Project:

Give Your Application Auto-Deploy Superpowers

Selling CI/CD to your Team/Organization

- **Explain the fundamentals and benefits of CI/CD to achieve, build, and deploy automation for cloud-based software products.**

The benefits of CI/CD include:

+ Speed up development: CI/CD helps speed up software development by automating build and deployment processes, reducing the time and effort required for manual tasks.

+ Quality Assurance: Automated and continuous testing during the CI/CD process ensures that the software is reliable, stable, and monitored for quality standards.

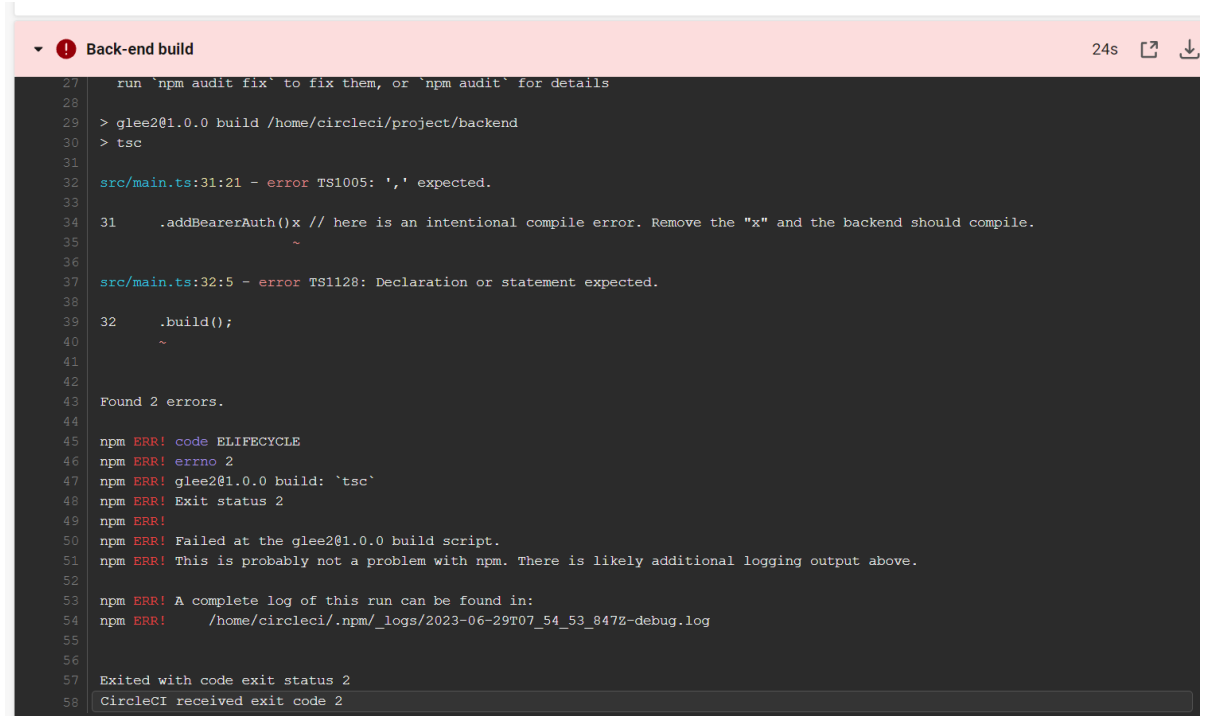
+ Risk Reduction: CI/CD helps to reduce the risk of errors arising during the development of software exploits by ensuring that changes are developed only after they have been tested and validated.

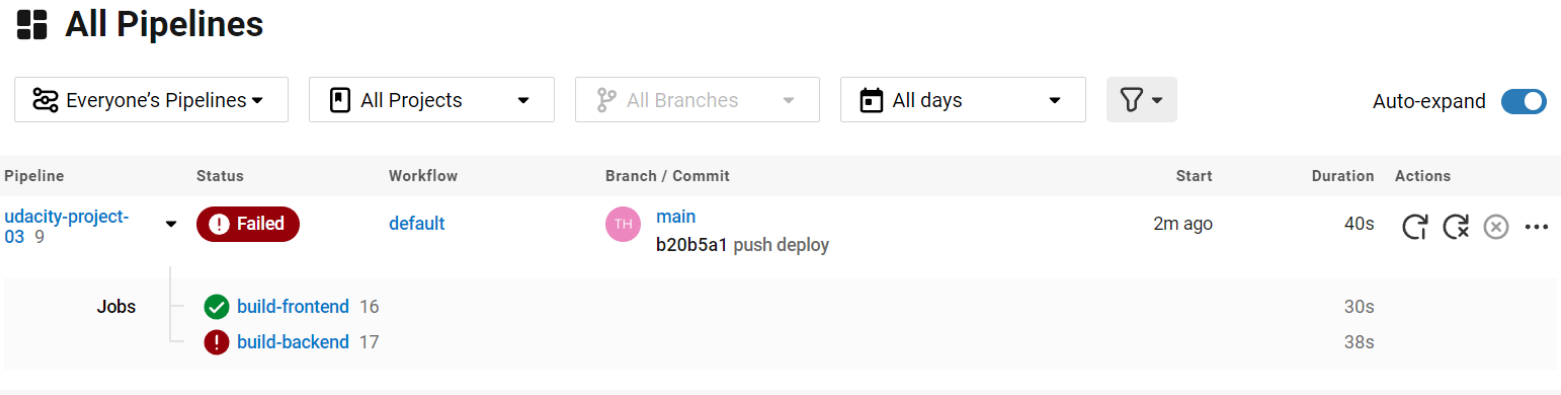
+ Increased scalability: The automation in CI/CD makes it easy to scale software systems, while ensuring consistency and reliability during deployment.

+ Continuous Integration: CI/CD facilitates continuous analysis of different software components, including source code, databases, and external services, which enhances interoperability and compatible interactions.

Build Phase

[SCREENSHOT01]

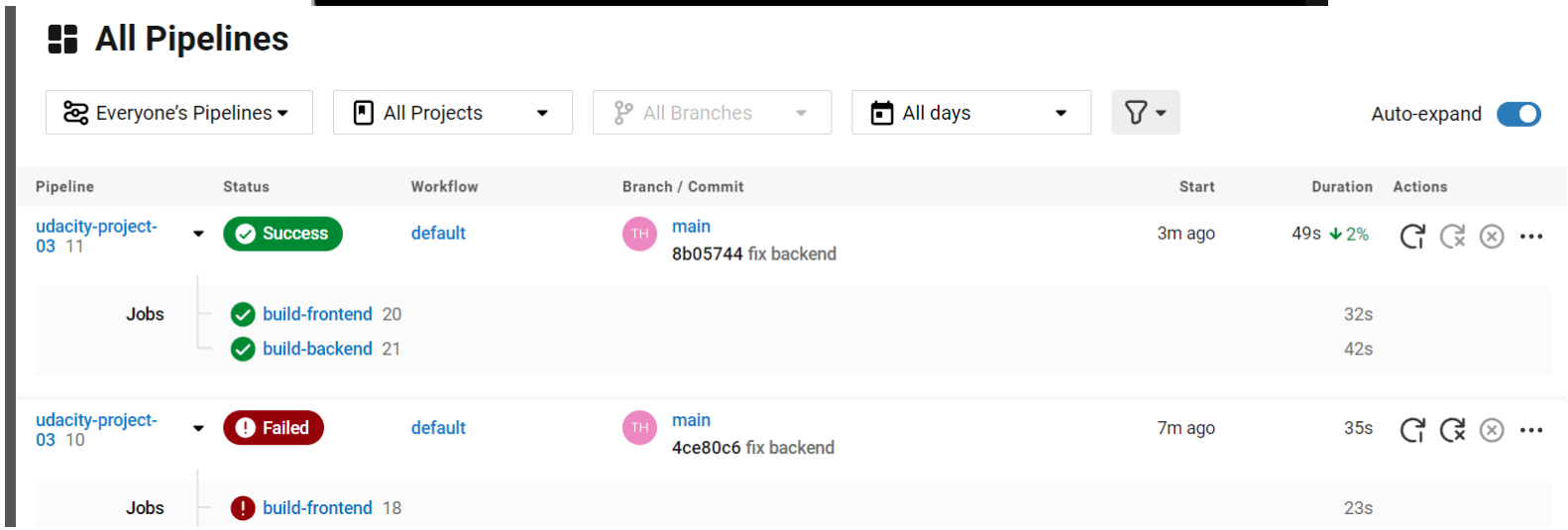




Build Phase

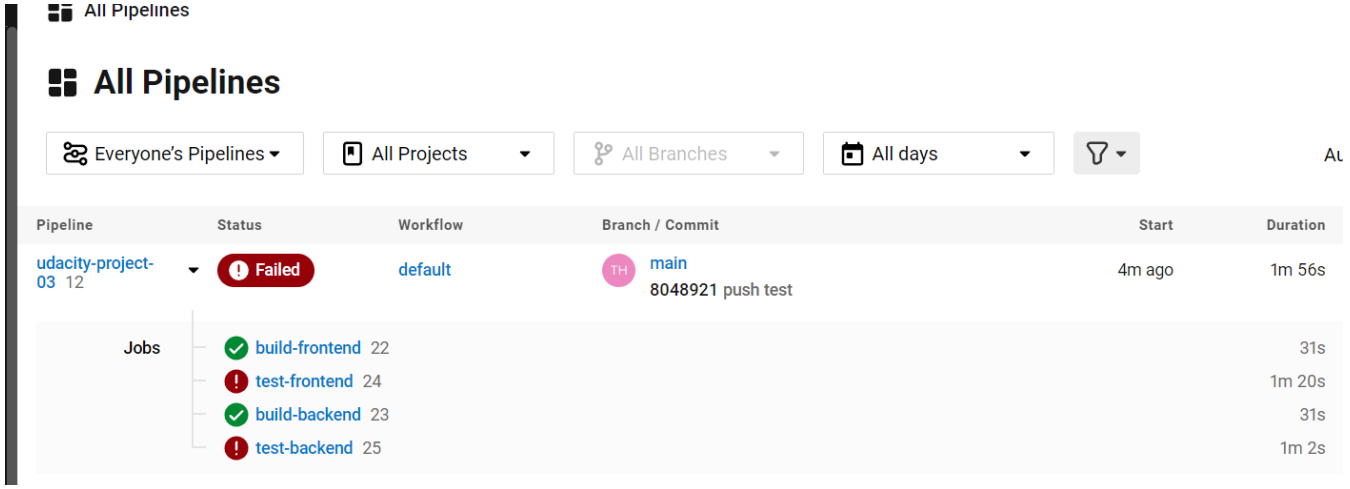
- Fix

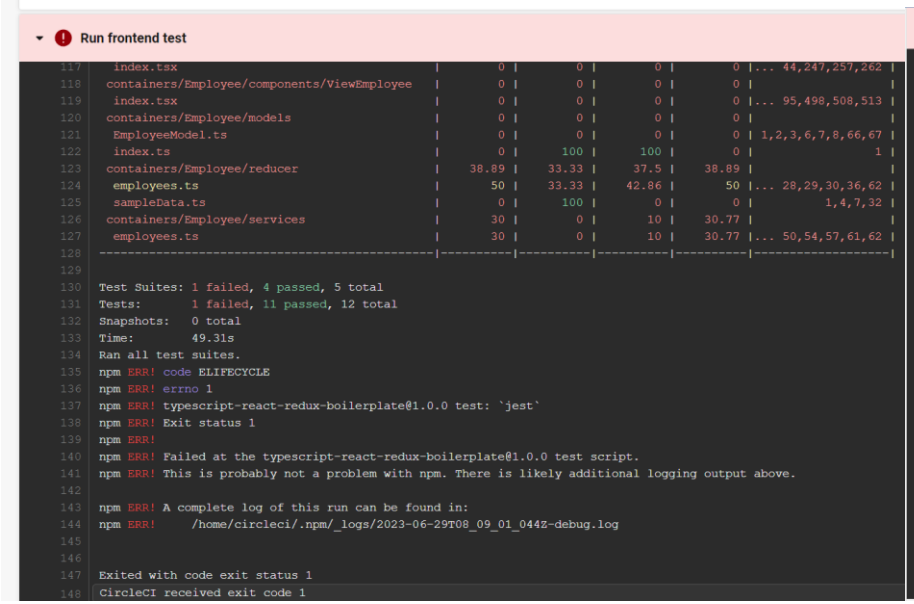
```
Administrator@DESKTOP-2361KFI MINGW64 ~/Desktop/AWS-Lerning/UDACITY-Project-03/c
dond-c3-projectstarter (main)
$ git commit -m "fix backend"
[main 8b05744] fix backend
 2 files changed, 18 insertions(+), 53 deletions(-)

Administrator@DESKTOP-2361KFI MINGW64 ~/Desktop/AWS-Lerning/UDACITY-Project-03/c
dond-c3-projectstarter (main)
$ git push origin main
Enumerating objects: 15, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 16 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 818 bytes | 818.00 KiB/s, done.
Total 8 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), completed with 6 local objects.
To https://github.com/phan-van-thuy/udacity-project-03.git
   4ce80c6..8b05744  main -> main

Administrator@DESKTOP-2361KFI MINGW64 ~/Desktop/AWS-Lerning/UDACITY-Project-03/c
dond-c3-projectstarter (main)
$
```

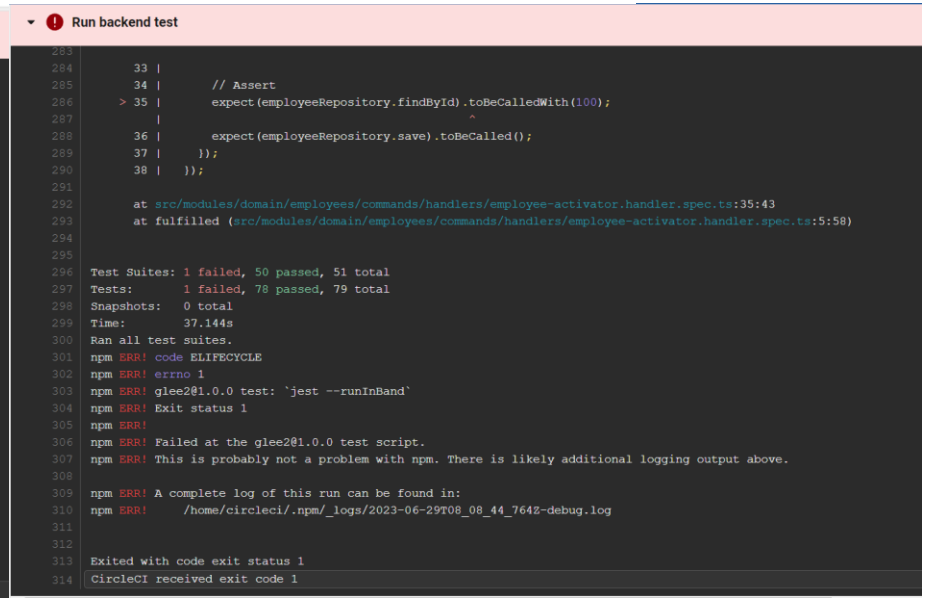


Test Phase

[SCREENSHOT02]



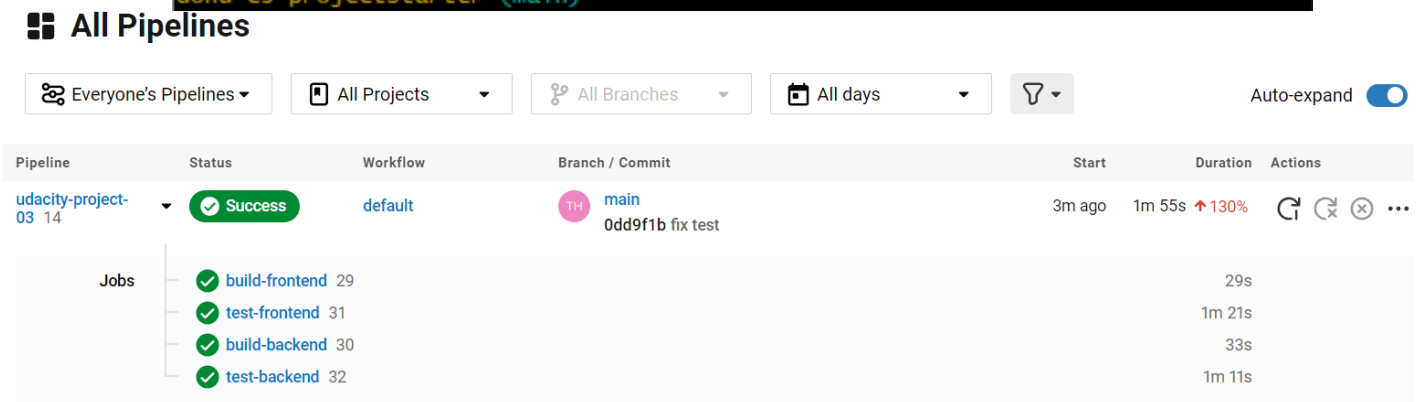




Test Phase

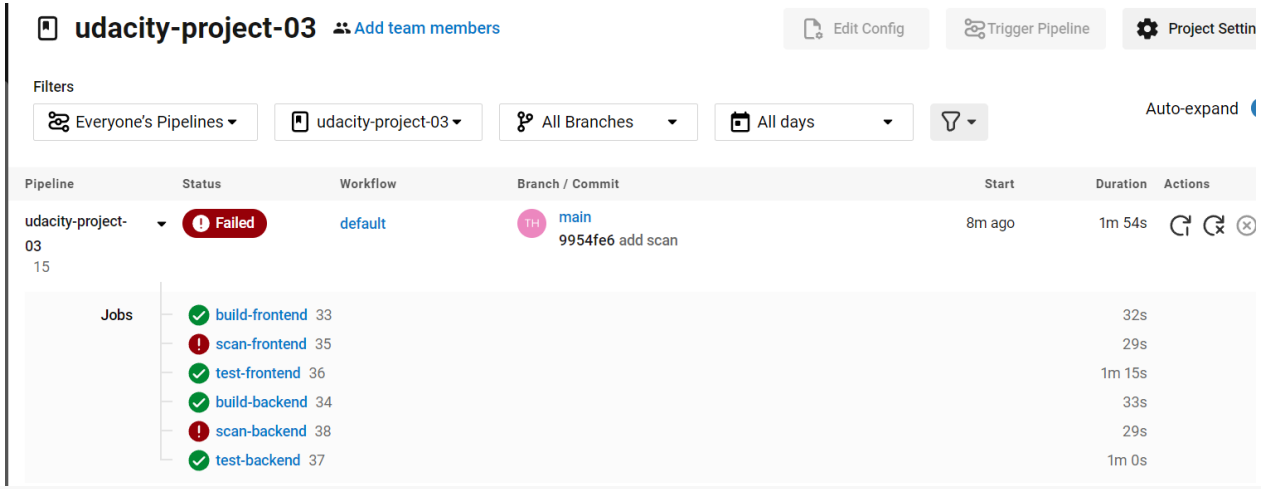
- FIX

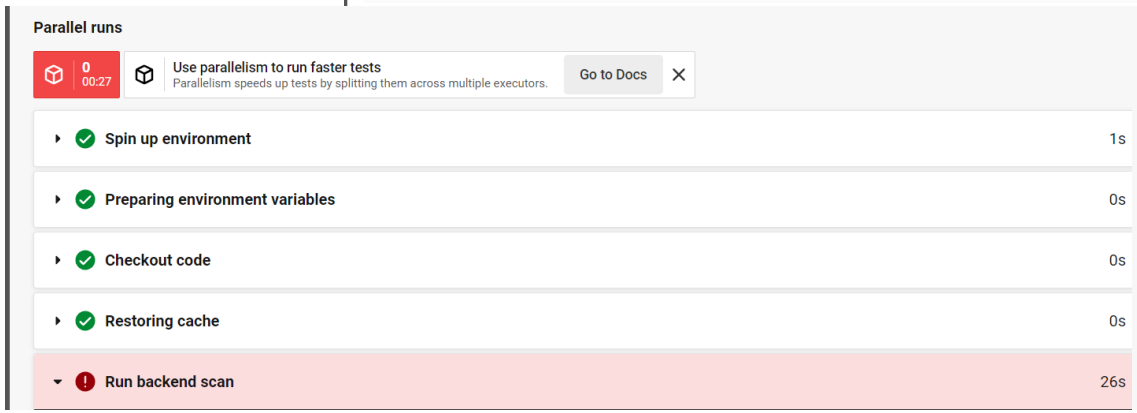
```
Administrator@DESKTOP-2361KFI MINGW64 ~/Desktop/AWS-Lerning/UDACITY-Project-03/c
dond-c3-projectstarter (main)
$ git commit -m "fix test"
[main 0dd9f1b] fix test
 1 file changed, 1 insertion(+), 1 deletion(-)

Administrator@DESKTOP-2361KFI MINGW64 ~/Desktop/AWS-Lerning/UDACITY-Project-03/c
dond-c3-projectstarter (main)
$ git push origin main
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 16 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 679 bytes | 679.00 KiB/s, done.
Total 8 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), completed with 6 local objects.
To https://github.com/phan-van-thuy/udacity-project-03.git
   7a6e798..0dd9f1b  main -> main

Administrator@DESKTOP-2361KFI MINGW64 ~/Desktop/AWS-Lerning/UDACITY-Project-03/c
dond-c3-projectstarter (main)
```

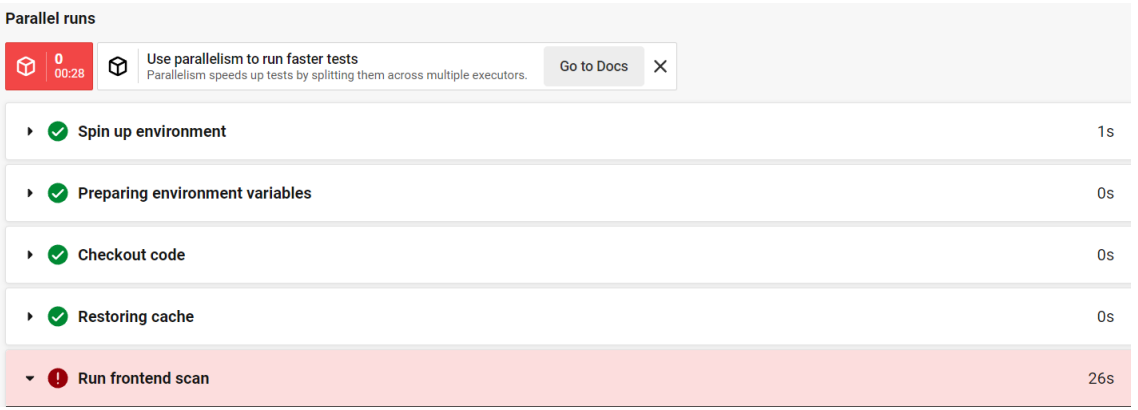


Analyze Phase

[SCREENSHOT03]



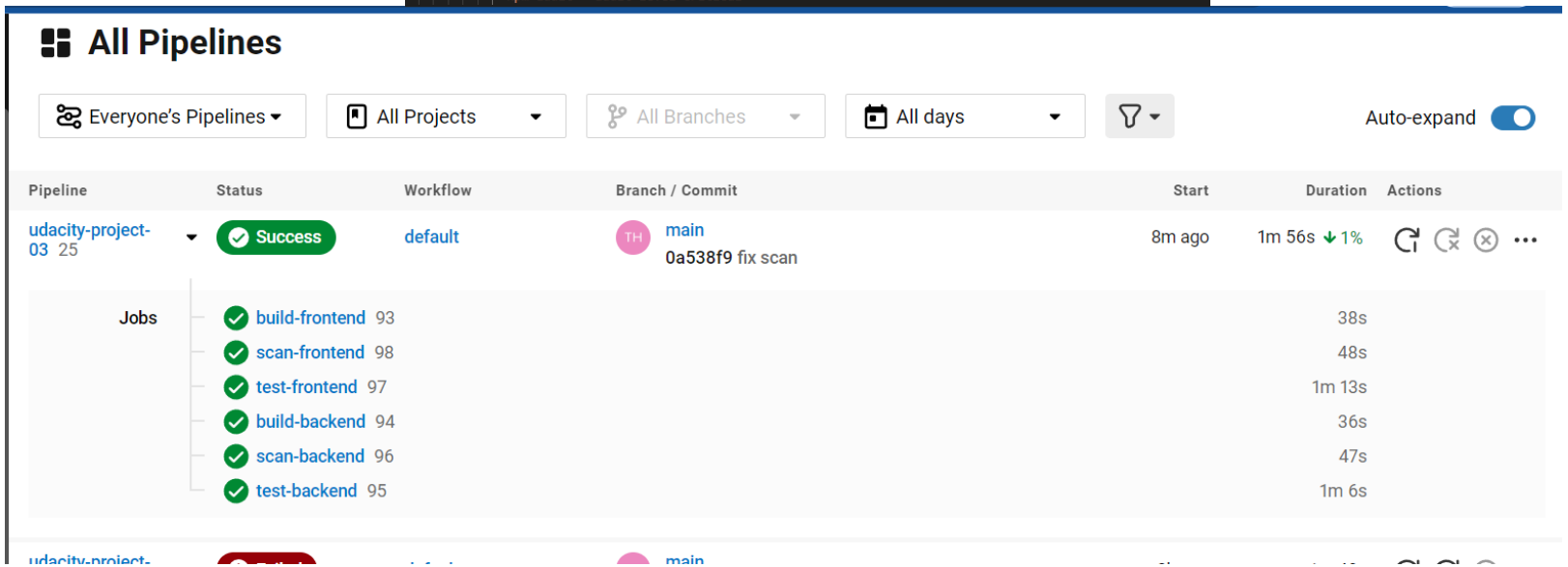




Analyze Phase

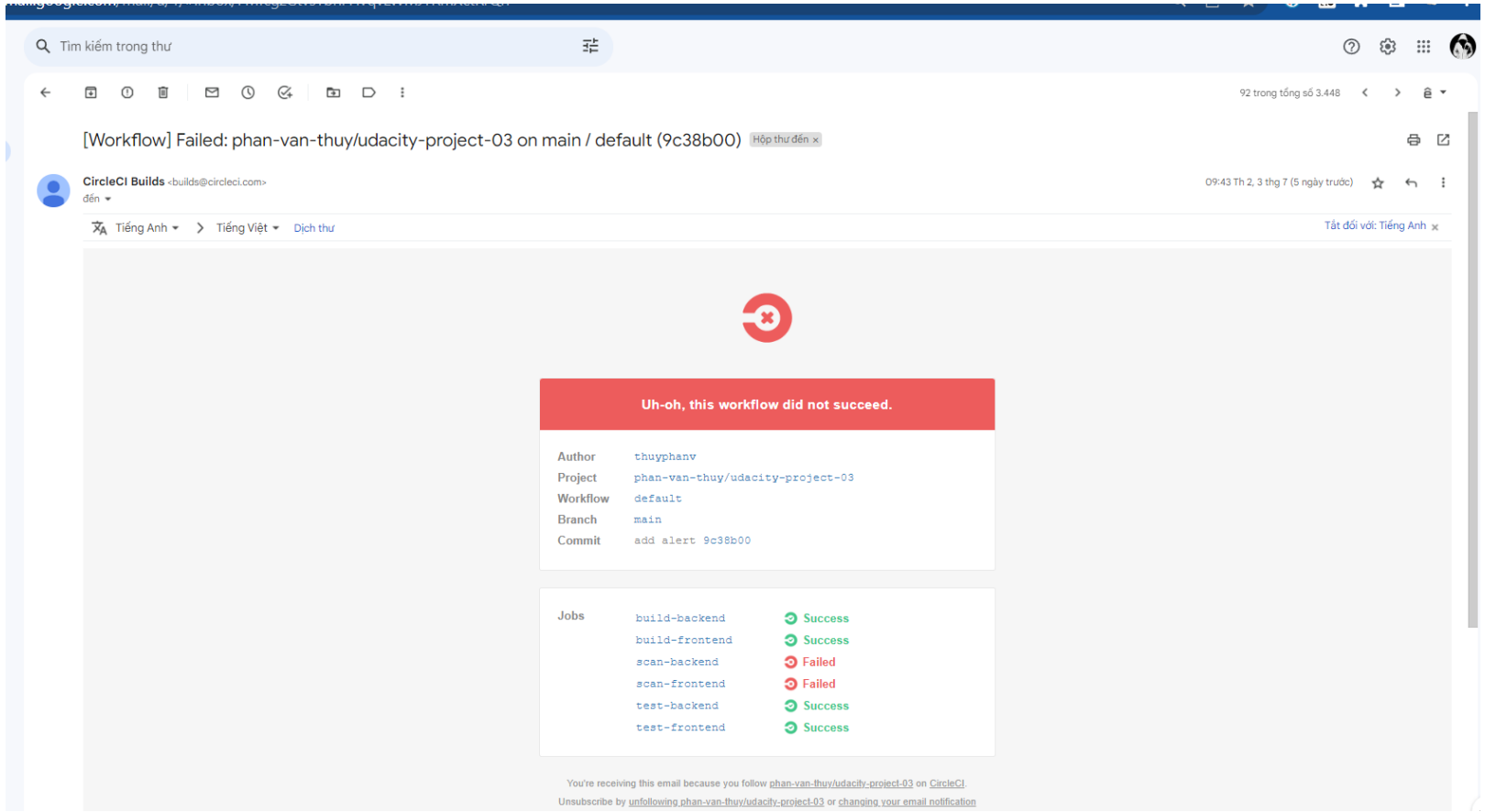
FIX

```
scan-frontend:
  docker:
    - image: circleci/node:13.8.0
  steps:
    - checkout
    - restore_cache:
        keys: [frontend-scan]
    - run:
        name: Run frontend scan
        command: |
          cd frontend
          npm install
          npm audit fix --audit-level=critical
          npm audit --audit-level=critical
scan-backend:
  docker:
    - image: circleci/node:13.8.0
  steps:
    - checkout
    - restore_cache:
        keys: [backend-scan]
    - run:
        name: Run backend scan
        command: |
          cd frontend
          npm install
          npm audit fix --audit-level=critical
          npm audit --audit-level=critical
```



Alerts

SCREENSHOT04



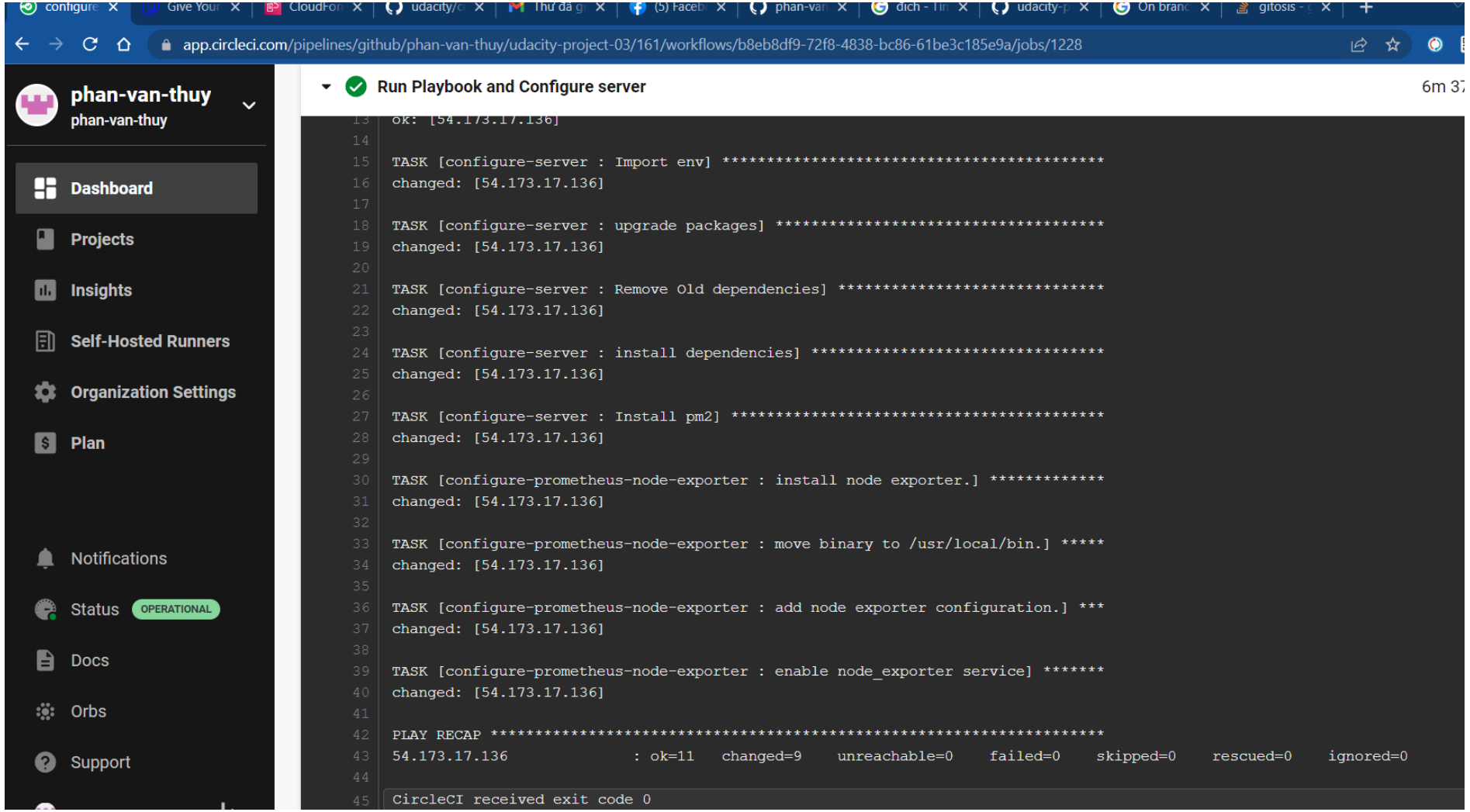
Create the Infrastructure

Create Backend

[SCREENSHOT05]

Configure Infrastructure

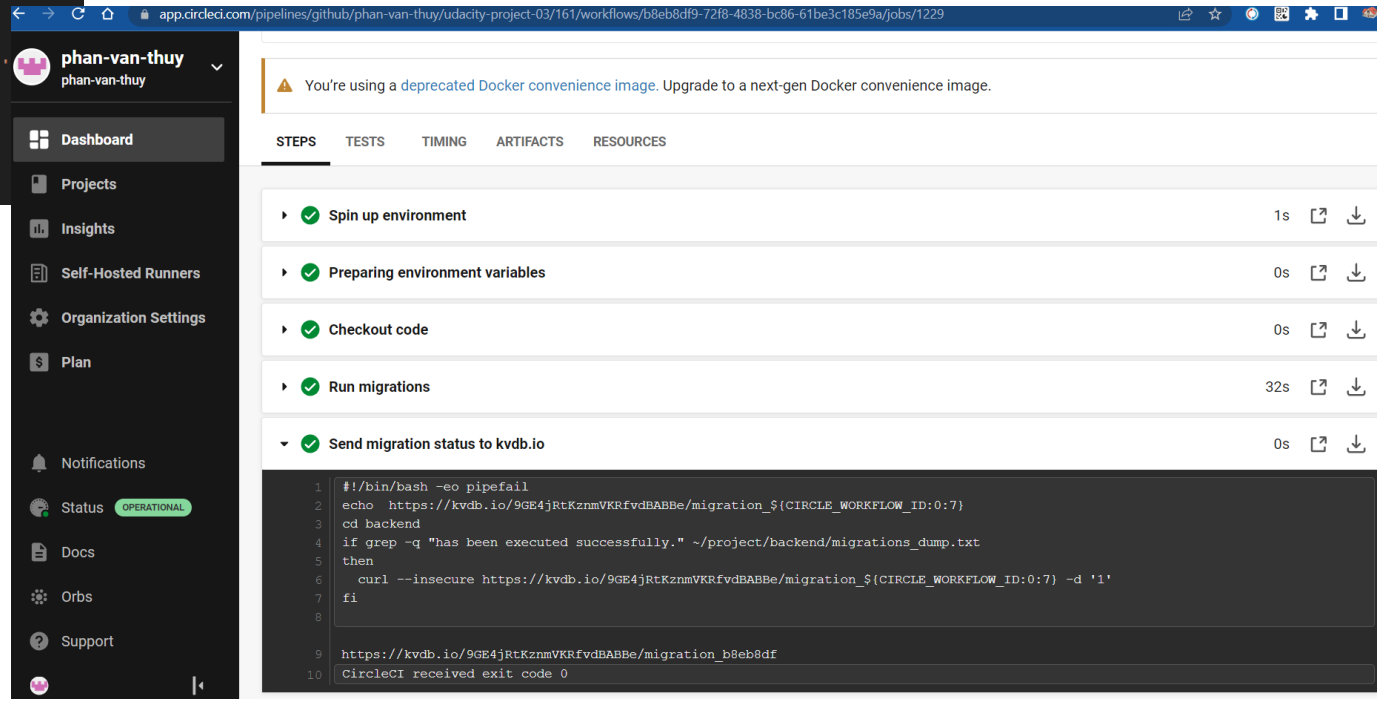
[URL01] https://github.com/phan-van-thuy/udacity-project-03/blob/main/.circleci/config.yml



Deploy Phase

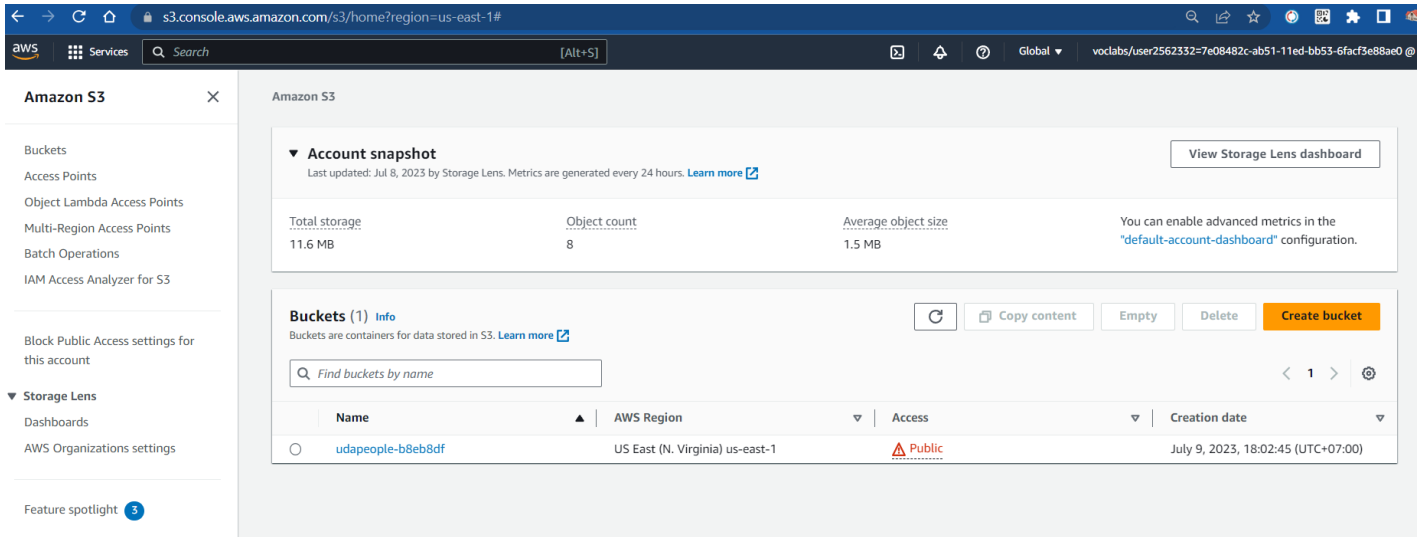
- **Database Migrations**

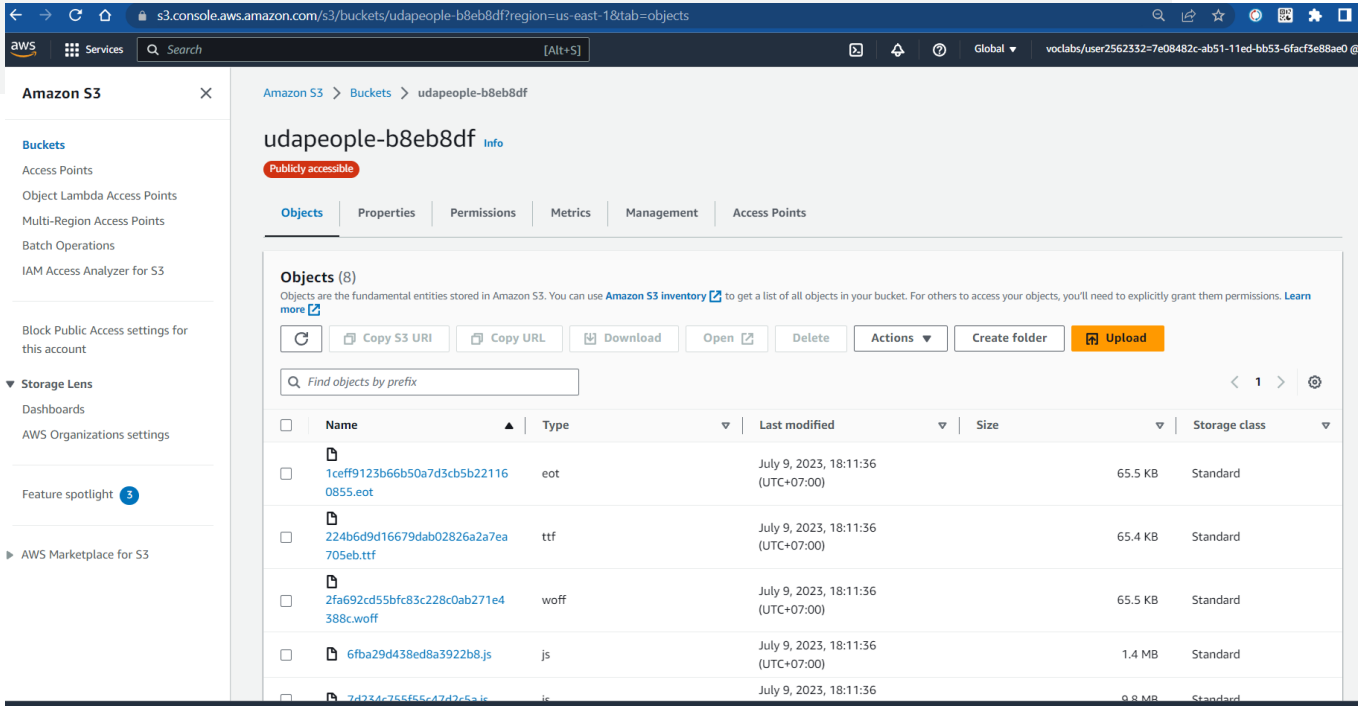
```yaml
run-migrations:
  docker:
    - image: circleci/node:13.8.0
  steps:
    - checkout
    - run:
        name: Run migrations
        command: |
          cd backend
          npm install
          npm run migrations > migrations_dump.txt
          cat migrations_dump.txt
    - run:
        name: Send migration status to kvdb.io
        command: |
          echo  https://kvdb.io/9GE4jRtKznmVKRfvdBABBe/migration_${CIRCLE_WORKFLOW_ID:0:7}
          cd backend
          if grep -q "has been executed successfully." ~/project/backend/migrations_dump.txt
          then
            curl --insecure https://kvdb.io/9GE4jRtKznmVKRfvdBABBe/migration_${CIRCLE_WORKFLOW_ID:0:7} -d '
          fi
  # Rollback if fail
    - destroy-environment
    - revert-migrations
```



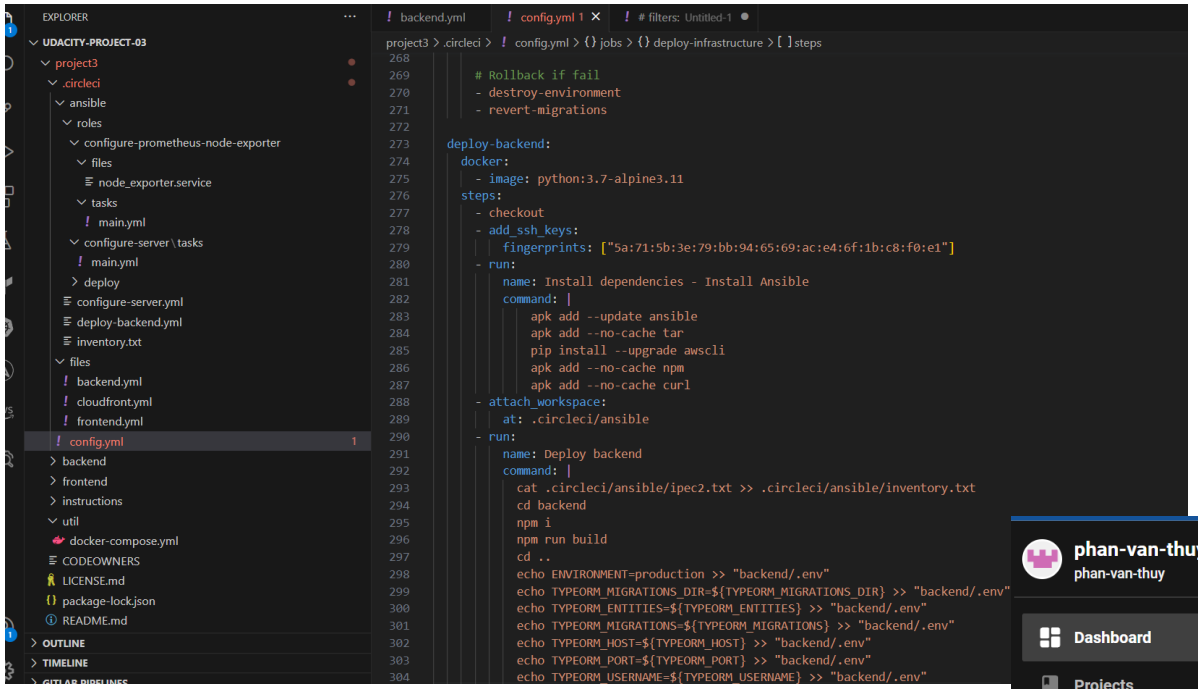
Deploy Frontend

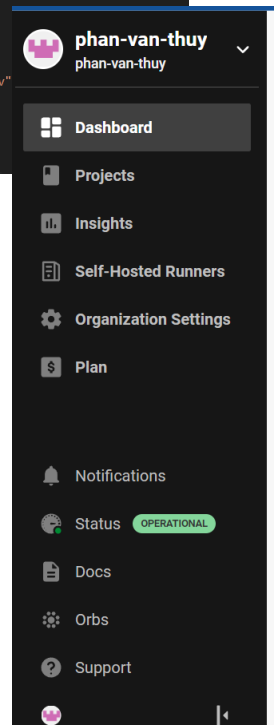
- **[URL02]**

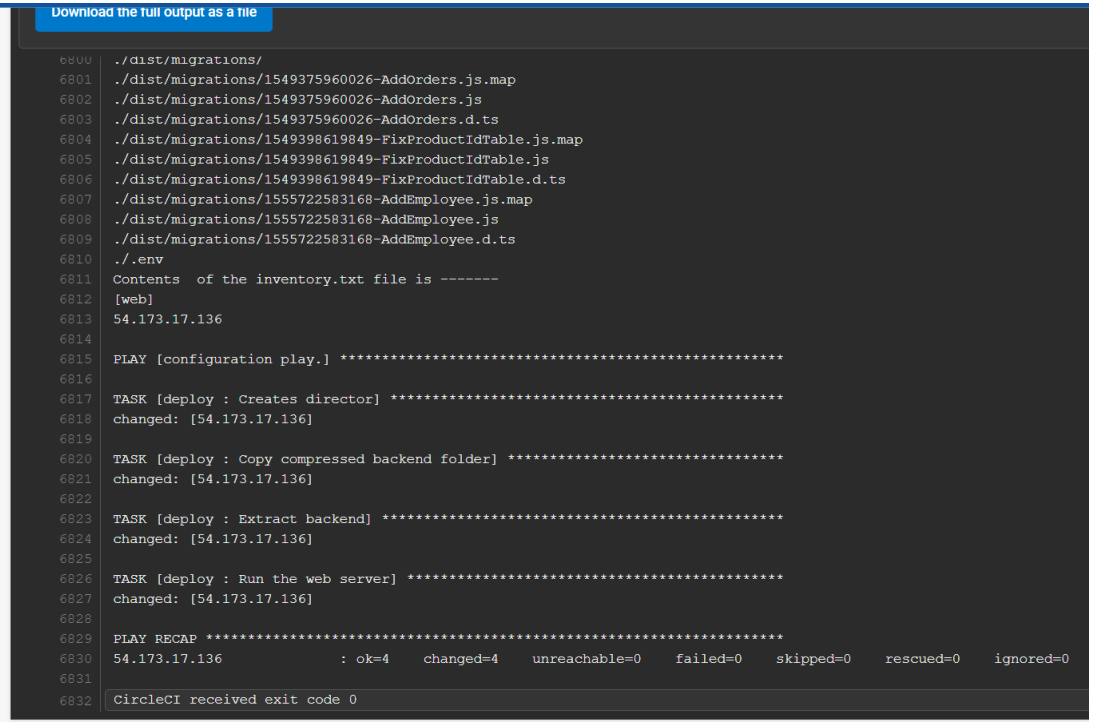




Deploy Backend



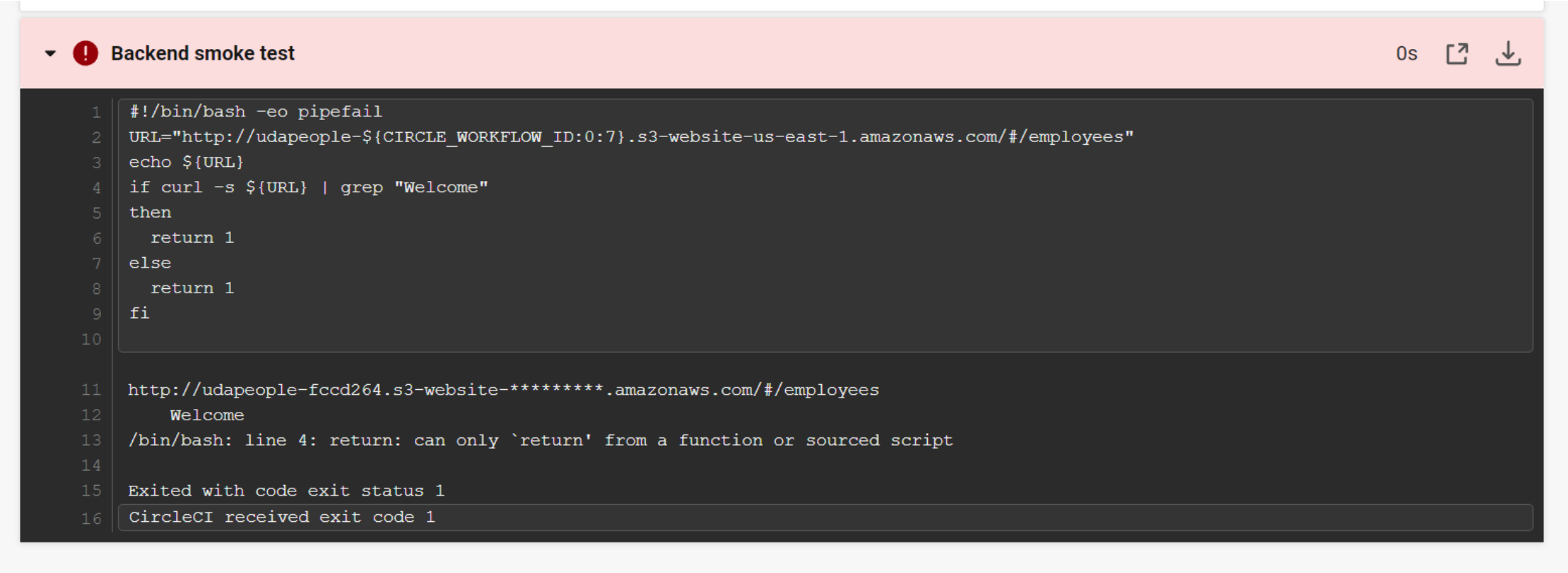




smoke-test

- Alerts

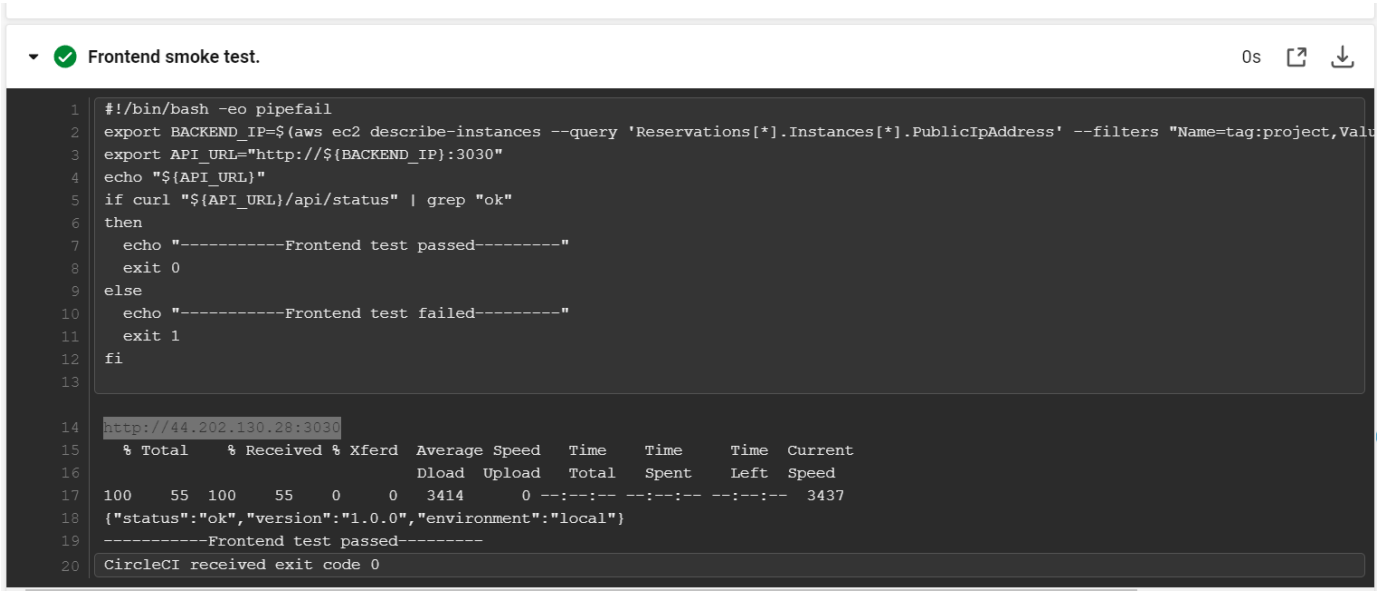
[SCREENSHOT06]



smoke-test

Fix smoke test

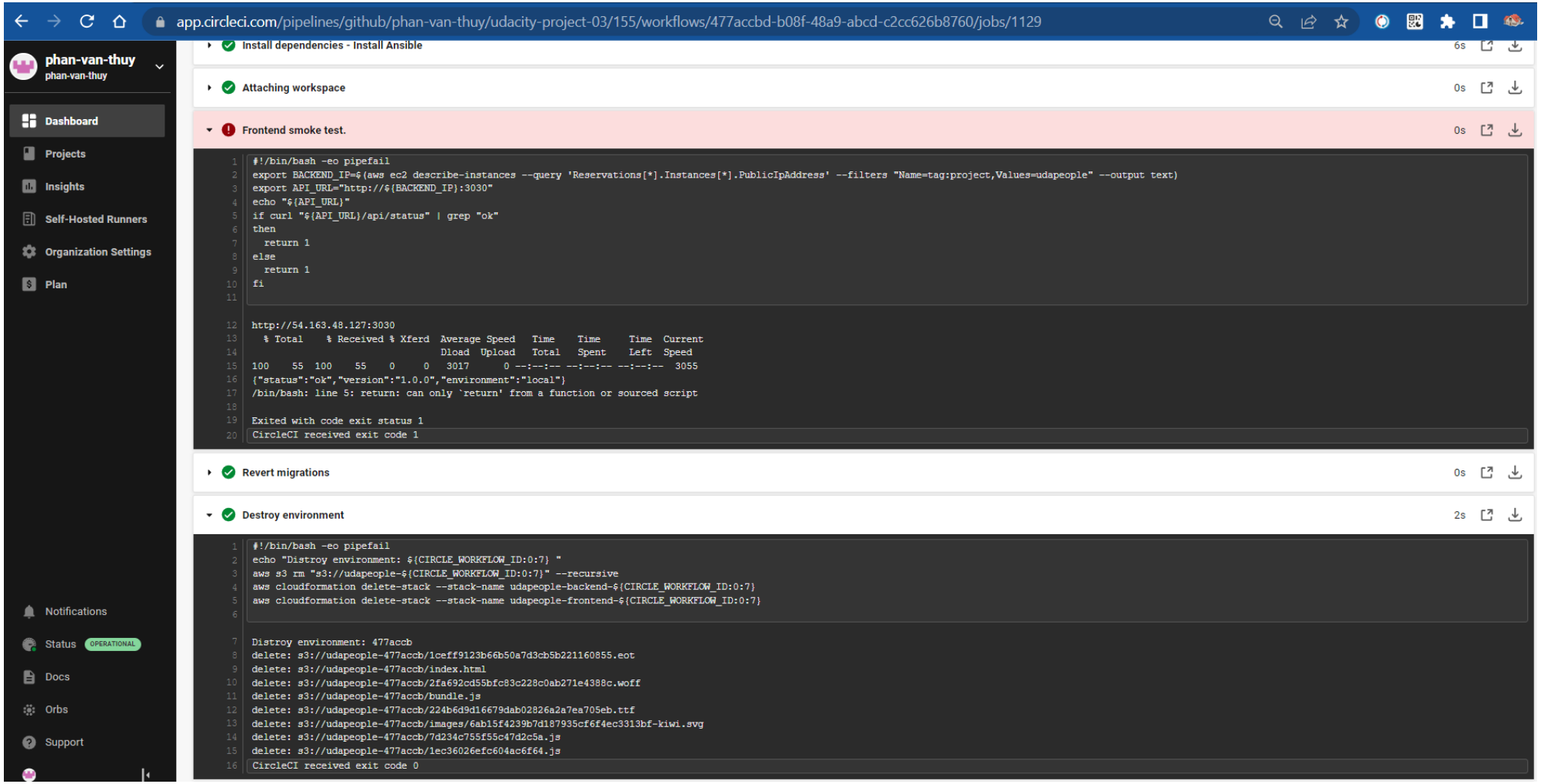
Frotend



Backend

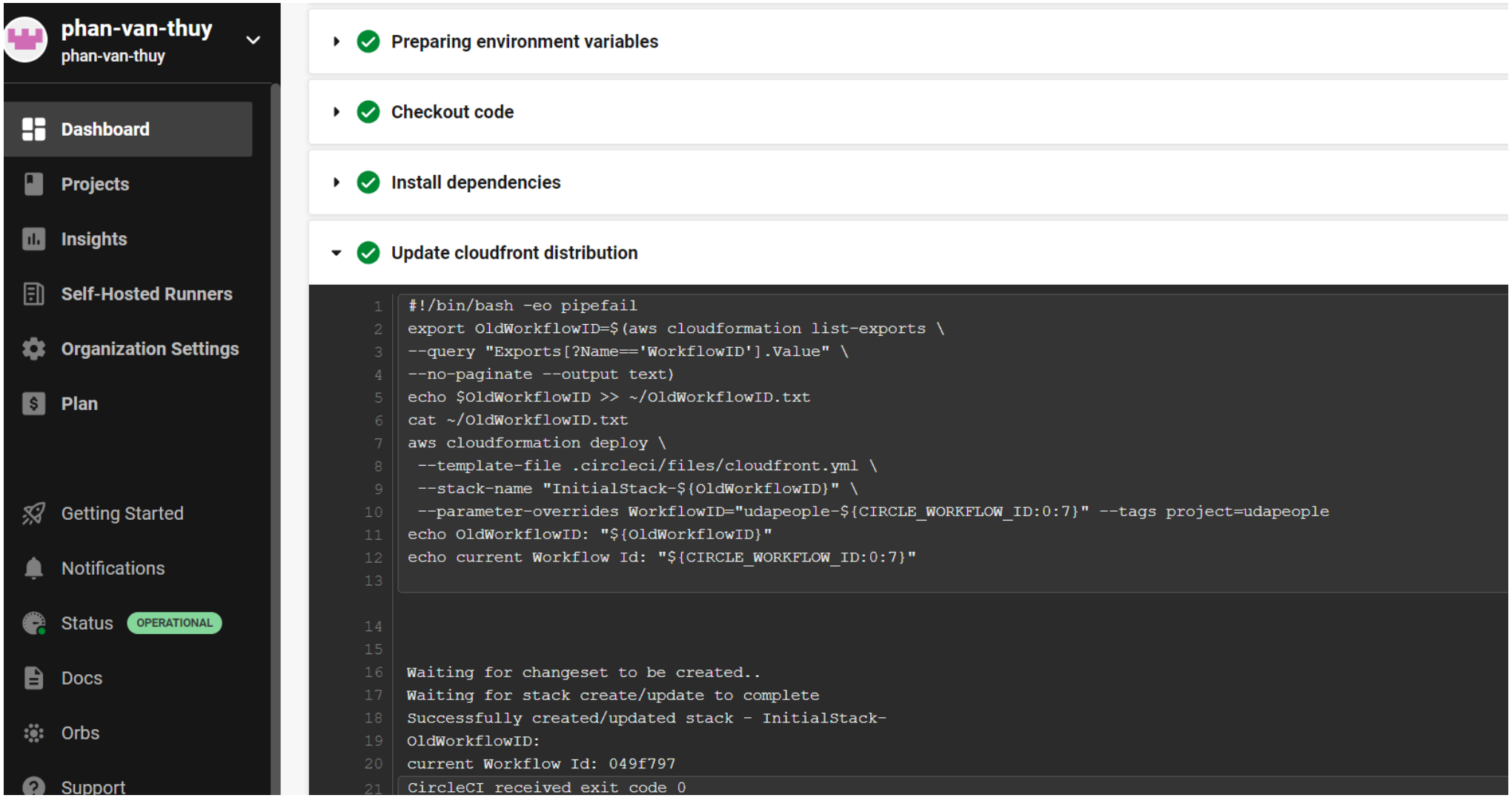
Rollback Phase

- **[SCREENSHOT07]**



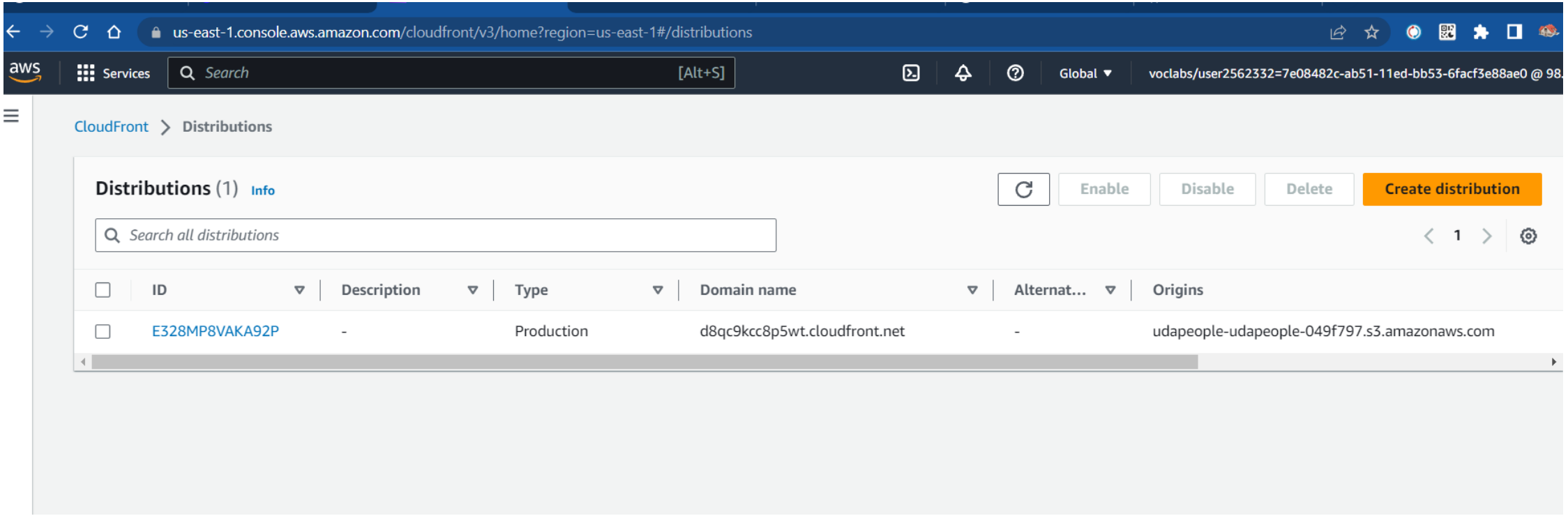
Promotion Phase

[SCREENSHOT08]



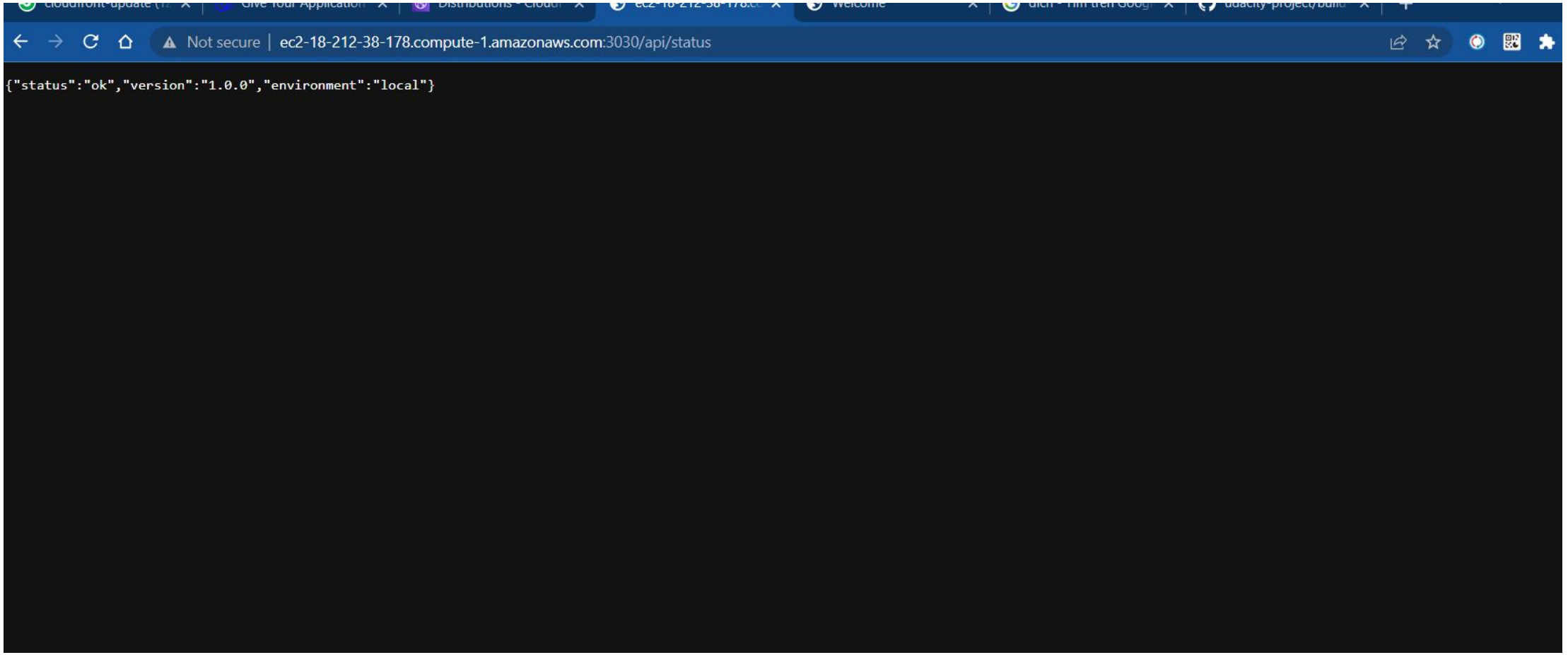
Promotion Phase

[URL03_SCREENSHOT]



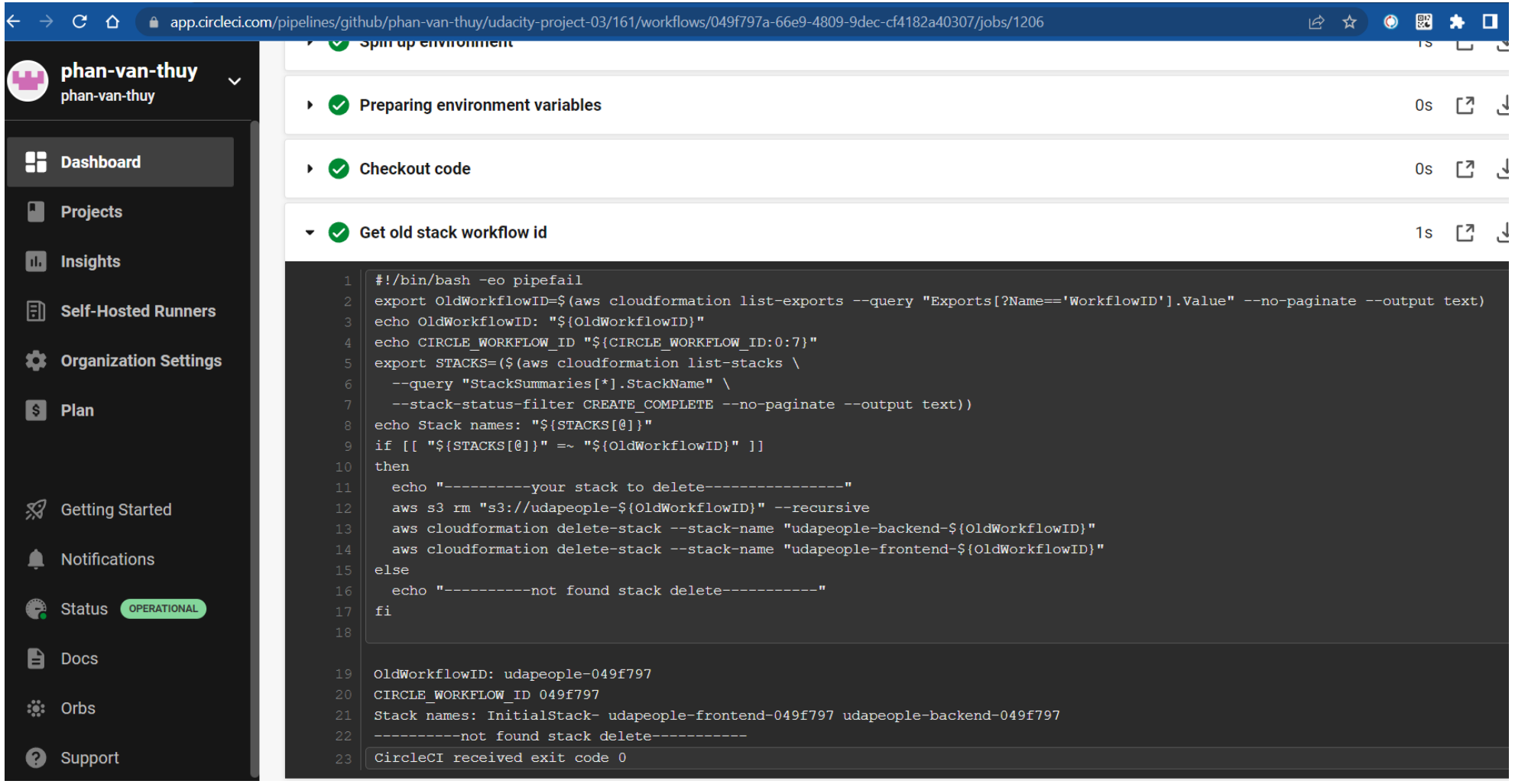
Promotion Phase

[URL04_SCREENSHOT]



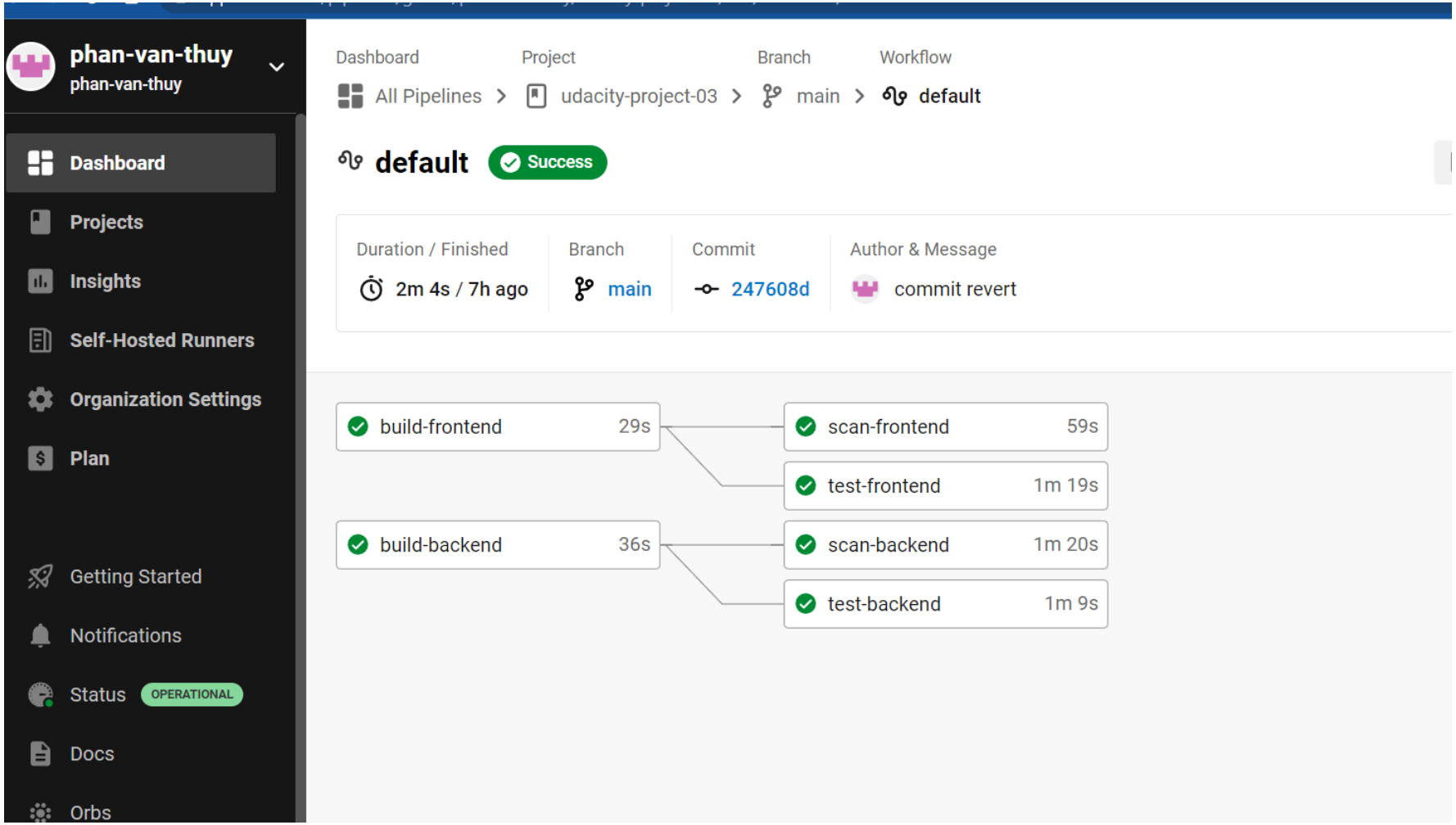
Cleanup Phase

[SCREENSHOT09]

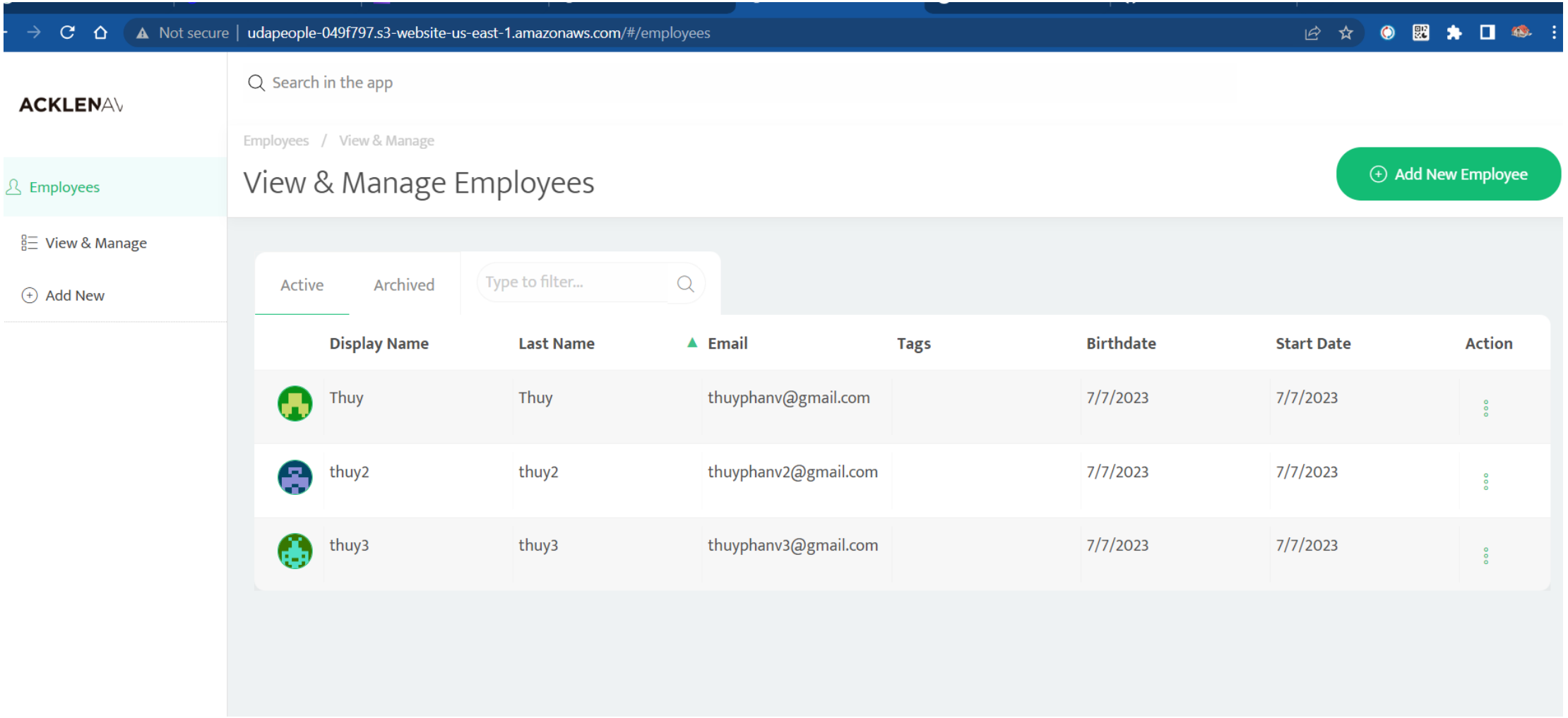


Other Considerations

- **[SCREENSHOT10]**



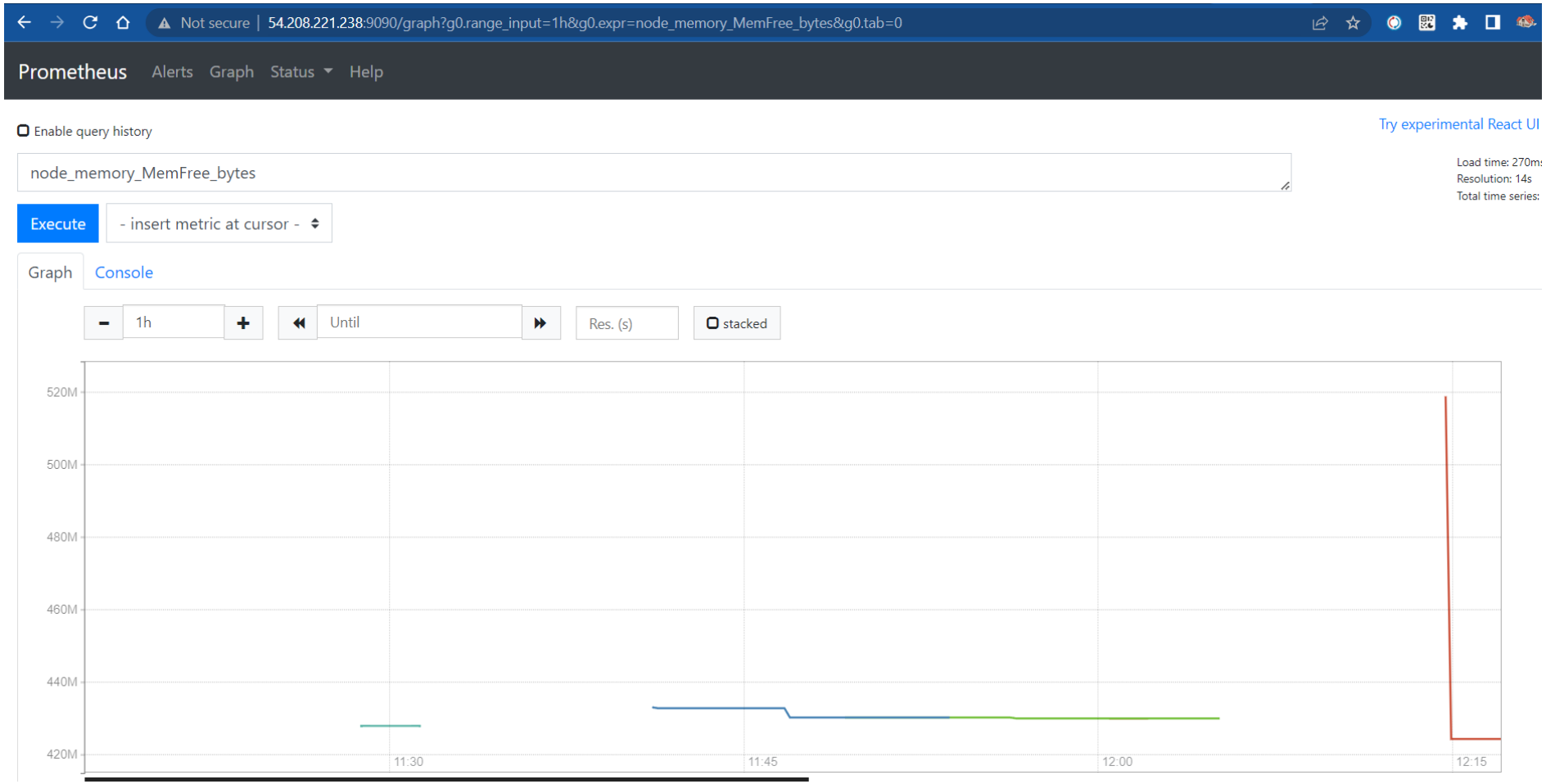
Run the application



Setup Back-End Monitoring

- Setup Back-End Monitoring

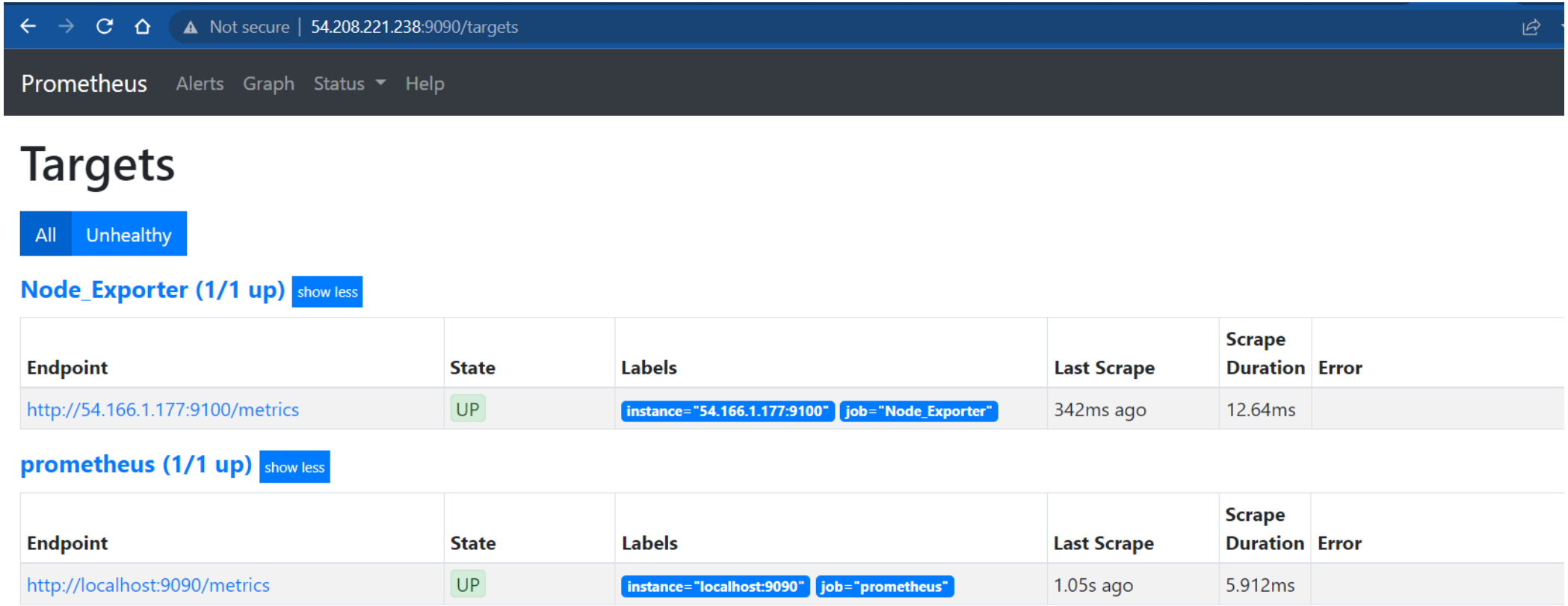
[SCREENSHOT11]



Setup Back-End Monitoring

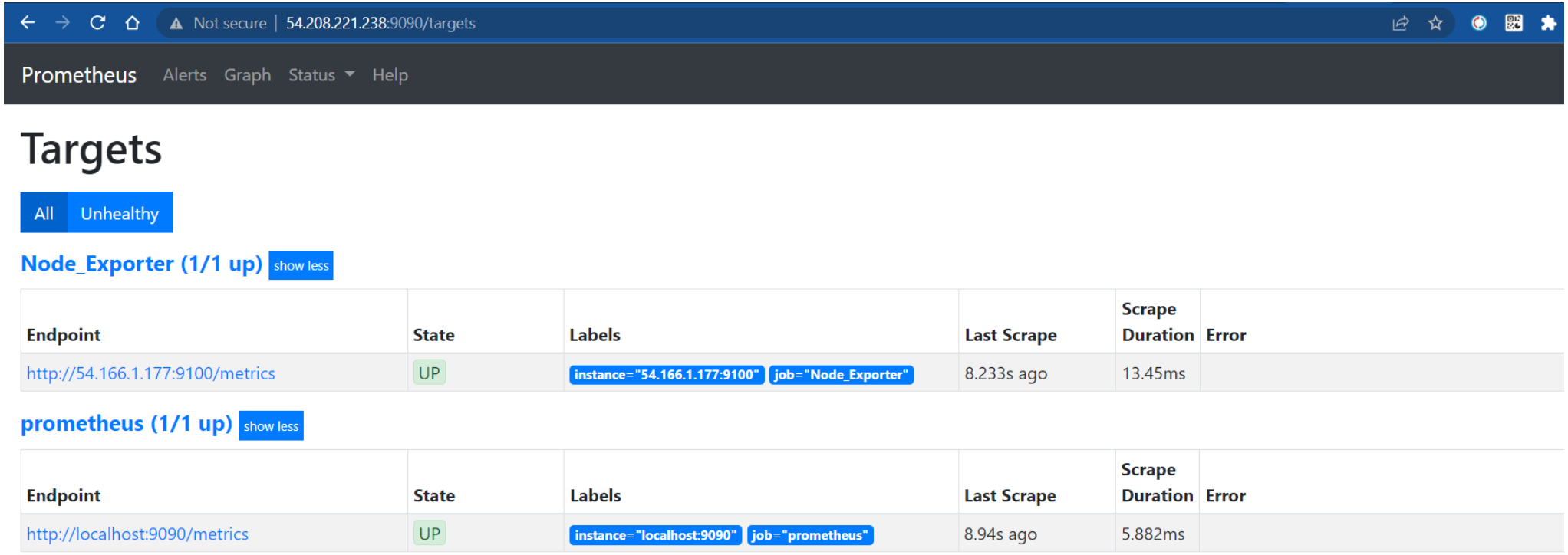
- **[URL05_SCREENSHOT]**

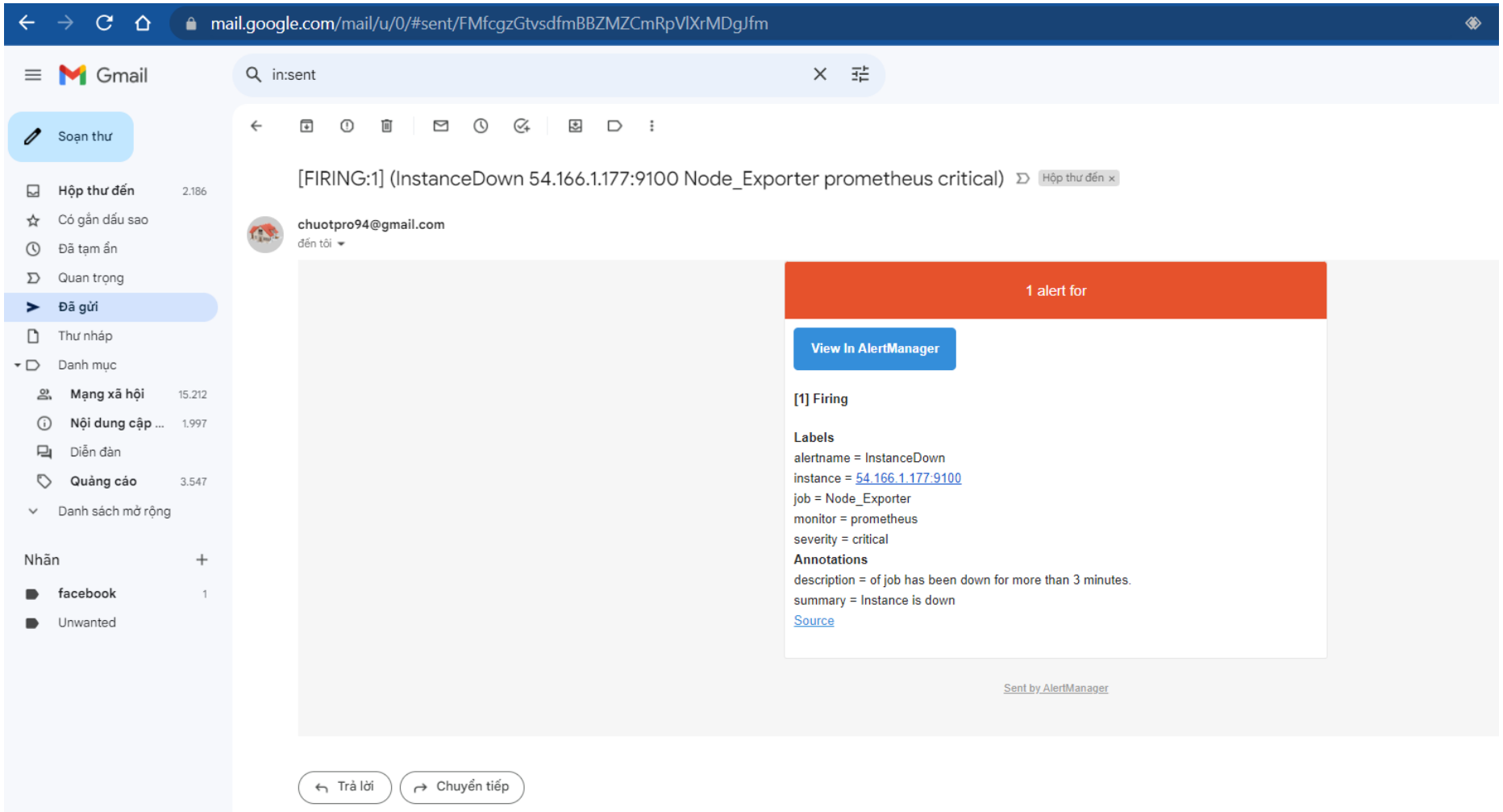
http://54.208.221.238:9090/targets



- Setup Back-End Monitoring

[URL05_SCREENSHOT]



Setup Alerts

- **[SCREENSHOT12]**