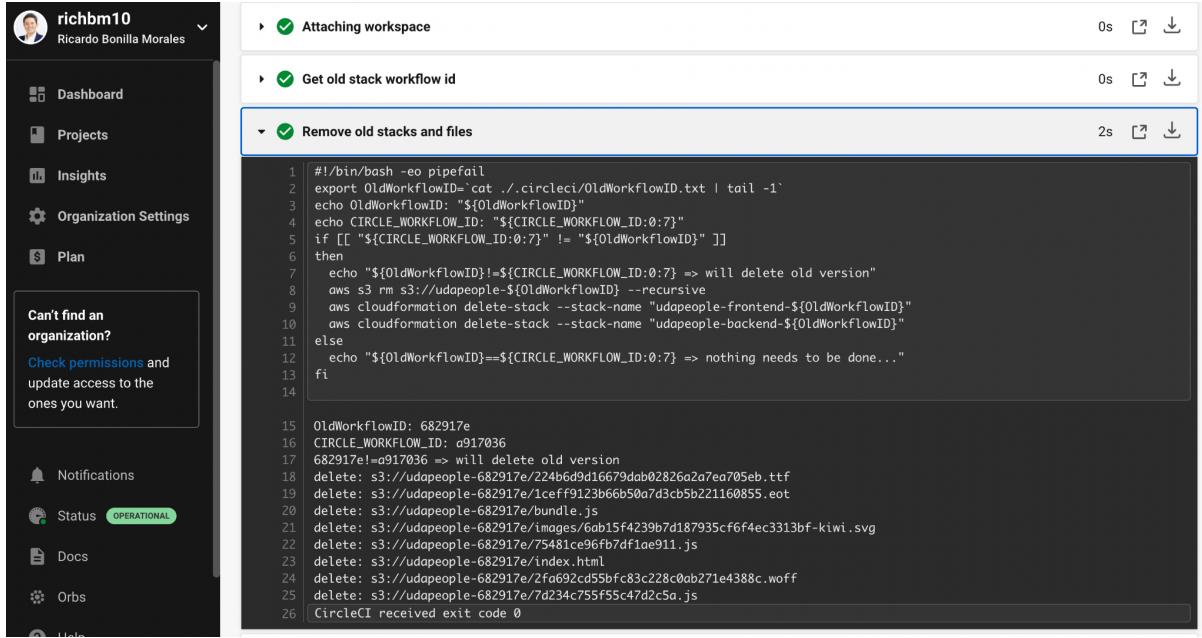
Updated less than a minute ago

Instance ID	i-0cbb9c6dd65dc5c6c (backend-01899fb)	Public IPv4 address	54.158.244.16	Private IPv4 addresses	172.31.23.177
IPv6 address	-	Instance state		Public IPv4 DNS	ec2-54-158-244-16.compute-1.amazonaws.com
Hostname type		Private IP DNS name (IPv4 only)	ip-172-31-23-177.ec2.internal	Answer private resource DNS name	-
IP name: ip-172-31-23-177.ec2.internal		Elastic IP addresses	-	VPC ID	vpc-2280e85f
Instance type	t2.micro	IAM Role	-	Subnet ID	subnet-44e20d08
AWS Compute Optimizer finding					
Platform	Ubuntu (Inferred)	AMI ID	ami-04505e74c0741db8d	Monitoring	disabled

## 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:

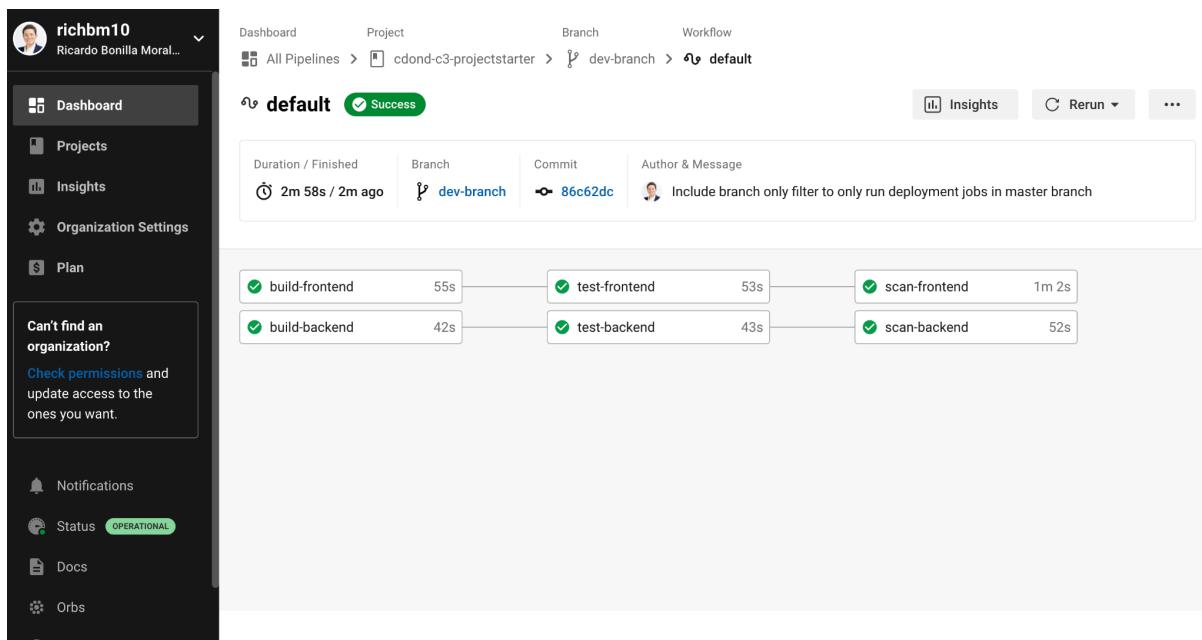
```
#!/bin/bash -eo pipefail
export OldWorkflowID=$(cat ./circleci/OldWorkflowID.txt | tail -1)
echo OldWorkflowID: "$OldWorkflowID"
echo CIRCLE_WORKFLOW_ID: "${CIRCLE_WORKFLOW_ID:0:7}"
if [[ "${CIRCLE_WORKFLOW_ID:0:7}" != "$OldWorkflowID" ]]
then
    echo "${OldWorkflowID}!=${CIRCLE_WORKFLOW_ID:0:7} => will delete old version"
    aws s3 rm s3://udapeople-$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`:

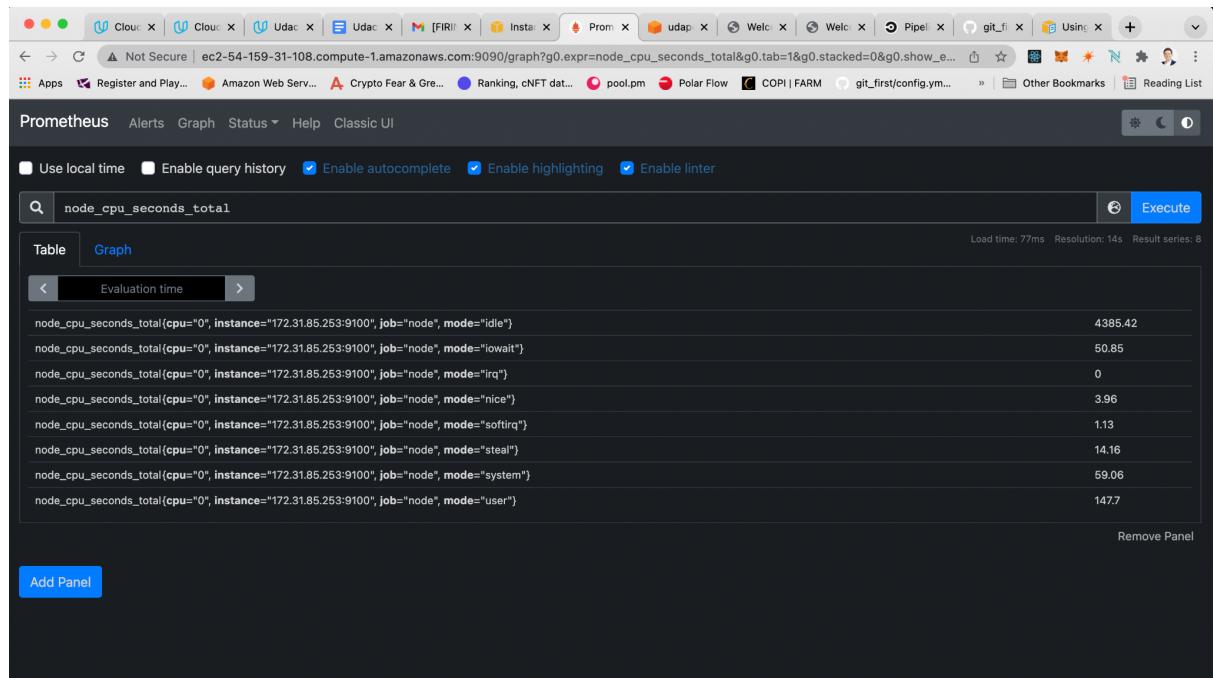
```
- job_name: 'node'  
  ec2_sd_configs:  
    - region: us-east-1  
      access_key: "IAM_KEY"  
      secret_key: "IAM_KEY"  
      port: 9100
```

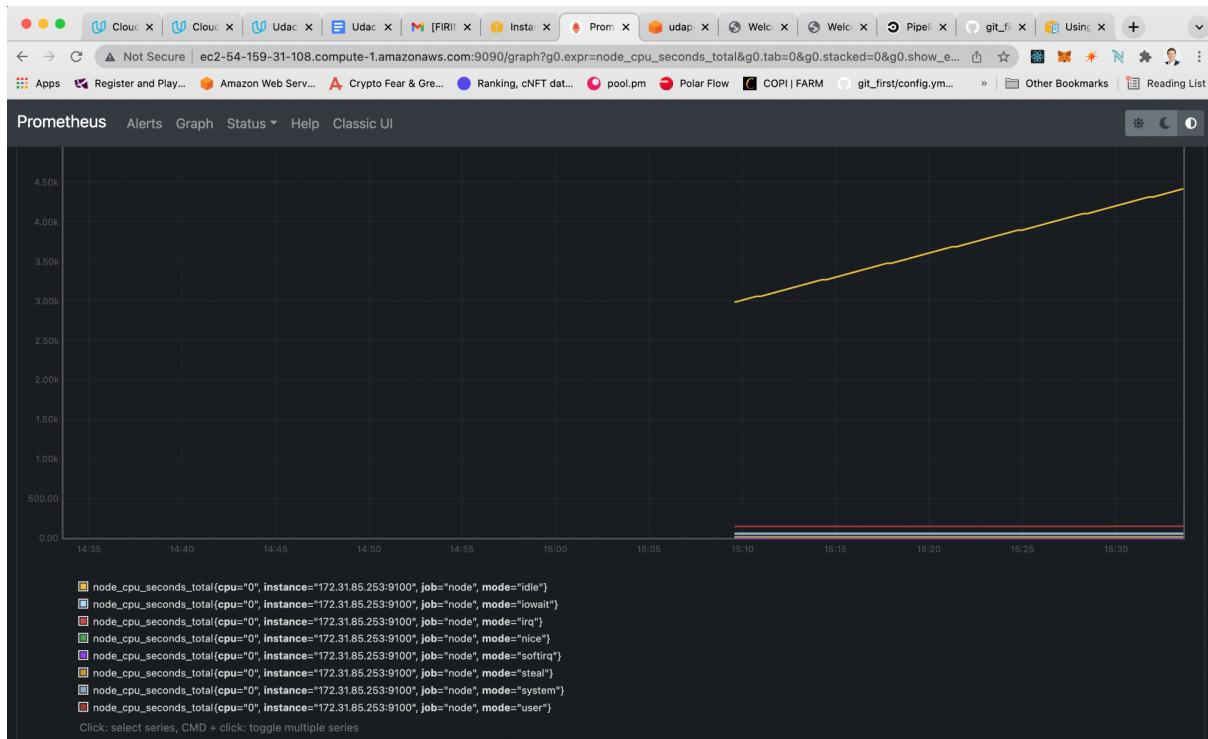
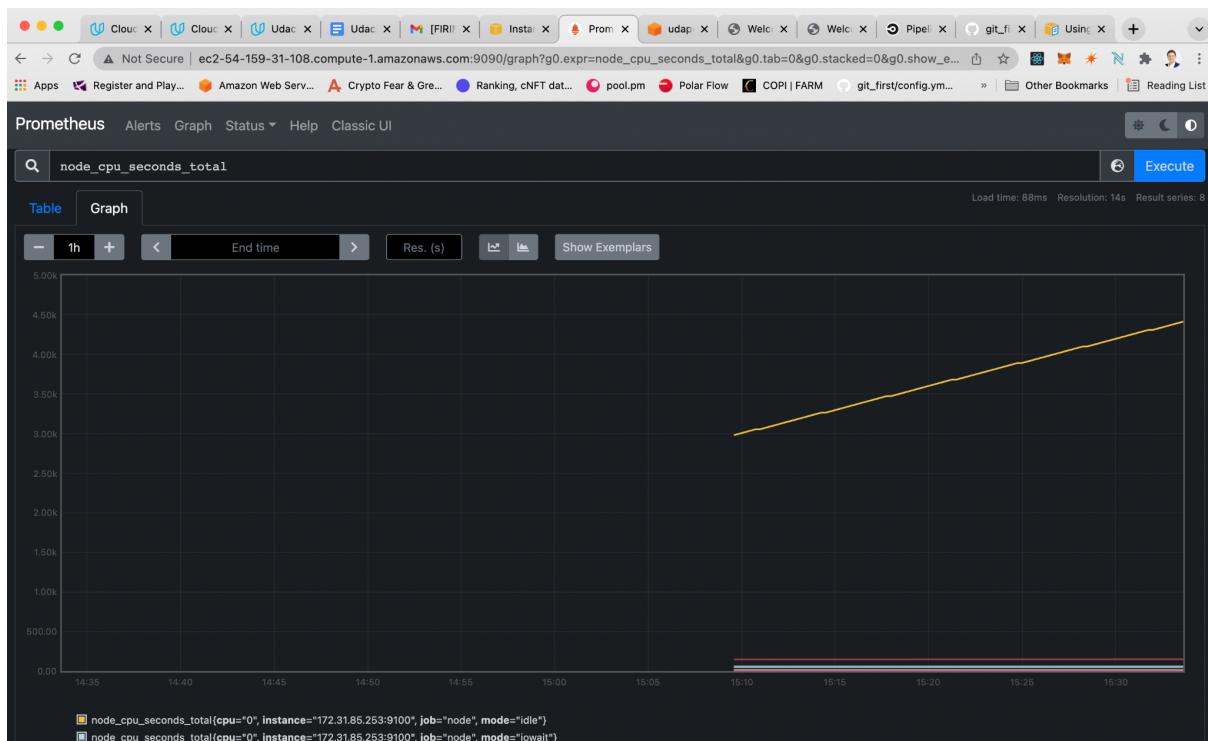
## [SCREENSHOT11]

### Node Free Memory Back End Instance Graph

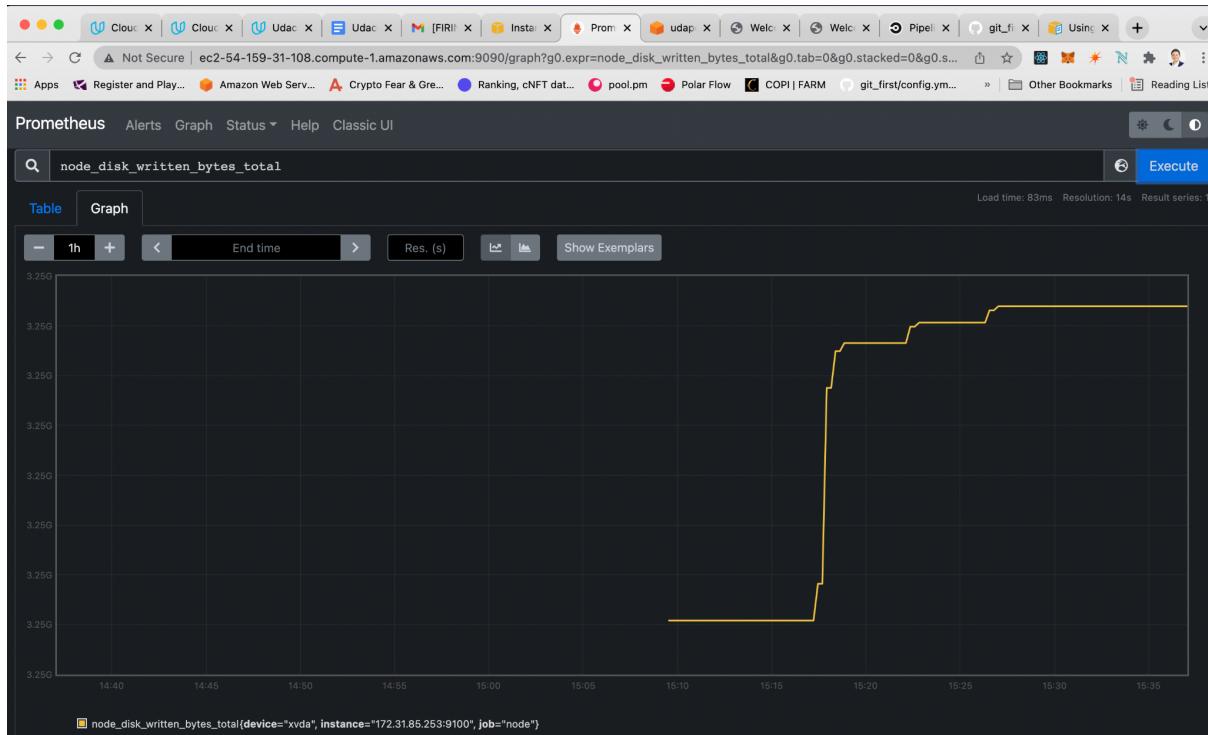


### Node CPU Usage Back End Instance

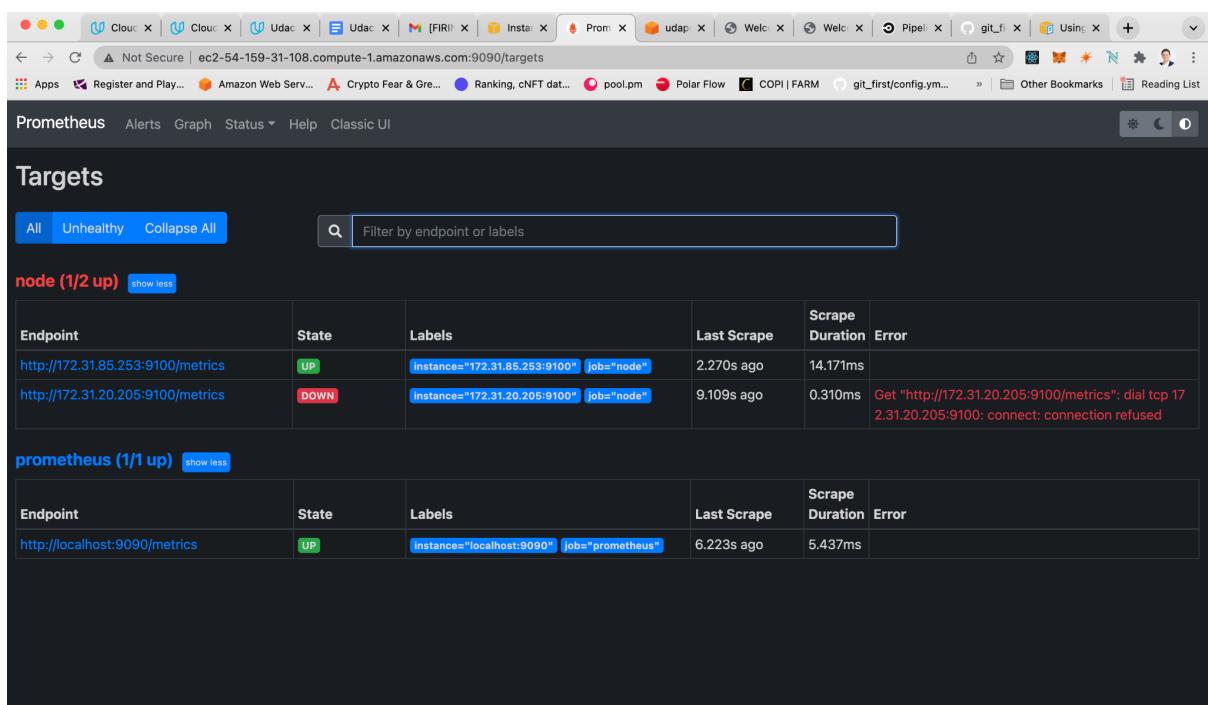




## Node Disk Usage Back End Instance Graph



## [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/1MfcgzGmv8tHNRdNNvxfPdVPCDbkqNfq](https://mail.google.com/mail/u/0/?tab=rm&ogbl#inbox/1MfcgzGmv8tHNRdNNvxfPdVPCDbkqNfq). 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.