Truong Hoang Minh (FHM.GHS)

FPT Software  MinhTH6@fsoft.com.vn

AWS DevOps Capstone Project

Contents

[My Github project: 2](#_Toc103875181)

[Step 1: Propose and Scope the Project 3](#_Toc103875182)

[Step 2: Use Jenkins or Circle CI, and implement blue/green or rolling deployment. 4](#_Toc103875183)

[Step 3: Pick AWS Kubernetes as a Service, or build your own Kubernetes cluster. 5](#_Toc103875184)

[Step 4: Build your pipeline 7](#_Toc103875185)

[Step 5: Test your pipeline 14](#_Toc103875186)

# My Github project:

<https://github.com/minh1302/aws-devops-capstone>

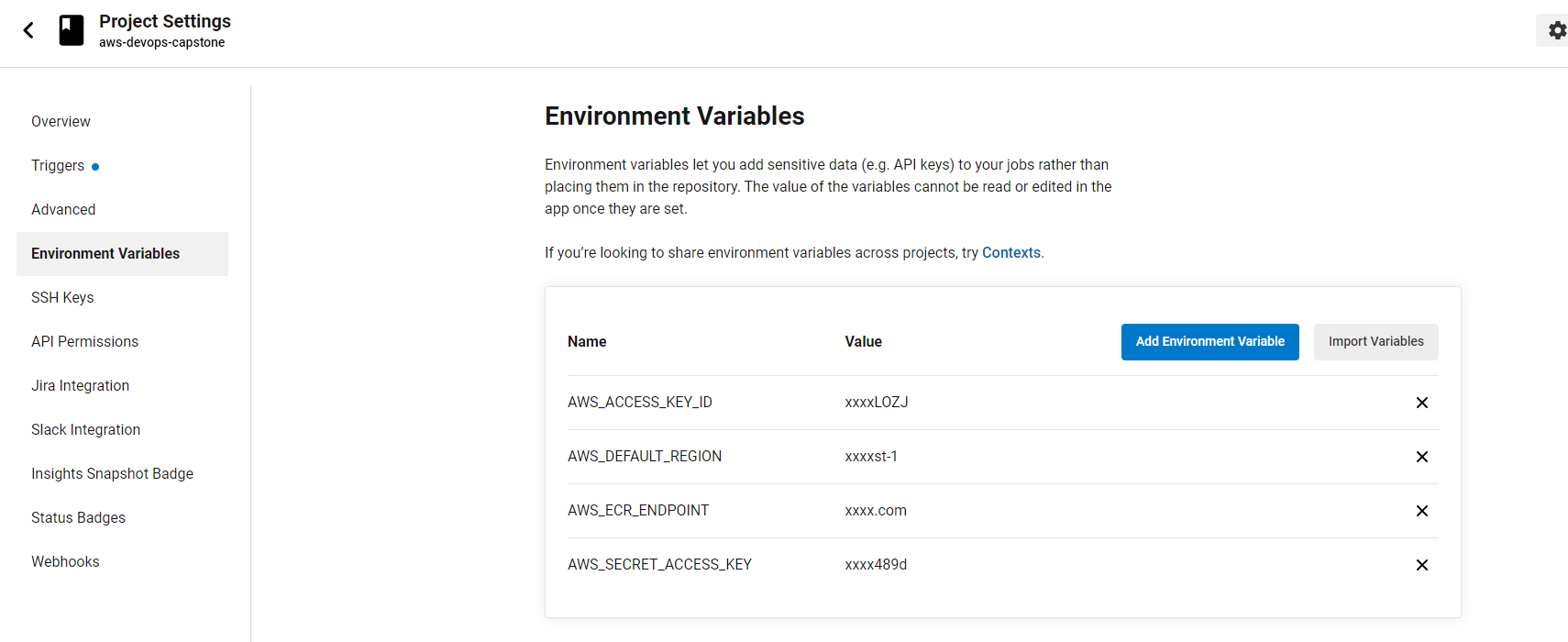
branch: master

# Step 1: Propose and Scope the Project

Plan what your pipeline will look like.

* Decide which options you will include in your Continuous Integration phase. Use either Circle CI or Jenkins.
* Circle CI
* Pick a deployment type - either rolling deployment or blue/green deployment.
* blue/green deployment
* For the Docker application you can either use an application which you come up with, or use an open-source application pulled from the Internet, or if you have no idea, you can use an Nginx “Hello World, my name is (student name)” application.
* Use Udapeople Project : <https://github.com/minh1302/udapeople>

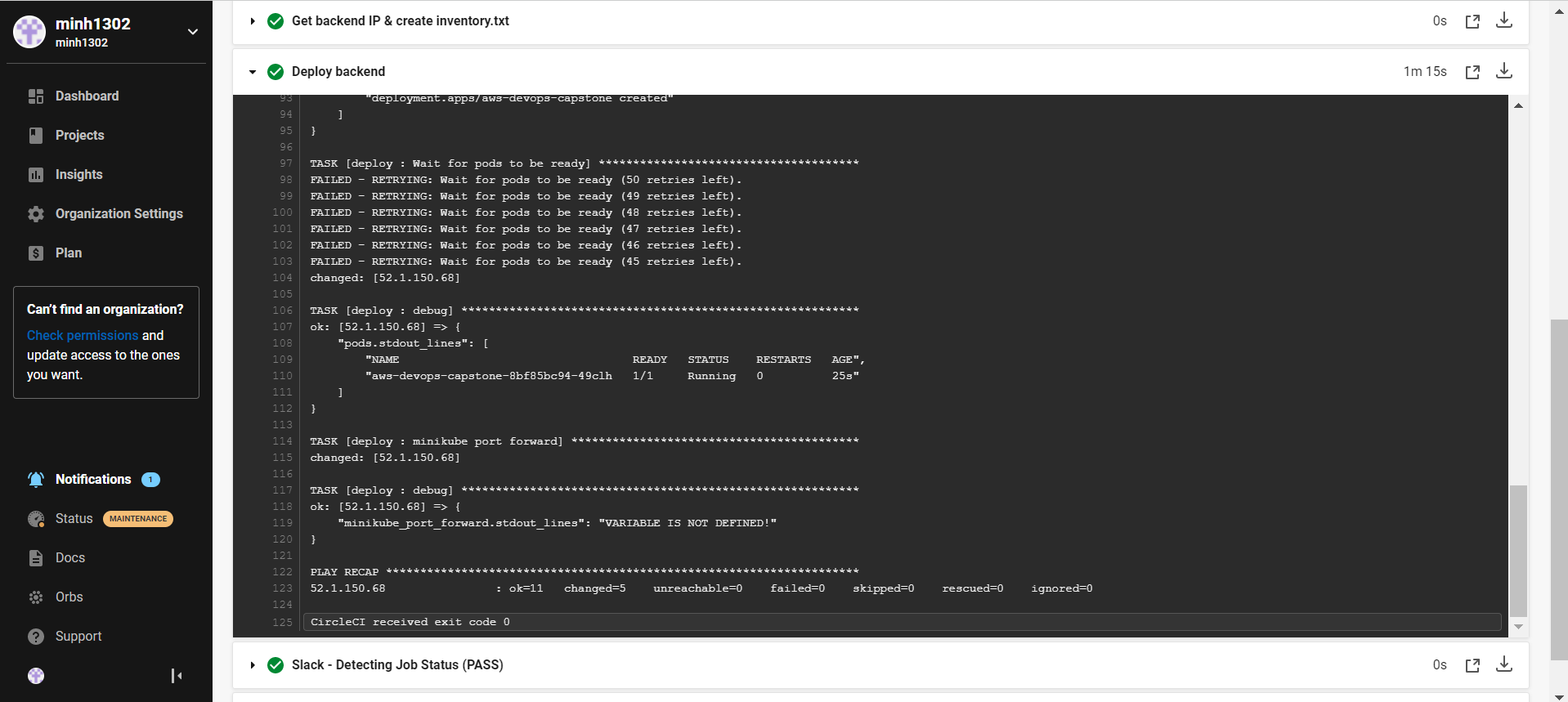
# Step 2: Use Jenkins or Circle CI, and implement blue/green or rolling deployment.

* If you're using Jenkins, create your Jenkins master box and install the plugins you will need.
* Not use Jenkins
* If you're using Circle CI, set up your circle CI account and connect your git repository.
* My Circle CI & Github ID: minh1302
* Setup CircleCI
  + SSH Key: from pem file (EC2 Key pair)
  + Context:
    - slack notification
    - Docker Hub (for local test)
  + Environment Variable:

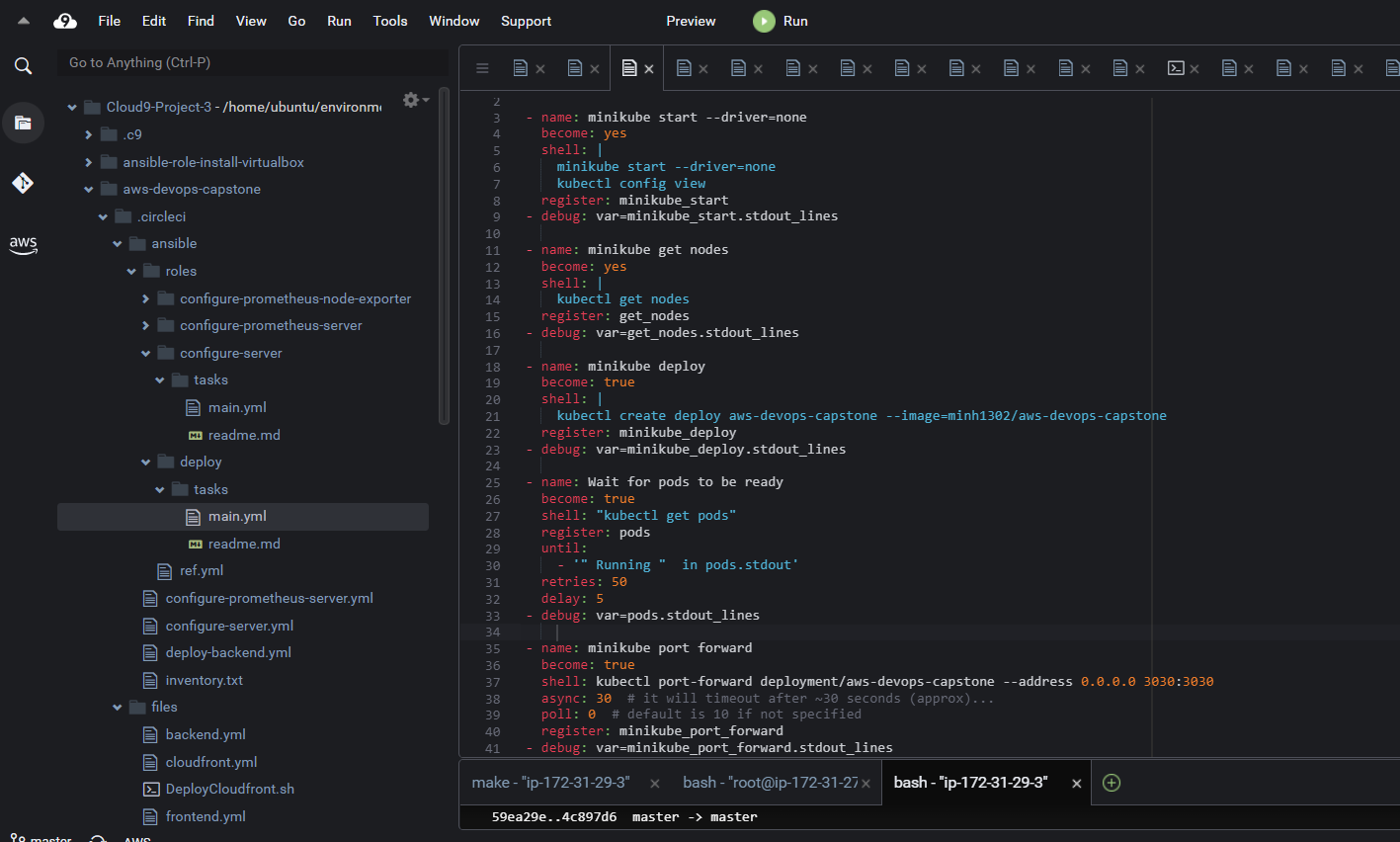
Set up your environment to which you will deploy code.

# Step 3: Pick AWS Kubernetes as a Service, or build your own Kubernetes cluster.

* Use Ansible or CloudFormation to build your “infrastructure”; i.e., the Kubernetes Cluster.
* It should create the EC2 instances (if you are building your own), set the correct networking settings, and deploy software to these instances.
* User cloudformation + ansible to build Kubernetes On EC2
* As a final step, the Kubernetes cluster will need to be initialized.

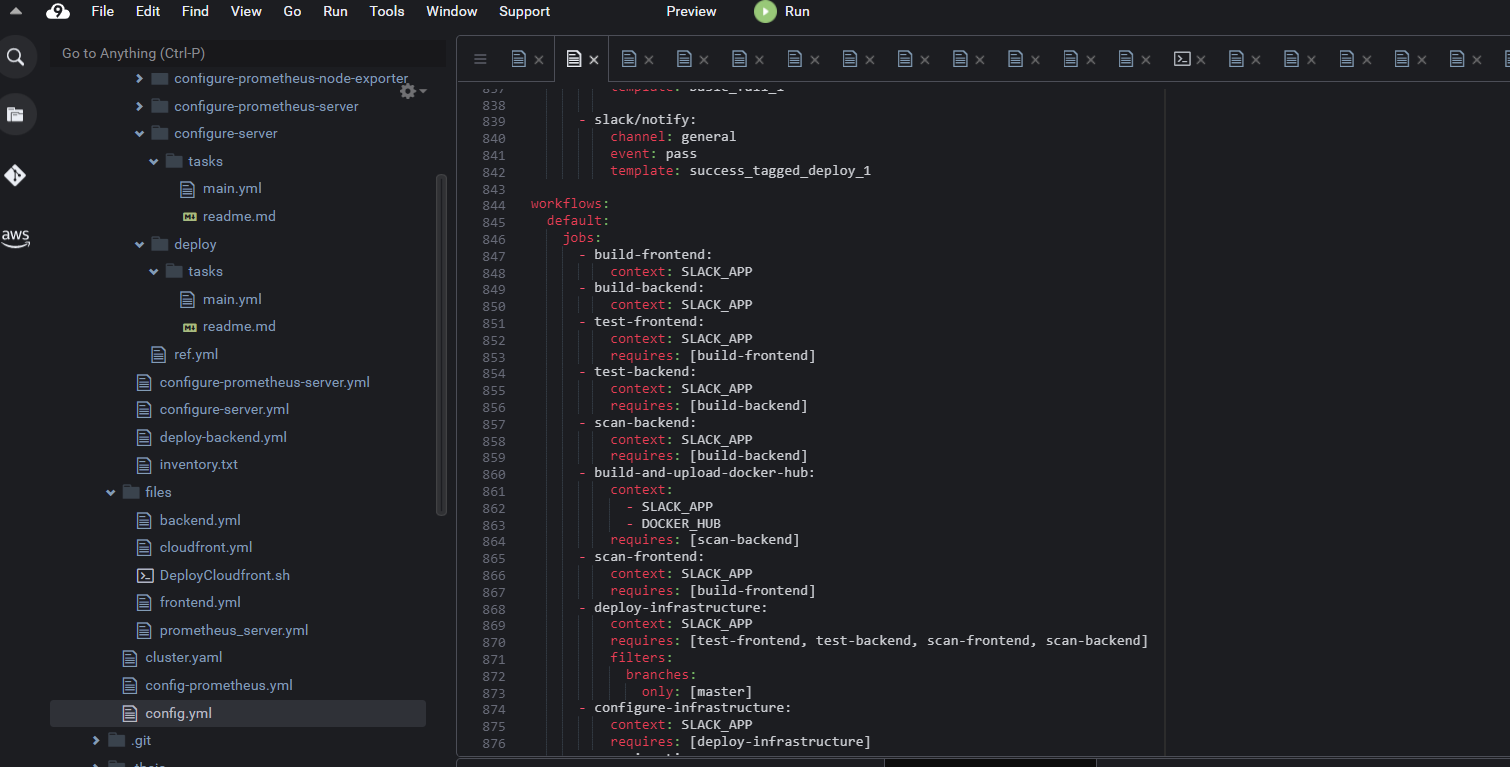


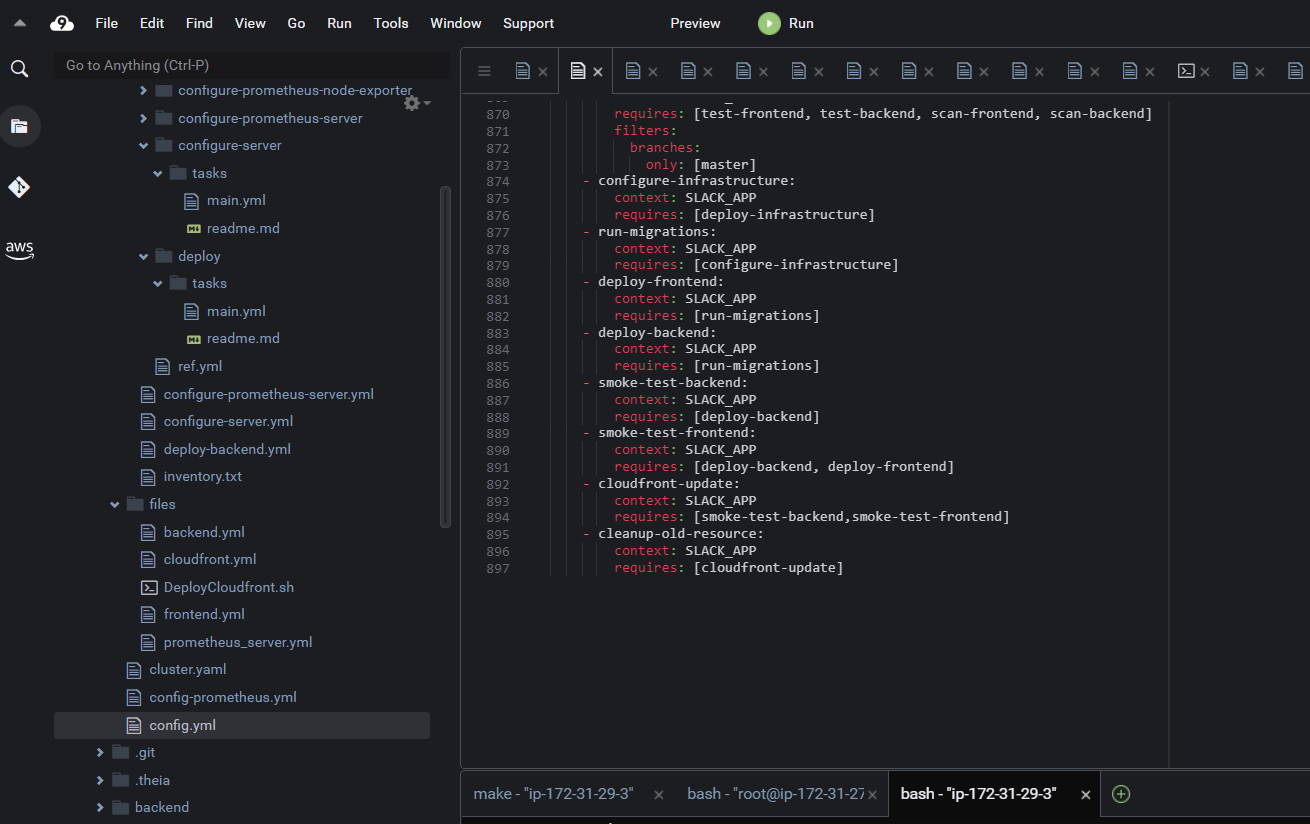
* The Kubernetes cluster initialization can either be done by hand, or with Ansible/Cloudformation at the student’s discretion.



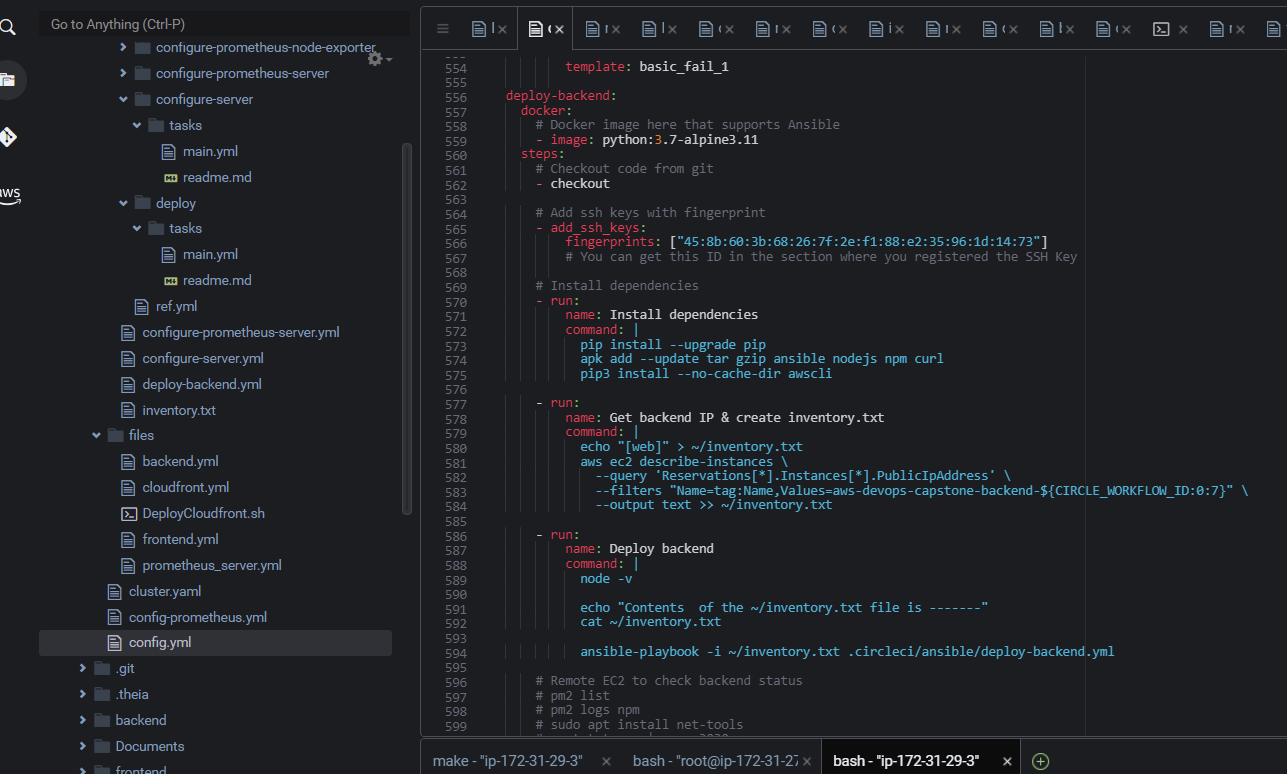
# Step 4: Build your pipeline

* Construct your pipeline in your GitHub repository.

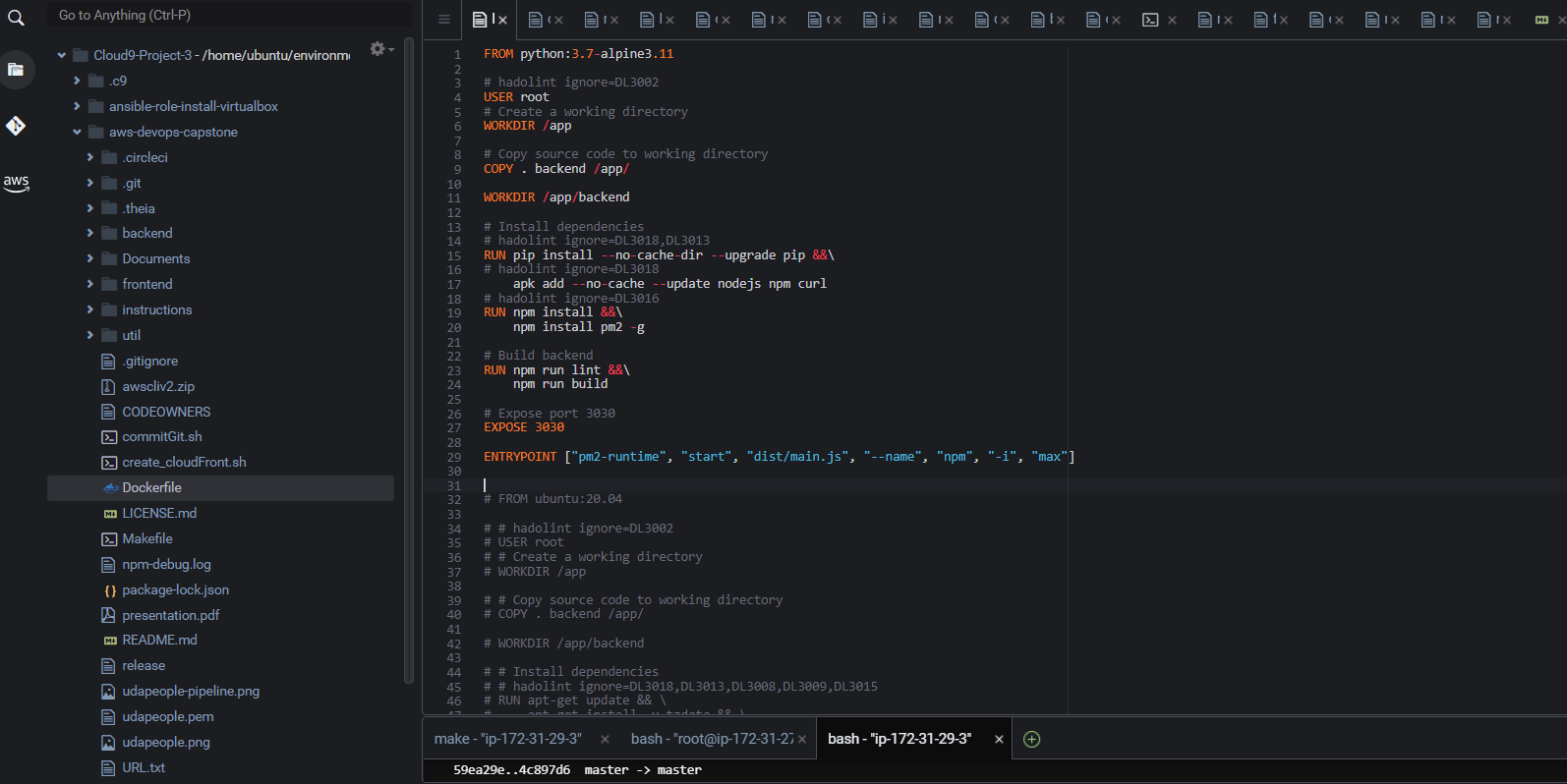




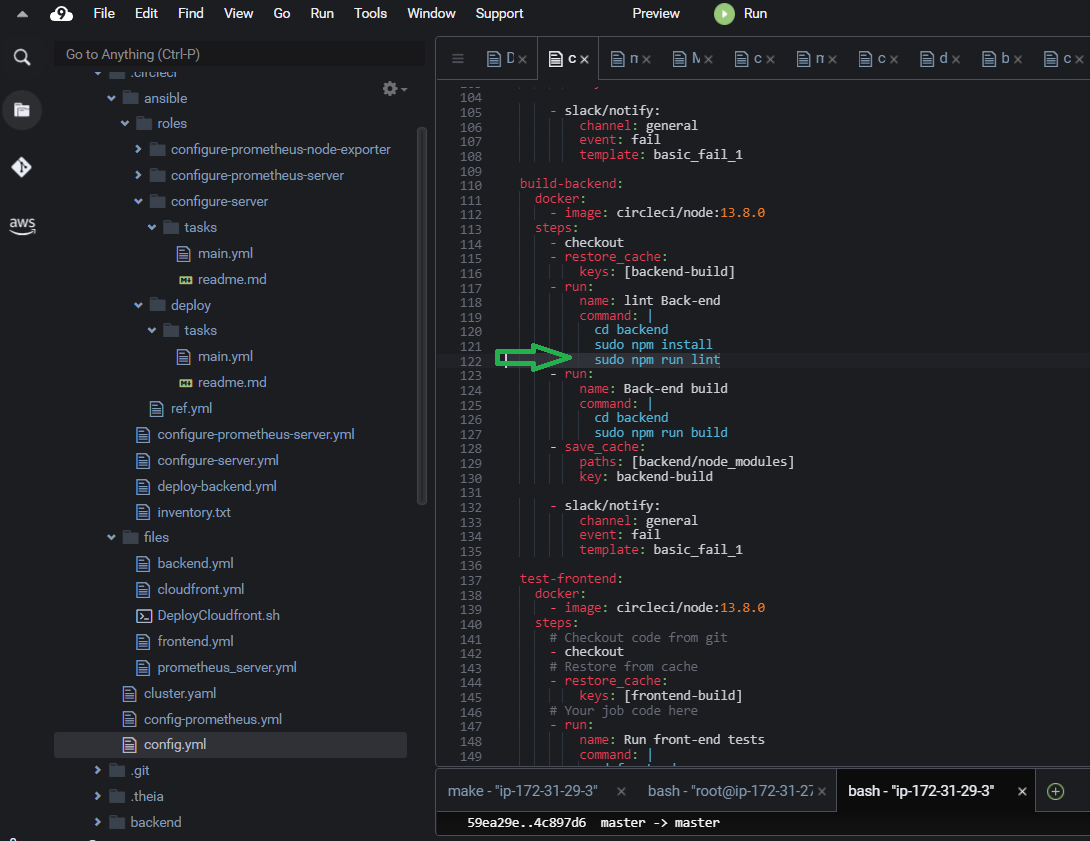
* Set up all the steps that your pipeline will include.
* Configure a deployment pipeline.

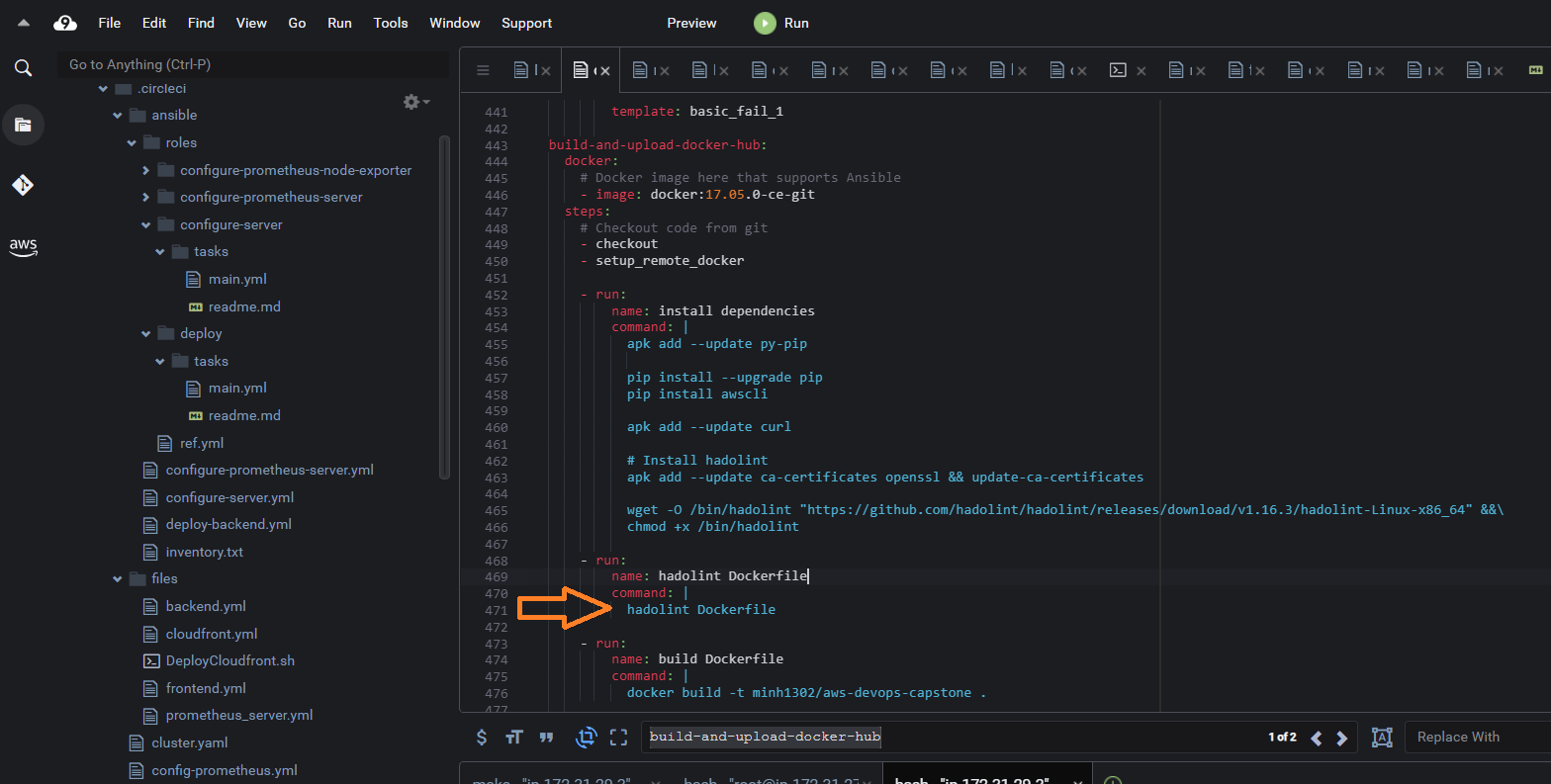


* Include your Dockerfile/source code in the Git repository.

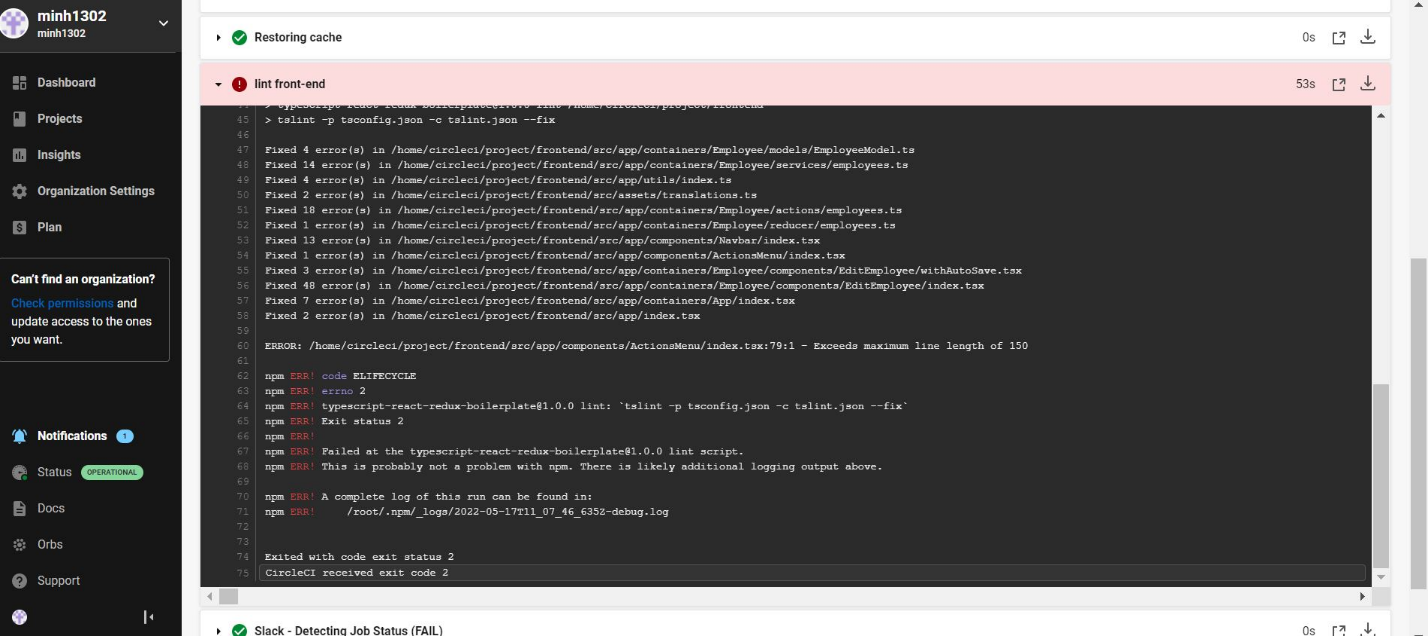


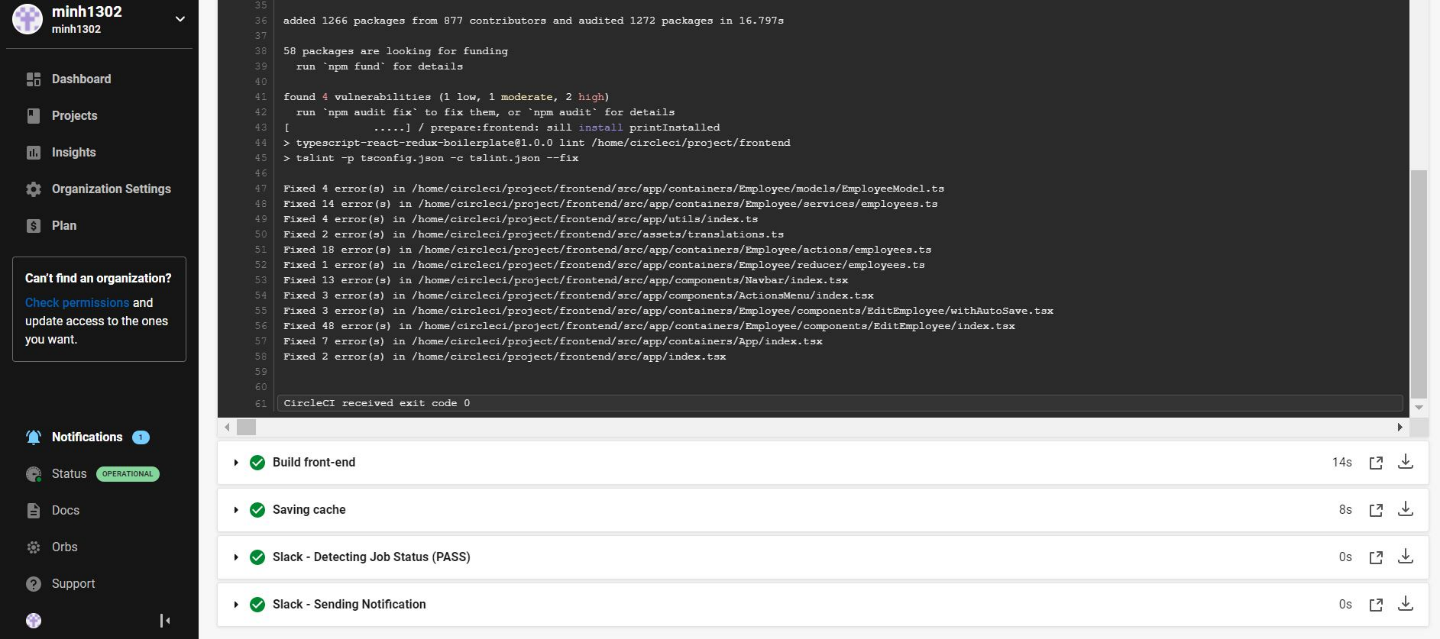
* Include with your Linting step both a failed Linting screenshot and a successful Linting screenshot to show the Linter working properly.



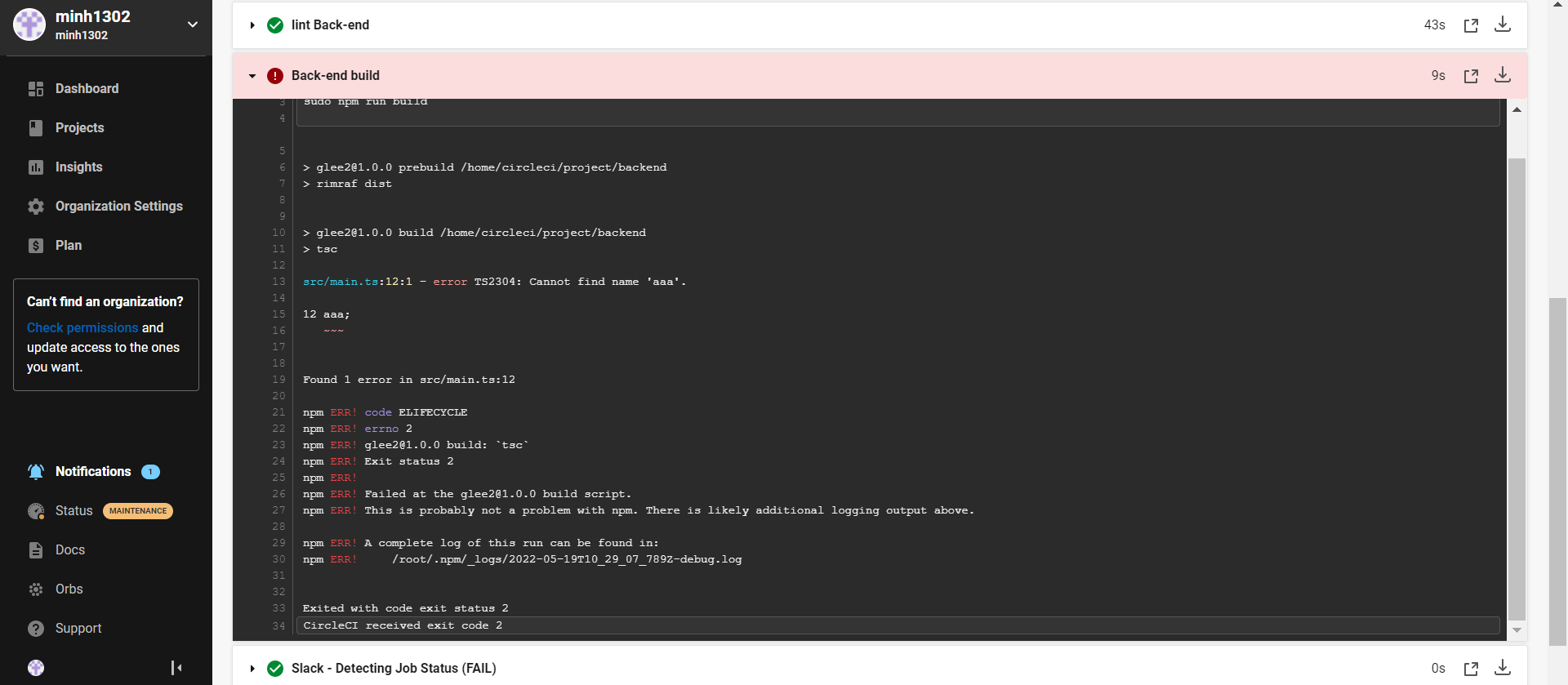


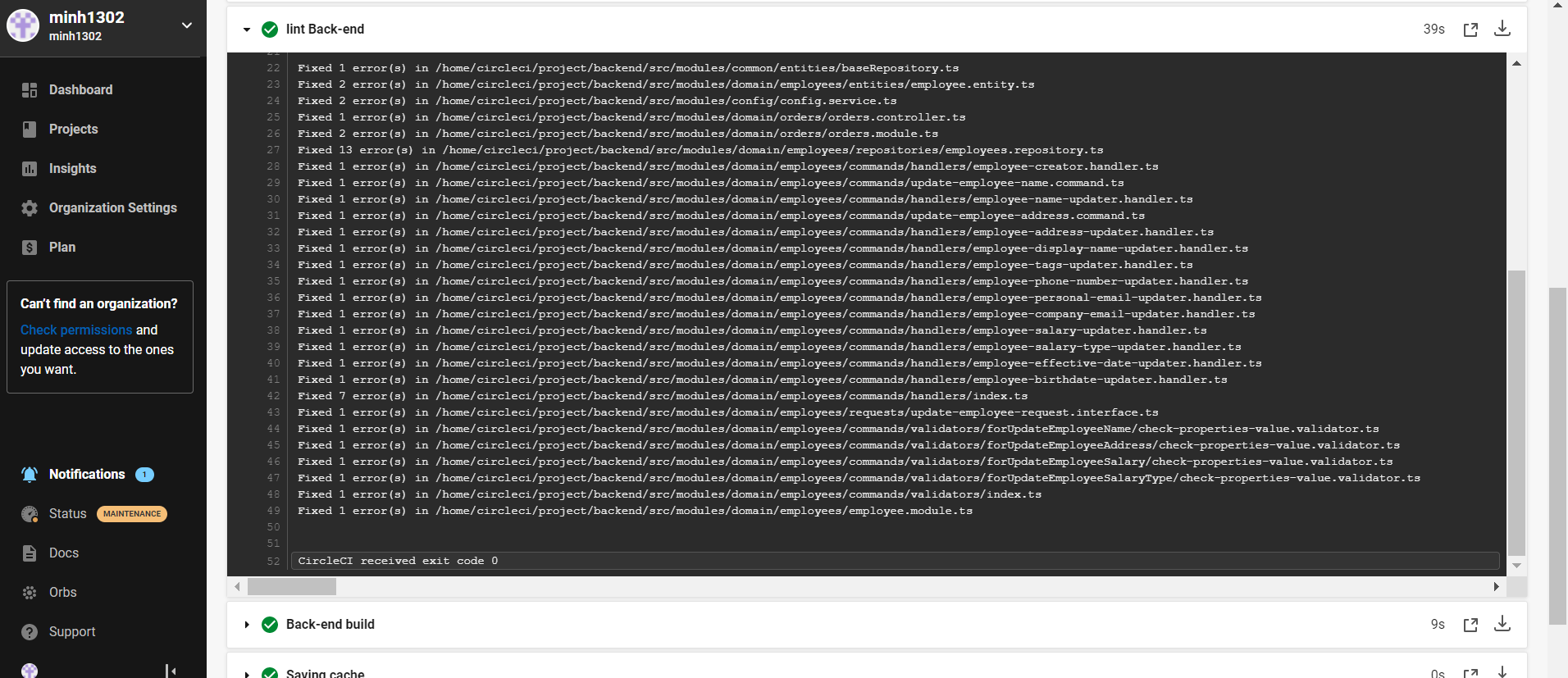
Lint frontend fail



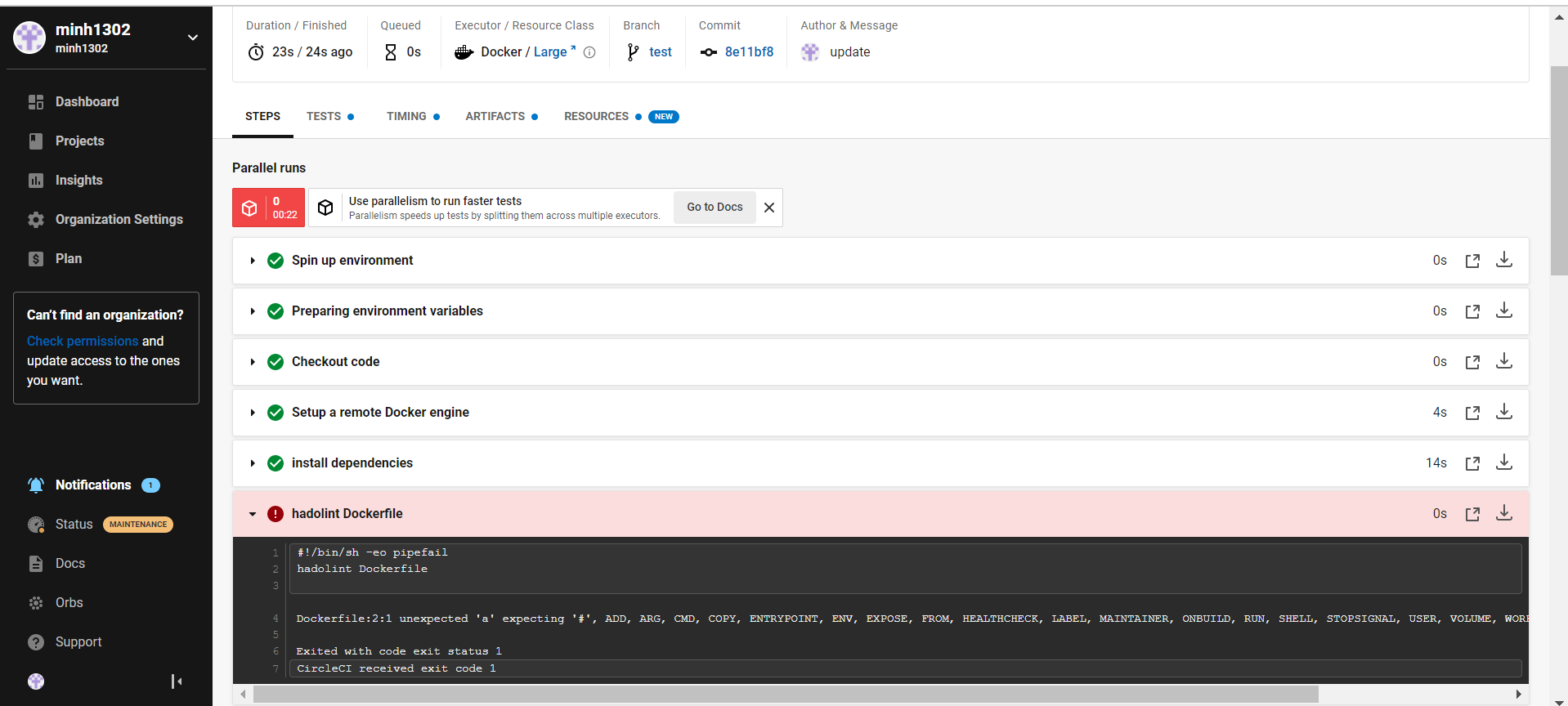
Lint frontend success

Lint backend fail

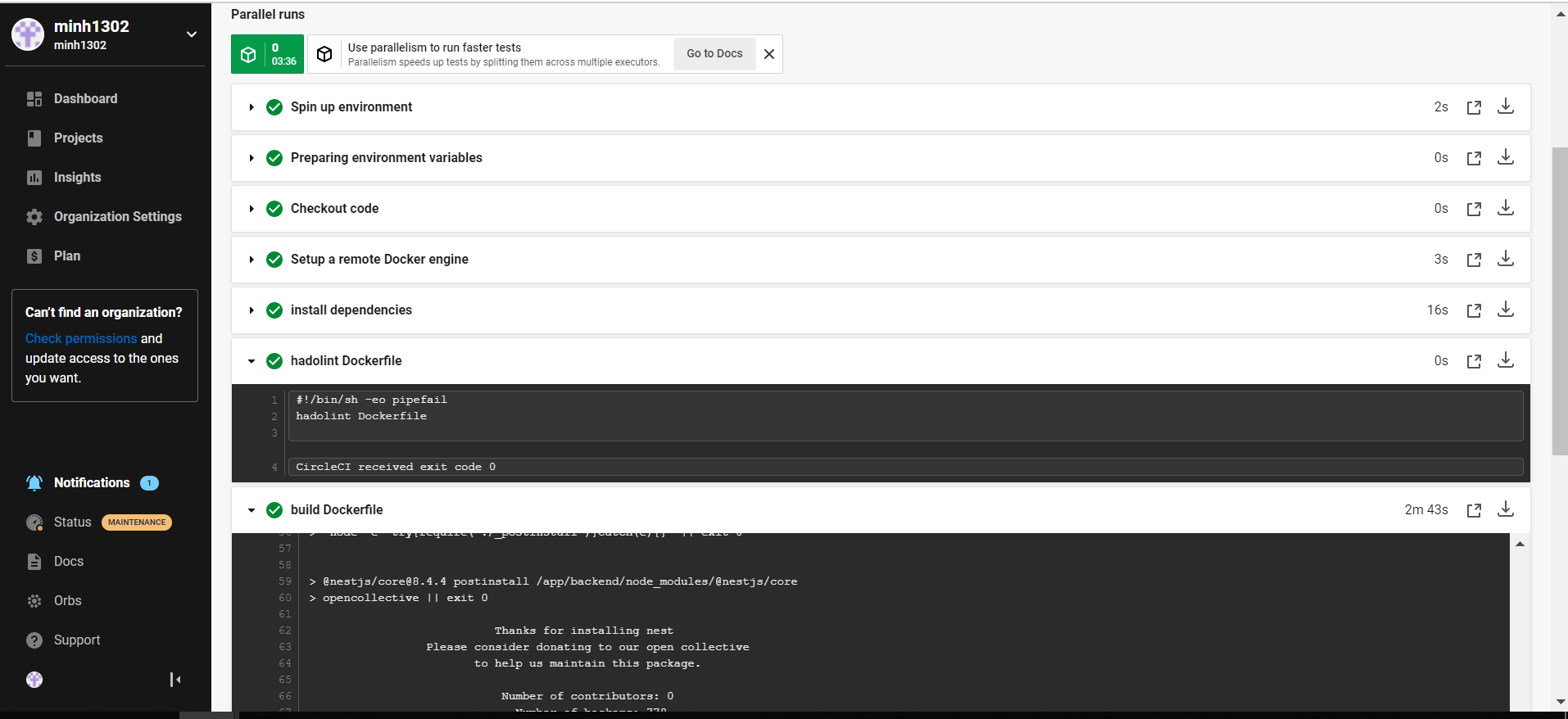


Lint backend success

Hadolint docker fail

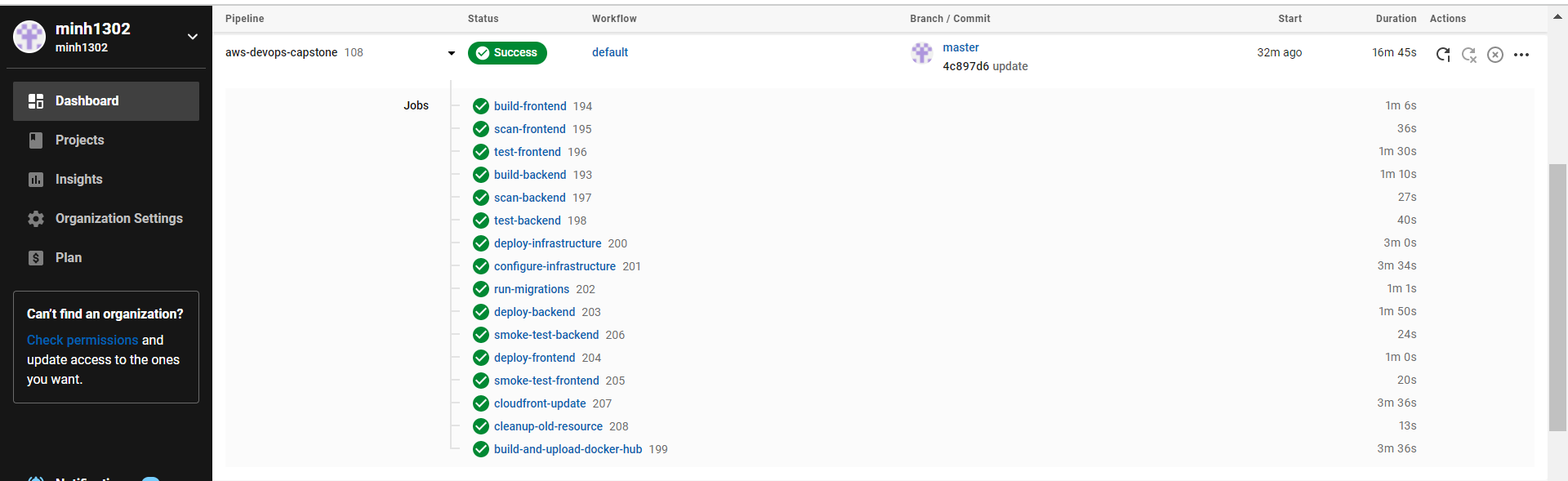


Hadolint docker success



# Step 5: Test your pipeline

* Perform builds on your pipeline.



* Verify that your pipeline works as you designed it.
* Yes
* Take a screenshot of the Circle CI or Jenkins pipeline showing deployment, and a screenshot of - your AWS EC2 page showing the newly created (for blue/green) or modified (for rolling) instances. Make sure you name your instances differently between blue and green deployments.

