

Linting failed

The screenshot shows a GitHub Actions workflow run for a project named 'bishoymaurice'. The workflow is titled 'Lint Dockerfile and app source code'. The run is currently in the 'Lint Dockerfile and app source code' step, which has failed. The failure message is: 'Makefile[12]: recipe for target 'lint' failed: make: *** [lint] Error 1'. The error occurred at line 12 of the Makefile. The workflow steps are: Preparing environment variables, Checkout code, Restoring cache, Install dependencies, Saving cache, and Lint Dockerfile and app source code. The Lint step is highlighted in red, indicating failure. The terminal output shows the installation of dependencies and the execution of the 'lint' command, which failed due to a Makefile error.

The screenshot shows a GitHub Actions workflow run for a project named 'bishoymaurice'. The workflow is titled 'run lint'. The run is currently in the 'run lint' step, which has failed. The failure message is: 'Makefile[6]: recipe for target 'lint' failed: make: *** [lint] Error 2'. The error occurred at line 6 of the Makefile. The workflow steps are: Preparing environment variables, Checkout code, Restoring cache, install dependencies, Saving cache, and run lint. The run lint step is highlighted in red, indicating failure. The terminal output shows the installation of dependencies and the execution of the 'lint' command, which failed due to a Makefile error.

Linting succeeded

 bishoymaurice
bishoy

Dashboard

Projects


Insights

Organization Settings

Plan

Status MAINTENANCE

Help



Preparing environment variables

Checkout code

Restoring cache

install dependencies

Saving cache

run lint

0s

0s


0s

15s

0s

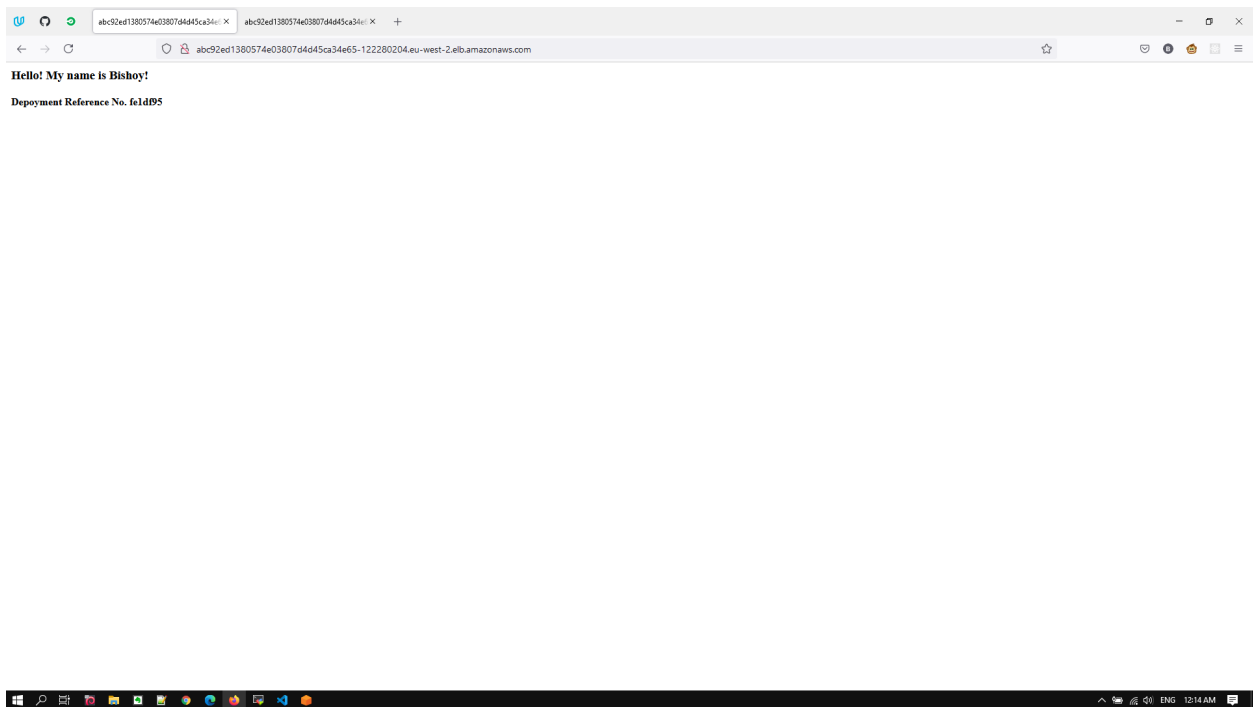
2s



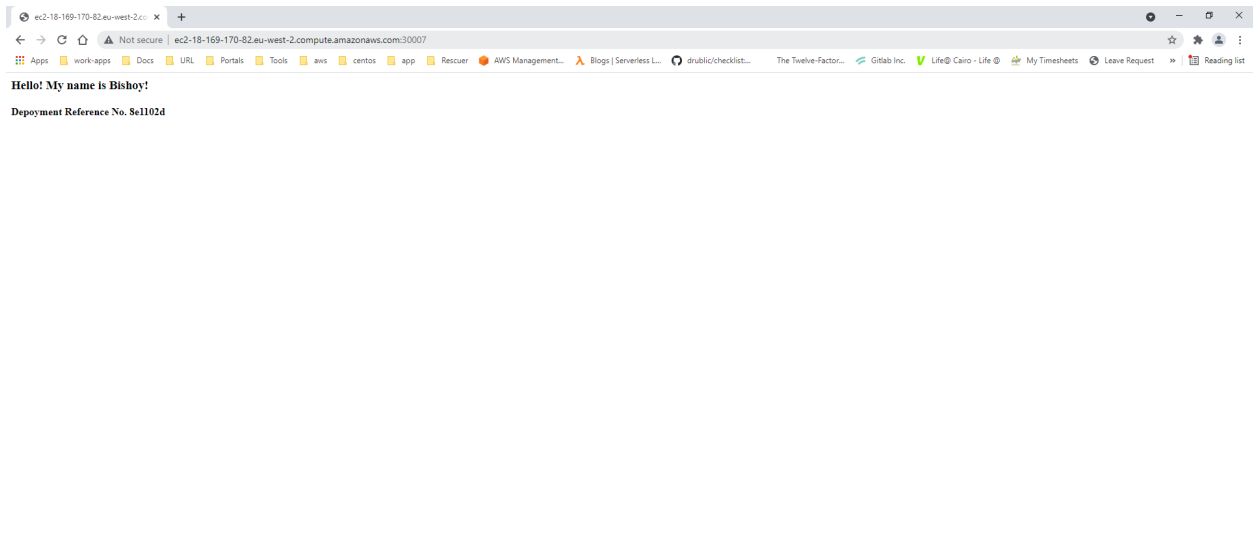


```
3 source ~/.venv/bin/activate
4 pip install pylint
5 make lint
6
7 Collecting pylint
8   Downloading pylint-2.8.3-py3-none-any.whl (357 kB)
9     | 357 kB 16.5 MB/s eta 0:00:01
10 Collecting astroid==2.8.6
11   Downloading astroid-2.8.6-py3-none-any.whl (219 kB)
12     | 219 kB 98.9 MB/s eta 0:00:01
13 Collecting mccabe<0.7,>=0.6
14   Downloading mccabe-0.6.1-py3-none-any.whl (8.6 kB)
15 Collecting toml==0.7.1
16   Downloading toml-0.10.2-py2.py3-none-any.whl (16 kB)
17 Collecting isort<4,>=4.2.5
18   Downloading isort-5.9.1-py3-none-any.whl (105 kB)
19     | 105 kB 40.9 MB/s eta 0:00:01
20 Collecting typed-ast<1.5,>=1.4.0
21   Downloading typed_ast-1.4.3-cp37-cp37m-manylinux1_x86_64.whl (743 kB)
22     | 743 kB 17.3 MB/s eta 0:00:01
23 Collecting lazy-object-proxy==1.4.0
24   Downloading lazy_object_proxy-1.6.0-cp37-cp37m-manylinux1_x86_64.whl (55 kB)
25     | 55 kB 20.9 MB/s eta 0:00:01
26 Collecting wrapt<1.13,>=1.11
27   Downloading wrapt-1.12.1-cp37-cp37m (27 kB)
28 Using legacy 'setup.py install' for wrapt, since package 'wheel' is not installed.
29 Installing collected packages: wrapt, typed-ast, lazy-object-proxy, toml, mccabe, isort, astroid, pylint
30 Running setup.py install for wrapt ... - \ done
31 Successfully installed astroid-2.8.6 isort-5.9.1 lazy-object-proxy-1.6.0 mccabe-0.6.1 pylint-2.8.3 toml-0.10.2 typed-ast-1.4.3 wrapt-1.12.1
32 hadolint Dockerfile
33 html_lint.py ../app/index.html
34 CircleCI received exit code 0
```

Green version



Blue version on EC2 instance



Workflow: AWS_DevOps/111/workflows/feld95a-db9a-469c-9111-60712d0f5717/jobs/209

- Checkout code
- Restoring cache
- Install dependencies
- Configure kubectl for AWS EKS
- Configure kubectl files
- Deploy blue instance
- Check blue instance

```
1 #!/bin/bash -eo pipefail
2 cd kube-deploy
3 make deploy-blue
4
5 kubectl apply -f ./app-deploy.yaml
6 deployment.apps/sampleapp-deployment-feld95 created
7 CircleCI received exit code 0
```

Deploy blue instance

```
1 #!/bin/bash -eo pipefail
2 cd kube-deploy
3 make check-blue
4
5 sh ./scripts/CheckBlue.sh
6 Wait 2 seconds ..
7 Wait 2 seconds ..
8 Wait 2 seconds ..
9 Wait 2 seconds ..
10 Wait 2 seconds ..
11 Wait 2 seconds ..
12 Wait 2 seconds ..
13 Wait 2 seconds ..
14 Wait 2 seconds ..
15 Wait 2 seconds ..
16 Wait 2 seconds ..
17 Wait 2 seconds ..
18 Wait 2 seconds ..
19 Wait 2 seconds ..
20 Wait 2 seconds ..
21 Wait 2 seconds ..
22 Wait 2 seconds ..
```

Hello! My name is Bishop!

Deployment Reference No. feld95

Workflow: AWS_DevOps/112/workflows/42f88669-8fdd-441f-8dc0-9d9bc7390ac1/jobs/211

STEPS TESTS ARTIFACTS

- Spin up environment
- Preparing environment variables
- Checkout code
- Restoring cache
- Install dependencies
- Configure kubectl for AWS EKS
- Configure kubectl files
- Deploy blue instance
- Check blue instance

```
1 #!/bin/bash -eo pipefail
2 cd kube-deploy
3 make deploy-blue
4
5 kubectl apply -f ./app-deploy.yaml
6 deployment.apps/sampleapp-deployment-42f8866 created
7 CircleCI received exit code 0
```

Deploy blue instance

```
1 #!/bin/bash -eo pipefail
2 cd kube-deploy
3 make check-blue
4
5 sh ./scripts/CheckBlue.sh
6 Wait 2 seconds ..
7 CircleCI received exit code 0
```

Hello! My name is Bishop!

Deployment Reference No. 42f8866

Pipeline

The screenshot shows the AWS DevOps Pipeline console for the project 'AWS_DevOps' on the 'main' branch. The interface includes a sidebar with navigation options like Dashboard, Projects, Insights, and Organization Settings. The main area displays a table of pipeline runs with columns for Pipeline ID, Status, Workflow, Branch/Commit, Start time, Duration, and Actions. The table lists five runs, with the first three being successful and the last two being canceled. Each run shows a list of jobs and their individual durations.

Pipeline	Status	Workflow	Branch / Commit	Start	Duration	Actions
AWS_DevOps 113	Success	default	main 031fa0b Add Stable Version	2m ago	1m 43s	[Refresh] [Cancel] [More]
						Jobs
						lint 212 35s
						build 213 19s
						deploy 214 38s
AWS_DevOps 112	Success	default	main e20cf0a Test	8m ago	1m 22s	[Refresh] [Cancel] [More]
						Jobs
						build 210 42s
						deploy 211 34s
AWS_DevOps 111	Success	default	main 09a8b62 Test	13m ago	1m 48s	[Refresh] [Cancel] [More]
						Jobs
						build 208 18s
						deploy 209 1m 24s
AWS_DevOps 110	Canceled	default	main bc8ff2e Test	16m ago	2m 15s	[Refresh] [Cancel] [More]
						Jobs
						lint 205 33s
						build 206 19s
						deploy 207 1m 11s
AWS_DevOps 109	Canceled	default	main 05290f6 Test	19m ago	2m 53s	[Refresh] [Cancel] [More]
						Jobs
						lint 202 30s

Lint

The screenshot shows the details of a specific pipeline run (AWS_DevOps 113) for the 'lint' job. The interface displays a list of steps in the job, including 'Preparing environment variables', 'Checkout code', 'Restoring cache', 'Install dependencies', 'Saving cache', and 'Lint Dockerfile and app source code'. The 'Lint Dockerfile and app source code' step is expanded, showing the command output. The output includes the installation of various Python packages and the execution of the 'make lint' command.

```
1 source ~/.env/bin/activate
2 pip install pylint
3 make lint
4
5
6
7 Collecting pylint
8   Downloading pylint-2.8.3-py3-none-any.whl (357 KB)
9     |#####| 257 KB 17.1 MB/s eta 0:00:01
10 Collecting toml==0.10.1
11   Downloading toml-0.10.1-py2.py3-none-any.whl (16 KB)
12 Collecting isort==4.3.4
13   Downloading isort-4.3.4-py3-none-any.whl (105 KB)
14     |#####| 105 KB 61.3 MB/s eta 0:00:01
15 Collecting astroid==2.5.6
16   Downloading astroid-2.5.6-py3-none-any.whl (219 KB)
17     |#####| 219 KB 91.8 MB/s eta 0:00:01
18 Collecting mccabe==0.7.0
19   Downloading mccabe-0.7.0-py3-none-any.whl (8.6 KB)
20 Collecting wrapt==1.12.1
21   Downloading wrapt-1.12.1.tar.gz (27 KB)
22 Collecting typed-ast==1.4.0
23   Downloading typed_ast-1.4.0-cp37-cp37m-manylinux1_x86_64.whl (749 KB)
24     |#####| 749 KB 89.4 MB/s eta 0:00:01
25 Collecting lazy-object-proxy==1.4.0
26   Downloading lazy_object_proxy-1.4.0-cp37-cp37m-manylinux1_x86_64.whl (65 KB)
27     |#####| 65 KB 28.6 MB/s eta 0:00:01
28 Using legacy 'setup.py install' for wrapt, since package 'wheel' is not installed.
29 Installing collected packages: wrapt, typed-ast, lazy-object-proxy, toml, mccabe, isort, astroid, pylint
30 Running setup.py install for wrapt ... - \ done
31 Successfully installed astroid-2.5.6 isort-4.3.4 lazy-object-proxy-1.4.0 mccabe-0.6.1 pylint-2.8.3 toml-0.10.1 typed-ast-1.4.3 wrapt-1.12.1
32
33 Makefile
34 html_lint.py ./app/index.html
35
36 CircleCI received exit code 0
```

Build

The screenshot shows the 'Build' step of a CI/CD pipeline. The left sidebar displays the user 'bishoymaurice' and navigation options: Dashboard, Projects, Insights, Organization Settings, and Plan. The main panel shows the 'Build Docker image' step with a terminal output. The terminal content includes:

```
24 hadolint Dockerfile
25 html_lint.py ./app/index.html
26
27 build:
28   docker build -t $(DOCKER_PATH) --tag=$(DOCKER_TAG) .
29
30 push:
31   echo "${DOCKERHUB_PASSWORD}" | docker login -u "${DOCKERHUB_USERNAME}" --password-stdin
32   docker build -t *****/sampleapp:7614511 --tag=7614511 .
33   Sending build context to Docker daemon  6.656kB
34   Step 1/3 : FROM nginx:1.21.0
35   1.21.0: Pulling from library/nginx
36
37   81a07f80: Pulling fs layer
38   1c9b01f5: Pulling fs layer
39   e65b9baf: Pulling fs layer
40   a2452751: Pulling fs layer
41   7f888feb: Pulling fs layer
42   Digest: sha256:f8e8bdcf064d280b0c4c78a429540c7c801e8e8c892778c0d5af1c09db
43   Status: Downloaded newer image for nginx:1.21.0
44   --> 4f380adfc1cf
45   Step 2/3 : COPY ./app/index.html /usr/share/nginx/html/index.html
46   --> 7c2d3ba3e14f
47   Step 3/3 : EXPOSE 80
48   --> Running in 83a181404712
49   --> 2d7658a5b207
50   Removing intermediate container 83a181404712
51   Successfully built 2d7658a5b207
52   Successfully tagged *****/sampleapp:7614511
53   Successfully tagged 7614511:latest
54   CircleCI received exit code 0
```

Push to docker hub

The screenshot shows the 'Push to docker hub' step of a CI/CD pipeline. The left sidebar is the same as the previous screenshot. The main panel shows the terminal output for the 'Push Docker image to docker hub' step. The terminal content includes:

```
34   Sending build context to Docker daemon  6.656kB
35   Step 1/3 : FROM nginx:1.21.0
36   1.21.0: Pulling from library/nginx
37
38   81a07f80: Pulling fs layer
39   1c9b01f5: Pulling fs layer
40   e65b9baf: Pulling fs layer
41   a2452751: Pulling fs layer
42   7f888feb: Pulling fs layer
43   Digest: sha256:f8e8bdcf064d280b0c4c78a429540c7c801e8e8c892778c0d5af1c09db
44   Status: Downloaded newer image for nginx:1.21.0
45   --> 4f380adfc1cf
46   Step 2/3 : COPY ./app/index.html /usr/share/nginx/html/index.html
47   --> 7c2d3ba3e14f
48   Step 3/3 : EXPOSE 80
49   --> Running in 83a181404712
50   --> 2d7658a5b207
51   Removing intermediate container 83a181404712
52   Successfully built 2d7658a5b207
53   Successfully tagged *****/sampleapp:7614511
54   Successfully tagged 7614511:latest
55   CircleCI received exit code 0
```

Below the terminal output, the 'Push Docker image to docker hub' step is shown with a status of '4s'. The terminal content for this step includes:

```
1  #!/bin/bash -eo pipefail
2  cd docker
3  make push
4
5  echo "*****" | docker login -u "*****" --password-stdin
6  WARNING: Your password will be stored unencrypted in /home/circleci/.docker/config.json.
7  Configure a credential helper to remove this warning. See
8  https://docs.docker.com/engine/reference/commandline/login/#credentials-store
9
10 Login Succeeded
11 docker push *****/sampleapp:7614511
12 The push refers to a repository [docker.io/*****/sampleapp]
13
14 54c463af: Preparing
15 4dc07fe7: Preparing
16 193a0cfe: Preparing
17 46c0c0d1: Preparing
18 8fcf3327: Preparing
19 679ea01f: Preparing
20 7614511: digest: sha256:90d1001f469085e94c7c2c8ddcf59e71821929048f33d4130c7fec0602cb25 size: 1777
21 CircleCI received exit code 0
```

Configure Kubectl for AWS EKS

The screenshot shows a CircleCI pipeline run for the job 'Configure Kubectl for AWS EKS'. The pipeline is part of a workflow named 'AWS_DevOps/113/workflows/7614511e-c16a-4a2b-af12-43ab7bee1e10/jobs/214'. The pipeline consists of several steps, all of which are completed successfully:

- Checkout code
- Restoring cache
- Install dependencies
- Configure kubectl for AWS EKS (66s)
- Configure kubectl files
- Deploy blue instance
- Check blue instance
- Switch load balancer to target the new app
- Destroy old version
- Saving cache

The 'Configure kubectl for AWS EKS' step shows the following commands and output:

```
1 #!/bin/bash --no pipefail
2 cd kube-deploy
3 make configure
4
5 export AWS_ACCESS_KEY_ID=*****
6 export AWS_SECRET_ACCESS_KEY=*****
7 export AWS_DEFAULT_REGION=*****
8 aws s3 ls
9 aws eks --region ***** update-kubeconfig --name eks-cluster
10 Added new context arn:aws:eks:*****:639584626680:cluster/eks-cluster to /root/.kube/config
11 kubectl get svc
12
13 NAME                                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
14 nginx-service-cluster-ip            ClusterIP            172.20.0.1       <none>            443/TCP          6b25m
15 nginx-service-loadbalancer          LoadBalancer         172.20.46.49     <none>            80/TCP           159m
16
17 kubectl get po
18
19 NAME                                READY    STATUS    RESTARTS   AGE
20 sampleapp-deployment-42f8666-74d8b97f-xrtkh 1/1      Running   0           5m42s
21
22 kubectl get rs
23
24 NAME                                DESIRED    CURRENT    READY    AGE
25 sampleapp-deployment-42f8666-74d8b97f      1          1          1        5m43s
26
27 CircleCI received exit code 0
```

Deploy blue version

The screenshot shows a CircleCI pipeline run for the job 'Deploy blue version'. The pipeline is part of a workflow named 'AWS_DevOps/113/workflows/7614511e-c16a-4a2b-af12-43ab7bee1e10/jobs/214'. The pipeline consists of several steps, all of which are completed successfully:

- Configure kubectl files
- Deploy blue instance (18s)
- Check blue instance
- Switch load balancer to target the new app
- Destroy old version
- Saving cache

The 'Deploy blue instance' step shows the following commands and output:

```
1 #!/bin/bash --no pipefail
2 cd kube-deploy
3 make deploy-blue
4
5 kubectl apply -f ./app-deploy.yaml
6 deployment.apps/sampleapp-deployment-7614511 created
7 CircleCI received exit code 0
```

Check blue version is up and running

The screenshot shows a CircleCI pipeline run for user 'bishoymaurice'. The pipeline is titled 'https://app.circleci.com/pipelines/github/bishoymaurice/AWS_DevOps/113/workflows/7614511e-c16a-4a2b-af12-43ab7bee1e10/jobs/214'. The pipeline consists of several steps, with the 'Check blue instance' step currently active. The terminal output for this step shows the following commands and results:

```
1 #!/bin/bash -eo pipefail
2 cd kube-deploy
3 make check-blue
4
5 kubectl apply -f ./app-deploy.yaml
6 deployment.apps/sampleapp-deployment-7614511 created
7 CircleCI received exit code 0
```

The 'Check blue instance' step has a duration of 4s and is marked as successful. The terminal output also shows the status of the Kubernetes cluster and the deployment:

```
11 kubectl get svc
12 NAME                                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
13 nginx-service-cluster-ip            ClusterIP            172.20.0.4       <none>            443/TCP          62s
14 nginx-service-loadbalancer          LoadBalancer        172.20.106.19    abc92ed1380574e03907d4d45ca3e65-122280204.*****.elb.amazonaws.com  80/TCP           159m
15
16 kubectl get po
17 NAME                                READY    STATUS    RESTARTS   AGE
18 sampleapp-deployment-42f8866-74d8b897f-xrtkh 1/1      Running   0           5m42s
19
20 kubectl get rs
21 NAME                                DESIRED    CURRENT    READY    AGE
22 sampleapp-deployment-42f8866-74d8b897f 1          1          1        5m43s
23 [CircleCI received exit code 0]
```

Switch load balancer

The screenshot shows a CircleCI pipeline run for user 'bishoymaurice'. The pipeline is titled 'https://app.circleci.com/pipelines/github/bishoymaurice/AWS_DevOps/113/workflows/7614511e-c16a-4a2b-af12-43ab7bee1e10/jobs/214'. The pipeline consists of several steps, with the 'Switch load balancer to target the new app' step currently active. The terminal output for this step shows the following commands and results:

```
1 #!/bin/bash -eo pipefail
2 cd kube-deploy
3 make switch-load-balancer
4
5 kubectl apply -f ./load-balancer.yaml
6 service/nginx-service-loadbalancer configured
7 CircleCI received exit code 0
```

The 'Switch load balancer to target the new app' step has a duration of 1s and is marked as successful. The terminal output also shows the status of the Kubernetes cluster and the deployment:

```
18 sampleapp-deployment-42f8866-74d8b897f-xrtkh 1/1      Running   0           5m42s
19 kubectl get rs
20 NAME                                DESIRED    CURRENT    READY    AGE
21 sampleapp-deployment-42f8866-74d8b897f 1          1          1        5m43s
22 [CircleCI received exit code 0]
```