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

- Reduce risk

Finding and fixing bugs late in the development process is expensive and time-consuming. This is especially true when there are issues with features that have already been released to production.

With a CI/CD pipeline, you can test and deploy code more frequently, giving testers the ability to detect issues as soon as they occur and to fix them immediately. You are essentially mitigating risks in real time.

- Deliver faster

Organizations are moving toward releasing features multiple times a day. This is not an easy task; only a handful of companies like Netflix, Amazon, and Facebook have been able to achieve this goal. But, with a seamless CI/CD pipeline, multiple daily releases can be made a reality.

Teams can build, test and deploy features automatically with almost no manual intervention. This is accomplished using various tools, frameworks, and systems like Travis CI, Docker, Kubernetes, and LaunchDarkly.

- Expend less manual effort

To align with the shift-left paradigm, we need automation right from the start. This is also a vital component of having a successful CI/CD implementation. Once you build features and check in code, tests should be automatically triggered to make sure that the new code does not break existing features and that the new features are working correctly.

After the tests run, the code gets deployed to different environments, including QA, staging and production. Throughout this process, you will be getting constant notifications through different channels, giving you plenty of information about the build, test and deploy cycles.

- Generate extensive logs

Observability is one of the biggest aspects of DevOps and CI/CD integration. If something is wrong, you need to understand why. You need a mechanism to study the system in production over time and identify key performance metrics. Observability is a technical solution that helps in this effort.

One key aspect of observability is logging information. Logs are a rich source of information to understand what is happening beneath the UI and study application behavior.

With a CI/CD pipeline, extensive logging information is generated in each stage of the development process. There are various tools available to analyze these logs effectively and get immediate feedback about the system.

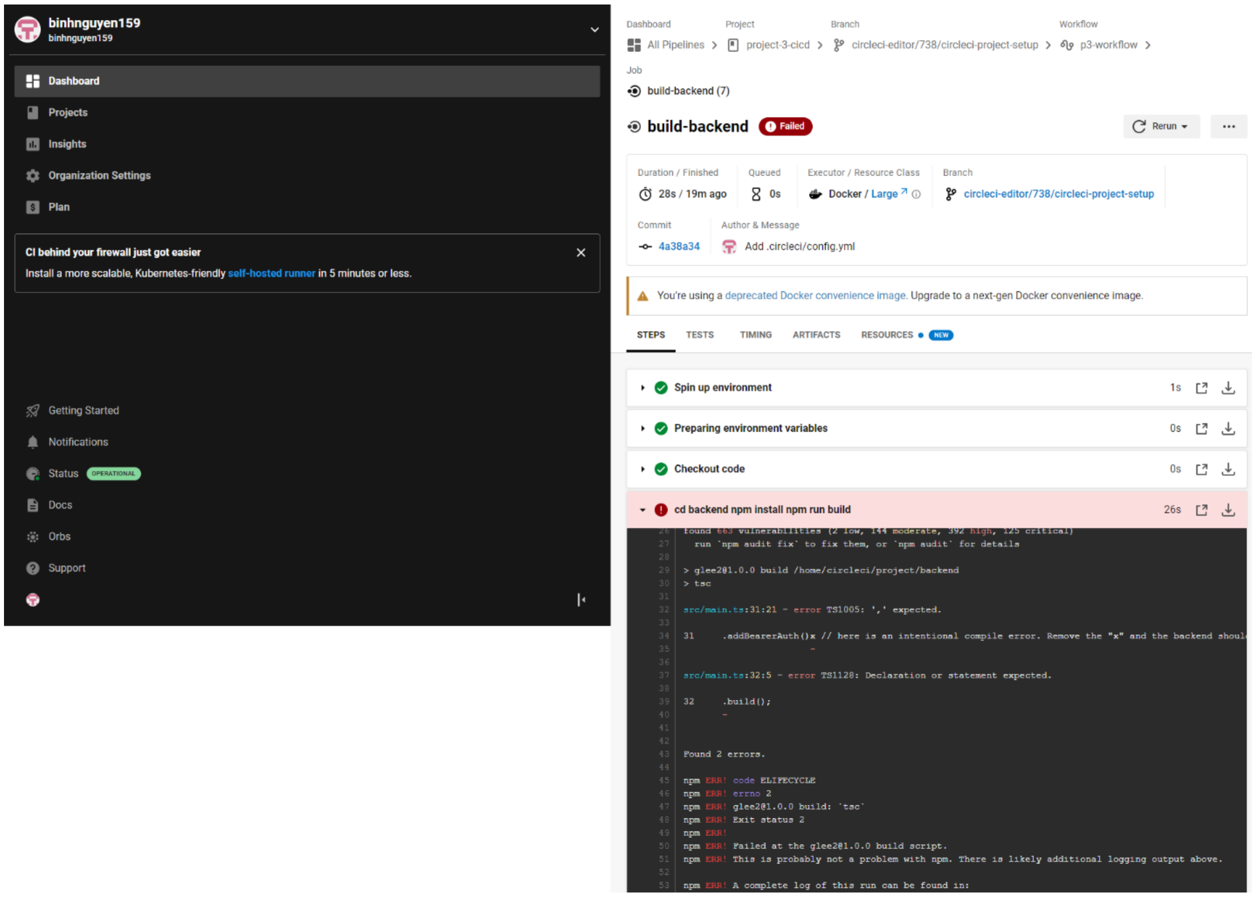
- Make easier rollbacks

One of the biggest advantages of a CI/CD pipeline is you can roll back changes quickly. If any new code changes break the production application, you can immediately return the application to its previous state. Usually, the last successful build gets immediately deployed to prevent production outages.

The world is moving toward rapid release cycles, and CI/CD pipelines have accelerated the release rate. With careful planning and implementation, such a pipeline can help you find defects faster, implement fixes immediately, and increase overall customer satisfaction.

2. **Section 2: Deploying Working, Trustworthy Software**

- Job failed because of compile errors. [SCREENSHOT01]



build-backend Failed

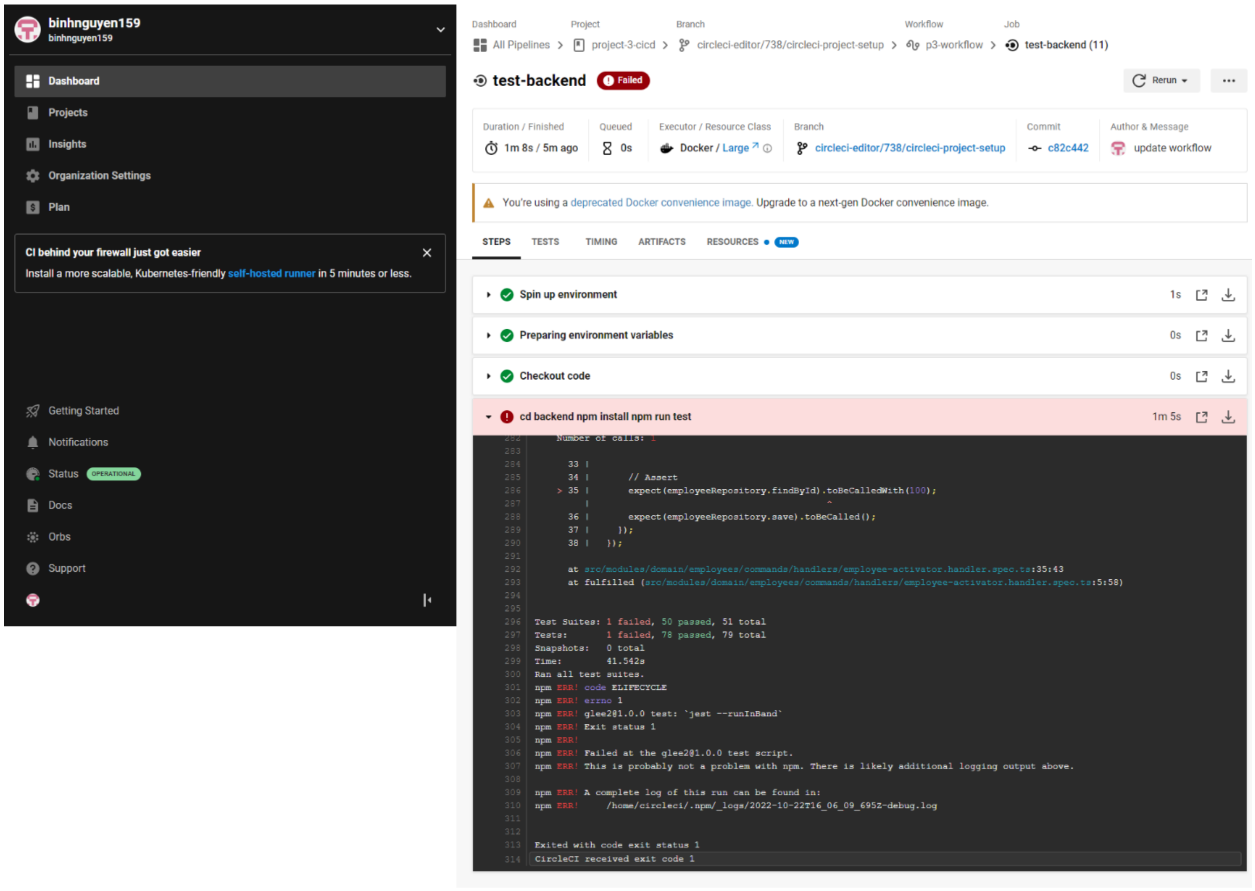
Duration / Finished: 28s / 19m ago
Queued: 0s
Executor / Resource Class: Docker / Large
Branch: circleci-editor/738/circleci-project-setup
Commit: 4a38a34
Author & Message: Add .circleci/config.yml

You're using a deprecated Docker convenience image. Upgrade to a next-gen Docker convenience image.

Spin up environment 1s
Preparing environment variables 0s
Checkout code 0s
cd backend npm install npm run build 26s

```
found 663 vulnerabilities (2 low, 144 moderate, 392 high, 125 critical)
  run `npm audit fix` to fix them, or `npm audit` for details

> glee2@1.0.0 build /home/circleci/project/backend
> tsc

src/main.ts:31:21 - error TS1005: ',' expected.

31     .addBearerAuth()x // here is an intentional compile error. Remove the "x" and the backend shoul
                       ~

src/main.ts:32:5 - error TS1128: Declaration or statement expected.

32     .build();
       ~


Found 2 errors.

npm ERR! code ELIFECYCLE
npm ERR! errno 2
npm ERR! glee2@1.0.0 build: `tsc`
npm ERR! Exit status 2
npm ERR!
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.

npm ERR! A complete log of this run can be found in:
```

- Job failed because of unit tests. [SCREENSHOT02]



CI behind your firewall just got easier
Install a more scalable, Kubernetes-friendly self-hosted runner in 5 minutes or less.

Dashboard: All Pipelines > Project: project-3-cicd > Branch: circleci-editor/738/circleci-project-setup > Workflow: p3-workflow > Job: test-backend (11)

test-backend Failed

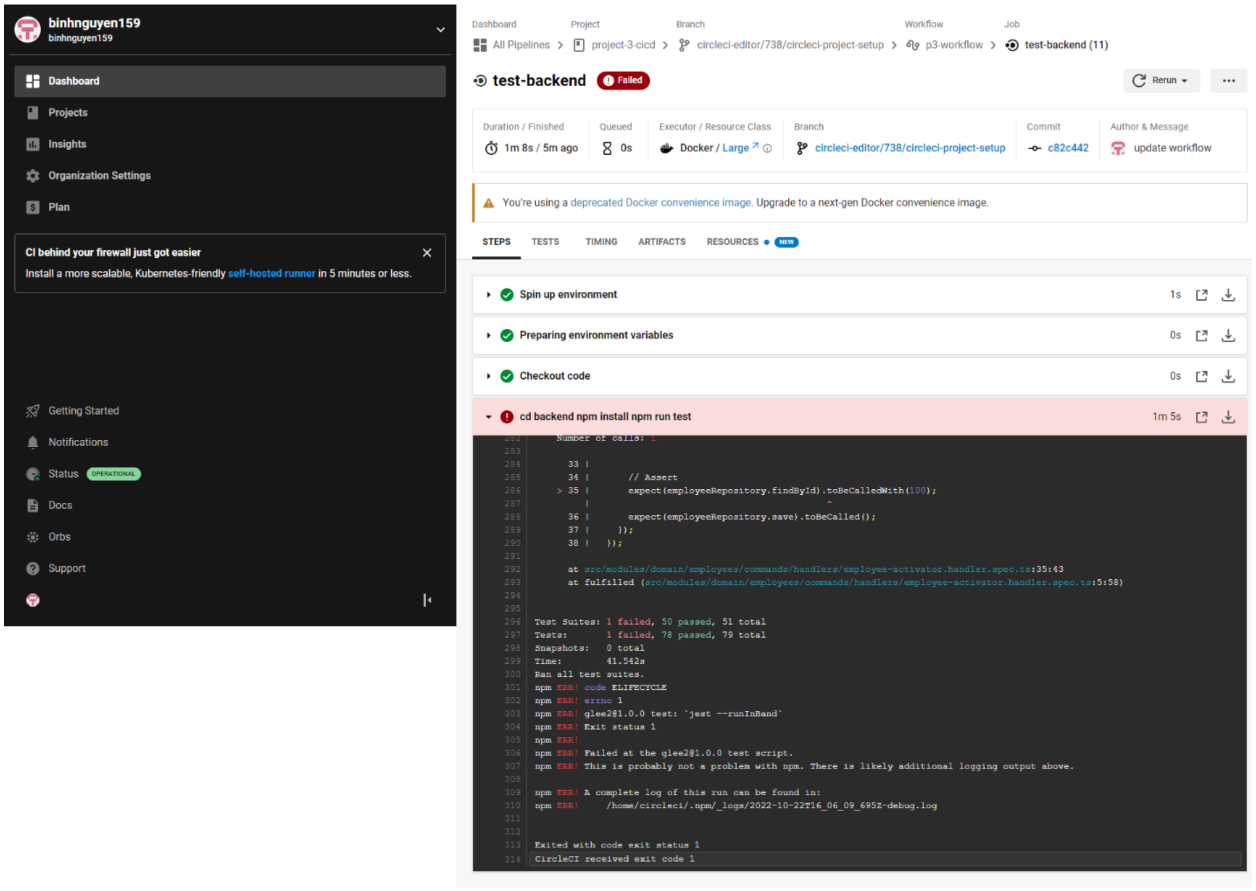
Duration / Finished: 1m 8s / 5m ago
Queued: 0s
Executor / Resource Class: Docker / Large
Branch: circleci-editor/738/circleci-project-setup
Commit: c82c442
Author & Message: update workflow

You're using a deprecated Docker convenience image. Upgrade to a next-gen Docker convenience image.

Spin up environment 1s
Preparing environment variables 0s
Checkout code 0s
cd backend npm install npm run test 1m 5s

```
    Number of calls: 1

      33 |
      34 |       // Assert
    > 35 |       expect(employeeRepository.findById).toBeCalledWith(100);
         |                                           ^
      36 |       expect(employeeRepository.save).toBeCalled();
      37 |     });
      38 |   });

      at src/modules/domain/employees/commands/handlers/employee-activator.handler.spec.ts:35:43
      at fulfilled (src/modules/domain/employees/commands/handlers/employee-activator.handler.spec.ts:5:58)


Test Suites: 1 failed, 50 passed, 51 total
Tests:       1 failed, 78 passed, 79 total
Snapshots:   0 total
Time:        41.542s
Ran all test suites.
npm ERR! code ELIFECYCLE
npm ERR! errno 1
npm ERR! glee2@1.0.0 test: `jest --runInBand`
npm ERR! Exit status 1
npm ERR!
npm ERR! Failed at the glee2@1.0.0 test script.
npm ERR! This is probably not a problem with npm. There is likely additional logging output above.

npm ERR! A complete log of this run can be found in:
npm ERR!     /home/circleci/.npm/_logs/2022-10-22T16_06_09_695Z-debug.log


Exited with code exit status 1
CircleCI received exit code 1
```

- Job that failed because of vulnerable packages. [SCREENSHOT03]



CI behind your firewall just got easier

Install a more scalable, Kubernetes-friendly self-hosted runner in 5 minutes or less.

test-backend Failed

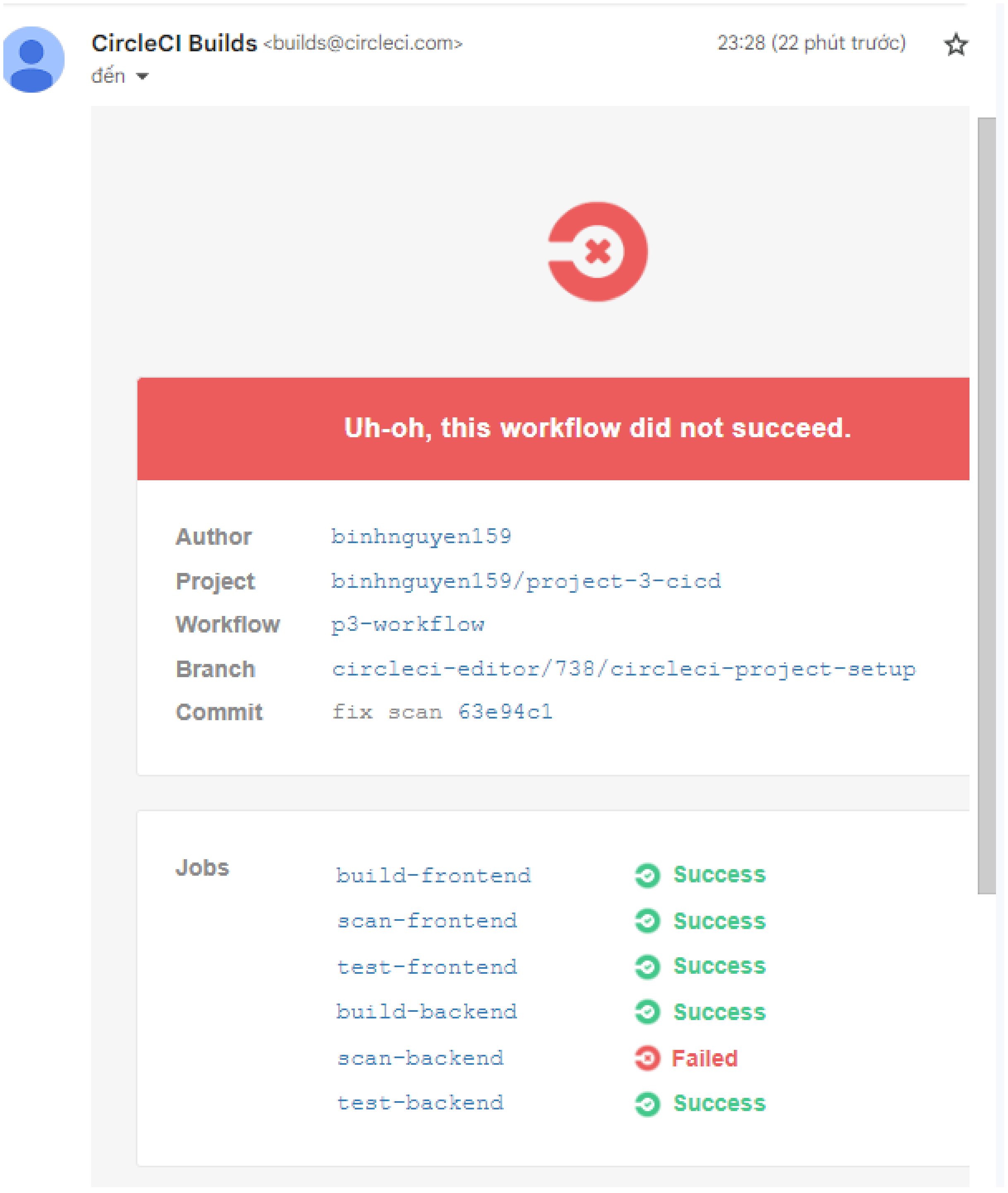
Duration / Finished	Queued	Executor / Resource Class	Branch	Commit	Author & Message
1m 8s / 5m ago	0s	Docker / Large	circleci-editor/738/circleci-project-setup	c82c442	update workflow

You're using a deprecated Docker convenience image. Upgrade to a next-gen Docker convenience image.

✓ Spin up environment	1s
✓ Preparing environment variables	0s
✓ Checkout code	0s
! cd backend npm install npm run test	1m 5s

```
    Number of calls: 1

      33 |
      34 |         // Assert
    > 35 |         expect(employeeRepository.findById).toBeCalledWith(100);
         |                                             ^
      36 |         expect(employeeRepository.save).toBeCalled();
      37 |       });
      38 |     });

      at src/modules/domain/employees/commands/handlers/employee-activator.handler.spec.ts:35:43
      at fulfilled (src/modules/domain/employees/commands/handlers/employee-activator.handler.spec.ts:5:58)


Test Suites: 1 failed, 50 passed, 51 total
Tests:       1 failed, 78 passed, 79 total
Snapshots:   0 total
Time:        41.542s
Ran all test suites.
npm ERR! code ELIFECYCLE
npm ERR! errno 1
npm ERR! glee2@1.0.0 test: `jest --runInBand`
npm ERR! Exit status 1
npm ERR!
npm ERR! Failed at the glee2@1.0.0 test script.
npm ERR! This is probably not a problem with npm. There is likely additional logging output above.

npm ERR! A complete log of this run can be found in:
npm ERR!     /home/circleci/.npm/_logs/2022-10-22T16_06_09_695Z-debug.log


Exited with code exit status 1
CircleCI received exit code 1
```

- An alert from one of your failed builds. [SCREENSHOT04]



CircleCI Builds <builds@circleci.com>
đến

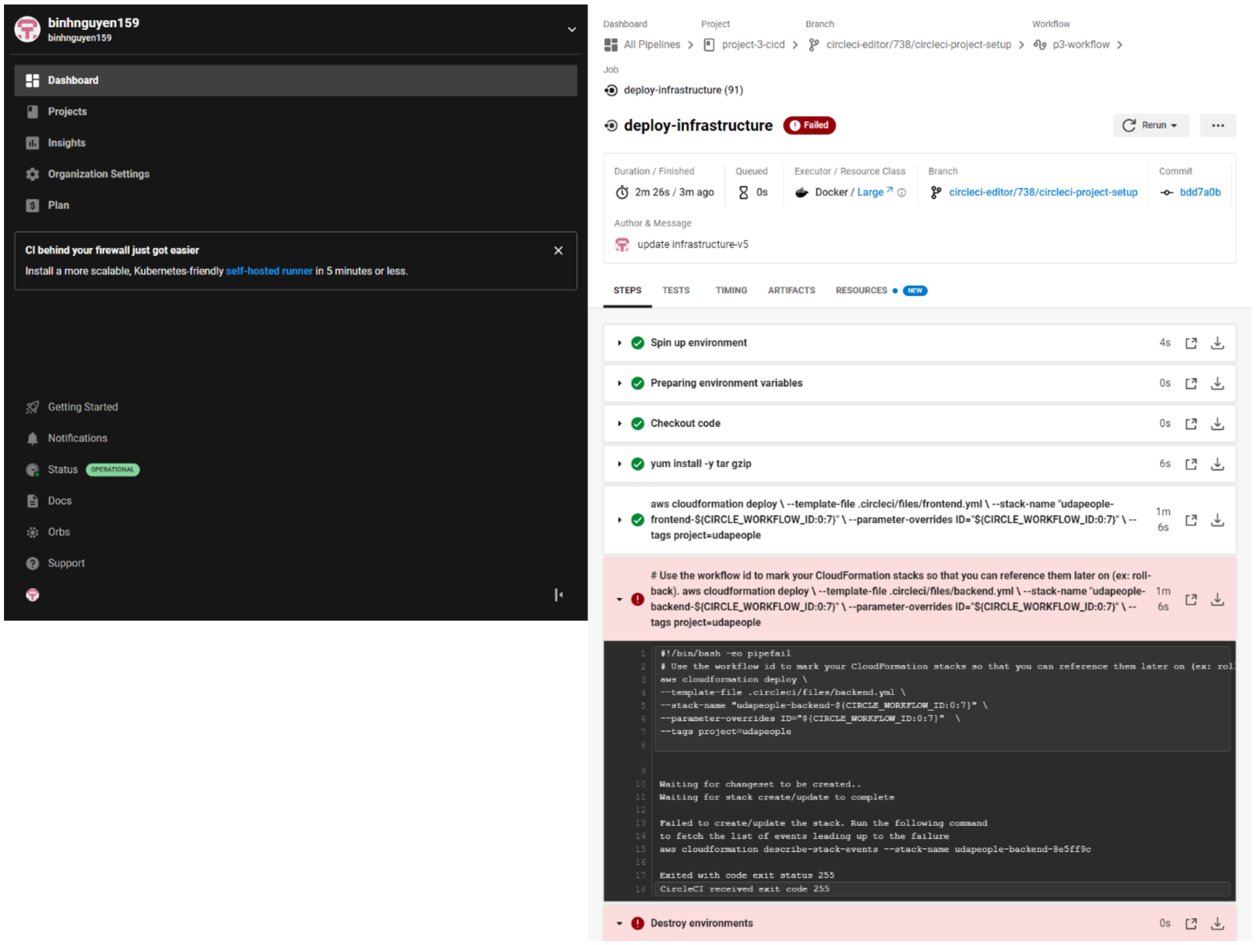
23:28 (22 phút trước)

Uh-oh, this workflow did not succeed.

Author	binhnguyen159
Project	binhnguyen159/project-3-cicd
Workflow	p3-workflow
Branch	circleci-editor/738/circleci-project-setup
Commit	fix scan 63e94c1

Jobs		
	build-frontend	Success
	scan-frontend	Success
	test-frontend	Success
	build-backend	Success
	scan-backend	Failed
	test-backend	Success

- Appropriate job failure for infrastructure creation. [SCREENSHOT05]



CI behind your firewall just got easier

Install a more scalable, Kubernetes-friendly self-hosted runner in 5 minutes or less.

deploy-infrastructure Failed

Duration / Finished	Queued	Executor / Resource Class	Branch	Commit
2m 26s / 3m ago	0s	Docker / Large	circleci-editor/738/circleci-project-setup	bdd7a0b

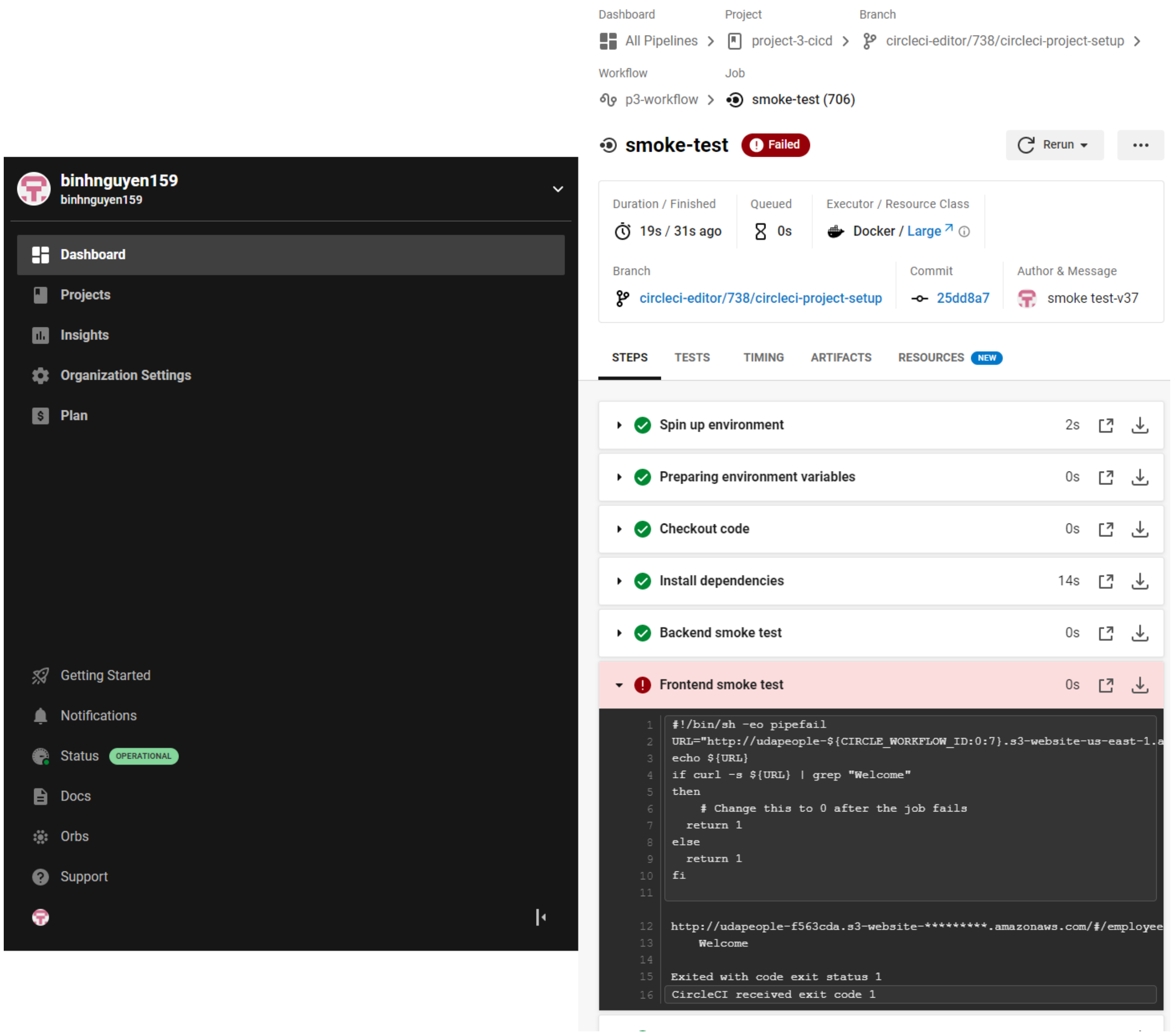
Author & Message: update infrastructure-v5

Step	Time
Spin up environment	4s
Preparing environment variables	0s
Checkout code	0s
yum install -y tar gzip	6s
aws cloudformation deploy \ --template-file .circleci/files/frontend.yml \ --stack-name "udapeople-frontend-${CIRCLE_WORKFLOW_ID:0:7}" \ --parameter-overrides ID="${CIRCLE_WORKFLOW_ID:0:7}" \ --tags project=udapeople	1m 6s
# Use the workflow id to mark your CloudFormation stacks so that you can reference them later on (ex: roll-back). aws cloudformation deploy \ --template-file .circleci/files/backend.yml \ --stack-name "udapeople-backend-${CIRCLE_WORKFLOW_ID:0:7}" \ --parameter-overrides ID="${CIRCLE_WORKFLOW_ID:0:7}" \ --tags project=udapeople	1m 6s

```
#!/bin/bash -eo pipefail
# Use the workflow id to mark your CloudFormation stacks so that you can reference them later on (ex: rol
aws cloudformation deploy \
--template-file .circleci/files/backend.yml \
--stack-name "udapeople-backend-${CIRCLE_WORKFLOW_ID:0:7}" \
--parameter-overrides ID="${CIRCLE_WORKFLOW_ID:0:7}"  \
--tags project=udapeople



Waiting for changeset to be created..
Waiting for stack create/update to complete

Failed to create/update the stack. Run the following command
to fetch the list of events leading up to the failure
aws cloudformation describe-stack-events --stack-name udapeople-backend-8e5ff9c

Exited with code exit status 255
CircleCI received exit code 255
```

Step	Time
Destroy environments	0s

- Appropriate job failure for the smoke test job. [SCREENSHOT06]



smoke-test Failed

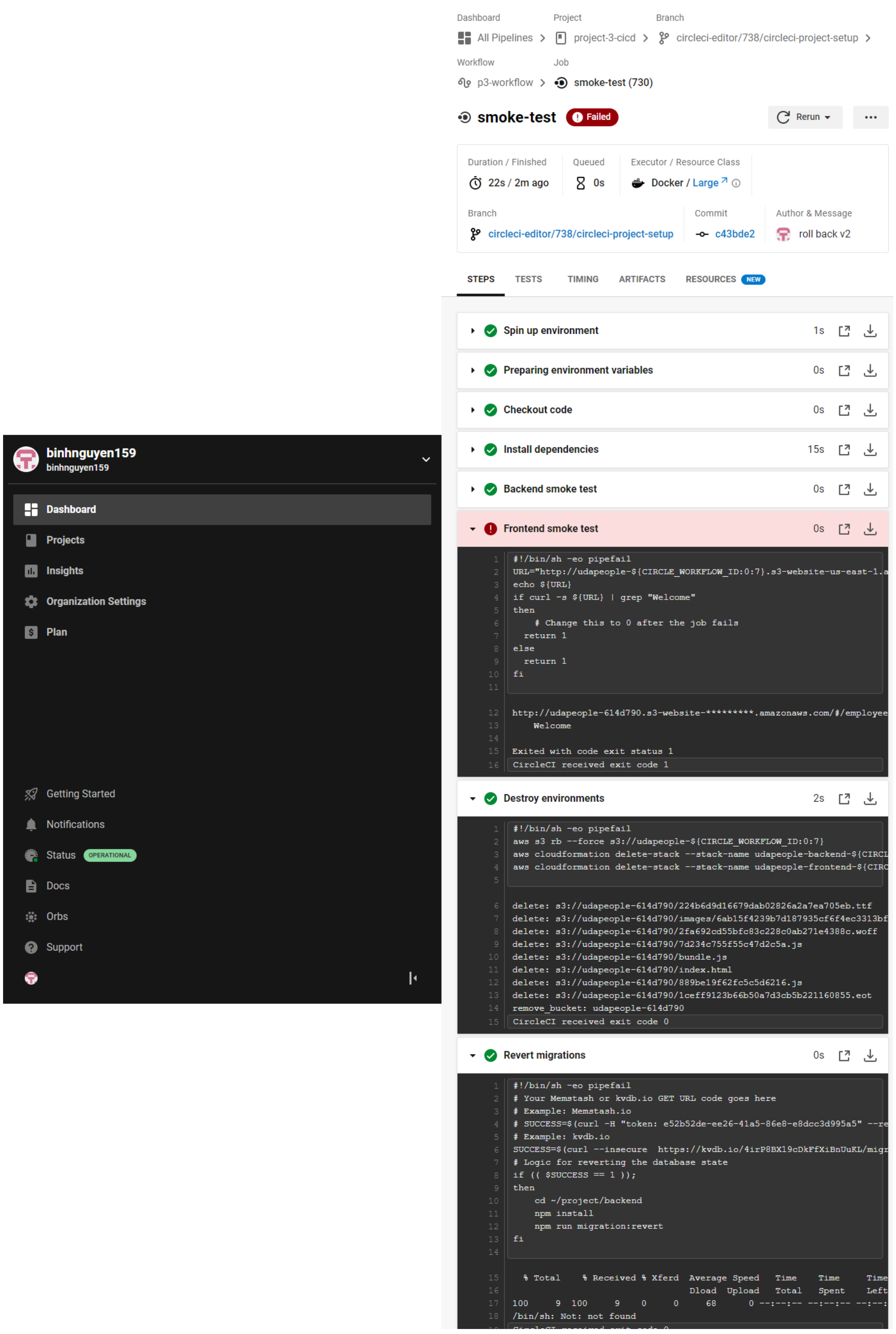
Duration / Finished	Queued	Executor / Resource Class
19s / 31s ago	0s	Docker / Large

Branch	Commit	Author & Message
circleci-editor/738/circleci-project-setup	25dd8a7	smoke test-v37

✓ Spin up environment	2s
✓ Preparing environment variables	0s
✓ Checkout code	0s
✓ Install dependencies	14s
✓ Backend smoke test	0s
! Frontend smoke test	0s

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


http://udapeople-f563cda.s3-website-*********.amazonaws.com/#/employee
    Welcome

Exited with code exit status 1
CircleCI received exit code 1
```

- Successful rollback after a failed smoke test. [SCREENSHOT07]



smoke-test Failed

Duration / Finished: 22s / 2m ago
Queued: 0s
Executor / Resource Class: Docker / Large
Branch: circleci-editor/738/circleci-project-setup
Commit: c43bde2
Author & Message: roll back v2

Spin up environment 1s
Preparing environment variables 0s
Checkout code 0s
Install dependencies 15s
Backend smoke test 0s
Frontend smoke test 0s

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


http://udapeople-614d790.s3-website-*********.amazonaws.com/#/employee
    Welcome

Exited with code exit status 1
CircleCI received exit code 1
```

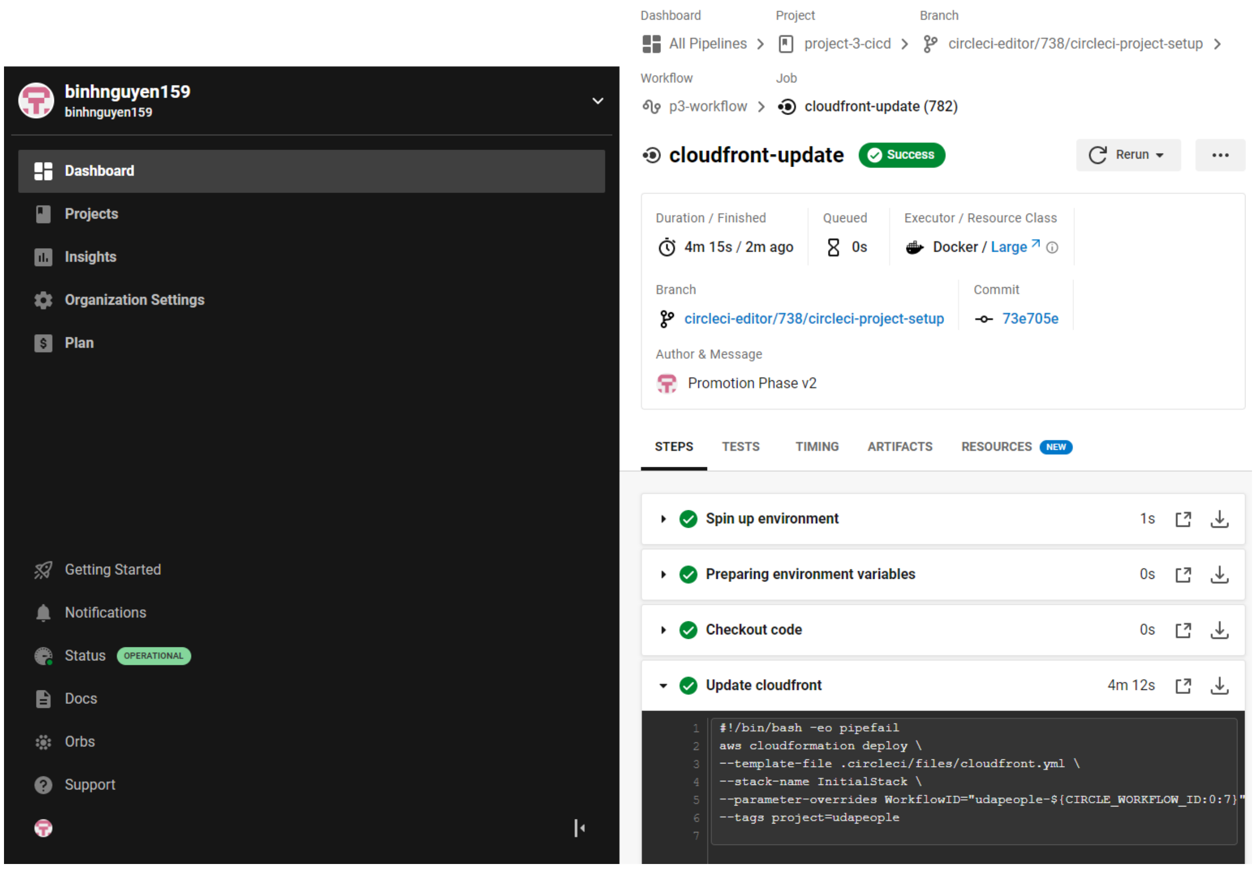
Destroy environments 2s

```
#!/bin/sh -eo pipefail
aws s3 rb --force s3://udapeople-${CIRCLE_WORKFLOW_ID:0:7}
aws cloudformation delete-stack --stack-name udapeople-backend-${CIRCL
aws cloudformation delete-stack --stack-name udapeople-frontend-${CIRC


delete: s3://udapeople-614d790/224b6d9d16679dab02826a2a7ea705eb.ttf
delete: s3://udapeople-614d790/images/6ab15f4239b7d187935cf6f4ec3313bf
delete: s3://udapeople-614d790/2fa692cd55bfc83c228c0ab271e4388c.woff
delete: s3://udapeople-614d790/7d234c755f55c47d2c5a.js
delete: s3://udapeople-614d790/bundle.js
delete: s3://udapeople-614d790/index.html
delete: s3://udapeople-614d790/889be19f62fc5c5d6216.js
delete: s3://udapeople-614d790/1ceff9123b66b50a7d3cb5b221160855.eot
remove_bucket: udapeople-614d790
CircleCI received exit code 0
```

Revert migrations 0s

```
#!/bin/sh -eo pipefail
# Your Memstash or kvdb.io GET URL code goes here
# Example: Memstash.io
# SUCCESS=$(curl -H "token: e52b52de-ee26-41a5-86e8-e8dcc3d995a5" --re
# Example: kvdb.io
SUCCESS=$(curl --insecure  https://kvdb.io/4irP8BX19cDkFfXiBnUuKL/migr
# Logic for reverting the database state
if (( $SUCCESS == 1 ));
then
    cd ~/project/backend
    npm install
    npm run migration:revert
fi


  % Total    % Received % Xferd  Average Speed   Time    Time     Time
                                 Dload  Upload   Total   Spent    Left
100     9  100     9    0     0     68      0 --:--:-- --:--:-- --:--:
/bin/sh: Not: not found
```

- Successful promotion job. [SCREENSHOT08]



cloudfront-update Success

Duration / Finished	Queued	Executor / Resource Class
4m 15s / 2m ago	0s	Docker / Large

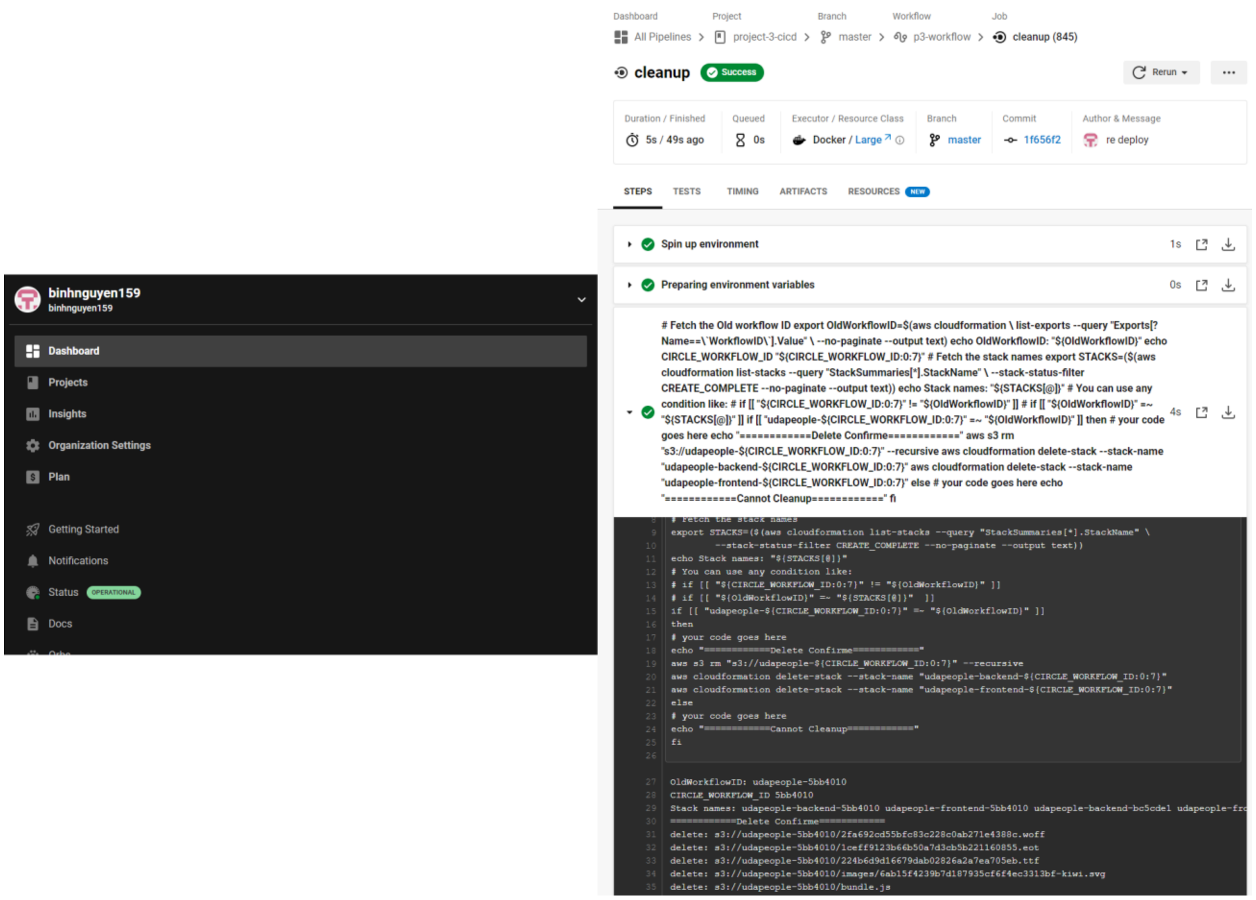
Branch	Commit
circleci-editor/738/circleci-project-setup	73e705e

Author & Message
Promotion Phase v2

Step	Time
Spin up environment	1s
Preparing environment variables	0s
Checkout code	0s
Update cloudfront	4m 12s

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

- Successful cleanup job. [SCREENSHOT09]



Duration / Finished: 5s / 49s ago
Queued: 0s
Executor / Resource Class: Docker / Large
Branch: master
Commit: 1f656f2
Author & Message: re deploy

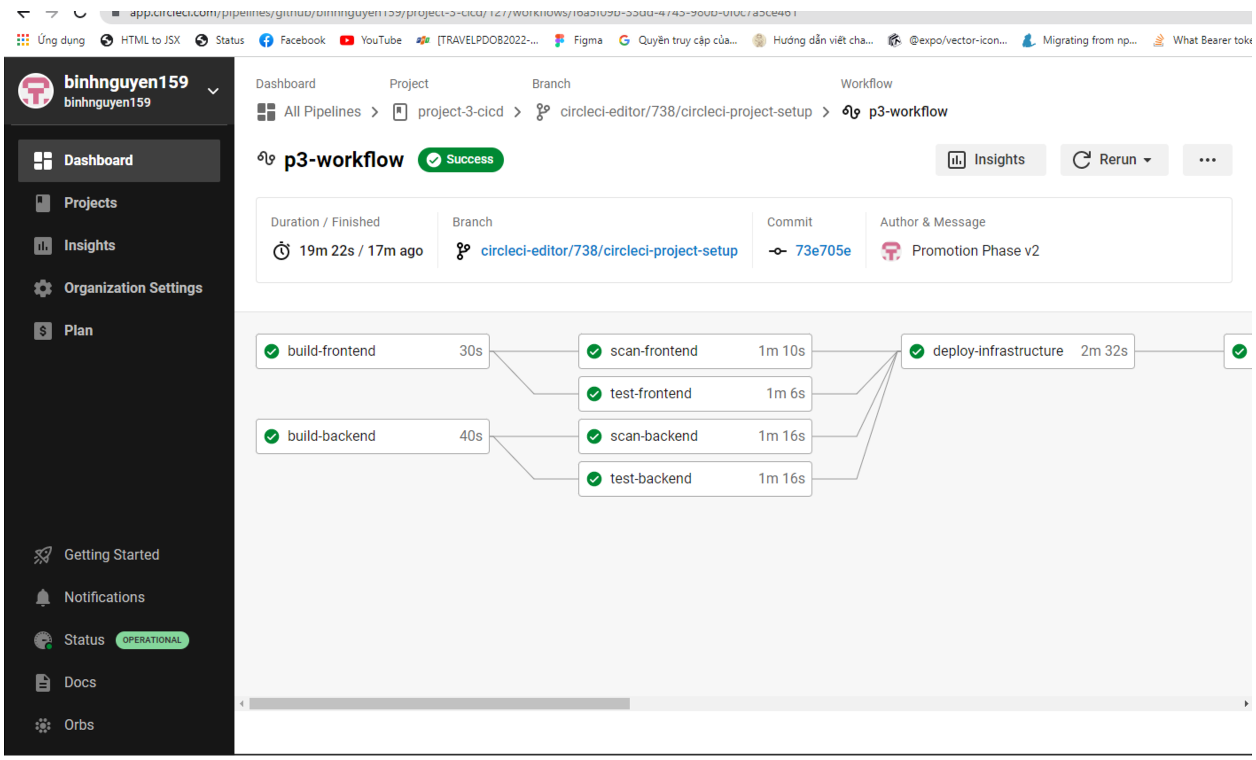
Spin up environment 1s

Preparing environment variables 0s

Fetch the Old workflow ID export OldWorkflowID=$(aws cloudformation \ list-exports --query "Exports[?Name==\`WorkflowID\`].Value" \ --no-paginate --output text) echo OldWorkflowID: "${OldWorkflowID}" echo CIRCLE_WORKFLOW_ID "${CIRCLE_WORKFLOW_ID:0:7}" # Fetch the stack names export STACKS=($(aws cloudformation list-stacks --query "StackSummaries[*].StackName" \ --stack-status-filter CREATE_COMPLETE --no-paginate --output text)) echo Stack names: "${STACKS[@]}" # You can use any condition like: # if [["${CIRCLE_WORKFLOW_ID:0:7}" != "${OldWorkflowID}"]] # if [["${OldWorkflowID}" =~ "${STACKS[@]}"]] if [["udapeople-${CIRCLE_WORKFLOW_ID:0:7}" =~ "${OldWorkflowID}"]] then # your code goes here echo "============Delete Confirme============" aws s3 rm "s3://udapeople-${CIRCLE_WORKFLOW_ID:0:7}" --recursive aws cloudformation delete-stack --stack-name "udapeople-backend-${CIRCLE_WORKFLOW_ID:0:7}" aws cloudformation delete-stack --stack-name "udapeople-frontend-${CIRCLE_WORKFLOW_ID:0:7}" else # your code goes here echo "============Cannot Cleanup============" fi 4s

```
# Fetch the stack names
export STACKS=($(aws cloudformation list-stacks --query "StackSummaries[*].StackName" \
        --stack-status-filter CREATE_COMPLETE --no-paginate --output text))
echo Stack names: "${STACKS[@]}"
# You can use any condition like:
# if [[ "${CIRCLE_WORKFLOW_ID:0:7}" != "${OldWorkflowID}" ]]
# if [[ "${OldWorkflowID}" =~ "${STACKS[@]}"  ]]
if [[ "udapeople-${CIRCLE_WORKFLOW_ID:0:7}" =~ "${OldWorkflowID}" ]]
then
# your code goes here
echo "============Delete Confirme============"
aws s3 rm "s3://udapeople-${CIRCLE_WORKFLOW_ID:0:7}" --recursive
aws cloudformation delete-stack --stack-name "udapeople-backend-${CIRCLE_WORKFLOW_ID:0:7}"
aws cloudformation delete-stack --stack-name "udapeople-frontend-${CIRCLE_WORKFLOW_ID:0:7}"
else
# your code goes here
echo "============Cannot Cleanup============"
fi

OldWorkflowID: udapeople-5bb4010
CIRCLE_WORKFLOW_ID 5bb4010
Stack names: udapeople-backend-5bb4010 udapeople-frontend-5bb4010 udapeople-backend-bc5cde1 udapeople-fr
============Delete Confirme============
delete: s3://udapeople-5bb4010/2fa692cd55bfc83c228c0ab271e4388c.woff
delete: s3://udapeople-5bb4010/1ceff9123b66b50a7d3cb5b221160855.eot
delete: s3://udapeople-5bb4010/224b6d9d16679dab02826a2a7ea705eb.ttf
delete: s3://udapeople-5bb4010/images/6ab15f4239b7d187935cf6f4ec3313bf-kiwi.svg
delete: s3://udapeople-5bb4010/bundle.js
```

- Only deploy on pushed to master branch. [SCREENSHOT10]

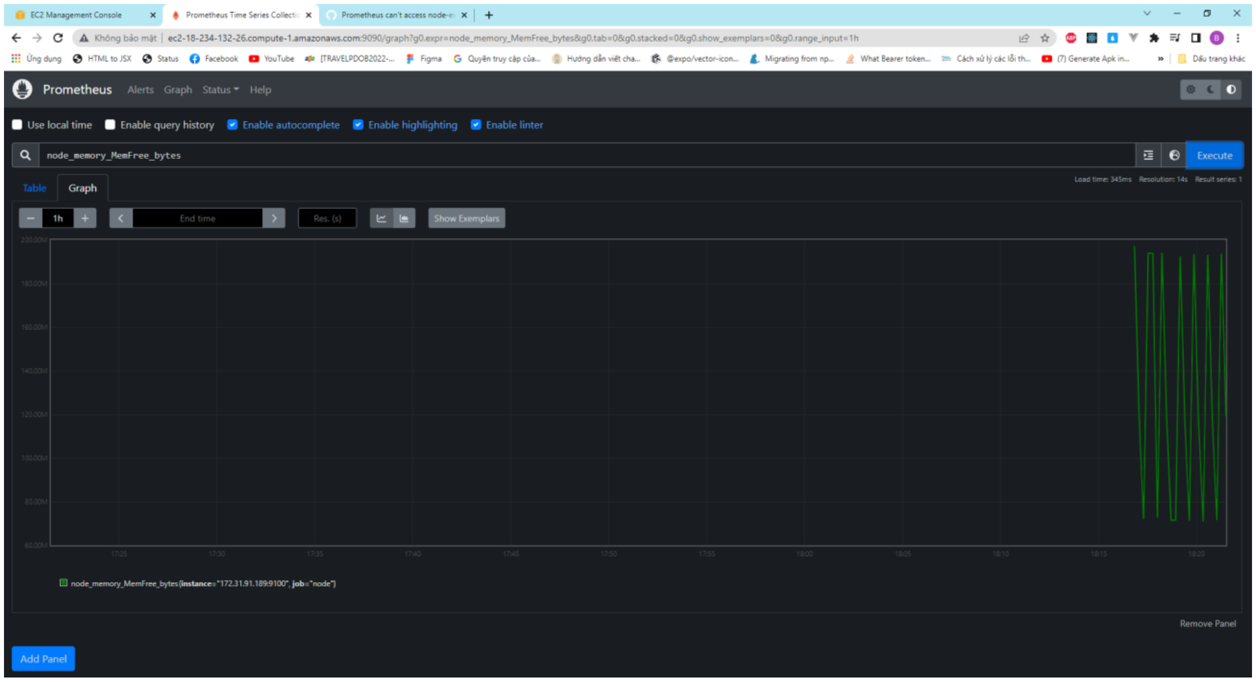


p3-workflow Success

Duration / Finished: 19m 22s / 17m ago
Branch: circleci-editor/738/circleci-project-setup
Commit: 73e705e
Author & Message: Promotion Phase v2

build-frontend 30s
build-backend 40s
scan-frontend 1m 10s
test-frontend 1m 6s
scan-backend 1m 16s
test-backend 1m 16s
deploy-infrastructure 2m 32s

- Provide a screenshot of a graph of your EC2 instance including available memory, available disk space, and CPU usage. [SCREENSHOT11]



ec2-18-234-132-26.compute-1.amazonaws.com:9090/graph?g0.expr=node_memory_MemFree_bytes&g0.tab=0&g0.stacked=0&g0.show_exemplars=0&g0.range_input=1h

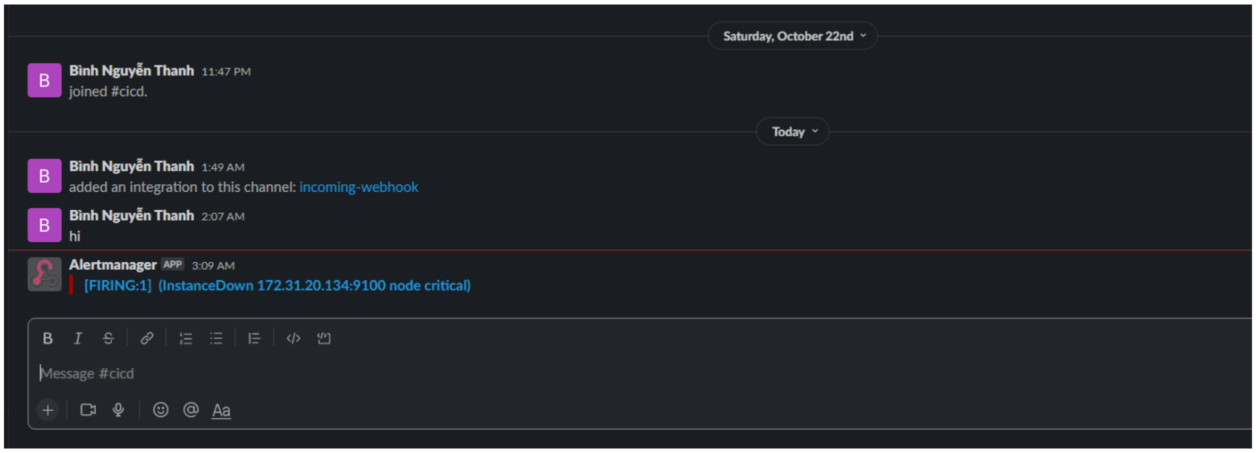
☐ Use local time ☐ Enable query history ☑ Enable autocomplete ☑ Enable highlighting ☑ Enable linter

```
node_memory_MemFree_bytes
```

Load time: 345ms Resolution: 14s Result series: 1

🟩 node_memory_MemFree_bytes{instance="172.31.91.189:9100", job="node"}

- Provide a screenshot of an alert that was sent by Prometheus. [SCREENSHOT12]



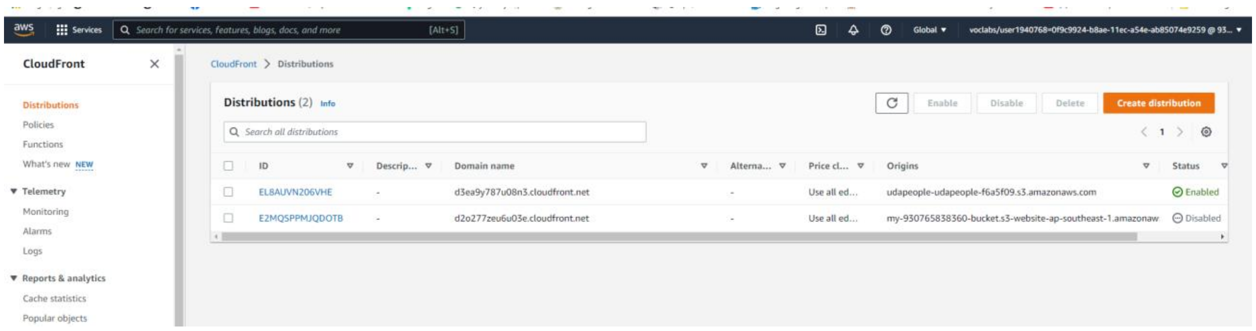
Binh Nguyễn Thanh 11:47 PM
joined #cicd.

Binh Nguyễn Thanh 1:49 AM
added an integration to this channel: incoming-webhook

Binh Nguyễn Thanh 2:07 AM
hi

Alertmanager APP 3:09 AM
[FIRING:1] (InstanceDown 172.31.20.134:9100 node critical)

- Provide a screenshot showing the evidence of deployed and functioning front-end application in CloudFront (aka, your production front-end). [URL03_SCREENSHOT]

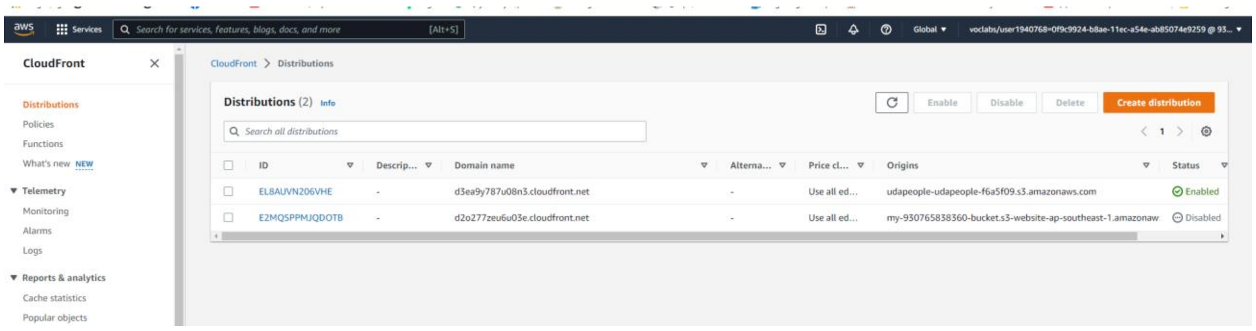


CloudFront > Distributions

Distributions (2)

ID	Descrip...	Domain name	Alterna...	Price cl...	Origins	Status
EL8AUVN206VHE	-	d3ea9y787u08n3.cloudfront.net	-	Use all ed...	udapeople-udapeople-f6a5f09.s3.amazonaws.com	Enabled
E2MQSPPMJQDOTB	-	d2o277zeu6u03e.cloudfront.net	-	Use all ed...	my-930765838360-bucket.s3-website-ap-southeast-1.amazonaw	Disabled

- Provide a screenshot showing the evidence of a healthy backend application. The backend endpoint status should show a healthy response. [URL04_SCREENSHOT]

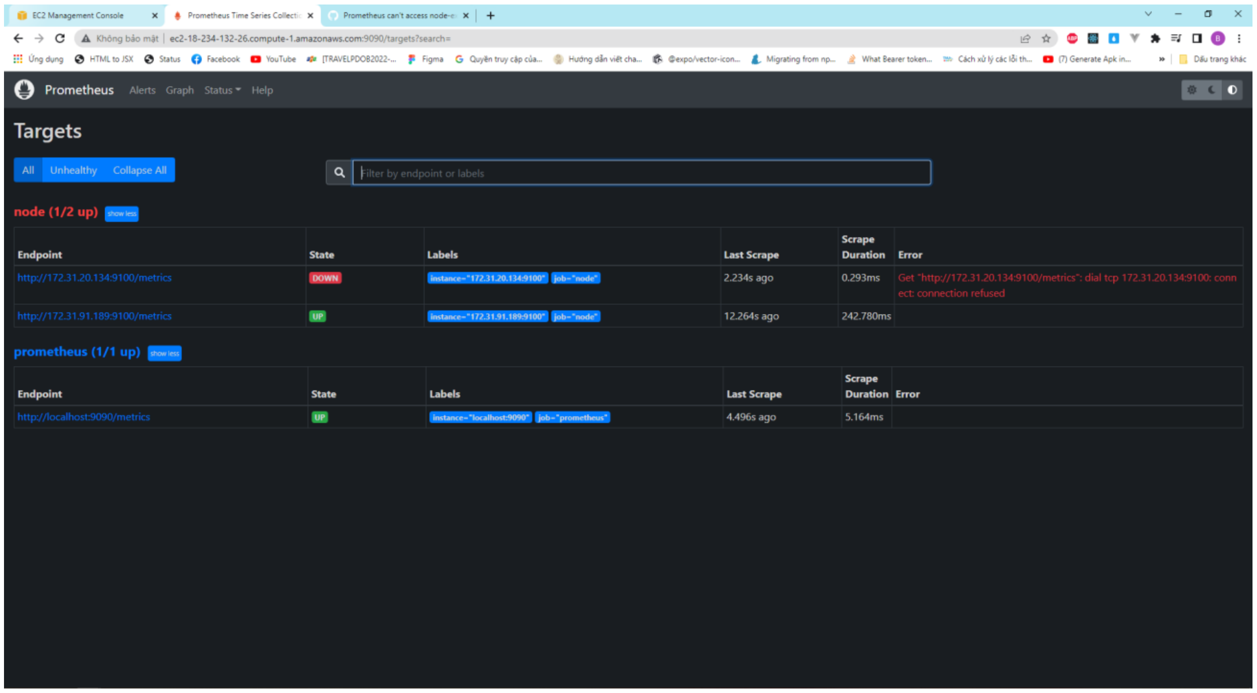


CloudFront > Distributions

Distributions (2)

ID	Descrip...	Domain name	Alterna...	Price cl...	Origins	Status
EL8AUVN206VHE	-	d3ea9y787u08n3.cloudfront.net	-	Use all ed...	udapeople-udapeople-f6a5f09.s3.amazonaws.com	Enabled
E2MQSPPMJQDOTB	-	d2o277zeu6u03e.cloudfront.net	-	Use all ed...	my-930765838360-bucket.s3-website-ap-southeast-1.amazonaw	Disabled

- Provide a screenshot of your Prometheus server showing UP state [URL05_SCREENSHOT]



ec2-18-234-132-26.compute-1.amazonaws.com:9090/targets?search=

Targets

node (1/2 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://172.31.20.134:9100/metrics	DOWN	instance="172.31.20.134:9100" job="node"	2.234s ago	0.293ms	Get "http://172.31.20.134:9100/metrics": dial tcp 172.31.20.134:9100: connect: connection refused
http://172.31.91.189:9100/metrics	UP	instance="172.31.91.189:9100" job="node"	12.264s ago	242.780ms	

prometheus (1/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	4.496s ago	5.164ms	