

# **Project: Give Your Application Auto-Deploy Superpowers CI/CD**

prepared by  
**Aya Rabih Mostafa**

[LinkedIn](#)

[GitHub](#)

## What is our plan?



**Let's start to run project**

### Step 1

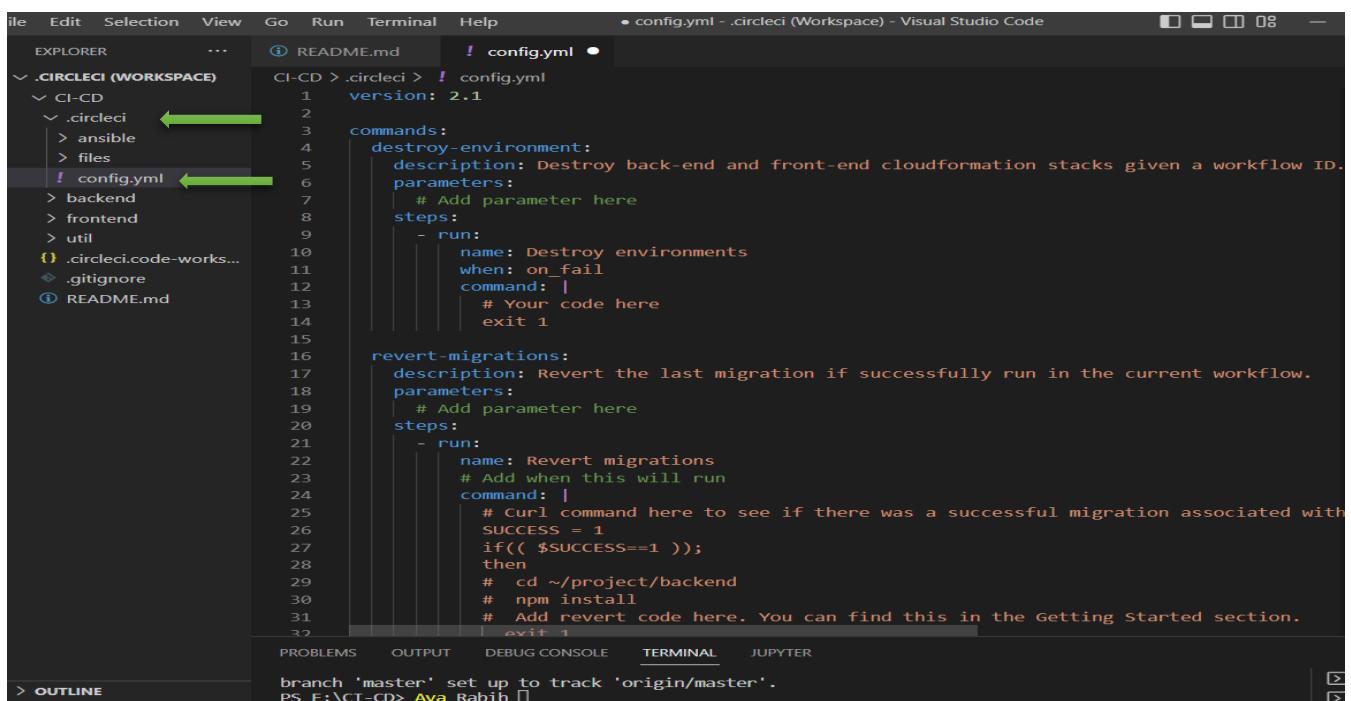
- **We will start in our project**
- First you should install git tool in this link <https://git-scm.com/>
- Second you should install node.js tool in this link
- Third you should install nvm tool in this link <https://github.com/coreybutler/nvm-windows/releases/tag/1.1.9>
- Command to git connect with git hub and your project
  1. Git init
  2. git remote add origin <https://github.com/ayarabih/udapeople.CICD.git>
  3. git add.
  4. git commit -m "initial commit"
  5. git push -u origin master
- now we upload our project in GitHub and we will get in second steps

### step 2

- we will open this link <https://app.circleci.com> it is tool for circles and select our project

The screenshot shows the CircleCI dashboard for the project 'udapeople.CICD'. The left sidebar has a dark theme with a user profile picture for 'ayarabih' and links for Dashboard, Projects, Insights, Organization Settings, and Plan. A callout box highlights a feature: 'CI behind your firewall just got easier. Install a more scalable, Kubernetes-friendly self-hosted runner in 5 minutes or less.' The main area shows the pipeline status for 'udapeople.CICD' with 1 job. A green arrow points to the error message in the log: 'ERROR IN CONFIG FILE: [#/jobs] 59 schema violations found. Any 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'.

➤ Now we will use image node:13.8.0

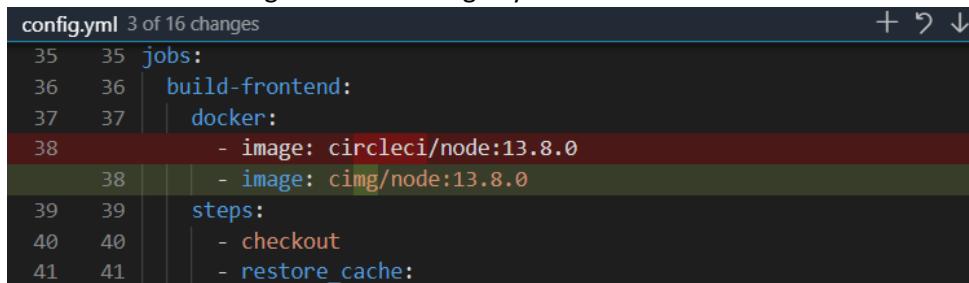


```
version: 2.1

commands:
  destroy-environment:
    description: Destroy back-end and front-end cloudformation stacks given a workflow ID.
    parameters:
      # Add parameter here
    steps:
      - run:
          name: Destroy environments
          when: on_fail
          command: |
            # Your code here
            exit 1

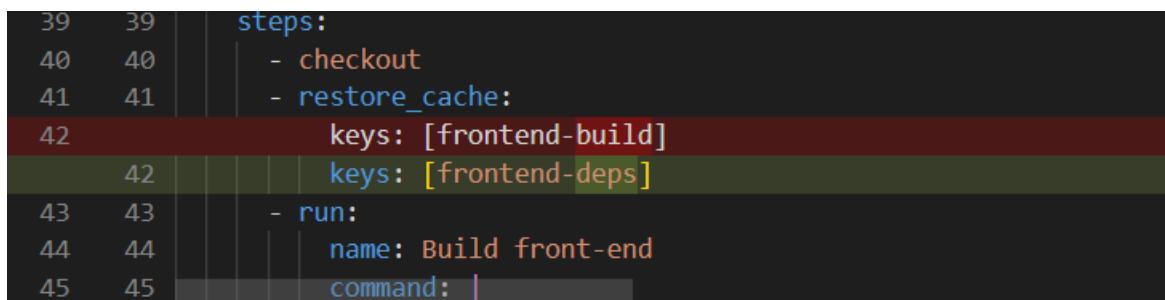
revert-migrations:
  description: Revert the last migration if successfully run in the current workflow.
  parameters:
    # Add parameter here
  steps:
    - run:
        name: Revert migrations
        # Add when this will run
        command: |
          # Curl command here to see if there was a successful migration associated with
          # SUCCESS = 1
          if(( $SUCCESS==1 )); then
            # cd ~/project/backend
            # npm install
            # Add revert code here. You can find this in the Getting Started section.
            exit 1
```

➤ Now we will change circleci to cimg keys frontend



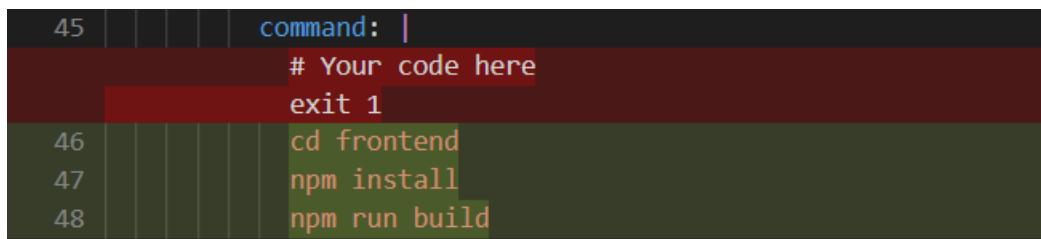
```
jobs:
  build-frontend:
    docker:
      - image: circleci/node:13.8.0
      - image: cimg/node:13.8.0
    steps:
      - checkout
      - restore_cache:
```

➤ Change key frontend build to deps to be more logic



```
steps:
  - checkout
  - restore_cache:
    keys: [frontend-build]
    keys: [frontend-deps]
  - run:
    name: Build front-end
    command: |
```

➤ Change command to install npm and run it



```
command: |
  # Your code here
  exit 1
  cd frontend
  npm install
  npm run build
```

- So we will change steps and command and key for frontend and backend

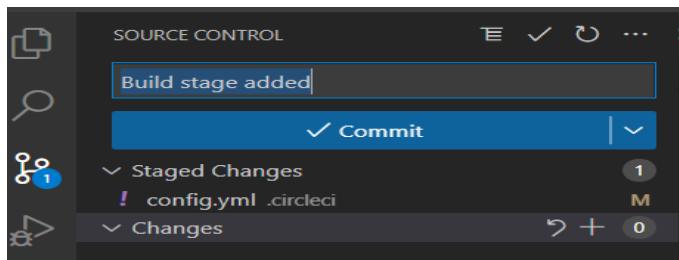
#Frontend

## #Backend

```
build-frontend:
  docker:
    - image: cimg/node:13.8.0
  steps:
    - checkout
    - restore_cache:
        keys: [frontend-deps]
    - run:
        name: Build front-end
        command: |
          cd frontend
          npm install
          npm run build
    - save_cache:
        paths: [frontend/node_modules]
        key: frontend-deps
```

```
build-backend:
  docker:
    - image: cimg/node:13.8.0
  steps:
    - checkout
    - restore_cache:
        keys: [backend-deps]
    - run:
        name: Back-end build
        command: |
          cd backend
          npm install
          npm run build
    - save_cache:
        paths: [backend/node_modules]
        key: backend-deps
```

- Now we will run by command #Build stage added



- Start to run

Pipeline	Status	Workflow	Branch / Commit	Start	Duration	Actions
udapeople.CICD_2	<span>Running</span>	default	 master b0ae504 Build stage added	11s ago	10s	   ...
udapeople.CICD_1	<span>WARN</span>	ERROR IN CONFIG FILE: [#/jobs] 59 schema violations found	Any 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			

- we found error in our project

The screenshot shows the CircleCI interface. On the left, there's a sidebar with user information (ayarabih, Aya Rabih) and navigation links (Dashboard, Projects, Insights, Organization Settings, Plan). A modal window is open with the message: "CI behind your firewall just got easier. Install a more scalable, Kubernetes-friendly self-hosted runner in 5 minutes or less." Below the sidebar, the main content area is titled "All Pipelines". It shows a table with one row for the pipeline "udapeople.CICD". The pipeline status is "Failed" (indicated by a red button). The workflow is "default". The branch is "master" (commit hash: b0ae504). The message "Build stage added" is shown. The timestamp is "2m ago". Below the table, an error message from the config file is displayed:

```

udapeople.CICD 1 ⚠️ ERROR IN CONFIG FILE:
D
  ▶ [#/jobs] 59 schema violations found
    Any 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

```

No more pipelines to load

- Our error in build-backend

The screenshot shows the CircleCI interface. The sidebar is identical to the previous one. The main content area shows the pipeline details for "udapeople.CICD". The workflow is "default" and is marked as "Failed". The duration is "45s / 1m ago". The branch is "master" (commit hash: b0ae504). The author message is "Build stage added". A green arrow points to the "build-backend" step, which is marked with a red exclamation icon and has a duration of "26s".

Step Status	Step Name	Duration
Success	build-frontend	43s
Failure	build-backend	26s

➤ Take screen for your error and read it

The screenshot shows a CircleCI build interface. At the top, a header bar displays the user 'ayarabih' (Aya Rabih), the branch 'master', and a build ID 'b0ae504'. Below the header, a sidebar on the left contains links for Dashboard, Projects, Insights, Organization Settings, Plan, and status information (Operational). A prominent message in the sidebar says: 'CI behind your firewall just got easier. Install a more scalable, Kubernetes-friendly self-hosted runner in 5 minutes or less.' The main area is titled 'Parallel runs' and shows a list of steps: 'Spin up environment' (1s), 'Preparing environment variables' (0s), 'Checkout code' (0s), 'Restoring cache' (0s), and 'Back-end build' (24s). The 'Back-end build' step is expanded, showing a terminal log. The log starts with a script to change directory and run npm commands. It then shows several postinstall scripts for dependencies like @nestjs/core and nodemon. A section of the log highlights a warning from npm audit: 'npm WARN glee2@1.0.0 No repository field.' and 'found 641 vulnerabilities (2 low, 144 moderate, 370 high, 125 critical)'. A green arrow points to the last line of the log.

```
#!/bin/bash -eo pipefail
cd backend
npm install
npm run build

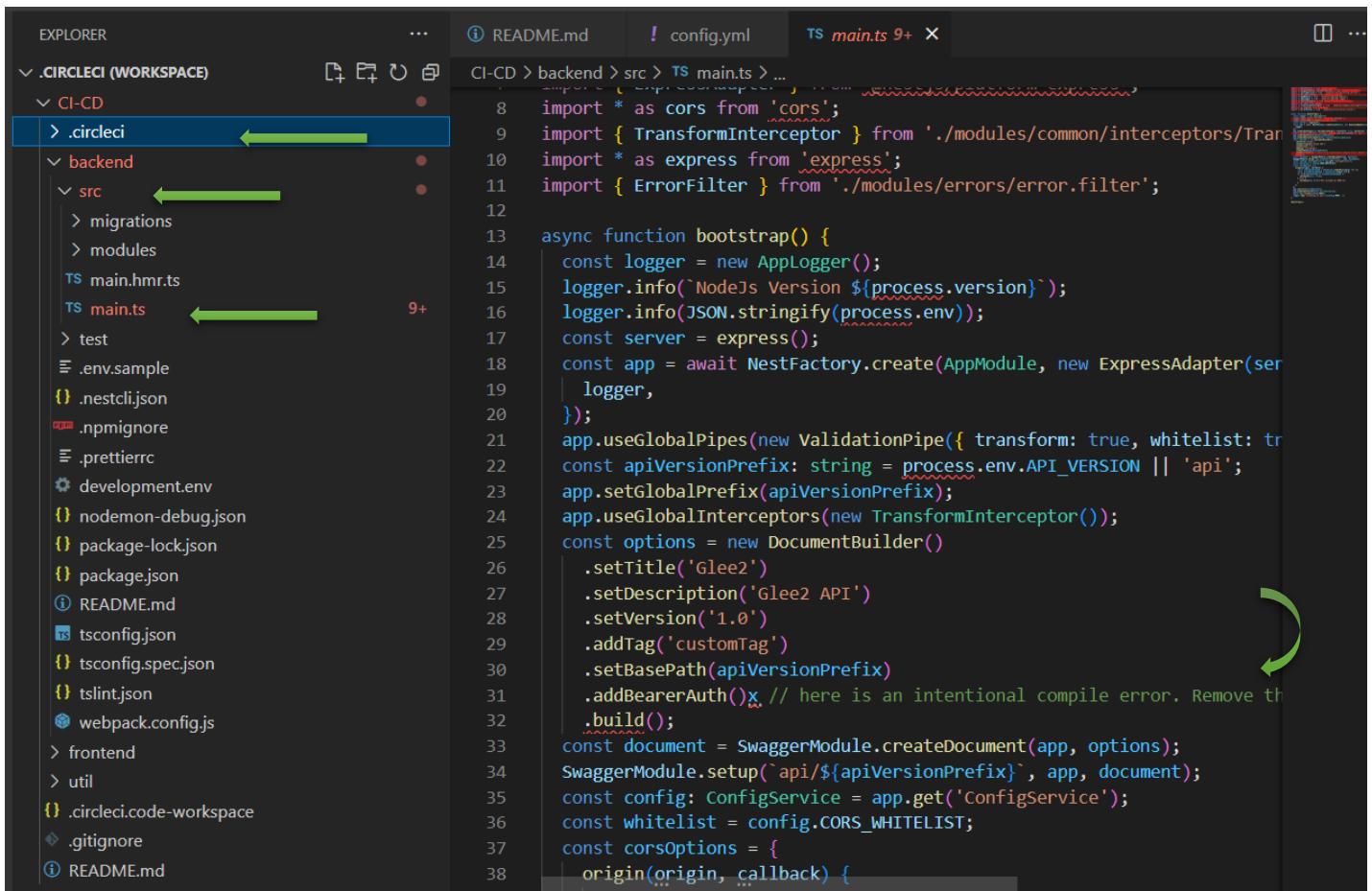
> @nestjs/core@6.11.11 postinstall /home/circleci/project/backend/node_modules/@nestjs/core
> opencollective || exit 0

> nodemon@1.19.4 postinstall /home/circleci/project/backend/node_modules/nodemon
> node bin/postinstall || exit 0

Love nodemon? You can now support the project via the open collective:
> https://opencollective.com/nodemon/donate

npm WARN glee2@1.0.0 No repository field.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.12: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","a
added 1311 packages from 1064 contributors and audited 1389 packages in 16.513s
23 packages are looking for funding
  run 'npm fund' for details

found 641 vulnerabilities (2 low, 144 moderate, 370 high, 125 critical)
  run 'npm audit fix' to fix them, or 'npm audit' for details
```



The screenshot shows a terminal window with a green arrow pointing from the command line to the pipeline configuration. The command is:

```
ci cd --config ./ci-cd/circleci.yml
```

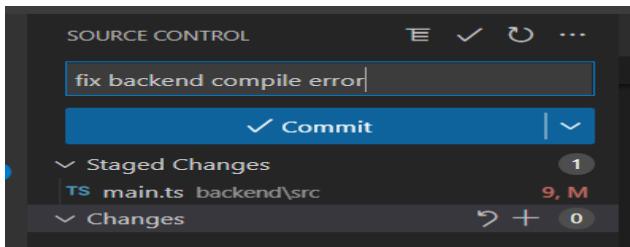
The pipeline configuration file contains the following code:

```

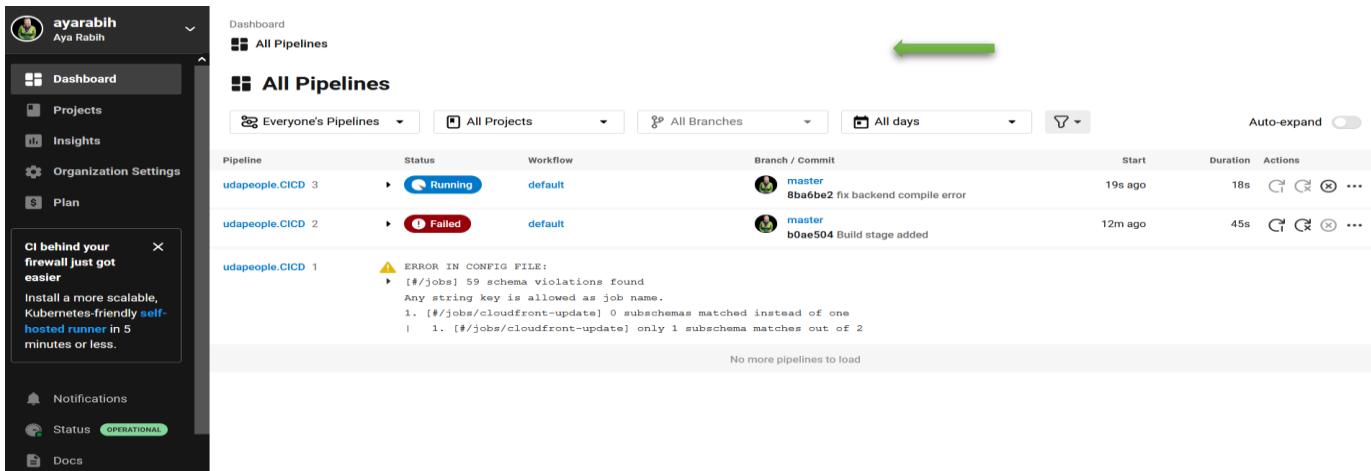
version: 2
jobs:
  build:
    steps:
      - run: npm install
      - run: npm run build
  test:
    steps:
      - run: npm test
  deploy:
    steps:
      - run: npm run deploy

```

➤ Now we will fix field job of backend with command #fix backend compile error



➤ Start to run



The screenshot shows the CircleCI dashboard under the "All Pipelines" section. Three pipelines are listed:

- udapeople.CI\_CD\_3**: Status: Running, Workflow: default, Branch / Commit: master b8ab6e2 fix backend compile error, Started: 19s ago, Duration: 18s.
- udapeople.CI\_CD\_2**: Status: Failed, Workflow: default, Branch / Commit: master b0ae504 Build stage added, Started: 12m ago, Duration: 45s.
- udapeople.CI\_CD\_1**: Status: Pending, Workflow: default, Branch / Commit: master b0ae504 Build stage added, Last updated: 12m ago.

A yellow warning icon is shown next to the udapeople.CI\_CD\_1 pipeline, indicating an error in the config file:

```

ERROR IN CONFIG FILE:
  • [#/jobs] 59 schema violations found
  Any 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

```

- Done we fix our error in build backend

The screenshot shows the CircleCI 'All Pipelines' interface. On the left is a sidebar with user info (ayarabih, Aya Rabih), navigation links (Dashboard, Projects, Insights, Organization Settings, Plan), and a note about CI behind firewalls. The main area displays three pipelines:

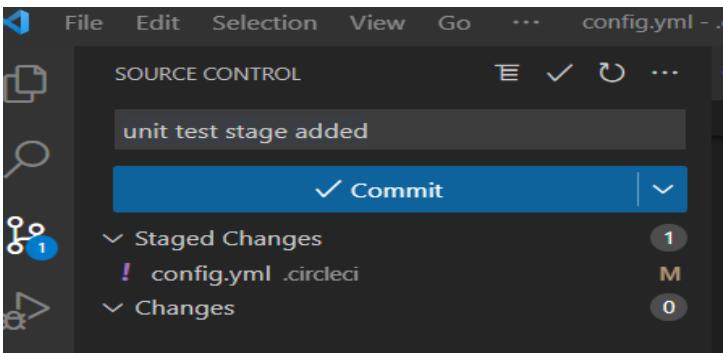
- udapeople.CI\_CD\_3**: Status Success, Workflow default, Branch master, Commit 8ba6be2 fix backend compile error. Contains jobs: build-frontend (3) and build-backend (4).
- udapeople.CI\_CD\_2**: Status Failed, Workflow default, Branch master, Commit b0ae504 Build stage added.
- udapeople.CI\_CD\_1**: Status Warning, Workflow default, Branch master. Shows an error in the config file: ERROR IN CONFIG FILE: [#/jobs] 59 schema violations found. Any 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.

No more pipelines to load.

- Let's test our file

```
test-frontend:
  docker:
    - image: cimg/node:13.8.0
  steps:
    - checkout
    - restore_cache:
        keys: [frontend-deps]
    - run:
        name: front-end unit test
        command: |
          cd frontend
          npm install
          npm test
test-backend:
  docker:
    - image: cimg/node:13.8.0
  steps:
    - checkout
    - restore_cache:
        keys: [backend-deps]
    - run:
        name: Back-end unit test
        command: |
          cd backend
          npm install
          npm test
```

- Now we will get unit test with command # unit test stage added



## ➤ Start to run

The screenshot shows the CircleCI All Pipelines dashboard. On the left, there's a sidebar with user information (ayarabih, Aya Rabih) and navigation links (Dashboard, Projects, Insights, Organization Settings, Plan). A sidebar message says "CI behind your firewall just got easier" and "Install a more scalable, Kubernetes-friendly self-hosted runner in 5 minutes or less." Below the sidebar are buttons for Notifications, Status (OPERATIONAL), and Docs.

The main area displays a table of pipelines:

Pipeline	Status	Workflow	Branch / Commit	Start	Duration	Actions				
udapeople.CICD 5	Running	default	master d578278 unit test stage added	34s ago	33s					...
udapeople.CI CD 4	<span style="color: yellow;">⚠</span> Unable to parse YAML while parsing a block mapping in 'string', line 71, column 5: docker: ^	default	master 8ba6be2 fix backend compile error	15m ago	37s					...
udapeople.CICD 3	Success	default	master b0ae504 Build stage added	27m ago	45s					...
udapeople.CI CD 2	<span style="color: red;">❗</span> Failed	default	master b0ae504 Build stage added	27m ago	45s					...
udapeople.CI CD 1	<span style="color: yellow;">⚠</span> ERROR IN CONFIG FILE: [#/jobs] 59 schema violations found Any string key is allowed as job name. 1. [#/jobs/cloudfront-update] 0 subschemas matched instead of one   1. [#/jobs/cloudfront-update] only 1 subschemas matches out of 2?	default	master b0ae504 Build stage added	27m ago	45s					...

## ➤ We found error in test for frontend and backend

This screenshot shows the same CircleCI All Pipelines dashboard as the previous one, but with a different pipeline state. The "udapeople.CICD 5" pipeline is now listed as Failed. The "Jobs" section shows the following details:

Job	Status	Start	Duration	Actions				
build-frontend 5	Success	2m ago	27s					...
test-frontend 8	Failure	2m ago	59s					...
build-backend 6	Success	2m ago	23s					...
test-backend 7	Failure	2m ago	48s					...

The other pipelines (udapeople.CI CD 4, udapeople.CICD 3, udapeople.CICD 2, udapeople.CICD 1) remain in their previous states from the first screenshot.

## ➤ Error for frontend

The screenshot shows the CircleCI web interface for a project named 'ayarabih'. On the left, there's a sidebar with navigation links like Dashboard, Projects, Insights, Organization Settings, and Plan. A prominent message box in the center says: 'CI behind your firewall just got easier' and 'Install a more scalable, Kubernetes-friendly self-hosted runner in 5 minutes or less.' Below this, there are buttons for Notifications, Status (labeled 'OPERATIONAL'), Docs, and Orbs.

The main area displays a build log for a 'front-end unit test'. The log shows the command: `#!/bin/bash -eo pipefail` followed by `cd frontend` and `npm install`. It then runs `npm test` which triggers several npm warnings:

```
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"any"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules/chokidar/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"any"})
audited 1896 packages in 6.128s
found 664 vulnerabilities (5 low, 184 moderate, 338 high, 137 critical)
  run 'npm audit fix' to fix them, or 'npm audit' for details
> typescript-react-redux-boilerplate@1.0.0 test /home/circleci/project/frontend
> jest
  • Deprecation Warning:
    Option "setupTestFrameworkScriptFile" was replaced by configuration "setupFilesAfterEnv", which supports multiple paths.
    PLEASE UPDATE YOUR CONFIGURATION
```

The total duration of the test is 51 seconds.

## ➤ Error for backend

This screenshot shows the same CircleCI interface for the 'ayarabih' project. The sidebar and central message box are identical to the first screenshot.

The main area displays a build log for a 'Back-end unit test'. The log shows the command: `#!/bin/bash -eo pipefail` followed by `cd backend` and `npm install`. It then runs `npm test` which triggers several npm warnings:

```
npm WARN @nestjs/cqrs@6.1.0 requires a peer of reflect-metadata@0.1.12 but none is installed. You must install peer dependencies yourself.
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.
npm WARN winston-slack-webhook@0.1.6 requires a peer of winston@2.x.x but none is installed. You must install peer dependencies yourself.
npm WARN glee2@1.0.0 No repository field.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.12: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"any"})
audited 1389 packages in 7.076s
24 packages are looking for funding
  run 'npm fund' for details
found 641 vulnerabilities (2 low, 144 moderate, 370 high, 125 critical)
  run 'npm audit fix' to fix them, or 'npm audit' for details
> glee2@1.0.0 test /home/circleci/project/backend
> jest --runInBand
  • Console
```

The total duration of the test is 42 seconds.

➤ Now we will fix error unit test for frontend and backend Frontend

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER:** Shows the project structure under ".CIRCLECI (WORKSPACE) / CI-CD / frontend / src / app / components / LoadingMessage". The file "LoadingMessage.spec.tsx" is selected.
- EDITOR:** Displays the content of "LoadingMessage.spec.tsx":

```
1 import * as React from 'react';
2 import { shallow } from 'enzyme';
3 import { LoadingMessage } from 'app/components>LoadingMessage';

4 describe('<LoadingMessage>', () => {
5   describe('Props', () => {
6     describe('message', () => {
7       it('Should render the props message', () => {
8         const message = 'Hello!';
9         const wrapper = shallow(<LoadingMessage message={message} />);
10        expect(wrapper.contains(<span>{message}</span>)).toBeTruthy();
11      });
12    });
13  });
14});
15});
16});
```
- STATUS BAR:** Shows "LoadingMessage.spec.tsx - .circleci (Workspace) - Visual Studio Code".

Green arrows highlight the path from the "LoadingMessage" folder in the Explorer to the "LoadingMessage.spec.tsx" file in the Editor, indicating the scope of the current test.

The screenshot shows the Visual Studio Code interface with the following details:

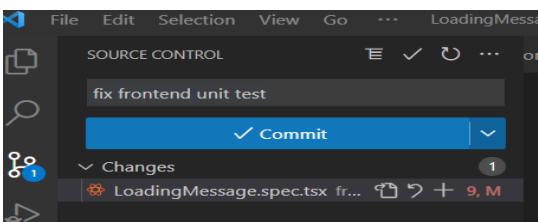
- EXPLORER:** Shows the project structure under ".CIRCLECI (WORKSPACE) / CI-CD / frontend / src / app / components / LoadingMessage". The file "LoadingMessage.spec.tsx" is selected.
- EDITOR:** Displays the content of "LoadingMessage.spec.tsx":

```
1 import * as React from 'react';
2 import { shallow } from 'enzyme';
3 import { LoadingMessage } from 'app/components>LoadingMessage';

4 describe('<LoadingMessage>', () => {
5   describe('Props', () => {
6     describe('message', () => {
7       it('Should render the props message', () => {
8         const message = 'Hello!';
9         const wrapper = shallow(<LoadingMessage message={message} />);
10        expect(wrapper.contains(<span>{message}</span>)).toBeTruthy();
11      });
12    });
13  });
14});
15});
16});
```
- STATUS BAR:** Shows "LoadingMessage.spec.tsx - .circleci (Workspace) - Visual Studio Code".

A green arrow points upwards from the bottom of the code editor towards the "LoadingMessage.spec.tsx" tab in the status bar, indicating a change or update to the file.

- Now we will fix field job of frontend with command #fix frontend unit test



- Start to run and we found error in backend

The screenshot shows the CircleCI dashboard for the project 'udapeople.CI\_CD'. A green arrow points to the 'Failed' status of the build. The build details show it was triggered by a pull request 'd2f1a48 fix frontend unit test' from the 'master' branch. The build duration was 2m ago, and the total time was 1m 31s. The jobs listed are: build-frontend (27s), test-frontend (1m 0s), build-backend (23s), and test-backend (48s). The test-backend job is marked with a red exclamation point, indicating failure.

- Look for error change code from 101 to 100

The screenshot shows the Visual Studio Code editor with the file 'employee-activator.handler.spec.ts' open. A green arrow points to the line of code where 'employeeId' is set to 101. The code is a Jest test for an EmployeeActivator handler. It imports EmployeeRepository, ActivateEmployee, and EmployeeActivator. It uses a MockEmployeeRepository to test the activateEmployee function. The test checks if the employee's isActive status is set to false when the employeeId is 101. A large green arrow points from the line with the error to the line where the value is being changed from 101 to 100.

```

import { EmployeeRepository } from '../repositories/employees.repository';
import { ActivateEmployee } from '../activate-employee.command';
import { EmployeeActivator } from './employee-activator.handler';

describe('Employee Remover', () => {
  describe('when a user activates an employee', () => {
    const MockEmployeeRepository = jest.fn(
      () =>
      {
        findById: jest.fn().mockResolvedValue([{}]),
        save: jest.fn(),
      } as any,
    );
    const employeeRepository = new MockEmployeeRepository();

    it('should activate the employee from the repository', async () => {
      // Arrange
      const handler = new EmployeeActivator(employeeRepository);

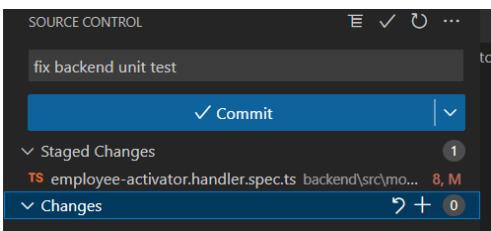
      const params = {
        employeeId: 101, //change this to 100 to make the test pass
        isActive: false,
      };

      const activateEmployeeCommand = new ActivateEmployee(
        params.employeeId,
        params.isActive,
      );

      // Act
      await handler.handle(activateEmployeeCommand);
    });
  });
})

```

- Now we will fix field job of backend with command # **fix backend unit test**



- Start to run and success

Pipeline	Status	Workflow	Branch / Commit	Start	Duration	Actions
udapeople.CICD 7	<span style="color: green;">Success</span>	default	master 417a4d0 fix backend unit test	2m ago	1m 30s	...
		Jobs	build-frontend 14 test-frontent 16 build-backend 13 test-backend 15		26s 59s 21s 58s	
udapeople.CICD 6	<span style="color: red;">Failed</span>	default	master d2f1a48 fix frontend unit test	11m ago	1m 31s	...
udapeople.CICD 5	<span style="color: red;">Failed</span>	default	master d578278 unit test stage added	31m ago	1m 30s	...

- Now we will scan front end and backend

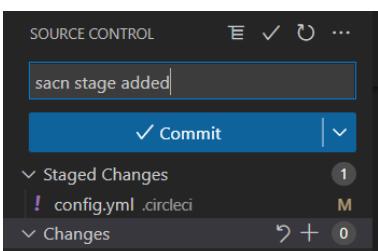
```

scan-frontend:
  docker:
    - image: cimg/node:13.8.0
  steps:
    - checkout
    - restore_cache:
        keys: [frontend-deps]
    - run:
        name: front-end scan
        command:
          cd frontend
          npm install
          npm audit --audit-level=critical

scan-backend:
  docker:
    - image: cimg/node:13.8.0
  steps:
    - checkout
    - restore_cache:
        keys: [backend-deps]
    - run:
        name: Back-end scan
        command:
          cd backend
          npm install
          npm audit --audit-level=critical

```

- Now we will fix field job of frontend with command # **scan stage added**



➤ Now we found error in scan in frontend and backend

The screenshot shows the CircleCI 'All Pipelines' interface. A single pipeline named 'udapeople.CI\_CD' is listed under the 'Everyone's Pipelines' tab. The status is 'Failed'. The workflow is 'default' and it was triggered by 'master' branch commit 'a527c41' with the message 'sacn stage added'. The pipeline started 8m ago and took 1m 48s. The jobs section details the following steps:

Job	Status	Duration
build-frontend	Passed	35s
scan-frontend	Failed	31s
test-frontend	Passed	1m 6s
build-backend	Passed	24s
scan-backend	Failed	25s
test-backend	Passed	58s

➤ Look for error for frontend

The screenshot shows the CircleCI pipeline logs for the 'front-end scan' step. The log output is as follows:

```
Spin up environment
Preparing environment variables
Checkout code
Restoring cache
front-end scan
Your output is too large to display in the browser.
Only the last 400000 characters are displayed.
Download the full output as a file
5501 | Moderate | Prototype Pollution in Ajv
5502 | Package | ajv
5503 | Patched in | >=6.12.3
5504 | Dependency of | glee-jvks-rsa
5505 | Path | glee-jvks-rsa > request > har-validator > ajv
5506 | More info | https://github.com/advisories/GHSA-w7cc-xwvz-8qzg
5507 |
5508 | High | node-fetch is vulnerable to Exposure of Sensitive Information so an Unauthorized Actor
5509 | Package | node-fetch
5510 | Patched in | >=2.6.7
5511 | Dependency of | react-foundation
5512 | Path | react-foundation > fbjs > isomorphic-fetch > node-fetch
5513 | More info | https://github.com/advisories/GHSA-r683-j2x4-v87g
5514 |
5515 | Moderate | node-fetch is vulnerable to Exposure of Sensitive Information so an Unauthorized Actor
5516 | Package | node-fetch
5517 | Patched in | >=2.6.7
5518 | Dependency of | react-foundation
5519 | Path | react-foundation > fbjs > isomorphic-fetch > node-fetch
5520 | More info | https://github.com/advisories/GHSA-r683-j2x4-v87g
5521 |
5522 | Moderate | found 664 vulnerabilities (5 low, 184 moderate, 338 high, 137 critical) in 1896 scanned packages
5523 | Run 'npm audit fix' to fix 484 of them.
5524 | 176 vulnerabilities require semver-major dependency updates.
5525 | 2 vulnerabilities require manual review. See the Full report for details.
5526 |
5527 | Exited with code exit status 1
5528 | CircleCI received exit code 1
```

➤ Look for error for backend

The screenshot shows the CircleCI pipeline logs for the 'Back-end scan' step. The log output is as follows:

```
Checkout code
Restoring cache
Back-end scan
Your output is too large to display in the browser.
Only the last 400000 characters are displayed.
Download the full output as a file
5501 | Moderate | Prototype Pollution in Ajv
5502 | Package | ajv
5503 | Patched in | >=6.12.3
5504 | Dependency of | glee-jvks-rsa
5505 | Path | glee-jvks-rsa > request > har-validator > ajv
5506 | More info | https://github.com/advisories/GHSA-w88q-czmm-v5xw
5507 |
5508 | Moderate | Prototype Pollution in Ajv
5509 | Package | ajv
5510 | Patched in | >=6.12.3
5511 | Dependency of | winston-loggly-bulk
5512 | Path | winston-loggly-bulk > node-loggly-bulk > request > har-validator > ajv
5513 | More info | https://github.com/advisories/GHSA-v88g-cgmx-v5xw
5514 |
5515 | Moderate | found 641 vulnerabilities (2 low, 144 moderate, 370 high, 125 critical) in 1389 scanned packages
5516 | Run 'npm audit fix' to fix 500 of them.
5517 | 134 vulnerabilities require semver-major dependency updates.
5518 | 7 vulnerabilities require manual review. See the Full report for details.
5519 |
5520 | Exited with code exit status 1
5521 | CircleCI received exit code 1
```

ayarabih  
Aya Rabih

**Dashboard**

**Projects**

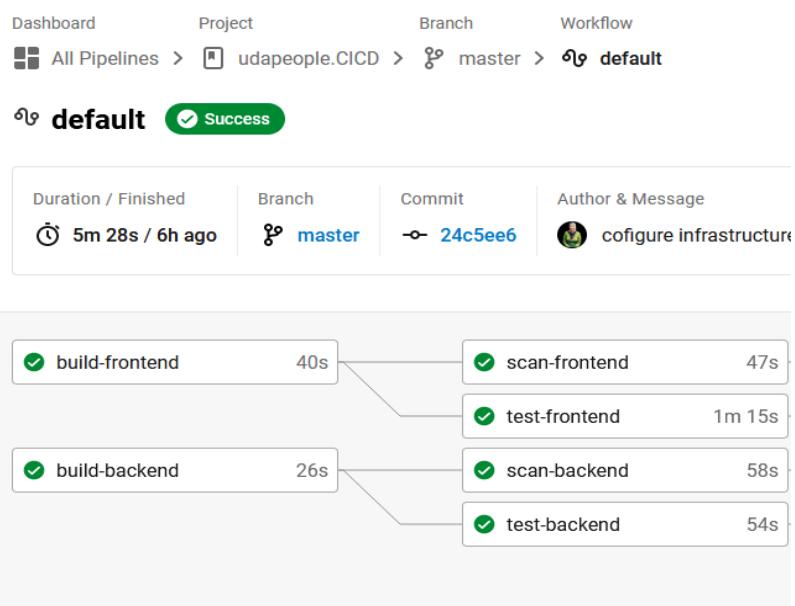
**Insights**

**Organization Settings**

**Plan**

**CI behind your firewall just got easier**

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



➤ Now we will fix vulnerability scan

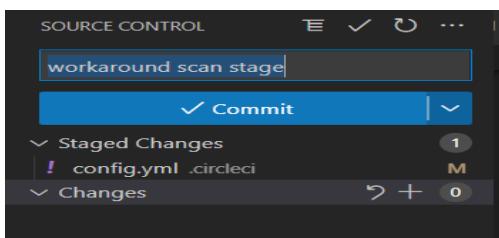
```

scan-frontend:
  docker:
    - image: cimg/node:13.8.0
  steps:
    - checkout
    - restore_cache:
        keys: [frontend-deps]
    - run:
        name: front-end scan
        command: |
          cd frontend
          npm install
          npm audit fix --force --audit-level=critical ←
          npm audit --audit-level=critical

scan-backend:
  docker:
    - image: cimg/node:13.8.0
  steps:
    - checkout
    - restore_cache:
        keys: [backend-deps]
    - run:
        name: Back-end scan
        command: |
          cd backend
          npm install
          npm audit fix --force --audit-level=critical ←
          npm audit fix --force --audit-level=critical
          npm audit --audit-level=critical

```

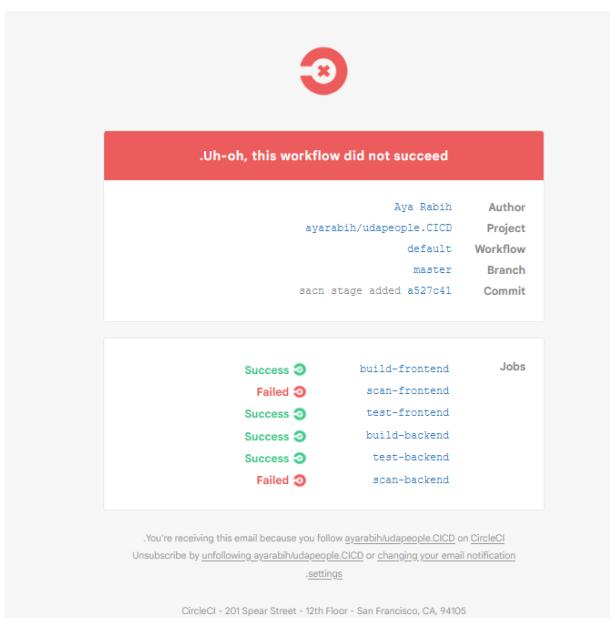
➤ Now we will fix field job vulnerability scan of frontend and backend with command # workaround scan stage



- Now we fix error in scan for backend and frontend

The screenshot shows the CircleCI All Pipelines dashboard. A sidebar on the left includes links for Dashboard, Projects, Insights, Organization Settings, and Plan. A message box in the center says "CI behind your firewall just got easier" and "Install a more scalable, Kubernetes-friendly self-hosted runner in 5". The main area displays a table of pipelines. One pipeline, "udapeople.CICD 10", is shown with a status of "Success", workflow "default", branch "master", and commit "641a10a workaround scan stage". The pipeline was started 2m ago and took 1m 34s. The table lists six jobs: build-frontend, scan-frontend, test-frontend, build-backend, scan-backend, and test-backend, all of which completed successfully.

- Remember to take screen in alarm notification form your email error



- In this like <https://api.slack.com/>
- we will create new app and with our project and change scope setting with this

## Scopes

A Slack app's capabilities and permissions are governed by the `scopes` it requests.

### Bot Token Scopes

Scopes that govern what your app can access.

OAuth Scope	Description	⋮
chat:write	Send messages as @Circle-CI	⋮
chat:write.public	Send messages to channels @Circle-CI isn't a member of	⋮
files:write	Upload, edit, and delete files as Circle-CI	⋮

[Add an OAuth Scope](#)

- Now we will do two environment variables with our slack project and we create channel with name for our project

The screenshot shows the 'Environment Variables' section of the CircleCI Project Settings. On the left sidebar, 'Environment Variables' is selected. The main area displays two environment variables:

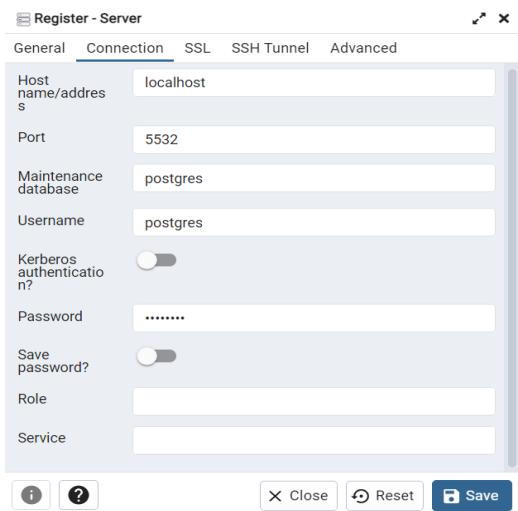
Name	Value	Action
MY_SLACK_ACSESS_TOKEN	xxxxwefC	X
MY_SLACK_DEFULL_CHANNEL	xxxxdone	X

Buttons for 'Add Environment Variable' and 'Import Variables' are visible at the top right of the table.

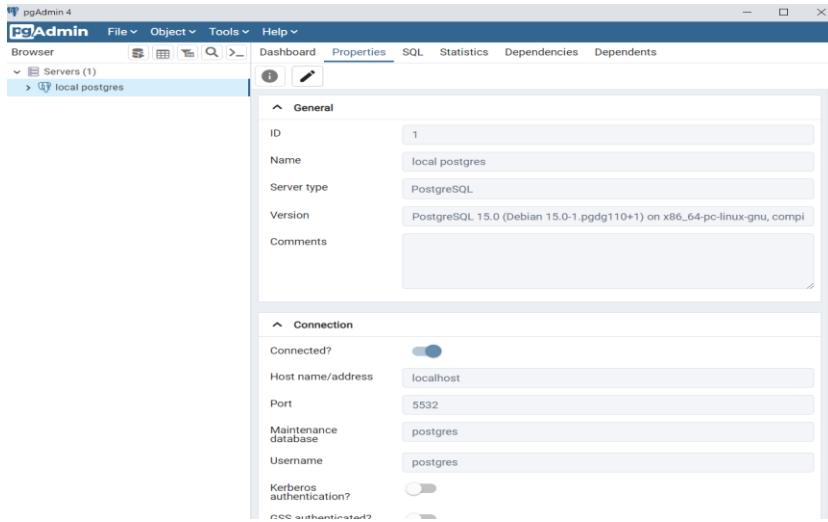
- git add .
- git commit -m "added slack notification orb"
- git push
- link will guide you when you do connect notification <https://github.com/CircleCI-Public/slack-orb/wiki/Setup>
- now we will run command #docker compose up

```
longest=0.560 s, average=0.005 s; distance=4217 kB, estimate=4217 kB
util-database-1 | 2022-10-16 21:00:41.758 UTC [49] LOG:  database system is shut down
util-database-1 | done
util-database-1 | server stopped
util-database-1 | PostgreSQL init process complete; ready for start up.
util-database-1 | 2022-10-16 21:00:41.984 UTC [1] LOG:  starting PostgreSQL 15.0 (Debian 15.0-1
.util-database-1 | .pgdg110+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 10.2.1-6) 10.2.1 20210110, 64-bit
util-database-1 | 2022-10-16 21:00:41.985 UTC [1] LOG:  listening on IPv4 address "0.0.0.0", po
rt 5432
util-database-1 | 2022-10-16 21:00:41.985 UTC [1] LOG:  listening on IPv6 address ":::", port 54
32
util-database-1 | 2022-10-16 21:00:42.101 UTC [1] LOG:  listening on Unix socket "/var/run/post
gresql/.s.PGSQL.5432"
util-database-1 | 2022-10-16 21:00:42.253 UTC [64] LOG:  database system was shut down at 2022-
10-16 21:00:41 UTC
util-database-1 | 2022-10-16 21:00:42.320 UTC [1] LOG:  database system is ready to accept conn
ections
```

- Now our database so we will install PostgreSQL by this link <https://www.postgresql.org/download/>  
By default port 5432 and in our project we will sit port 5532 in configuration



- Now we create database and run it in port 5532



- Now make sure that you use # node -version v13.8.0
- Now we will run command for backend and front end # npm install
  - backend

```
> nodemon@1.19.4 postinstall E:\CI-CD\backend\node_modules\nodemon
> node bin/postinstall || exit 0

Love nodemon? You can now support the project via the open
collective:
  > https://opencollective.com/nodemon/donate

npm [WARN] glee2@1.0.0 No repository field.
npm [WARN] optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node_modules\fsevents):
npm [WARN] notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.12
: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

added 1311 packages from 1064 contributors and audited 1389 packages in 337.762s
23 packages are looking for funding
  run `npm fund` for details

found 641 vulnerabilities (2 low, 144 moderate, 370 high, 125 critical)
  run `npm audit fix` to fix them, or `npm audit` for details
```

- frontend

```
npm [WARN] optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules\fsevents)
:
npm [WARN] notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9:
  wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm [WARN] optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules\chokidar\node_modules\fsevents):
npm [WARN] notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9:
  wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

added 1755 packages from 1104 contributors and audited 1895 packages in 486.775s
found 664 vulnerabilities (5 low, 184 moderate, 338 high, 137 critical)
  run `npm audit fix` to fix them, or `npm audit` for details
```

DELL@DESKTOP-SEA7AN9 MINGW64 /e/CI-CD/frontend (master)

- now we will site environment variables
- create file .env in backend and cope data from development.env to .env

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under ".CIRCLECI (WORKSPACE)" with files like ".circleci", "backend", "node\_modules", "src", "test", ".env", ".env.sample", ".nestcli.json", ".npmignore", ".prettierrc", and "development.env".
- Code Editor:** Displays the content of the ".env" file, which is a copy of "development.env". The "development.env" file contains environment variable definitions for a PostgreSQL database connection.
- Bottom Status Bar:** Shows "packages in 486.775s".
- Terminal:** Labeled "bash" at the bottom right.

```

1 NODE_ENV=local
2 VERSION=1
3 TYPEORM_CONNECTION=postgres
4 TYPEORM_MIGRATIONS_DIR=./src/migrations
5 TYPEORM_ENTITIES=./src/modules/domain/**/*.entity.ts
6 TYPEORM_MIGRATIONS=./src/migrations/*.ts
7
8 # Things you can change if you wish...
9 TYPEORM_HOST=localhost
10 TYPEORM_PORT=5532
11 TYPEORM_USERNAME=postgres
12 TYPEORM_PASSWORD=password
13 TYPEORM_DATABASE=glee|

```

- Same we will do that in frontend

npm run migrations-win for backend

```

DELL@DESKTOP-SEA7AN9 MINGW64 /e/CI-CD/backend (master)
$ npm run migrations-win

> glee2@1.0.0 migrations-win E:\CI-CD\backend
> ts-node node_modules/typeorm/cli.js migration:run

query: SELECT * FROM "information_schema"."tables" WHERE "table_schema" = current_
query: CREATE TABLE "migrations" ("id" SERIAL NOT NULL, "timestamp" bigint NOT NU
b734260ea46be" PRIMARY KEY ("id"))
query: SELECT * FROM "migrations" "migrations" ORDER BY "id" DESC
0 migrations are already loaded in the database.
3 migrations were found in the source code.
3 migrations are new migrations that needs to be executed.
query: START TRANSACTION
query: CREATE TABLE "product" ("id" SERIAL NOT NULL, "description" character varyi
Y ("id"))
query: CREATE TABLE "order" ("id" uuid NOT NULL, CONSTRAINT "PK_1031171c1313010249
query: CREATE TABLE "order_products_product" ("orderId" uuid NOT NULL, "productId"
ARY KEY ("orderId", "productId"))
query: ALTER TABLE "order_products_product" ADD CONSTRAINT "FK_1f9ea0b0e59e0d98ade
E CASCADE
query: ALTER TABLE "order_products_product" ADD CONSTRAINT "FK_d6c66c08b9c7e84a1b6
query: ALTER TABLE "order_products_product" ADD CONSTRAINT "PK_59f5d41216418eba313ed3c7d7c" PRIMARY KEY ("orderId", "productId")
query: ALTER TABLE "order_products_product" ADD CONSTRAINT "FK_d6c66c08b9c7e84a1b657797dff" FOREIGN KEY ("productId") REFERENCES "product"( "id") ON DELETE CASCADE
query: INSERT INTO "migrations"("timestamp", "name") VALUES ($1, $2) -- PARAMETERS: [1549398619849, "FixProductIdTable1549398619849"]
Migration FixProductIdTable1549398619849 has been executed successfully.

query: CREATE TABLE "employee" ("id" SERIAL NOT NULL, "firstName" character varying(100) NOT NULL, "middleName" character varying(100), "lastName" character varying(100) NOT NULL, "secondLastName" character varying(100), "displayName" character varying(100), "companyEmail" character varying(50) NOT NULL DEFAULT '', "personalEmail" character varying(100), "DEFAULT '', "birthdate" TIMESTAMP, "startDate" TIMESTAMP NOT NULL, "address" character varying(200), "phoneNumber" character varying(100), "bankName" character varying(100), "ccountNumber" character varying(100), "gender" character varying, "tags" json NOT NULL DEFAULT '{}', "country" character varying(100) NOT NULL, "region" character varying(100) NOT NULL, "city" character varying(100) NOT NULL, "effectiveDate" TIMESTAMP NOT NULL, "salary" numeric NOT NULL, "salaryType" character varying NOT NULL, "isActive" boolean NOT NULL, "workingHoursPerWeek" integer NOT NULL DEFAULT 40, CONSTRAINT "PK_3c2bc72f03fd5abb5ac169498" PRIMARY KEY ("id"))
query: CREATE INDEX "IDX_1f9ea0b0e59e0d98adef2d5e97" ON "order_products_product" ("orderId")
query: CREATE INDEX "IDX_dcc66c08b9c7e84a1b657797dff" ON "order_products_product" ("productId")
query: INSERT INTO "migrations"("timestamp", "name") VALUES ($1, $2) -- PARAMETERS: [1555722583168, "AddEmployee1555722583168"]
Migration AddEmployee1555722583168 has been executed successfully.
query: COMMIT

```

➤ Our data in pgadmin

The screenshot shows the pgAdmin interface. On the left, the 'Browser' pane lists various database objects under 'Schemas (1) / public'. A green arrow points from the 'Tables (5)' section in the browser to the main content area. The main area displays a table with five rows, each representing a table: employee, migrations, order, order\_products\_product, and product. The table has columns for Name, Owner, Partitioned table?, and Comment.

Name	Owner	Partitioned table?	Comment
employee	postgres	Off	
migrations	postgres	Off	
order	postgres	Off	
order_products_product	postgres	Off	
product	postgres	Off	

➤ Run command # npm start for backend

```
debug: Nest application successfully started
debug: Listening on port 3030.
```

Run now your <http://localhost:3030/api/status>

The screenshot shows a web browser window with the URL 'localhost:3030/api/status' in the address bar. The page content is a JSON object with three key-value pairs: 'status: "ok"', 'version: "1.0.0"', and 'environment: "local"'. The browser interface includes standard navigation buttons, a search bar, and a tab bar with other bookmarks.

```
status: "ok"
version: "1.0.0"
environment: "local"
```

- Run command # npm start for frontend

<http://localhost:3000/api/status>

- Now we will run our production
- \$ npm run build

```
DELL@DESKTOP-SEA7AN9 MINGW64 /e/CI-CD (master)
$ cd backend

DELL@DESKTOP-SEA7AN9 MINGW64 /e/CI-CD/backend (master)
$ npm run build

> glee2@1.0.0 build E:\CI-CD\backend
> tsc

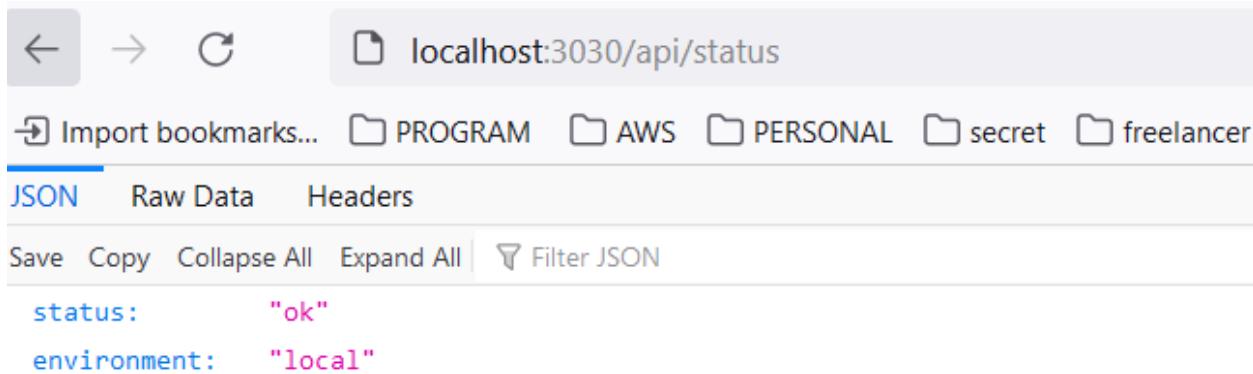
DELL@DESKTOP-SEA7AN9 MINGW64 /e/CI-CD/backend (master)
$ 
```

- We will run this commeds

```
export NODE_ENV=production
export VERSION=1
export TYPEORM_CONNECTION=postgres
export TYPEORM_MIGRATIONS_DIR=./migration
export TYPEORM_ENTITIES=./modules/domain/**/*.{entity{.ts,.js}}
export TYPEORM_MIGRATIONS=./migrations/*{.ts,.js}
export TYPEORM_HOST=localhost
export TYPEORM_PORT=5532
export TYPEORM_USERNAME=postgres
export TYPEORM_PASSWORD=password
export TYPEORM_DATABASE=glee
```

- cd dist

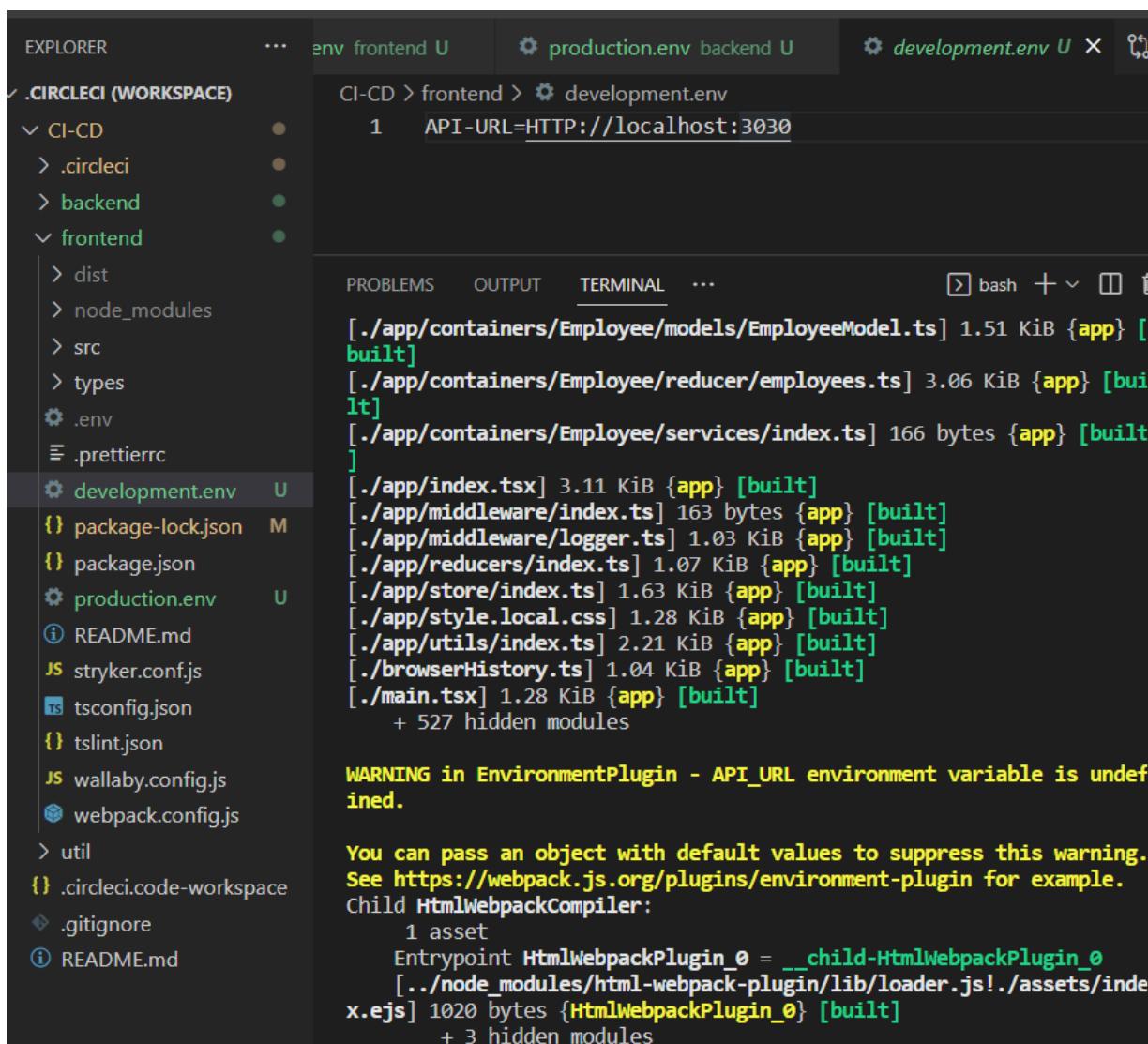
➤ node main.js



```
localhost:3030/api/status

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

➤ npm run build for fronten



```
API-URL=HTTP://localhost:3030

[ ./app/containers/Employee/models/EmployeeModel.ts ] 1.51 KiB {app} [built]
[ ./app/containers/Employee/reducer/employees.ts ] 3.06 KiB {app} [built]
[ ./app/containers/Employee/services/index.ts ] 166 bytes {app} [built]
[ ./app/index.tsx ] 3.11 KiB {app} [built]
[ ./app/middleware/index.ts ] 163 bytes {app} [built]
[ ./app/middleware/logger.ts ] 1.03 KiB {app} [built]
[ ./app/reducers/index.ts ] 1.07 KiB {app} [built]
[ ./app/store/index.ts ] 1.63 KiB {app} [built]
[ ./app/style.local.css ] 1.28 KiB {app} [built]
[ ./app/utils/index.ts ] 2.21 KiB {app} [built]
[ ./browserHistory.ts ] 1.04 KiB {app} [built]
[ ./main.tsx ] 1.28 KiB {app} [built]
+ 527 hidden modules

WARNING in EnvironmentPlugin - API_URL environment variable is undefined.

You can pass an object with default values to suppress this warning.
See https://webpack.js.org/plugins/environment-plugin for example.
Child HtmlWebpackPlugin:
    1 asset
      Entrypoint HtmlWebpackPlugin_0 = __child-HtmlWebpackPlugin_0
        [ ./node_modules/html-webpack-plugin/lib/loader.js!./assets/index.ejs ] 1020 bytes {HtmlWebpackPlugin_0} [built]
        + 3 hidden modules
```

- we create here s3 bucket and we create before key pair

The screenshot shows the AWS S3 Buckets page. At the top, there's an 'Account snapshot' section with a link to 'View Storage Lens dashboard'. Below it, a table lists a single bucket:

Name	AWS Region	Access	Creation date
udapeople-my	US East (N. Virginia) us-east-1	Objects can be public	October 18, 2022, 15:40:17 (UTC+02:00)

- We create database postgres

The screenshot shows the AWS RDS Databases page. At the top, there's a 'Databases' section with a 'Group resources' button and a 'Create database' button. Below it, a table lists a single database instance:

DB identifier	Role	Engine	Region & AZ	Size	Status	CPU	Current activity	Maintenance	VPC
database-1	Instance	PostgreSQL	us-east-1c	db.t3.micro	Available	16.30%	0.00 sessions	none	vpc-0da

➤ Now we will create our environment

**Add Environment Variable**

Enter a name starting with a letter or \_ with no spaces or special characters. Then enter a value consisting of valid POSIX characters. Note \$ must be escaped by \. Example: usd\\$

Name\*

Value\*

**Add Environment Variable**

Enter a name starting with a letter or \_ with no spaces or special characters. Then enter a value consisting of valid POSIX characters. Note \$ must be escaped by \. Example: usd\\$

Name\*

Value\*

**Add Environment Variable**

Enter a name starting with a letter or \_ with no spaces or special characters. Then enter a value consisting of valid POSIX characters. Note \$ must be escaped by \. Example: usd\\$

Name\*

Value\*

**Add Environment Variable**

Enter a name starting with a letter or \_ with no spaces or special characters. Then enter a value consisting of valid POSIX characters. Note \$ must be escaped by \. Example: usd\\$

Name\*

Value\*

**Add Environment Variable**

Enter a name starting with a letter or \_ with no spaces or special characters. Then enter a value consisting of valid POSIX characters. Note \$ must be escaped by \. Example: usd\\$

Name\*

Value\*

**Add Environment Variable**

Enter a name starting with a letter or \_ with no spaces or special characters. Then enter a value consisting of valid POSIX characters. Note \$ must be escaped by \. Example: usd\\$

Name\*

Value\*

**Add Environment Variable Value**

Enter a name starting with a letter or \_ with no spaces or special characters. Then enter a value consisting of valid POSIX characters. Note \$ must be escaped by \. Example: usd\$

Name\*

Value\*

**Add Environment Variable**

Enter a name starting with a letter or \_ with no spaces or special characters. Then enter a value consisting of valid POSIX characters. Note \$ must be escaped by \. Example: usd\$

Name\*

Value\*

**Add Environment Variable**

Enter a name starting with a letter or \_ with no spaces or special characters. Then enter a value consisting of valid POSIX characters. Note \$ must be escaped by \. Example: usd\$

Name\*

Value\*

**Add Environment Variable**

Enter a name starting with a letter or \_ with no spaces or special characters. Then enter a value consisting of valid POSIX characters. Note \$ must be escaped by \. Example: usd\$

Name\*

Value\*

Now our environment ready to show every thing in cloud

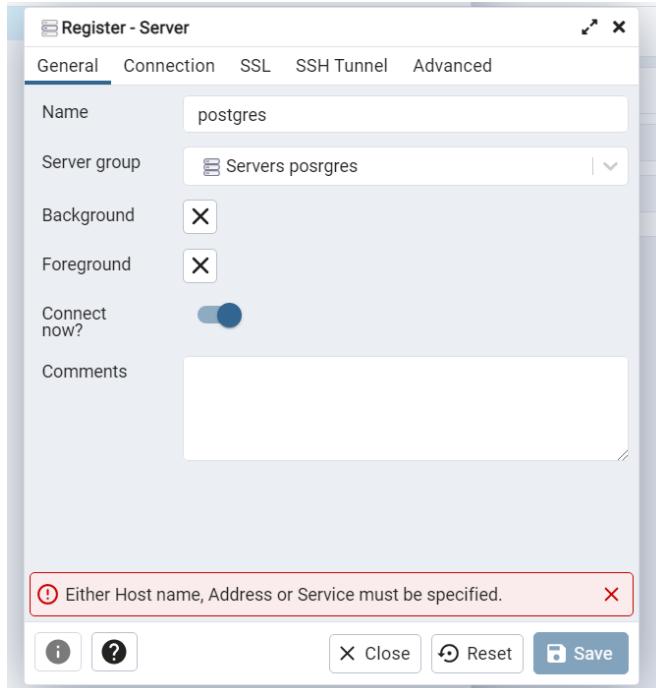
Name	Value	Add Environment Variable	Import Variables
AWS_ACCESS_KEY_ID	xxxxCQXV	x	
AWS_DEFAULT_REGION	xxxxst-1	x	
AWS_SECRET_ACCESS_KEY	xxxxJwQx	x	
MY_SLACK_ACSESS_TOKEN	xxxxweFC	x	
SLACK_DEFAULT_CHANNEL	xxxxdone	x	
TYPEORM_CONNECTION	xxxxgres	x	
TYPEORM_DATABASE	xxxxgres	x	
TYPEORM_ENTITIES	xxxxy.ts	x	
TYPEORM_HOST	xxxx.com	x	
TYPEORM.Migrations	xxxx*.ts	x	
TYPEORM.Migrations_DIR	xxxxions	x	
TYPEORM_PASSWORD	xxxxgres	x	
TYPEORM_PORT	xxxx32	x	
TYPEORM_USERNAME	xxxxgres	x	

- We need to run database endpoint in curl to make sure that we can reach it

```
DELL@DESKTOP-SEA7AN9 MINGW64 ~ (master)
$ curl Endpoint
% Total    % Received % Xferd  Average Speed   Time   Time     Time  Current
          Dload  Upload Total   Spent   Left Speed
  0      0     0      0       0      0      0 --:--:--  0:00:01 --:--:-- 0cu
curl: (6) Could not resolve host: Endpoint

DELL@DESKTOP-SEA7AN9 MINGW64 ~ (master)
```

When we will start to reach database throw pg admin it will give us this error so we need to open security group



➤ So we go to security group to open it to make sure that we are in port 5432

The screenshot shows the AWS Management Console with the EC2 service selected. The left sidebar shows various AWS services like Reserved Instances, Images, and Network & Security. Under Network & Security, 'Security Groups' is selected. The main pane displays a table titled 'Security Groups (1/2) Info' with two rows:

Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count	Outbound rules count
<input checked="" type="checkbox"/> -	sg-0a6249c46ce21015e	postgress	vpc-0da5aef03a77ef47	Created by RDS manag...	957923177470	1 Permission entry	1 Permission entry
<input type="checkbox"/>	sg-0b615d2bd5ac8c46	default	vpc-0da5aef03a77ef47	default VPC security gr...	957923177470	1 Permission entry	1 Permission entry

A dropdown menu for the selected 'postgress' group is open, showing options: 'Edit security group', 'Actions', 'Export security groups to CSV', and 'Create security group'.

The screenshot shows the details page for the 'postgress' security group. The top navigation bar includes 'EC2 > Security Groups > sg-0a6249c46ce21015e - postgress'. The left sidebar is identical to the previous screenshot. The main content area shows the 'Details' section with fields:

Security group name <input checked="" type="checkbox"/> postgress	Security group ID <input checked="" type="checkbox"/> sg-0a6249c46ce21015e	Description <input checked="" type="checkbox"/> Created by RDS management console	VPC ID <input checked="" type="checkbox"/> vpc-0da5aef03a77ef47
Owner <input checked="" type="checkbox"/> 957923177470	Inbound rules count <input checked="" type="checkbox"/> 1 Permission entry	Outbound rules count <input checked="" type="checkbox"/> 1 Permission entry	

Below this are tabs for 'Inbound rules', 'Outbound rules', and 'Tags'. A green arrow points down to the 'Inbound rules' table, which lists one rule:

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input checked="" type="checkbox"/> -	sgr-0d02eefc553b23c7b	IPv4	PostgreSQL	TCP	5432	154.176.159.4/32	-

A message at the bottom says: 'You can now check network connectivity with Reachability Analyzer' with a 'Run Reachability Analyzer' button.

Now we will give it ip 0.0.0.0/0 to get any request

Screenshot of the AWS Security Groups 'Edit inbound rules' page. The page shows a single rule for a PostgreSQL instance. The rule details are:

- Security group rule ID:** sgr-0d02eefc553b23c7b
- Type:** PostgreSQL
- Protocol:** TCP
- Port range:** 5432
- Source:** Custom, 154.176.159.4/32
- Description - optional:** (empty)

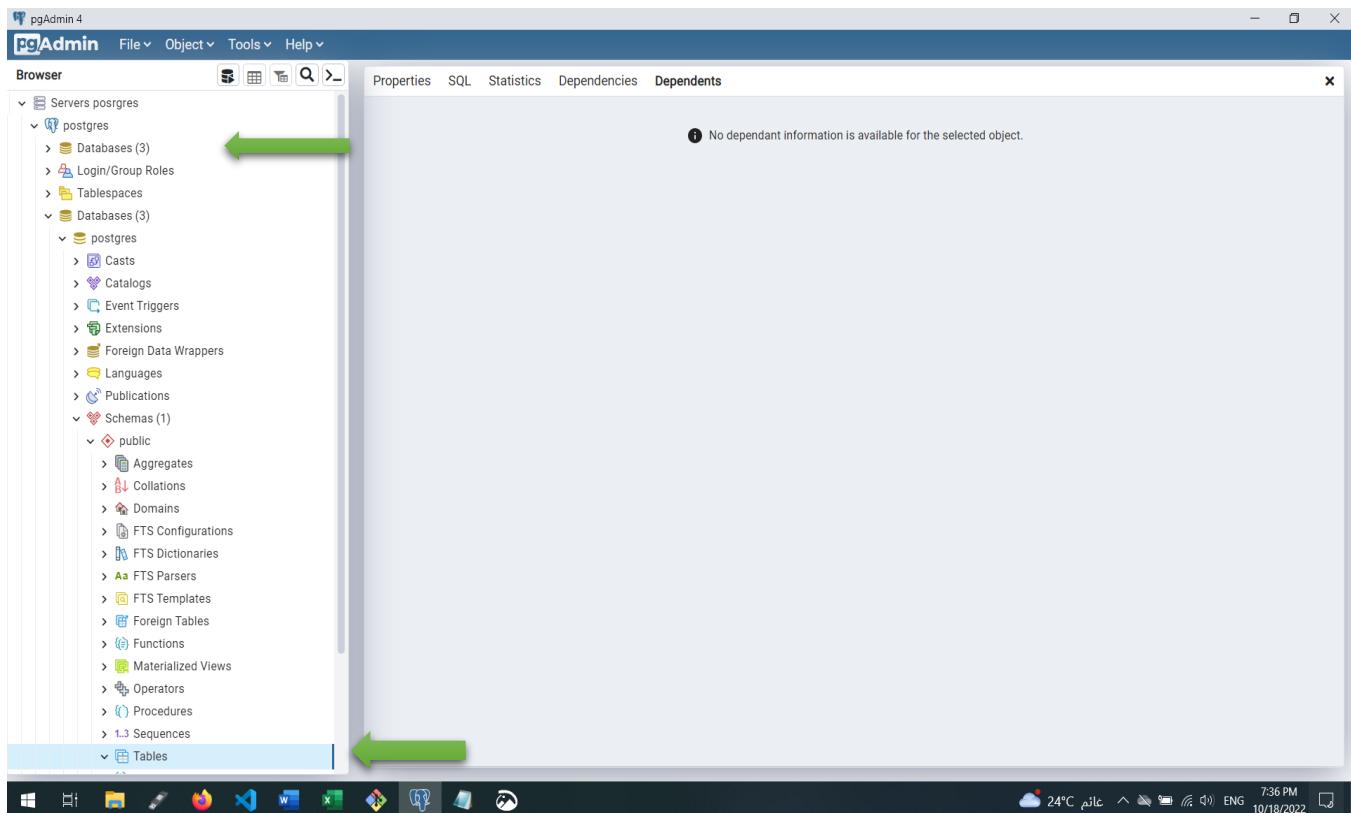
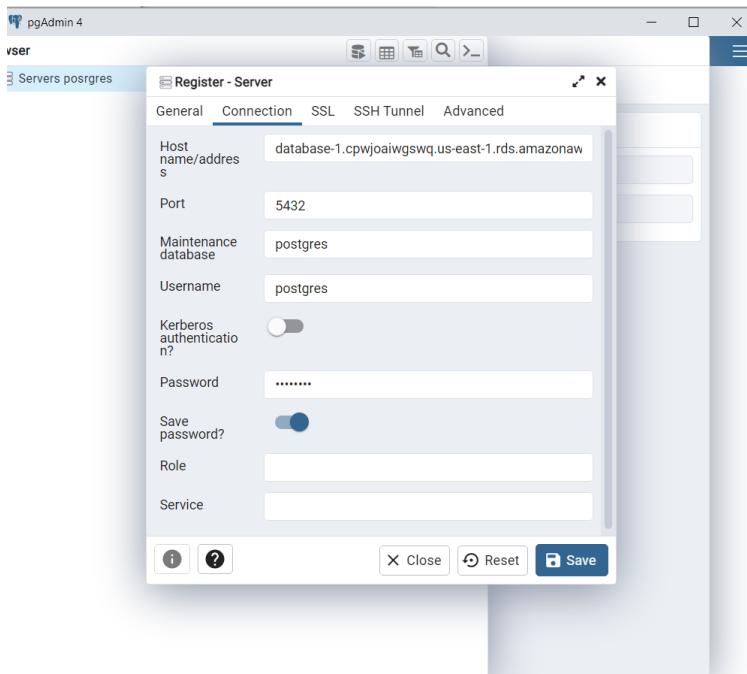
Buttons at the bottom include 'Add rule', 'Cancel', 'Preview changes', and 'Save rules'.

Screenshot of the AWS Security Groups 'Edit inbound rules' page. The page shows a single rule for a PostgreSQL instance. The rule details are:

- Security group rule ID:** sgr-0d02eefc553b23c7b
- Type:** PostgreSQL
- Protocol:** TCP
- Port range:** 5432
- Source:** Anywhere-IPv4
- Description - optional:** (empty)

A green arrow points down to the 'Source' dropdown field, which now contains '0.0.0.0/0'. Buttons at the bottom include 'Add rule', 'Cancel', 'Preview changes', and 'Save rules'.

➤ Now give pgadmin all date need and save it



## ➤ Let's show our database

The screenshot shows the AWS RDS console for a PostgreSQL database. The left sidebar lists various options like Databases, Query Editor, and Snapshots. The main panel displays connectivity details:

Endpoint & port	Networking	Security
Endpoint database-1.cpwjoaiwgswq.us-east-1.rds.amazonaws.com	Availability Zone us-east-1c	VPC security groups postgress (sg-0a6249c46ce21015e) Active
Port 5432	Subnet group default-vpc-0da5aefa03a77ef47	Publicly accessible Yes
	Subnets subnet-0d387d868a2e6e	Certificate authority rds-ca-2019
		Certificate authority date

The screenshot shows the AWS Security Groups console for a specific security group. The path is EC2 > Security Groups > sg-0a6249c46ce21015e - postgres > Edit inbound rules.

**Edit inbound rules**

Inbound rules control the incoming traffic that's allowed to reach the instance.

**Inbound rules**

Inbound rule 1	Delete	
Security group rule ID sgr-0d02eefc553b23c7b	Type Info PostgreSQL	Protocol Info TCP
Port range Info 5432	Source type Info Custom	Source 0.0.0.0/0 X
Description - optional	Info	
<input type="button" value="Add rule"/>		

Cancel

- We will create keypair and get the code in ssh in CICD and we don't need to get any name for it



**Key pairs (2) [Info](#)**  [Actions ▾](#) [Create key](#)

<input type="checkbox"/>	Name	Type	Created
<input type="checkbox"/>	udacity	rsa	2022/10/18 19:34 GMT+2

**Project Settings** [udapeople.CiCD](#)

Slack Integration	<p> A deploy key is a repo-specific SSH key. GitHub has the public key, and we store the private key. The deployment key gives CircleCI access to a single repository. If you want to push to your repository from builds, please add a user key as described below or manually add a <a href="#">read-write deployment key</a>.</p>
Insights Snapshot Badge	
Status Badges	
Webhooks	<p><b>User Key</b> A user key is a user-specific SSH key. GitHub has the public key, and we store the private key. Possession of the private key gives the ability to act as that user, for purposes of 'git' access to projects.</p> <p><a href="#">Authorize with GitHub</a></p> <hr/> <p> If a deploy key can't access all of your project's private dependencies, we can configure it to use an SSH key with the same level of access to GitHub repositories that you have.</p> <p>In order to do so, you'll need to grant authorization from GitHub to the "admin:public_key" scope. This will allow us to add a new authorized public key to your GitHub account.</p>

## Additional SSH Keys

Add keys to the build VMs that you need to deploy to your machines. [If the hostname field is blank, the key will be used for all hosts.](#)

Hostname	Fingerprint	<a href="#">Add SSH Key</a>
[REDACTED]		<a href="#"></a>

The screenshot shows the GitHub 'SSH keys' settings page for user 'Aya Rabih'. On the left, there's a sidebar with account management options like 'Public profile', 'Account', 'Appearance', 'Accessibility', 'Notifications', 'Access', 'Billing and plans', 'Emails', 'Password and authentication', 'SSH and GPG keys' (which is selected), 'Organizations', and 'Moderation'. The main area displays an 'SSH keys' section with a 'New SSH key' button. It lists one key: 'SSH Key' (SSH), added on Oct 18, 2022, which is 'Never used — Read/write'. Below this, there's a note about generating SSH keys or troubleshooting common SSH problems.

➤ We should run this code before run deploy

```
DELL@DESKTOP-SEA7AN9 MINGW64 /e/CI-CD (master)
$ cd .circleci/files

DELL@DESKTOP-SEA7AN9 MINGW64 /e/CI-CD/.circleci/files (master)
$ aws cloudformation deploy \
> --template-file cloudfront.yaml \
> --stack-name InitialStack \
> --parameter-overrides WorkflowID=udapeople-kk1j287dhjppmz437

Waiting for changeset to be created..
Waiting for stack create/update to complete
Successfully created/updated stack - InitialStack

DELL@DESKTOP-SEA7AN9 MINGW64 /e/CI-CD/.circleci/files (master)
$ [ ]
```

➤ Run deploy-infrastructure with ssh

The screenshot shows the CircleCI 'All Pipelines' dashboard for user 'ayarabih'. The sidebar includes 'Dashboard', 'Projects', 'Insights', 'Organization Settings', 'Plan', and a message about CI being behind a firewall. The main area shows the 'All Pipelines' section with a table. One pipeline, 'udapeople.CI\_CD', is listed with a status of 'Success' and a workflow named 'default'. It was started 17m ago and completed 12m 25s ago with a duration of 600%. The table details the jobs: build-frontend (106), scan-frontend (109), test-frontend (110), build-backend (105), scan-backend (107), test-backend (108), and deploy-infrastructure (113), all of which were successful.

Pipeline	Status	Workflow	Branch / Commit	Start	Duration	Actions
udapeople.CI_CD 53	Success	default	master b3401fc workaround scan stage	17m ago	12m 25s ↑ 600%	
	Jobs					
		build-frontend 106			42s	
		scan-frontend 109			53s	
		test-frontend 110			1m 19s	
		build-backend 105			28s	
		scan-backend 107			59s	
		test-backend 108			57s	
		deploy-infrastructure 113			12m 23s	

A screenshot of the CircleCI web interface. On the left, there's a sidebar with user information (ayarabih, Aya Rabih), navigation links (Dashboard, Projects, Insights, Organization Settings, Plan), and a note about CI behind firewalls. The main area shows a pipeline step titled "Parallel runs" with a duration of 0:38. It lists several steps: "Spin up environment" (1s), "Preparing environment variables" (0s), "Checkout code" (0s), and "Ensure back-end infrastructure exists" (36s). The last step shows a terminal session with AWS CloudFormation deployment commands:

```

1 #!/bin/bash -eo pipefail
2 aws cloudformation deploy \
3   --template-file ./circleci/files/backend.yml \
4   --stack-name "udapeople-backend-8$(CIRCLE_WORKFLOW_ID:0:7)" \
5   --parameter-overrides ID="$(CIRCLE_WORKFLOW_ID:0:7)" \
6   --tags 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-fd01635
14
15 Exited with code exit status 255
16 CircleCI received exit code 255
17

```

The system status bar at the bottom indicates it's 24°C, 5:12 PM, 10/19/2022, and the battery level is at 100%.

A screenshot of the AWS CloudFormation console. The left sidebar includes links for AWS Services, EC2, S3, IAM, RDS, CloudFormation, CloudFront, and VPC. The main content shows a table of stacks:

Stack name	Status	Created time	Description
udapeople-frontend-0f04eb4	CREATE_COMPLETE	2022-10-20 01:32:42 UTC+0200	UdaPeople frontend stack.
udapeople-backend-0f04eb4	CREATE_COMPLETE	2022-10-20 01:31:30 UTC+0200	UdaPeople backend stack.
udapeople-frontend-12b07db	CREATE_COMPLETE	2022-10-20 01:22:30 UTC+0200	UdaPeople frontend stack.
udapeople-backend-12b07db	CREATE_COMPLETE	2022-10-20 01:21:23 UTC+0200	UdaPeople backend stack.
InitialStack	CREATE_COMPLETE	2022-10-19 23:57:30 UTC+0200	Cloudfront distribution for UdaPeople.

A screenshot of the AWS EC2 Instances page. The left sidebar includes links for New EC2 Experience, EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (3), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, and Elastic Block Store. The main content shows a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP
backend-	i-05ea455340647de50	Terminated	t2.micro	-	No alarms	us-east-1b	-	-	-
backend-12b07db	i-0b28692f779adba417	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-54-226-59-148.co...	54.226.59.148	-
backend-0f04eb4	i-0ff351f1dd58b3f2	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-54-86-67-160.com...	54.86.67.160	-

➤ We need to get credential for code from this link curl -d "email=your email" https://kvdb.io

```

config.yml - CI-CD - Visual Studio Code
config.yml
  config.yml
    .circleci > ! config.yml
      .circleci
        286   # - install_awscli
        287   - run:
          288     name: Run migrations
          289     command: |
          290       cd backend
          291       npm install
          292       npm run migrations > migrations_dump.txt
          293   - run:
            294     name: Send migration status to kvdb.io
            295     command: |
            296       if grep -q "has been executed successfully." ~/project/backend/migrations_dump.txt
            297         then
            298           | curl https://kvdb.io/${KVDB_BUCKET}/migration_${CIRCLE_WORKFLOW_ID:0:7} -d '1'
            299         fi
            300       - destroy-environment
            301       - revert-migrations
    config.yml
      backend
        > dist
        > node_modules
        > src
        > test
        .env.sample
        nestcli.json
        .npmignore
        .prettierrc
        development.env
        nodemon-debug.json
        package-lock.json
        package.json
        README.md
    OUTLINE
    TIMELINE
  PROBLEMS OUTPUT TERMINAL
  bash bash powershell
  On branch master
  Your branch is up to date with 'origin/master'.
  nothing to commit, working tree clean
  DELL@DESKTOP-SEA7AN9 MINGW64 /e/CI-CD/.circleci (master)
  $ git push
  Everything up-to-date
  DELL@DESKTOP-SEA7AN9 MINGW64 /e/CI-CD/.circleci (master)
  $ curl -d "email=ayarabih2@gmail.com" https://kvdb.io
  GET /xvYmjaG55svvDQ
  DELL@DESKTOP-SEA7AN9 MINGW64 /e/CI-CD/.circleci (master)
  $ []

```

The screenshot shows the Visual Studio Code interface with an orange border. The left sidebar displays a file tree for a 'CI-CD' project, including files like 'myenv.sh', 'config.yml', 'main.yml', 'readme.md', 'deploy-backend.yml', and 'inventory.txt'. The main editor area shows the contents of 'config.yml'. The terminal tab is active, displaying the output of a CircleCI workflow run. The terminal output includes:

```

$ myenv.sh
$ config.yml
...
286   # - install_awscli
287   - run:
288     name: Run migrations
289     command: |
290       cd backend
291       npm install
292       npm run migrations > migrations_dump.txt
293   - run:
294     name: Send migration status to kvdb.io
295     command: |
296       if grep -q "has been executed successfully." ~/project/backend/migrations_dump.txt
297       then
298         curl https://kvdb.io/GFm9xtXVYmmja65svvrDPQ/migration_${CIRCLE_WORKFLOW_ID:0:7} -d '1'
299       fi
300   - destroy-environment
301   - revert-migrations

```

The terminal also shows a successful git push and a curl command to kvdb.io. The status bar at the bottom indicates the terminal has 22 selected lines, the date is 10/20/2022, and the time is 8:11 PM.

- When we run we find error so we start to make sure that our code it okay

The screenshot shows the CircleCI web interface. The left sidebar has a dark theme with sections for 'Dashboard', 'Projects', 'Insights', 'Organization Settings', and 'Plan'. A message box on the left says 'CI behind your firewall? just get easier! Install a more scalable, Kubernetes-friendly self-hosted runner in 5 minutes or less.' Below the sidebar are links for 'Getting Started', 'Notifications', 'Status (OPERATIONAL)', 'Docs', 'Orbs', and 'Support'.

The main area is titled 'All Pipelines' and lists four pipeline runs for 'udapeople.CI\_CD':

- Pipeline 130: Failed, default branch, master, b5d7e05, 8m ago, duration 6m 36s. Jobs: build-frontend, scan-frontend, test-frontend, build-backend, scan-backend, test-backend, deploy-infrastructure, configure-infrastructure, run-migrations.
- Pipeline 129: Failed, default branch, master, b5d7e05, 8m ago, duration 6m 36s. Error: Error calling workflow: 'default'.
- Pipeline 128: Failed, default branch, master, b5d7e05, 8m ago, duration 6m 36s. Error: Error calling workflow: 'default'.
- Pipeline 127: Failed, default branch, master, b5d7e05, 8m ago, duration 6m 36s. Error: Error calling workflow: 'default'.

A search bar at the bottom left contains 'workflow:default'. The status bar at the bottom right shows the date 10/20/2022, time 10:35 PM, and weather 21°C.

The screenshot shows a CircleCI pipeline interface. On the left, there's a sidebar with user information (ayarabih, Aya Rabih) and navigation links for Dashboard, Projects, Insights, Organization Settings, Plan, and Support. A message in the Support section says "CI behind your firewall just got easier". Below the sidebar is a "Getting Started" section with links for Notifications, Status (OPERATIONAL), Docs, Orbs, and Support.

The main area displays a pipeline step titled "Install Ansible" which has failed. The log output shows several errors related to package installations and dependencies. The status bar at the bottom indicates "Workflow: 'default'" and shows 12 of 16 matches found in the logs.

```

0% [0 Packages stored 0 B] [0 Packages] 0B/0B 2410s [0 Packages stored 0 B] [14 Packages] 570 kB/1,222 kB 47%
0% [9 Packages stored 0 B] [14 Packages] 570 kB/1,222 kB 47% 0% [14 Packages] 854 kB/1,222 kB 70%
Get:16 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [27.4 kB]
Get:17 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [55.2 kB]
96% [10 Packages stored 0 B] 96% [Connecting to ppa.launchpad.net (195.125.190.52)] 96% [10 Packages sto
96% [Working] 96% [14 Packages stored 0 B] 96% [Working] 96% [15 Packages
Reading package lists... Done
Building dependency tree
Reading state information... Done
Reading package lists... Done
0 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree
Reading state information... Done
Reading state information... Done
software-properties-common is already the newest version (0.99.9.8).
0 upgraded, 0 newly installed, 0 to remove and 61 not upgraded.
Hit:1 https://download.docker.com/linux/ubuntu focal InRelease
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:3 http://archive.ubuntu.com/ubuntu focal InRelease
Get:4 http://ppa.launchpad.net/ansible/ubuntu/focal/main amd64 Packages [18.0 kB]
Hit:5 http://archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:6 http://archive.ubuntu.com/ubuntu focal-security InRelease
Hit:7 http://ppa.launchpad.net/gilt-core/ppa/ubuntu focal InRelease
Get:8 http://ppa.launchpad.net/ansible/ubuntu/focal/main amd64 Packages [1,086 B]
Fetched 19.1 kB in (27.3 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package ansible
49
50  Exited with code exit status 100
51  CircleCI received exit code 100

```

At the bottom, there are buttons for "Destroy environments" and a status bar showing "2s" and various system icons.

➤ So we fix it this error

This screenshot shows the "All Pipelines" page in CircleCI. The sidebar on the left is identical to the previous one. The main area lists several pipelines:

- udapeople.CICD 171**: Status Success, Workflow default, Branch master, Started 10m ago, Duration 5m 57s +47%, Last commit 3d19c47, Action: configure infrastructure. Jobs: build-frontend, scan-frontend, test-frontend, build-backend, scan-backend, test-backend, deploy-infrastructure, configure-infrastructure, run-migrations.
- udapeople.CICD 170**: Status Failed, Workflow default, Branch master, Started 21m ago, Duration 5m 20s, Last commit cdc7a4e, Action: configure infrastructure.
- udapeople.CICD 169**: Status Failed, Workflow default, Branch master, Started 30m ago, Duration 5m 25s, Last commit 4dd45ed, Action: configure infrastructure.
- udapeople.CICD 168**: Status Failed, Workflow default, Branch master, Started 41m ago, Duration 9m 22s, Last commit f84d1fd, Action: configure infrastructure.
- udapeople.CICD 167**: Status Failed, Workflow default, Branch master, Started 2h ago, Duration 15m 22s, Last commit d696530, Action: configure infrastructure.
- udapeople.CICD 166**: Status Failed, Workflow default, Branch master, Started 2h ago, Duration 15m 22s, Last commit 2a65ff8, Action: configure infrastructure. Error message: "Error calling workflow: 'default' Error calling job: 'run-migrations' Cannot find a definition for command named revert-migrations".
- udapeople.CICD 165**: Status Success, Workflow default, Branch master, Started 2h ago, Duration 15m 22s, Last commit 2a65ff8, Action: configure infrastructure. Message: "Reconnected to Internet. Please refresh for updates."

At the bottom, there are system status icons and a status bar showing "20°C" and "9:50 PM" on 10/21/2022.

➤ So we start to run code deploy backend and frontend

The screenshot shows the CircleCI All Pipelines dashboard. On the left, there's a sidebar with user information (ayarabih) and navigation links like Dashboard, Projects, Insights, Organization Settings, and Plan. A note says "CI behind your firewall just got easier". Below the sidebar is a "Getting Started" section with links to Notifications, Status (OPERATIONAL), Docs, Orbs, and Support.

The main area displays a table of pipelines. One pipeline, "utapeople.CI\_CD 172", is shown as "Success" with a green button. Another pipeline, "utapeople.CI\_CD 171", is also "Success". Several other pipelines are listed as "Failed" with red buttons. The table includes columns for Pipeline, Status, Workflow, Branch / Commit, Start, Duration, and Actions. The "Actions" column contains icons for viewing logs, rerunning, and deleting.

At the bottom right of the dashboard, a message says "Reconnected to Internet. Please refresh for updates." The system status bar at the bottom indicates it's 20°C, 10:26 PM, and 10/21/2022.

➤ Now we run smoke test

This screenshot shows the CircleCI Step details page for a specific job. The sidebar and "Getting Started" section are identical to the previous dashboard. The main content shows a "STEPS" tab with several steps listed:

- Spin up environment (1s)
- Preparing environment variables (0s)
- Checkout code (0s)
- Install AWS CLI v2 (2s)
- Install Node.js 13 (28s)
- Frontend smoke test (0s)

The "Frontend smoke test" step is highlighted in pink. It shows a terminal session with the following command and output:

```
#!/bin/bash -eo pipefail
URL="http://utapeople-41c10.rhcloud.com:8070"
if curl -s $URL | grep "Welcome"
then
    # Change this to 0 after the job fails
    return 1
else
    return 0
fi

http://utapeople-3098525.s3-website-us-east-1.amazonaws.com/#/employees
Welcome
/bin/bash: line 5: return: can only "return" from a function or sourced script
Exited with code exit status 1
CircleCI received exit code 1
```

The system status bar at the bottom indicates it's 23°C, 4:40 PM, and 10/22/2022.

- When we check that we have node that mean EC2 it is run

```
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.15.0-1019-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Sun Oct 23 20:36:15 UTC 2022

System load: 0.01          Processes:           103
Usage of /: 37.5% of 7.57GB Users logged in: 0
Memory usage: 42%          IPv4 address for eth0: 172.31.88.96
Swap usage: 0%             

0 updates can be applied immediately.

New release '22.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Sun Oct 23 20:29:18 2022 from 54.158.221.253
ubuntu@ip-172-31-88-96:~$ node -v
v10.19.0
ubuntu@ip-172-31-88-96:~$ 
```

- Now we create cloud front-update

- When we install dependents for cloud front step

Servicess Search for services, features, blogs, docs, and more [Alt+S]

EC2 S3 IAM CloudFormation RDS

### CloudFront

Distributions Policies Functions What's new Telemetry Monitoring Alarms Logs Reports & analytics

CloudFront > Distributions

**Distributions (1) Info**

ID	Description	Domain name	Alternate domain names	Origins	Status
EPGI4H5SWO03C	-	d3azzqvbo48d1.cloudfront.net	-	udepeople-udepeople-kk1j287dh	Enabled

aws Services Search for services, features, blogs, docs, and more [Alt+S]

EC2 S3 IAM CloudFormation RDS

### New EC2 Experience Tell us what you think

EC2 Dashboard EC2 Global View Events Tags Limits

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

Images AMIs AMI Catalog

Elastic Block Store Volumes Snapshots Lifecycle Manager

CloudFront > Distributions

**Instances (1/2) Info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
backend-56a6d43	i-058cbf889fbf31435	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-3-80-54-124.comp...	3.80.54.124	-
backend-ff9acce	i-056c50a0463bd696e	Terminated	t2.micro	-	No alarms	us-east-1a	-	-	-

**Instance: i-058cbf889fbf31435 (backend-56a6d43)**

**Details** Security Networking Storage Status checks Monitoring Tags

**Instance summary** Info

Instance ID	Public IPv4 address	Private IPv4 addresses
i-058cbf889fbf31435 (backend-56a6d43)	3.80.54.124   open address	ec2-3-80-54-124.compute-1.amazonaws.com   open address
IPv6 address	Running	
Hostname type	Private IP DNS name (IPv4 only)	Elastic IP addresses
IP name: ip-172-31-26-250.ec2.internal	ip-172-31-26-250.ec2.internal	-
Answer private resource DNS name	t2.micro	AWS Compute Optimizer finding
-	VPC ID	Opt-in to AWS Compute Optimizer for recommendations.   Learn more
Auto-assigned IP address	vpc-0c490718fa2cc7a8c	
3.80.54.124 [Public IP]		

## When we need to access to EC2 with ssh

```
ubuntu@ip-172-31-26-250: ~
terraform/
DELL@DESKTOP-SEA7A9: ~ (master)
$ cd Desktop/
DELL@DESKTOP-SEA7A9: ~ (master)
$ ssh -i udacity.pem ubuntu@ec2-3-80-54-124.compute-1.amazonaws.com
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.15.0-1019-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 System information as of Tue Oct 25 00:37:31 UTC 2022

 System load: 0.0      Processes:          102
 Usage of /: 37.5% of 7.57GB   Users logged in:    0
 Memory usage: 42%           IPv4 address for eth0: 172.31.26.250
 Swap usage:  0%          

 * Ubuntu Pro delivers the most comprehensive open source security and
 compliance features.

 https://ubuntu.com/aws/pro

0 updates can be applied immediately.

New release '22.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Tue Oct 25 00:15:38 2022 from 44.204.190.183
ubuntu@ip-172-31-26-250:~$
```

➤ Run command docker pull prom/Prometheus

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\DELL> docker -v
PS C:\Users\DELL> docker pull prom/prometheus
Using default tag: latest
latest: Pulling from prom/prometheus
50783e0dfb64: Pull complete
daafab1bca260: Pull complete
72d3569fdc6f: Pull complete
13afa930da33: Pull complete
6ef28183cda8: Pull complete
4ad7245dbb40: Pull complete
26e6063b72b5: Pull complete
d859dd8f8ba9: Pull complete
583221d3597c: Pull complete
b4e477a4eb49: Pull complete
0b0ad5fc938d: Pull complete
53ddffa5a7d1: Pull complete
Digest: sha256:4748e26f9369ee7270a7cd3fb9385c1adb441c05792ce2bce2f6dd622fd91d38
Status: Downloaded newer image for prom/prometheus:latest
docker.io/prom/prometheus:latest
PS C:\Users\DELL> docker image
PS C:\Users\DELL> docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
postgres            latest   901a82b310d3  10 days ago   377MB
prom/prometheus     latest   6b9895947e9e  2 weeks ago   220MB
docker/getting-started  latest   cb90f98fd791  6 months ago  28.8MB
```

➤ docker run -p 9090:9090 prom/Prometheus

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Digest: sha256:4748e26f9369ee7270a7cd3fb9385c1adb441c05792ce2bce2f6dd622fd91d38
Status: Downloaded newer image for prom/prometheus:latest
docker.io/prom/prometheus:latest
PS C:\Users\DELL> docker image
PS C:\Users\DELL> docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
postgres            latest   901a82b310d3  10 days ago   377MB
prom/prometheus     latest   6b9895947e9e  2 weeks ago   220MB
docker/getting-started  latest   cb90f98fd791  6 months ago  28.8MB
PS C:\Users\DELL> docker run -p 9090:9090 prom/prometheus
ts=2022-10-25T01:39:24.820Z caller=main.go:499 level=info msg="No time or size retention was set so using the default time retention" duration=15d
ts=2022-10-25T01:39:24.820Z caller=main.go:543 level=info msg="Starting Prometheus Server" mode=server version="(version=2.39.1, branch=HEAD, revision=dcd6af9e0d56165c6f5c64ebbc1fae798d24933a)"
ts=2022-10-25T01:39:24.821Z caller=main.go:548 level=info build_context="(go=go1.19.2, user=root@273d60c69592, date=20221007-15:57:09)"
ts=2022-10-25T01:39:24.821Z caller=main.go:549 level=info host_details="(Linux 5.10.16.3-microsoft-standard-WSL2 #1 SMP Fri Apr 2 22:23:49 UTC 2021 x86_64 c58d3ef01f33 localdomain)"
ts=2022-10-25T01:39:24.822Z caller=main.go:550 level=info fd_limits="(soft=1048576, hard=1048576)"
ts=2022-10-25T01:39:24.823Z caller=main.go:551 level=info vm_limits="(soft=unlimited, hard=unlimited)"
ts=2022-10-25T01:39:24.835Z caller=web.go:559 level=info component=web msg="Start listening for connections" address=0.0.0.0:9090
ts=2022-10-25T01:39:24.840Z caller=main.go:980 level=info msg="Starting TSDB ..." ts=2022-10-25T01:39:24.848Z caller=tls_config.go:195 level=info component=web msg="TLS is disabled." http2=false
ts=2022-10-25T01:39:24.851Z caller=head.go:551 level=info component=tsdb msg="Replaying on-disk memory mappable chunks if any"
ts=2022-10-25T01:39:24.851Z caller=head.go:595 level=info component=tsdb msg="On-disk memory mappable chunks replay completed" duration=5.3μs
ts=2022-10-25T01:39:24.851Z caller=head.go:601 level=info component=tsdb msg="Replaying WAL, this may take a while"
ts=2022-10-25T01:39:24.853Z caller=head.go:672 level=info component=tsdb msg="WAL segment loaded" segment=0 maxSegment=0
ts=2022-10-25T01:39:24.853Z caller=head.go:709 level=info component=tsdb msg="WAL replay completed" checkpoint_replay_duration=342.3μs wbl_replay_duration=1.2054ms wbl_replay_duration=200ns total_replay_duration=1.6045ms
ts=2022-10-25T01:39:24.855Z caller=main.go:1001 level=info fs_type=EXT4_SUPER_MAGIC
ts=2022-10-25T01:39:24.855Z caller=main.go:1004 level=info msg="TSDB started"
ts=2022-10-25T01:39:24.855Z caller=main.go:1184 level=info msg="Loading configuration file" filename=/etc/prometheus/prometheus.yml
ts=2022-10-25T01:39:24.860Z caller=main.go:1221 level=info msg="Completed loading of configuration file" filename=/etc/prometheus/prometheus.yml totalDuration=4.364ms db_storage=30.1μs remote_storage=2.4μs web_handler=900ns query_engine=1.3μs scrape=3.3243ms scrape_sd=54.9μs notify=54.7μs notify_sd=31μs rules=1.9μs tracing=18.3μs
ts=2022-10-25T01:39:24.860Z caller=main.go:965 level=info msg="Server is ready to receive web requests."
ts=2022-10-25T01:39:24.860Z caller=manager.go:943 level=info component="rule manager" msg="Starting rule manager..."
```

- When we will run Prometheus local will see that

The screenshot shows the Prometheus Targets page. At the top, there's a search bar with the URL "192.168.1.2:9090/targets?search=". Below the search bar is a navigation bar with links: Import bookmarks..., PROGRAM, AWS, PERSONAL, secret, freelancer, update program, language, work, ترجمة, Dashboard, FWD, ...ayara, and البريد الوارد - ayara. The main title is "Prometheus" with a logo. Below the title are tabs: Alerts, Graph, Status, and Help.

## Targets

Filter by endpoint or labels

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

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	13.911s ago	4.514ms	

- Now we run steps for step clean

The screenshot shows the CircleCI interface. On the left, there's a sidebar with user information ("ayarabih", "Aya Rabih") and navigation links: Dashboard, Projects, Insights, Organization Settings, and Plan. A message box says "CI behind your firewall just got easier". The main area shows a pipeline named "cleanup (1326)" under "All Pipelines > udapeople.CI\_CD > master > my-workflow > cleanup (1326)". The "cleanup" step is marked as "Success".

**cleanup** Success

Duration / Finished: 8s / 21m ago | Queued: 0s | Executor / Resource Class: Docker / Large | Branch: master | Commit: 74a393a | Author & Message: configure infrastructure

**STEPS TESTS TIMING ARTIFACTS RESOURCES NEW**

**Parallel runs**

0 00:08 Use parallelism to run faster tests Parallelism speeds up tests by splitting them across multiple executors. Go to Docs X

- Spin up environment
- Preparing environment variables
- Checkout code

**Spin up environment** 1s ✓

**Preparing environment variables** 0s ✓

**Checkout code** 0s ✓

**Install the latest pip** 17s ✓

**Install [ awscli ] through pip3.** 5s ✓

**final cleanup on success** 0s ✓

```

1 #!/bin/bash -eo pipefail
2 export OldWorkflowID=$(aws cloudformation list-exports \
3   --query "Exports[?Name=~'WorkflowID'].Value" \
4   --no-paginate --output text)
5 echo OldWorkflowID: "$OldWorkflowID"
6 echo CIRCLE_WORKFLOW_ID="udapeople-$CIRCLE_WORKFLOW_ID:0:7"
7 export STACKS=$(aws cloudformation list-stacks \
8   --query "StackSummaries[*].StackName" \
9   --stack-status-filter CREATE_COMPLETE --no-paginate --output text)
10 echo Stack names: "$STACKS[@]"
11
12 if [[ "$CIRCLE_WORKFLOW_ID:0:7" != "$OldWorkflowID" ]]
13 then
14   aws s3 rm "s3://$OldWorkflowID" --recursive
15   aws cloudformation delete-stack --stack-name "$STACKS[0]" \
16     & aws cloudformation delete-stack --stack-name "$STACKS[1]"
17 else
18   echo "nothing to cleanup!"
19 fi
20
21 OldWorkflowID: a25da79
22 CIRCLE_WORKFLOW_ID udapeople-a25da79
23 Stack names: udapeople-frontend-a25da79 udapeople-backend-a25da79
24 nothing to cleanup!
25 CircleCI received exit code 0

```

➤ It is done now

The screenshot shows a CI/CD pipeline interface with the following details:

- Pipeline:** ut/people.CI\_CD (262)
- Status:** Success
- Workflow:** my-workflow
- Branch / Commit:** master  
74a393a configure infrastructure
- Start:** 21m ago
- Duration:** 14m 50s
- Actions:** Auto-expand
- Jobs:** A list of 25 jobs, all marked as successful (green checkmarks). The jobs are:
  - build-frontend 1313
  - scan-frontend 1317
  - test-frontend 1318
  - build-backend 1314
  - scan-backend 1315
  - test-backend 1316
  - deploy-infrastructure 1319
  - configure-infrastructure 1320
  - run-migrations 1321
  - deploy-backend 1323
  - deploy-frontend 1322
  - smoke-test 1324
  - cloudfront-update 1325
  - cleanup 1326
- Timeline:** A detailed timeline showing the duration of each job and the overall pipeline run.

➤ Now we will check than backend listening for front end we will create user

The screenshot shows an employee management application with the following details:

- Employee:** aya
- Last Name:** rabih
- Email:** ayarabih21@gmail.com
- Birthday:** 10/23/2022
- Start Date:** 10/23/2022
- Action:** A green button with three dots.

➤ That all steps for our CI/CD

The screenshot shows a detailed view of a CI/CD pipeline with the following details:

- Pipeline:** my-workflow
- Status:** Success
- Duration / Finished:** 14m 50s / 47m ago
- Branch:** master
- Commit:** 74a393a configure infrastructure
- Author & Message:** ayarabih21 configure infrastructure
- Workflow Diagram:** A flowchart showing the sequence of 25 jobs. The main flow starts with build-frontend (27s), followed by scan-frontend (51s), deploy-infrastructure (2m 22s), configure-infrastructure (15s), run-migrations (42s), deploy-backend (4m 25s), deploy-frontend (34s), smoke-test (34s), cloudfront-update (4m 22s), and cleanup (9s). There are several parallel branches: one from build-frontend to test-frontend (1m 11s) and another from build-backend to test-backend (1m 9s and 1m 7s).

Now we will make premeasures for new 2EC2 and we have for them 1 security group and one keypair

```
ubuntu@ip-172-31-84-175:~$ sudo systemctl start node-exporter
ubuntu@ip-172-31-84-175:~$ sudo systemctl status node-exporter
● node-exporter.service - Prometheus Node Exporter Service
  Loaded: loaded (/etc/systemd/system/node-exporter.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2022-10-25 23:09:33 UTC; 9s ago
    Main PID: 3814 (node_exporter)
      Tasks: 3 (limit: 1143)
     Memory: 1.7M
        CPU: 0.000 CPU(s) since start
       CGroup: /system.slice/node-exporter.service
           └─3814 /usr/local/bin/node_exporter

ubuntu@ip-172-31-84-175:~$ sudo systemctl restart prometheus
ubuntu@ip-172-31-84-175:~$ sudo systemctl status prometheus
● prometheus.service - Prometheus
  Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2022-10-25 23:10:51 UTC; 3s ago
    Main PID: 3828 (prometheus)
      Tasks: 6 (limit: 1143)
     Memory: 15.2M
        CPU: 0.000 CPU(s) since start
       CGroup: /system.slice/prometheus.service
           └─3828 /usr/local/bin/prometheus --config.file /etc/prometheus/prometheus.yml --storage.tsdb.path /var/lib/prometheus/tsdb

Oct 25 23:10:51 ip-172-31-84-175 prometheus[3828]: level=info ts=2022-10-25T23:10:51.662Z caller=head.go:645 component=web msg="Starting up"
Oct 25 23:10:51 ip-172-31-84-175 prometheus[3828]: level=info ts=2022-10-25T23:10:51.663Z caller=web.go:524 component=>http msg="HTTP server listening on :9100"
Oct 25 23:10:51 ip-172-31-84-175 prometheus[3828]: level=info ts=2022-10-25T23:10:51.666Z caller=head.go:706 component=web msg="HTTP server successfully started"
Oct 25 23:10:51 ip-172-31-84-175 prometheus[3828]: level=info ts=2022-10-25T23:10:51.666Z caller=head.go:706 component=web msg="Metrics endpoint successfully started"
Oct 25 23:10:51 ip-172-31-84-175 prometheus[3828]: level=info ts=2022-10-25T23:10:51.667Z caller=head.go:709 component=web msg="Log file successfully opened"
Oct 25 23:10:51 ip-172-31-84-175 prometheus[3828]: level=info ts=2022-10-25T23:10:51.669Z caller=main.go:694 fs_type=ext4
Oct 25 23:10:51 ip-172-31-84-175 prometheus[3828]: level=info ts=2022-10-25T23:10:51.669Z caller=main.go:695 msg="TSDB: open /var/lib/prometheus/tsdb: permission denied"
Oct 25 23:10:51 ip-172-31-84-175 prometheus[3828]: level=info ts=2022-10-25T23:10:51.669Z caller=main.go:799 msg="Load balancer successfully started"
Oct 25 23:10:51 ip-172-31-84-175 prometheus[3828]: level=info ts=2022-10-25T23:10:51.673Z caller=main.go:827 msg="Compaction goroutine successfully started"
Oct 25 23:10:51 ip-172-31-84-175 prometheus[3828]: level=info ts=2022-10-25T23:10:51.673Z caller=main.go:646 msg="Service metrics successfully registered"
Lines 1-19/19 (END)...skipping...
```

Prometheus    Alerts    Graph    Status ▾    Help

## Targets

All    Unhealthy

**node\_exporter (1/1 up)** show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="#">http://ec2-54-242-175-40.compute-1.amazonaws.com:9100/metrics</a>	UP	instance="ec2-54-242-175-40.compute-1.amazonaws.com:9100" job="node_exporter"	13.864s ago	14.15ms	

Prometheus    Alerts    Graph    Status ▾    Help

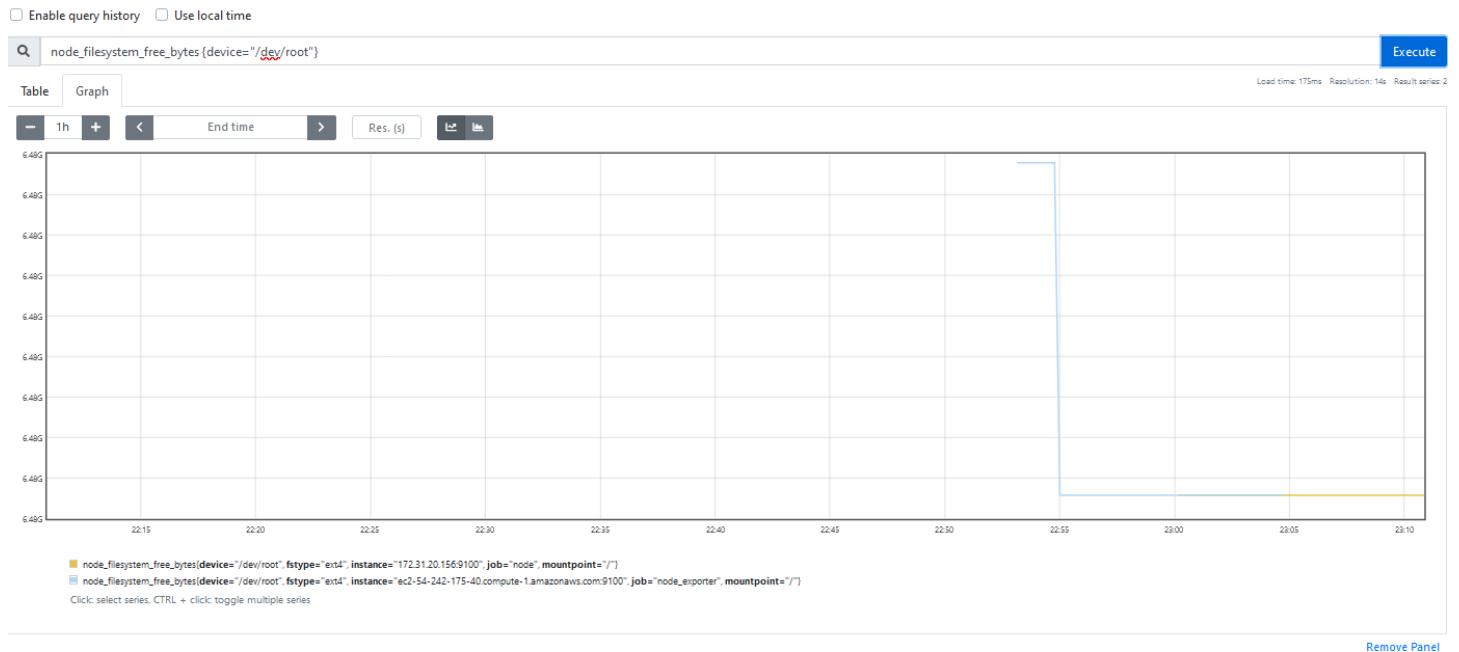
## Targets

All    Unhealthy

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

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="#">http://172.31.20.156:9100/metrics</a>	UP	instance="172.31.20.156:9100" job="node"	929ms ago	14.41ms	
<a href="#">http://172.31.23.221:9100/metrics</a>	DOWN	instance="172.31.23.221:9100" job="node"	482ms ago	390us	Get "http://172.31.23.221:9100/metrics": dial tcp 172.31.23.221:9100: connect connection refused

## this is dygram for node



Remove Panel

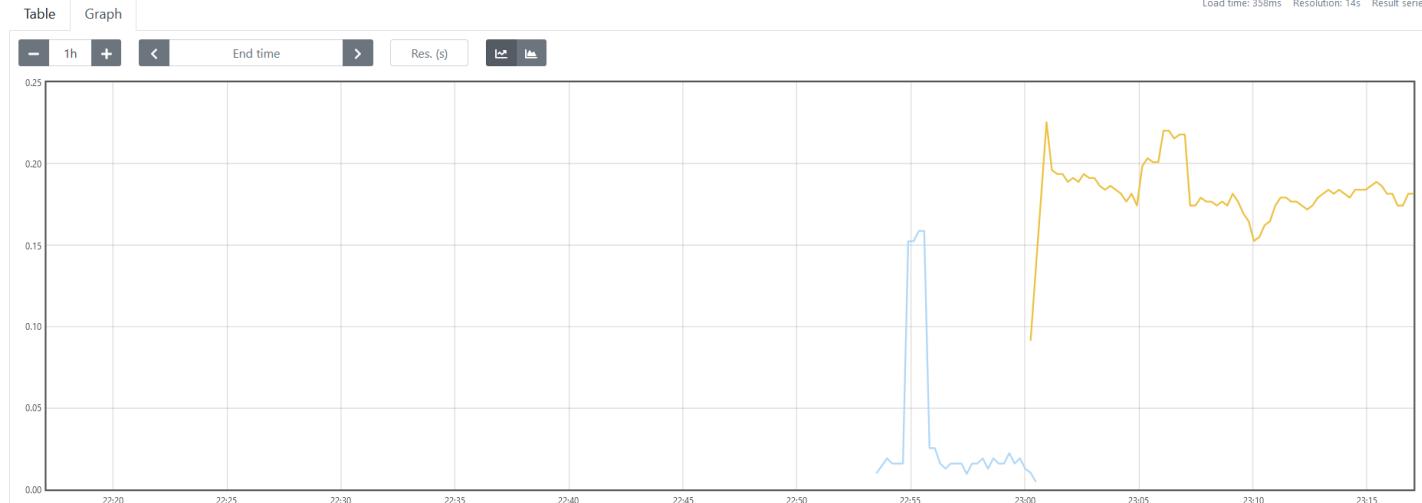
Add Panel

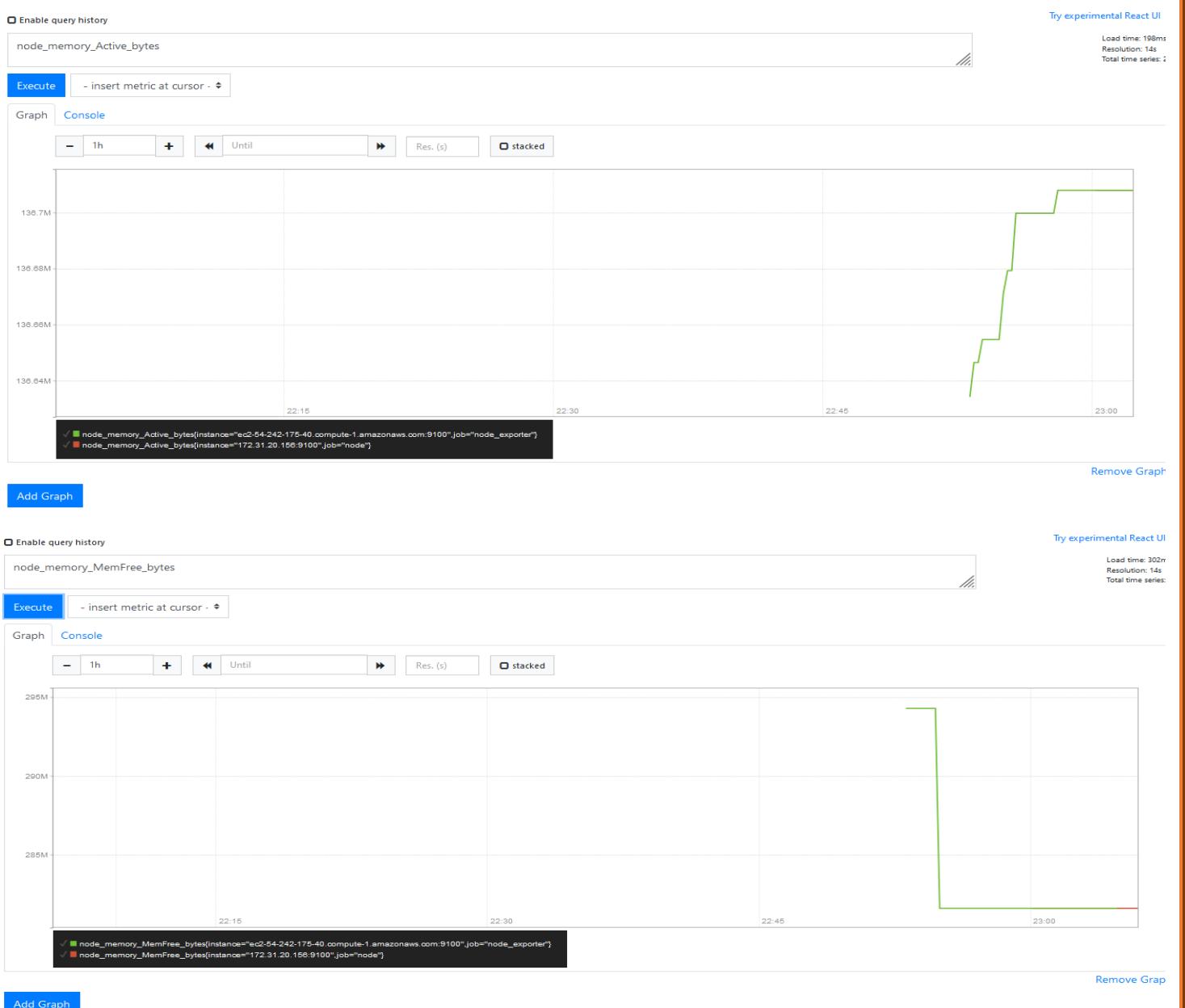
Prometheus Alerts Graph Status Help Classic UI

Enable query history  Use local time

avg(rate(node\_cpu\_seconds\_total[model="idle"])[1m]) by (instance) \* 100 Execute

Load time: 350ms Resolution: 14s Result series: 1





- we will now try to sent notification for our email when instance node down

Instances (2) Info

C Connect Instance state Actions Launch instances

Find instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Private IP
<input type="checkbox"/>	node	i-01e4c7b618c569b68	Stopped	t2.micro	-	No alarms	us-east-1b	-	-
<input type="checkbox"/>	server	i-0a82daa1047ecaa91	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-34-201-145-226.co...	34.201.145.226

- We here stop node instance

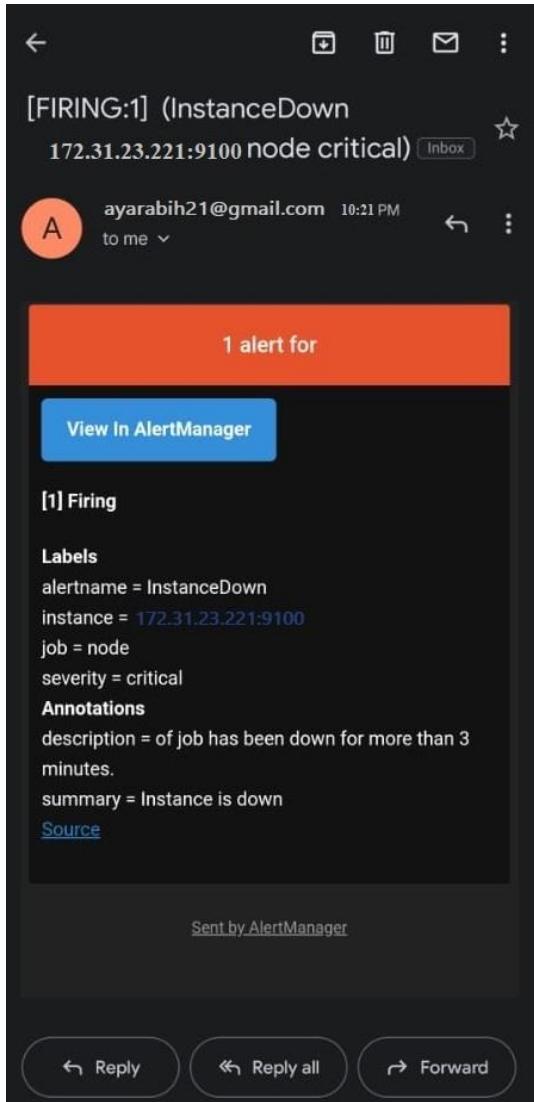
The screenshot shows the Prometheus Alerting interface. At the top, there are three filter buttons: 'Inactive (0)', 'Pending (0)', and 'Firing (1)'. Below them is a search bar with the text '/etc/prometheus/rules.yml > Down'. To the right of the search bar is a button labeled 'firing (2)'. A pink header bar indicates '2 active' alerts. The main area displays the configuration for the 'InstanceDown' alert:

```
name: InstanceDown
expr: up == 0
labels:
  severity: critical
annotations:
  summary: Instance is down
```

Below this, a table lists the active alerts:

Labels	State	Active Since	Value
alertname=InstanceDown,instance=172.31.20.156:9100,job=node,severity=critical	FIRING	2022-10-26T23:31:50.214668377Z	0e+00
alertname=InstanceDown,instance=172.31.23.221:9100,job=node,severity=critical	FIRING	2022-10-26T23:28:26.214668377Z	0e+00

- This is notification for email trailer



Project is done!