

Cluster

- Cluster Roles
- Namespaces
- Nodes
- Persistent Volumes
- Service Accounts N
- Storage Classes

Workloads N


- Cron Jobs
- Daemon Sets
- Deployments
- Jobs
- Pods

Logs from api-gateway in api-gateway-deployment-54d886dc96-rj7lz  

```
> api-gateway@1.46.0 build /usr/src/app
> babel ./src --out-dir build
Successfully compiled 180 files with Babel (3610ms).
> api-gateway@1.46.0 start:prod /usr/src/app
> NODE_ENV=production node -r dotenv-flow/config ./build/server.js
(node:40) Warning: Setting the NODE_TLS_REJECT_UNAUTHORIZED environment variable to '0' makes TLS connections and HTTPS requests insecure by disabling certificate verification.
Server running at: http://localhost:5000
```

Events



Message	Source	Sub-object	Count	First Seen	Last Seen
 Readiness probe failed: Get "http://10.244.0.197:5000/v1/hc": dial tcp 10.244.0.197:5000: connect: connection refused	kubelet pool-rzjssyt3-3vjsv	spec.containers(api-gateway)	417	Feb 1, 2021	Feb 1, 2021

service.yaml

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: api-gateway-lb
5    annotations:
6      service.beta.kubernetes.io/do-loadbalancer-name: "NAMESPACE-api-gateway-lb"
7  spec:
8    ports:
9      - port: 80
10      targetPort: 5000
11      protocol: TCP
12    externalTrafficPolicy: Local
13    selector:
14      name: api-gateway
15    type: LoadBalancer
16
```

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: api-gateway-deployment
5    labels:
6      app: api-gateway
7  spec:
8    replicas: 2
9    selector:
10   matchLabels:
11     name: api-gateway
12   template:
13     metadata:
14       labels:
15         name: api-gateway
16     spec:
17       containers:
18         - name: api-gateway
19           image: BUILD_IMAGE
20           ports:
21             - containerPort: 80
22           readinessProbe:
23             httpGet:
24               path: /v1/hc
25               port: 5000
26               scheme: HTTP
27             initialDelaySeconds: 30
28             periodSeconds: 10
29             timeoutSeconds: 5
30           envFrom:
31             - secretRef:
32                 name: api-gateway-env-secret
33           imagePullSecrets:
34             - name: regcred
35
```

Dockerfile x

```
1  ▶ FROM node:12.14.1-alpine3.9
2
3  ENV SERVER_HOST=0.0.0.0
4  ENV SERVER_PORT=5000
5
6  WORKDIR /usr/src/app
7
8  COPY package.json ./
9  COPY package-lock.json ./
10
11 RUN npm ci
12
13 COPY . .
14
15 EXPOSE 5000
16 CMD npm run build && npm run start:prod
17
```



docker-compose.yml

```
1  version: '3'
2  services:
3    api:
4      container_name: api-gateway
5      ports:
6        - "5000:5000"
7      build: .
8
```

.gitlab-ci.yml

```
1  image: docker:18.09.7
2  services:
3  - docker:18.09.7-dind
4
5  stages:
6  - build
7  - lint
8  - test
9  - deploy
10
11 variables:
12 DOCKER_DRIVER: overlay
13 TEST_NAMESPACE: test
14 PROD_NAMESPACE: prod
15 BUILD_IMAGE: ${CI_REGISTRY_IMAGE}:${CI_COMMIT_REF_NAME}_${CI_COMMIT_SHORT_SHA}
16
17 build:
18 stage: build
19 script:
20 - docker login -u gitlab-ci-token -p ${CI_BUILD_TOKEN} ${CI_REGISTRY}
21 - docker build --pull -t ${BUILD_IMAGE} .
22 - docker push ${BUILD_IMAGE}
23 only:
24 - develop
25 - production
26
27 lint:
28 stage: lint
29 script:
30 - docker login -u gitlab-ci-token -p ${CI_BUILD_TOKEN} ${CI_REGISTRY}
31 - docker run --env NODE_ENV=test --name testing -d ${BUILD_IMAGE}
32 - docker exec -i testing npm run lint
33 only:
34 - develop
35 - production
36 when: manual
```

```
34 - develop
35 - production
36 when: manual
37
38 test:
39   stage: test
40   script:
41     - docker login -u gitlab-ci-token -p ${CI_BUILD_TOKEN} ${CI_REGISTRY}
42     - docker run --env NODE_ENV=test --name testing -d ${BUILD_IMAGE}
43     - docker exec -i testing npm run test-db:refresh
44     - docker exec -i testing npm run test
45   only:
46     - develop
47     - production
48   when: manual
49
50 deploy to test:
51   stage: deploy
52   environment:
53     name: test
54   image:
55     name: bitnami/kubectl:latest
56     entrypoint: [""]
57   script:
58     - sed -ri "s~BUILD_IMAGE~${BUILD_IMAGE}~" ./manifest/k8s/deployment.yaml
59     - sed -ri "s~BUILD_IMAGE~${BUILD_IMAGE}~" ./manifest/k8s/service.yaml
60     - sed -ri "s~NAMESPACE~${TEST_NAMESPACE}~" ./manifest/k8s/service.yaml
61     - kubectl apply -f ./manifest/k8s/deployment.yaml -n ${TEST_NAMESPACE}
62     - kubectl apply -f ./manifest/k8s/service.yaml -n ${TEST_NAMESPACE}
63   only:
64     - develop
65
66 deploy to prod:
67   stage: deploy
68   environment:
69     name: prod
```



```
52  environment:
53      name: test
54  image:
55      name: bitnami/kubectl:latest
56      entrypoint: [""]
57  script:
58      - sed -ri "s~BUILD_IMAGE~${BUILD_IMAGE}~" ./manifest/k8s/deployment.yaml
59      - sed -ri "s~BUILD_IMAGE~${BUILD_IMAGE}~" ./manifest/k8s/service.yaml
60      - sed -ri "s~NAMESPACE~${TEST_NAMESPACE}~" ./manifest/k8s/service.yaml
61      - kubectl apply -f ./manifest/k8s/deployment.yaml -n ${TEST_NAMESPACE}
62      - kubectl apply -f ./manifest/k8s/service.yaml -n ${TEST_NAMESPACE}
63  only:
64      - develop
65
66  deploy to prod:
67      stage: deploy
68      environment:
69          name: prod
70      image:
71          name: bitnami/kubectl:latest
72          entrypoint: [""]
73      script:
74          - sed -ri "s~BUILD_IMAGE~${BUILD_IMAGE}~" ./manifest/k8s/deployment.yaml
75          - sed -ri "s~BUILD_IMAGE~${BUILD_IMAGE}~" ./manifest/k8s/service.yaml
76          - sed -ri "s~NAMESPACE~${PROD_NAMESPACE}~" ./manifest/k8s/service.yaml
77          - kubectl apply -f ./manifest/k8s/deployment.yaml -n ${PROD_NAMESPACE}
78          - kubectl apply -f ./manifest/k8s/service.yaml -n ${PROD_NAMESPACE}
79      only:
80          - production
81      when: manual
82
```