

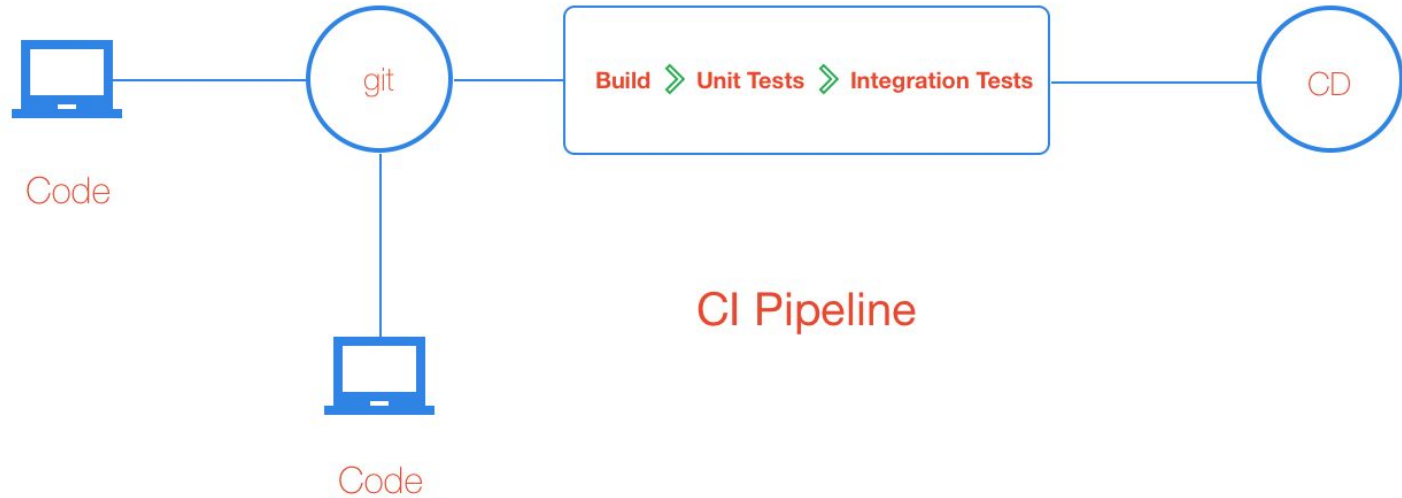
A CI/CD Pipeline using Kubernetes, Deis, Helm and Gitlab



Adnaan Badr

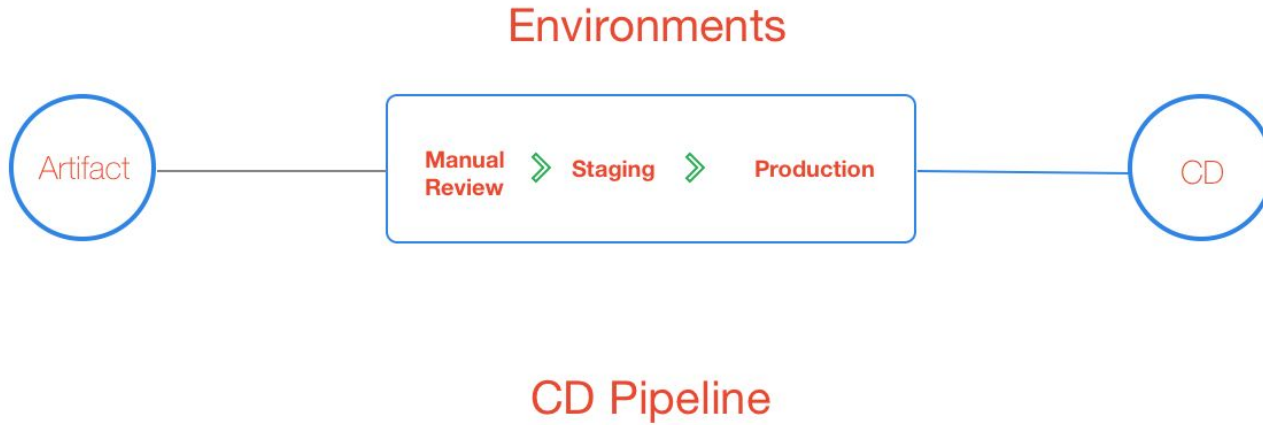
github.com/adnaan

Continuous Integration



Continuously merging quality code

Continuous Deployment



Continuously deploying quality code artifacts

The Pipeline

CI

- Build
- Test
- Code Quality

CD

- Manual Review
- Deploy
- Monitor

The Pipeline - Tools

CI

- Build - Gitlab Runners
- Test - Unit Test: Gitlab Runners, Integration Tests: Gitlab Runners, Deis Workflow
- Code Quality: Gitlab Runners

CD

- Manual Review: Deis Workflow
- Deploy: Deis Workflow
- Monitor: Kubernetes, Prometheus, Gitlab UI

*Deis Workflow is a PAAS which runs on top of Kubernetes

*Gitlab Runner runs jobs.

The Pipeline - Configuration

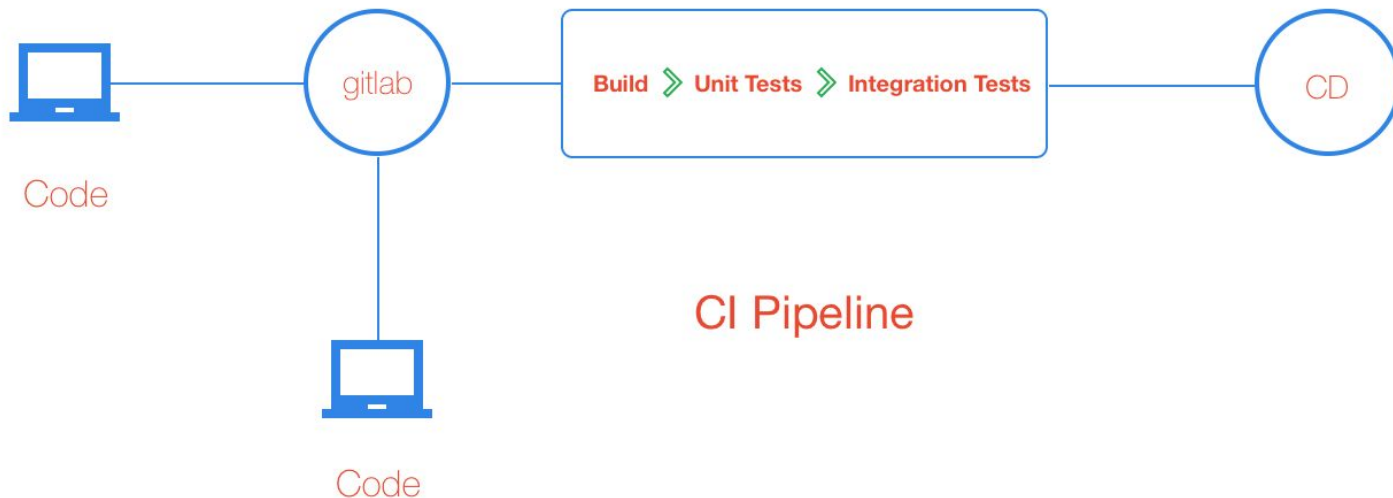
CI

- Build - .gitlab-ci.yaml
- Test - .gitlab-ci.yaml
- Code Quality - .gitlab-ci.yaml

CD

- Manual Review - .gitlab-ci.yaml, gitlab secret variables
- Deploy - .gitlab-ci.yaml, gitlab secret variables
- Monitor - gitlab ui

Continuous Integration Components: Gitlab Runners

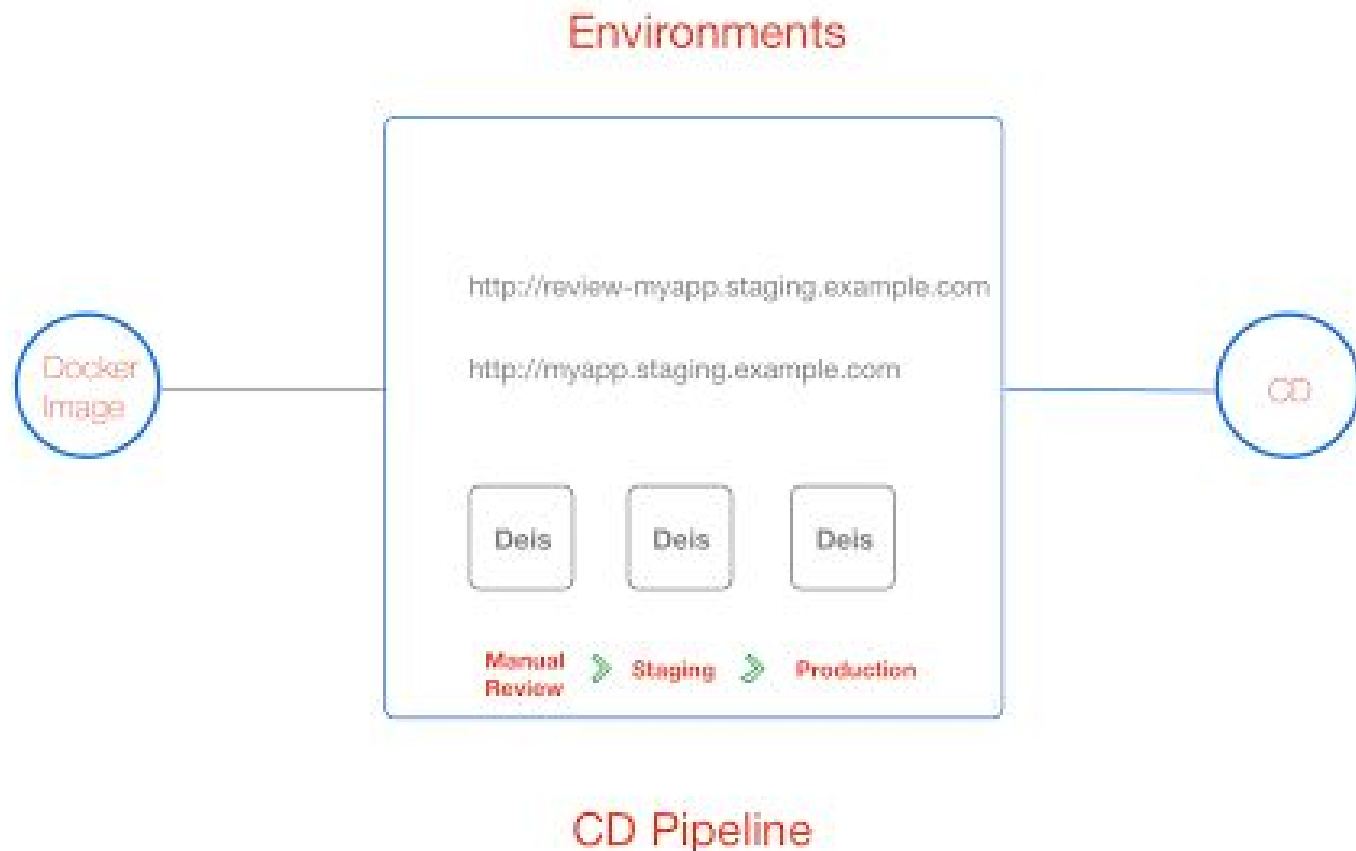


```
1  variables:
2    DEIS_APP_NAME: $APP_NAME-$CI_BUILD_REF_SLUG
3    APP_IMAGE_TAG: $APP_NAME:$CI_BUILD_REF_SLUG
4    NGINX_IMAGE: 'nginx-upstream:latest'
5
6  stages:
7    - build
8    - docker
9    - integration-tests
10   - integration-tests-cleanup
11   - review
12   - staging
13   - production
14
15  go_test_build_branch:
16    stage: build
17    image: golang:latest
18    before_script:
19      - ln -s /builds /go/src/gitlab.com
20      - cd /go/src/gitlab.com/zenius/$APP_NAME
21    script:
22      - curl https://glide.sh/get | sh
23      - glide install
24      - go test -race $(go list ./... | grep -v /vendor/)
25      - CGO_ENABLED=0 GOOS=linux GOARCH=amd64 go build
26        -ldflags "-extldflags '-static'" -o $APP_NAME
27  artifacts:
28    paths:
29      - $APP_NAME
30    expire_in: 1 hour
31
```

```
run-integration-tests:
  stage: integration-tests
  variables:
    GIT_STRATEGY: none
  dependencies: []
  before_script:
    - mkdir -p tmp && cd tmp
    - curl -sSL http://deis.io/deis-cli/install-v2.sh | bash
  script:
    - ./deis login $DEIS_CONTROLLER --username=$DEIS_REVIEW_USER --password=$DEIS_REVIEW_PWD
    - ./deis create $DEIS_APP_NAME --no-remote #Create the app
    - ./deis config:set PORT=3333 --app=$DEIS_APP_NAME
    - ./deis config:set AppVersion=V1-$CI_BUILD_REF_SLUG --app=$DEIS_APP_NAME
    - ./deis pull $ECR_URL/$APP_IMAGE_TAG --app=$DEIS_APP_NAME
    - ./deis routing:disable --app=$DEIS_APP_NAME
    - sleep 10
    - ./deis create integration-$DEIS_APP_NAME --no-remote #Create the temporary integration test app
    - ./deis config:set PORT=80 --app=integration-$DEIS_APP_NAME
    - ./deis config:set AppVersion=V1Integration --app=integration-$DEIS_APP_NAME
    - ./deis config:set TEST_HOSTNAME=$DEIS_APP_NAME.$DEIS_APP_NAME --app=integration-$DEIS_APP_NAME
    - ./deis pull $ECR_URL/$APP_IMAGE_TAG-integration --app=integration-$DEIS_APP_NAME
    - ./deis routing:disable --app=integration-$DEIS_APP_NAME
    - sleep 5
    - ./deis run '/usr/bin/integration' --app=integration-$DEIS_APP_NAME # run tests
    - sleep 2
    - echo "integration tests done!!"

only:
  - branches
except:
  - master
```


Continuous Deployment Component: Deis



Continuous Deployment: Gitlab Environments

Available (4) Stopped (3) [New environment](#)

Environment	Deployment	Job	Commit	Updated	
production	#10 by	production #190500148	be8328f8 Merge branch 'empty_deps' into 'ma...	2 months ago	Re-deploy
+ review (3)					
review/master		No deployments yet			
review/test_fix		No deployments yet			
Show all					
staging	#9 by	staging #190600047	be8328f8 Merge branch 'empty_deps' into 'ma...	2 months ago	Re-deploy

Continuous Deployment: Gitlab Pipelines

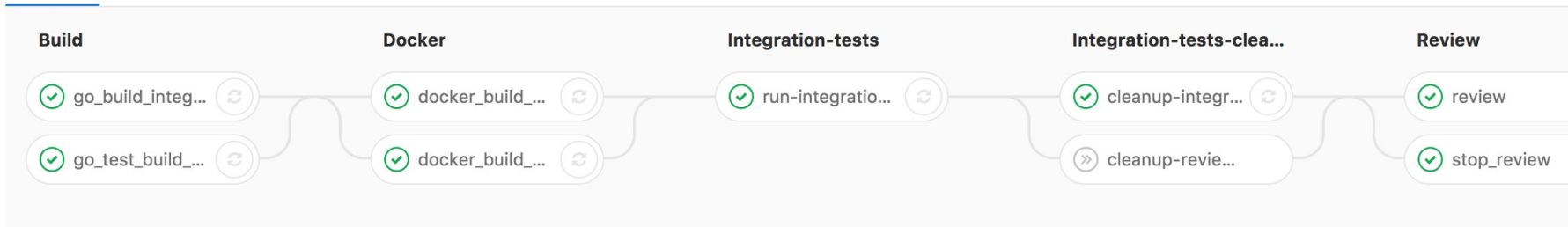
passed Pipeline #9139926 triggered 2 months ago by  Md Adnaan Badr

empty dependencies for later stages

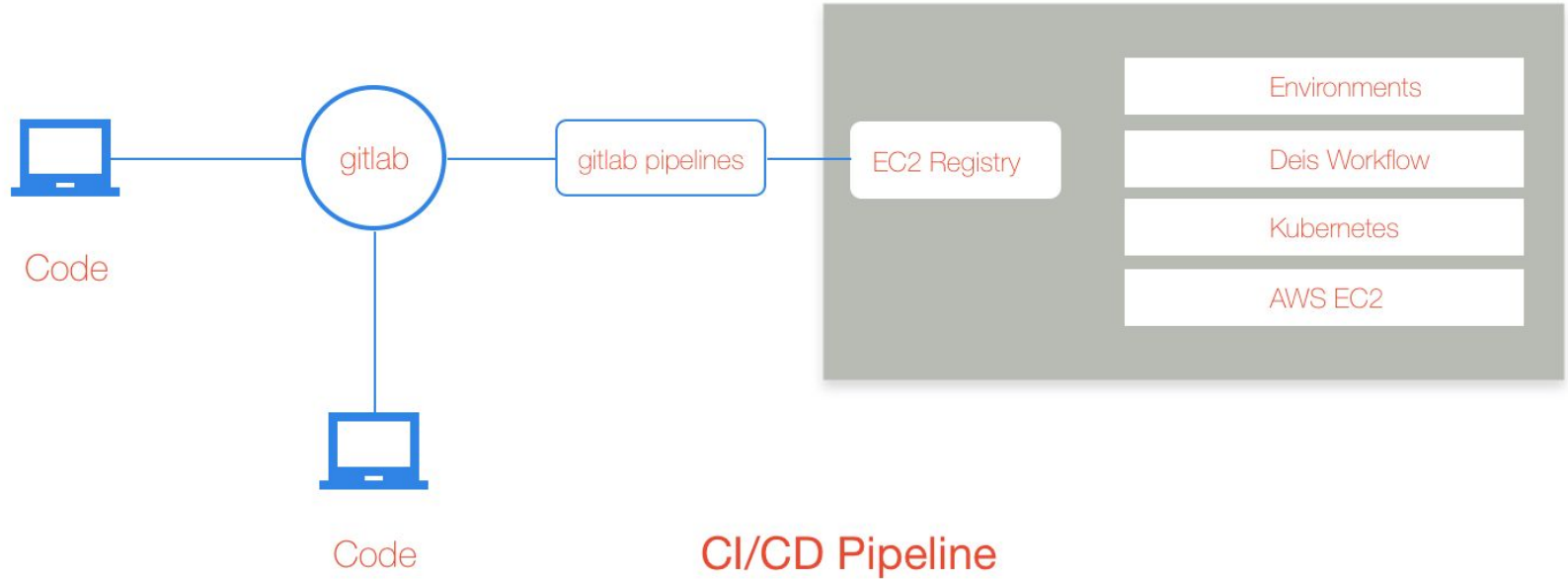
9 jobs from `empty_deps` in 9 minutes 46 seconds (queued for 7 seconds)

 [9a740507](#)  

Pipeline Jobs 9



CI/CD Pipeline

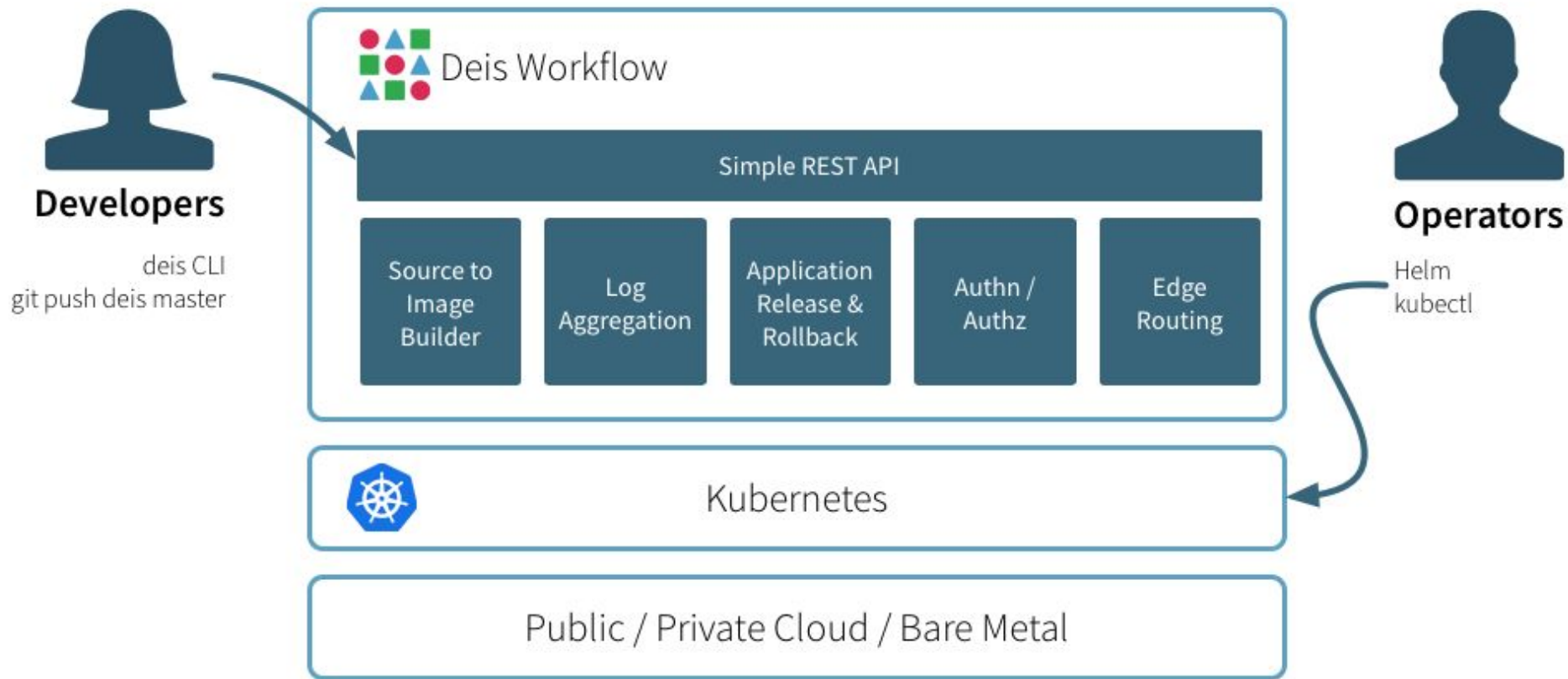


Setting up the Deployment Stack: Kubernetes and Deis Workflow

Prerequisites:

- AWS EC2(3 t2-medium instances)
- AWS EC2 Container Registry
- ELB
- DNS
- S3
- AWS CLI
- Kubectl
- Kops
- Helm

Deis Workflow and Kubernetes



Create Kubernetes Cluster

```
$ aws configure
```

```
# create sub-domain name for cluster in route53: staging.example.com
```

```
# create s3 bucket "clusters.staging.example.com"
```

```
$ export KOPS_STATE_STORE=s3://clusters.staging.example.com
```

```
$ export NAME=staging.example.com
```

```
# Installing Kops
```

```
# https://github.com/kubernetes/kops/blob/master/docs/aws.md
```

```
$ KOPS_STATE_STORE=s3://clusters.staging.example.com kops create cluster \
```

```
--name=staging.example.com \
```

```
--zones=us-west-2a \
```

```
--master-size=t2.medium
```

Installing Helm

Helm is used to:

- Make configurable releases.
- Upgrade, delete, inspect releases.

```
$ helm init
```


Installing Deis Workflow

Deis Workflow is a PAAS. It includes capabilities for:

- building and deploying from source via git push,
- simple application configuration,
- creating and rolling back releases,
- managing domain names and SSL certificates,
- providing seamless edge routing,
- aggregating logs,
- and sharing applications with teams.

```
$ helm install deis/workflow --namespace deis-staging -f values.yaml
```

[docs](#)

Configure DNS

```
$ kubectl --namespace=staging describe svc deis-router | egrep LoadBalancer
```

Type: LoadBalancer

LoadBalancer Ingress: afd27ef1b523d11e7bad306274faeacf-1159166960.us-west-2.elb.amazonaws.com

Create a wildcard CNAME record with the above load balancer hostname at the DNS registry

After DNS change has propagated. Test:

```
$ dig ns +short anydomain.staging.example.com
```

```
a1f80462354a111e7bad306274faeacf-1159166960.us-west-2.elb.amazonaws.com.
```

Answer should match the load balancer hostname above.

Configure Load Balancer

```
$ aws elb describe-load-balancers --query 'LoadBalancerDescriptions[*].LoadBalancerName'
```

```
[  
  "a1f80462354a111e7bad306274faeacf"  
]
```

```
$ aws elb modify-load-balancer-attributes \  
  --load-balancer-name a1f80462354a111e7bad306274faeacf \  
  --load-balancer-attributes "{\\"ConnectionSettings\\":{\\"IdleTimeout\\":1200}}"
```

Up and Running

Test deis workflow API service

```
$ curl deis.staging.example.com/v2/ && echo  
{"detail":"Authentication credentials were not provided."}
```

Register admin user:

```
$ deis register http://deis.staging.example.com
```

Git Push and Create Merge Request

```
$ git push origin test_fix
```

On creating the merge request from UI/CLI. The build stage is automatically triggered, followed by tests, review and deployment.

☐ Merge request approvals 

☒ Only allow merge requests to be merged if the pipeline succeeds
Pipelines need to be configured to enable this feature. 

Request to merge `test_fix`  into `master`

Check out branch




 Pipeline #11579650  running for 03f032d3.

 ☒ Merge when pipeline succeeds ☐ Remove source branch

You can merge this merge request manually using the [command line](#)

Complete Pipeline

✓ passed Pipeline #9139926 triggered 2 months ago by  Md Adnaan Badr

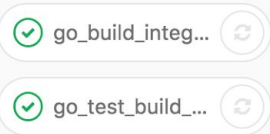
empty dependencies for later stages

🕒 9 jobs from [empty_deps](#) in 9 minutes 46 seconds (queued for 7 seconds)

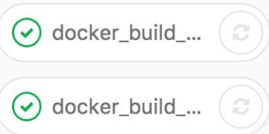
🔗 [9a740507](#) ... 📄

Pipeline Jobs 9

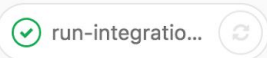
Build



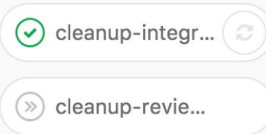
Docker



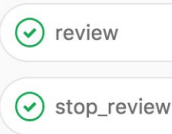
Integration-tests



Integration-tests-clea...



Review















Manage Deployments

production

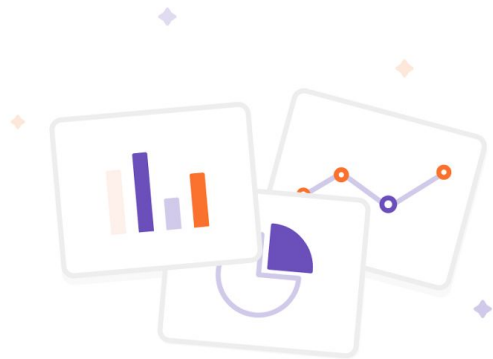
[View deployment](#)

[Monitoring](#)

[Edit](#)

ID	Commit	Job	Created	
#10	 master  be0320fb  Merge branch 'empty_deps' into 'master'	production (#19050048) by 	2 months ago	Re-deploy
#7	 master  a4b934bd  Merge branch 'split_stages' into 'master'	production (#19044608) by 	2 months ago	Rollback
#4	 master  c3292afb  Merge branch 'gitlab_ci_environ' into 'master'	production (#19040748) by 	2 months ago	▶ ▼ Rollback

Environment: **production**



Get started with performance monitoring

Stay updated about the performance and health of your environment by configuring Prometheus to monitor your deployments.

[Configure Prometheus](#)

Thank you

github.com/adnaan

github.com/adnaan/talks

badr.adnaan@gmail.com

Twitter: @adnaanx

Technical Lead

Systems Engineering, Myntra

References:

[Booting Kubernetes on Amazon Elastic Compute with kops](#)

[Installing Deis Workflow on Amazon Web Services](#)

[Gitlab Continuous Integration](#)

[Autodevops](#)