

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 syntax error in the Dockerfile.

Preparing environment variables 0s

Checkout code 0s

Restoring cache 0s

Install dependencies 20s

Saving cache 0s

Lint Dockerfile and app source code 3s

Makefile[12]: recipe for target 'lint' failed: make: *** [lint] Error 1

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 syntax error in the Dockerfile.

Preparing environment variables 0s

Checkout code 0s

Restoring cache 0s

install dependencies 22s

Saving cache 0s

run lint 3s

Makefile[6]: recipe for target 'lint' failed: make: *** [lint] Error 2

Linting succeeded

 bishoymaurice
bishoy

Dashboard

Projects


Insights

Organization Settings

Plan


Status MAINTENANCE

Help

 |


Preparing environment variables

0s


Checkout code

0s


Restoring cache

0s

install dependencies

15s


Saving cache

0s

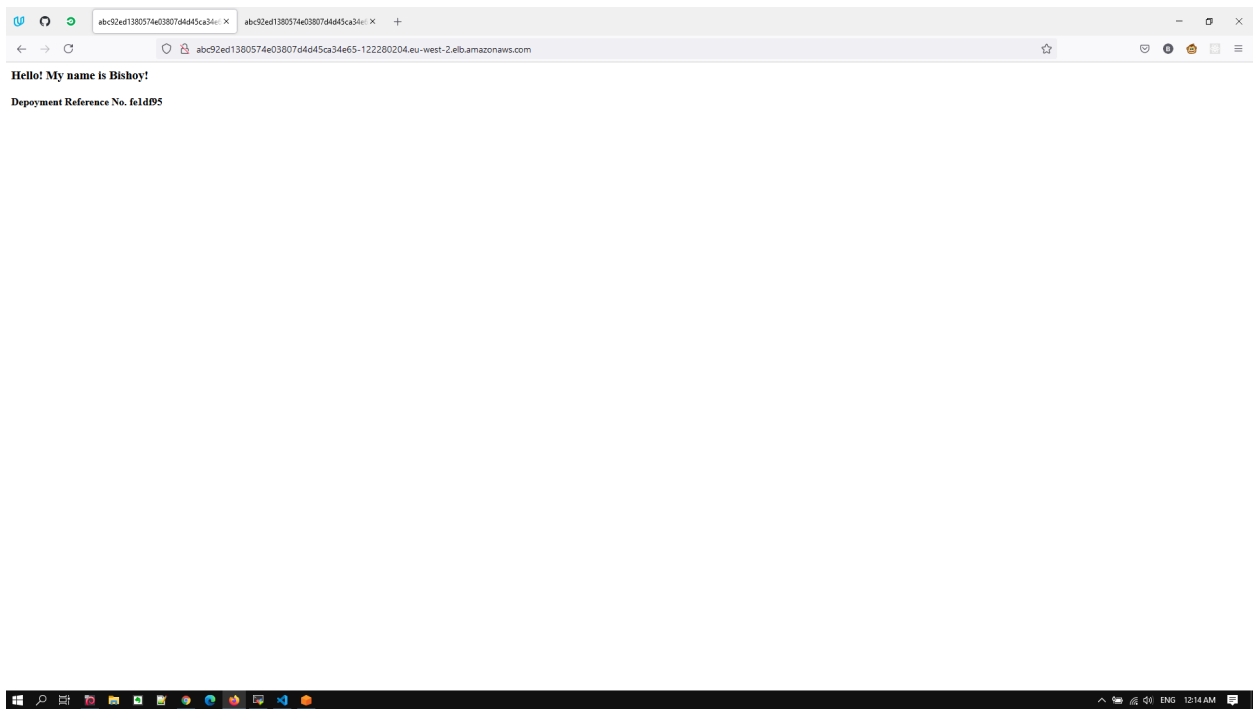
run lint

2s

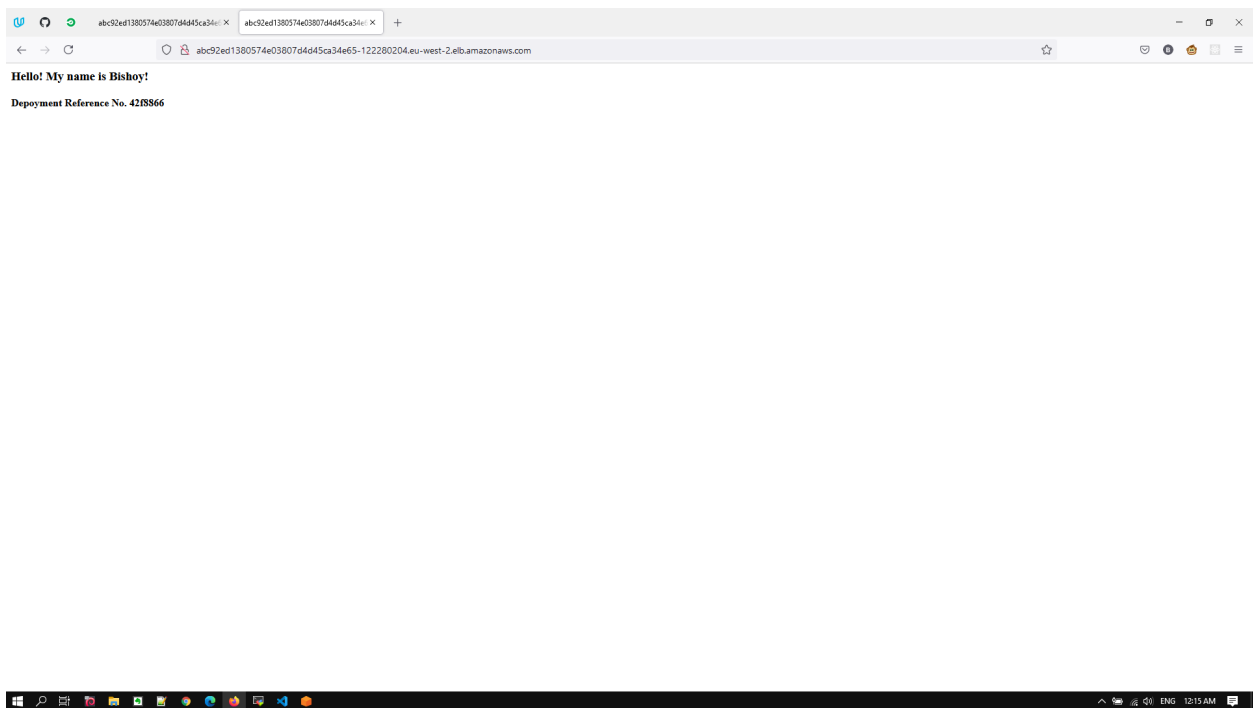
 

```
3 source ~/.venv/bin/activate
4 pip install pylint
5 make lint
6
7 Collecting pylint
8   Downloading pylint-2.8.3-py3-none-any.whl (357 kB)
9     | 357 kB 16.5 MB/s eta 0:00:01
10 Collecting astroid==2.5.6
11   Downloading astroid-2.5.6-py3-none-any.whl (219 kB)
12     | 219 kB 98.9 MB/s eta 0:00:01
13 Collecting mocabe<0.7,>=0.6
14   Downloading mocabe-0.6.1-py2.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, mocabe, isort, astroid, pylint
30 Running setup.py install for wrapt ... - \ done
31 Successfully installed astroid-2.5.6 isort-5.9.1 lazy-object-proxy-1.6.0 mocabe-0.6.1 pylint-2.8.3 toml-0.10.2 typed-ast-1.4.3 wrapt-1.12.1
32 html_lint.py ../app/index.html
33
34 CircleCI received exit code 0
```

Green version



Blue version



CircleCI Pipeline View for bishoymaurice/bishoy

Steps:

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

Deploy blue instance script:

```
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/feldf99 created
7 CircleCI received exit code 0
```

Check blue instance script:

```
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 ..
```

Output:

Hello! My name is Bishoy!

Deployment Reference No. feldf99

CircleCI Pipeline View for bishoymaurice/bishoy

Steps:

- 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
- Switch load balances to favor the new app

Deploy blue instance script:

```
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
```

Check blue instance script:

```
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
```

Output:

Hello! My name is Bishoy!

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, Status, Workflow, Branch / Commit, Start, 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==0.11.1
11   Downloading toml-0.10.2-py2.py3-none-any.whl (16 KB)
12 Collecting isort==4.3.21
13   Downloading isort-5.9.1-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-0.6.1-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-1.6.0-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-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
33 # Dockerfile
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 a list of steps: Checkout code, Restoring cache, Setup a remote Docker engine, and Build Docker image. The 'Build Docker image' step is expanded, showing a terminal window with the following commands and output:

```
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   --> 4f380adfc10f
45   Step 2/3 : COPY ./app/index.html /usr/share/nginx/html/index.html
46   --> 7c2d3b8e44
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 in the previous screenshot. The main panel shows the 'Push Docker image to docker hub' step expanded, displaying a terminal window with the following commands and output:

```
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   --> 4f380adfc10f
46   Step 2/3 : COPY ./app/index.html /usr/share/nginx/html/index.html
47   --> 7c2d3b8e44
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
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

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 job is currently running and has a duration of 66 seconds. The terminal output shows the following steps:

- Checkout code
- Restoring cache
- Install dependencies
- Configure kubectl for AWS EKS

The terminal output for the 'Configure kubectl for AWS EKS' step is as follows:

```
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       172.20.48.49     <none>            80/TCP           159m
16 kubectl get po
17
18 NAME                                READY    STATUS    RESTARTS   AGE
19 sampleapp-deployment-42f8666-74d8b97f-xrtkh  1/1      Running   0           5m42s
20 kubectl get rs
21
22 NAME                                DESIRED    CURRENT    READY    AGE
23 sampleapp-deployment-42f8666-74d8b97f      1          1          1        5m43s
24 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 job is currently running and has a duration of 18 seconds. The terminal output shows the following steps:

- Configure kubectl files
- Deploy blue instance

The terminal output for the 'Deploy blue instance' step is as follows:

```
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 left sidebar shows the user's profile and navigation links: Dashboard, Projects, Insights, Organization Settings, and Plan. The main area displays the pipeline steps and their execution details.

Steps and Execution Details:

- Configure kubectl files** (0s):
 - 8 `aws s3 ls`
 - 9 `aws eks --region ***** update-kubeconfig --name eks-cluster`
 - 10 `Added new context arn:aws:eks:*****:639284526680:cluster/eks-cluster to /root/.kube/config`
 - 11 `kubectl get svc`
- Deploy blue instance** (1s):
 - 12 `kubectl get svc`
 - 13 `kubectl get po`
 - 14 `NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE`
 - 15 `nginx-service-cluster-ip ClusterIP 172.20.45.69 <none> 443/TCP 62s`
 - 16 `nginx-service-loadbalancer LoadBalancer 172.20.106.19 abc92ed1380574e03907d4d45ca3e65-122280204-*****.elb.amazonaws.com 80:32747/TCP 160m`
 - 17 `kubectl get po`
 - 18 `NAME READY STATUS RESTARTS AGE`
 - 19 `sampleapp-deployment-42f8866-74d8b97f-xrtkh 1/1 Running 0 5m42s`
 - 20 `kubectl get rs`
 - 21 `NAME DESIRED CURRENT READY AGE`
 - 22 `sampleapp-deployment-42f8866-74d8b97f 1 1 1 5m42s`
 - 23 `CircleCI received exit code 0`
- Check blue instance** (4s):
 - 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`
- Switch load balancer to target the new app** (1s):
 - 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`
- Destroy old version** (3s):
 - 1 `#!/bin/bash -eo pipefail`
 - 2 `cd kube-deploy`
 - 3 `make destroy-old-version`
 - 4
 - 5 `kubectl apply -f ./load-balancer.yaml`
 - 6 `service/nginx-service-loadbalancer configured`
 - 7 `CircleCI received exit code 0`
- Saving cache** (0s):
 - 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`

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 left sidebar shows the user's profile and navigation links: Dashboard, Projects, Insights, Organization Settings, and Plan. The main area displays the pipeline steps and their execution details.

Steps and Execution Details:

- Configure kubectl files** (0s):
 - 8 `aws s3 ls`
 - 9 `aws eks --region ***** update-kubeconfig --name eks-cluster`
 - 10 `Added new context arn:aws:eks:*****:639284526680:cluster/eks-cluster to /root/.kube/config`
 - 11 `kubectl get svc`
- Deploy blue instance** (1s):
 - 12 `kubectl get svc`
 - 13 `kubectl get po`
 - 14 `NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE`
 - 15 `nginx-service-cluster-ip ClusterIP 172.20.45.69 <none> 443/TCP 62s`
 - 16 `nginx-service-loadbalancer LoadBalancer 172.20.106.19 abc92ed1380574e03907d4d45ca3e65-122280204-*****.elb.amazonaws.com 80:32747/TCP 160m`
 - 17 `kubectl get po`
 - 18 `NAME READY STATUS RESTARTS AGE`
 - 19 `sampleapp-deployment-42f8866-74d8b97f-xrtkh 1/1 Running 0 5m42s`
 - 20 `kubectl get rs`
 - 21 `NAME DESIRED CURRENT READY AGE`
 - 22 `sampleapp-deployment-42f8866-74d8b97f 1 1 1 5m42s`
 - 23 `CircleCI received exit code 0`
- Check blue instance** (4s):
 - 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`
- Switch load balancer to target the new app** (1s):
 - 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`
- Destroy old version** (3s):
 - 1 `#!/bin/bash -eo pipefail`
 - 2 `cd kube-deploy`
 - 3 `make destroy-old-version`
 - 4
 - 5 `kubectl apply -f ./load-balancer.yaml`
 - 6 `service/nginx-service-loadbalancer configured`
 - 7 `CircleCI received exit code 0`
- Saving cache** (0s):
 - 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`

Destroy old version

The screenshot shows a GitHub Actions workflow run for the repository 'bishoymaurice/sampleapp'. The workflow is named 'Destroy old version' and is currently in a 'Completed' state. The run ID is '42f866'. The workflow consists of several steps, each with a job name and a duration. The steps are: 'Check blue instance' (4s), 'Switch load balancer to target the new app' (1s), 'Destroy old version' (3s), and 'Saving cache' (0s). The 'Destroy old version' step is the focus of the screenshot, showing the execution of a script that destroys the old deployment.

Workflow Steps:

- Check blue instance** (4s):
 - 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-7614911 created
 - 7. CircleCI received exit code 0
- Switch load balancer to target the new app** (1s):
 - 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
- Destroy old version** (3s):
 - 1. #!/bin/bash -eo pipefail
 - 2. cd kube-deploy
 - 3. make destroy-old-version
 - 4.
 - 5. sh ./scripts/DestroyBlue.sh
 - 6. NAME READY UP-TO-DATE AVAILABLE AGE
 - 7. sampleapp-deployment-42f866 1/1 1 1 1s2s
 - 8. Old deployment is: "OLD_DEPLOYMENT" and will be destroyed
 - 9. deployment.apps "sampleapp-deployment-42f866" deleted
 - 10. CircleCI received exit code 0
- Saving cache** (0s):

Workflow Summary:

Step	Job	Duration
Check blue instance	4s	
Switch load balancer to target the new app	1s	
Destroy old version	3s	
Saving cache	0s	

Workflow Status: Completed

Workflow URL: https://app.circleci.com/pipelines/github/bishoymaurice/AWS_DevOps/113/workflows/7614511e-c16a-4a28-af12-43ab7bee1e10/jobs/214