

به نام خدا

پروژه پایانی

مبانی رایانش ابری

سینا کریمی

۹۹۳۱۰۵۰

فاطمه الماسیاز

۹۹۳۱۰۶۵

بخش اول

در این بخش سرور و نیازمندی های آن شامل دیتابیس پیاده سازی شده اند.

سرویس اول برای ثبت سفارشات:

POST /submit

Parameters

Cancel

Name	Description
orderDTO * required	
object (query)	<pre>{ "orderId": 24, "productName": "Macbook Pro", "count": 5, "address": "tehran"}</pre>

Execute Clear

Responses

Curl

```
curl -X 'POST' \  "http://localhost:8080/submit?orderId=24&productName=Macbook%20Pro&count=5&address=tehran" \  -H 'accept: */*' \  -d ''
```

Request URL

http://localhost:8080/submit?orderId=24&productName=Macbook%20Pro&count=5&address=tehran

Server response

Code	Details
200	<p>Response body</p> <pre>{}</pre> <p>Response headers</p> <pre>connection: keep-alive content-type: application/json date: Tue, 11 Dec 2024 19:03:26 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre>

می‌توانیم ببینیم که این سرویس شناسه سفارش را برگردانده است.
سرویس دوم برای تایید سفارشات:

PUT /{orderId}

Parameters

Cancel

Name	Description
orderId * required	
integer (int64) (path)	2
status * required	
string (query)	APPROVED

Execute Clear

Responses

Curl

```
curl -X 'PUT' \  "http://localhost:8080/2/status=APPROVED" \  -H 'accept: */*' \  -d ''
```

Request URL

http://localhost:8080/2/status=APPROVED

Server response

Code	Details
200	<p>Response headers</p> <pre>connection: keep-alive content-length: 0 date: Tue, 11 Dec 2024 19:04:35 GMT keep-alive: timeout=60</pre>

می‌توانیم ببینیم که سرویس مورد نظر با کد 200 خروجی داده‌است.
در نهایت سرویس سوم برای بررسی وضعیت سفارشات:

GET `/orderId/status`

Parameters

Name	Description
orderId * required integer(\$int64) (path)	2

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/2/status' \
  -H 'accept: */*'
```

Request URL

`http://localhost:8080/2/status`

Server response

Code	Details
200	<div>Response body</div> <div>PENDING</div> <div>Response headers</div> <pre>connection: keep-alive content-length: 7 content-type: text/plain; charset=UTF-8 date: Tue, 31 Dec 2024 19:05:57 GMT keep-alive: timeout=60</pre>

در ابتدا برای شناسه 2 مقدار PENDING بازگردانده می‌شود. پس از آپدیت وضعیت خروجی به زیر تغییر می‌کند:

GET `/orderId/status`

Parameters

Name	Description
orderId * required integer(\$int64) (path)	2

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/2/status' \
  -H 'accept: */*'
```

Request URL

`http://localhost:8080/2/status`

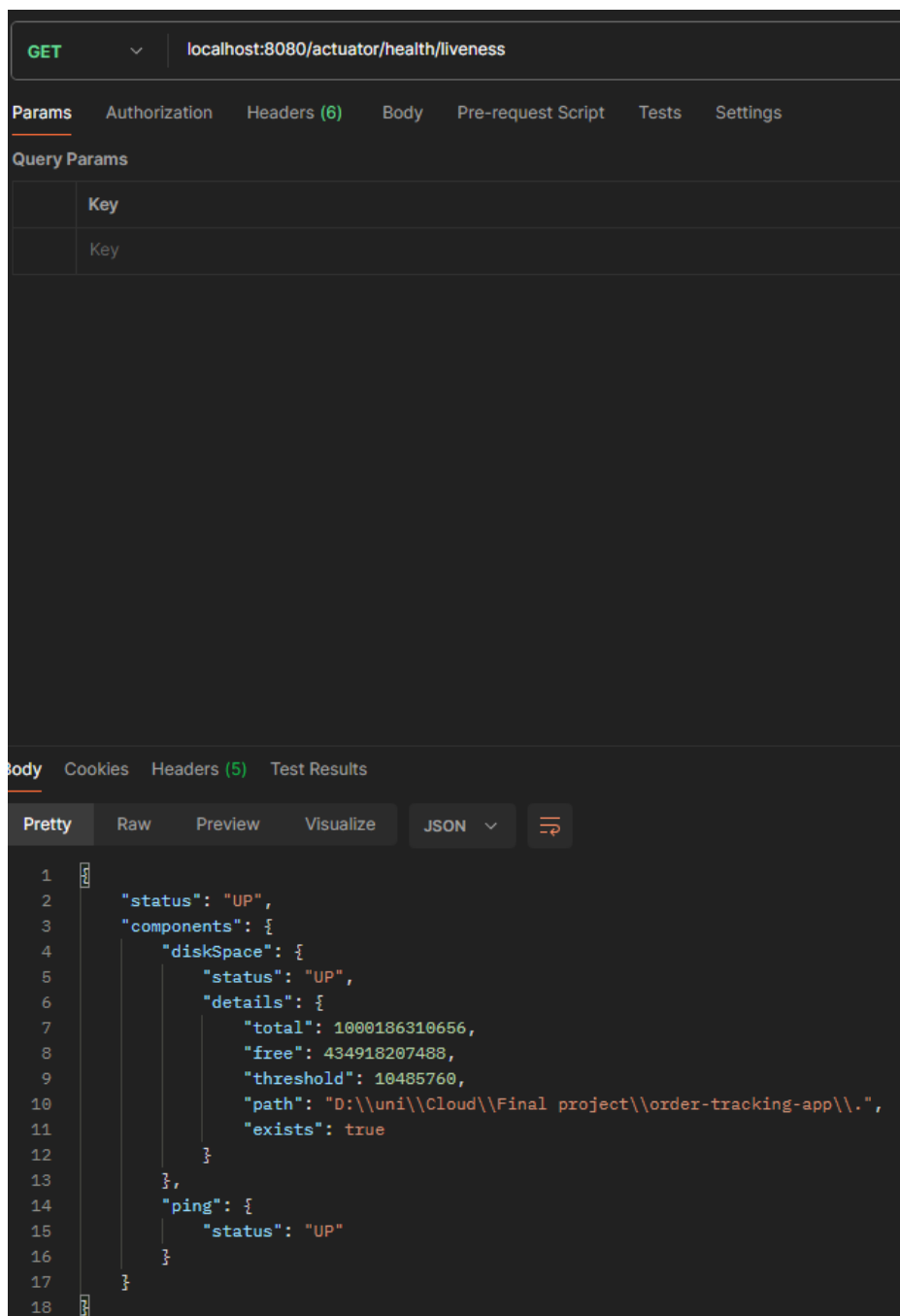
Server response

Code	Details
200	<div>Response body</div> <div>APPROVED</div> <div>Response headers</div> <pre>connection: keep-alive content-length: 8 content-type: text/plain; charset=UTF-8 date: Tue, 31 Dec 2024 19:07:31 GMT keep-alive: timeout=60</pre>

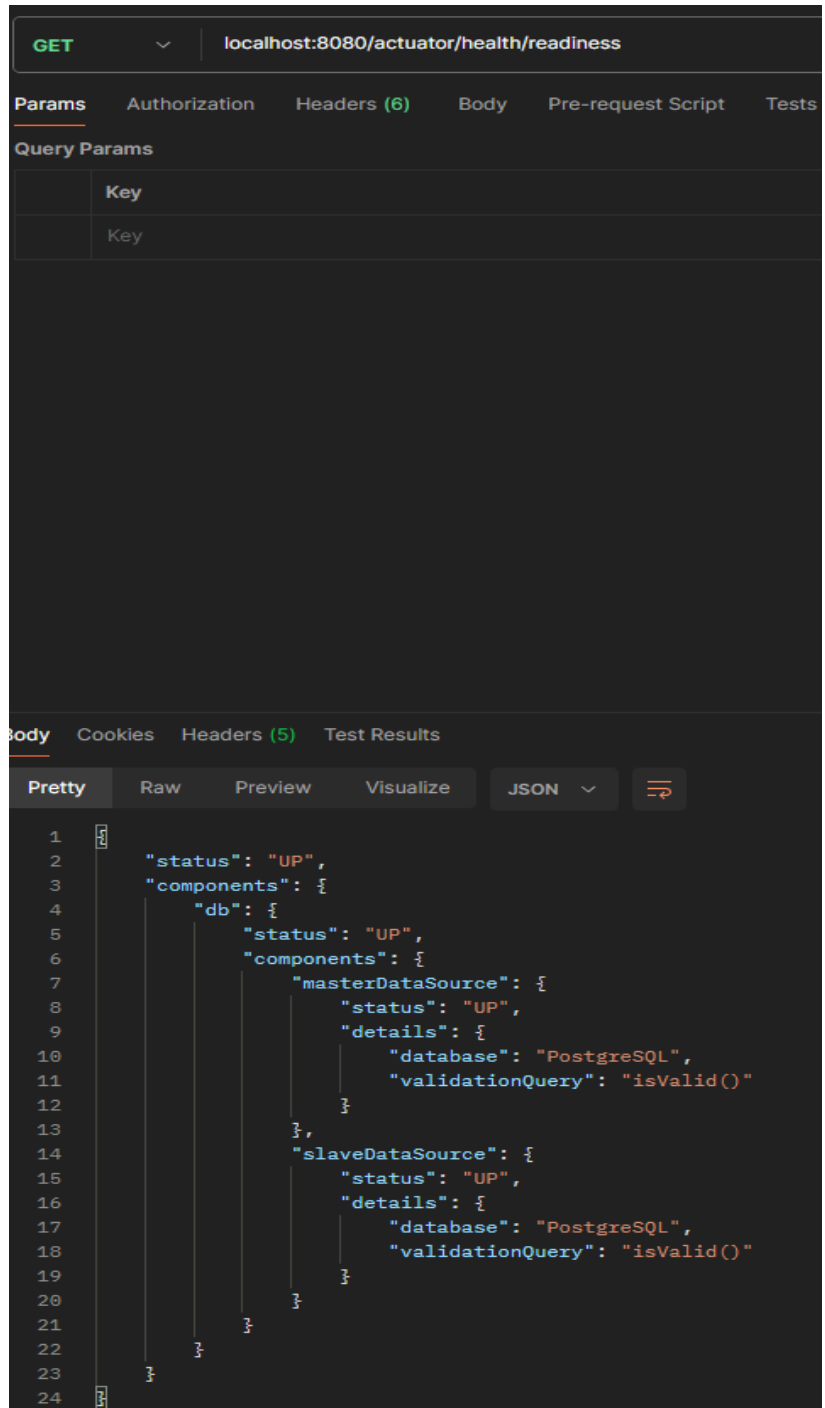
برای بخش امتیازی سرویس های `liveness` و `readiness` به سرور اضافه شدند.

از `liveness` برای بررسی سالم بودن برنامه داخل کانتینر درون Pod استفاده می‌شود و اگر خروجی مناسب بازگردانده نشود، `kubernetes` کانتینر را ریست می‌کند. بیشتر برای سناریوهایی استفاده می‌شود که برنامه به `deadlock` خورده است و یا `unresponsive` شده است.

برای این منظور یک مسیر در سرور در نظر گرفته شده است که با فراخوانی آذ نتیجه ping کردن سرور و فضای disk را به کاربر نمایش می دهد:



Readiness تقریباً مانند liveness عمل می کند با این تفاوت که صحت سرویس های thirdparty مانند پایگاه داده ها را بررسی می کند و عموماً برای نشان دادن آمادگی جهت ارائه سرویس از سوی سرور استفاده می شود و با استفاده از آن kubernetes تصمیم می گیرد که آیا ترافیک را به سمت این سرور ارسال بکند یا نه. برای این مورد هم یک سرویس دیگر در نظر گرفته شده است که وضعیت اتصال به پایگاه داده را نمایش می دهد.



بخش دوم

```
PS D:\uniCloud\Final project\order-tracking-app\order-tracking-app> helm list
NAME      NAMESPACE      REVISION      UPDATED           STATUS      CHART              APP VERSION
project   default         1             2025-01-12 22:16:05.2480606 +0330 +0330 deployed order-tracking-app-0.1.0 1.16.0

PS D:\uniCloud\Final project\order-tracking-app\order-tracking-app> kubectl get pods -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP            NODE       NOMINATED NODE   READINESS GATES
master-postgres-dcd487bbf-z2xqj     1/1     Running   0           81s   10.244.0.74   minikube   <none>            <none>
order-tracking-app-backend-deployment-d788cd4cd-djppd 1/1     Running   0           81s   10.244.0.73   minikube   <none>            <none>
slave-postgres-66ff4cb77-jzzqp      1/1     Running   0           81s   10.244.0.75   minikube   <none>            <none>
```

می‌توانیم ببینیم که پادها بالا آمده‌اند.

پاد سرور:

```
PS D:\Uni\Cloud\Final project\order-tracking-app\order-tracking-app> kubectl describe pod order-tracking-app-backend-deployment-d788cd4cd-djpqd
Name: order-tracking-app-backend-deployment-d788cd4cd-djpqd
Namespace: default
Priority: 0
Service Account: default
Node: minikube/192.168.49.2
Start Time: Sun, 12 Jan 2025 22:16:05 +0330
Labels: app=spring-backend
         pod-template-hash=d788cd4cd
Annotations: <none>
Status: Running
IP: 10.244.0.73
IPs:
  IP: 10.244.0.73
Controlled By: ReplicaSet/order-tracking-app-backend-deployment-d788cd4cd
Init Containers:
  wait-for-master-db:
    Container ID: docker://1626b6ac4061b7da02195159be81f4e52b0c899e737a0525c24336ea8937d23f
    Image: busybox:1.28
    Image ID: docker-pullable://busybox@sha256:141c253bc4c3fd0a201d32dc1f493bcf3fff003b6df416dea4f41046e0f37d47
    Port: <none>
    Host Port: <none>
    Command:
      sh
      -c
      until nc -z master-svc 5432; do
        echo "Waiting for master database to be ready ..."
        sleep 2
      done
      echo "Master database is ready!"
    State: Terminated
    Reason: Completed
    Exit Code: 0
    Started: Sun, 12 Jan 2025 22:16:06 +0330
    Finished: Sun, 12 Jan 2025 22:16:42 +0330
    Ready: True
    Restart Count: 0
    Environment: <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-j4zgn (ro)
  wait-for-slave-db:
    Container ID: docker://8f6848491b91481b7d69fcc215750c77f342858d61657257a2fb0022b1bd98ae
    Image: busybox:1.28
    Image ID: docker-pullable://busybox@sha256:141c253bc4c3fd0a201d32dc1f493bcf3fff003b6df416dea4f41046e0f37d47
    Port: <none>
    Host Port: <none>
    Command:
      sh
      -c
      until nc -z slave-svc 5432; do
        echo "Waiting for slave database to be ready ..."
        sleep 2
      done
      echo "Slave database is ready!"
    State: Terminated
    Reason: Completed
    Exit Code: 0
    Started: Sun, 12 Jan 2025 22:16:42 +0330
    Finished: Sun, 12 Jan 2025 22:16:44 +0330
    Ready: True
    Restart Count: 0
    Environment: <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-j4zgn (ro)
Containers:
  order-tracking-container:
    Container ID: docker://1f03a878fb01676d3d12c8e2b80cfaa28d475138106caca909eeeb4908a55f67
    Image: sinak2002/order-tracking:latest
    Image ID: docker://sha256:bb483ebb448616e28f03d90f9a732ba563a200f7731f70f261380704df8ace54
    Port: 8080/TCP
    Host Port: 0/TCP
    State: Running
    Started: Sun, 12 Jan 2025 22:16:45 +0330
    Ready: True
    Restart Count: 0
    Liveness: http-get http://:8080/actuator/health/liveness delay=30s timeout=5s period=10s #success=1 #failure=3
    Readiness: http-get http://:8080/actuator/health/readiness delay=30s timeout=5s period=10s #success=1 #failure=3
    Startup: http-get http://:8080/actuator/health/liveness delay=30s timeout=20s period=10s #success=1 #failure=1
    Environment Variables from:
      backend-conf ConfigMap Optional: false
    Environment: <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-j4zgn (ro)
Conditions:
  Type Status
  PodReadyToStartContainers True
  Initialized True
  Ready True
  ContainersReady True
  PodScheduled True
Volumes:
  kube-api-access-j4zgn:
    Type: Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName: kube-root-ca.crt
    ConfigMapOptional: <nil>
    DownwardAPI: true
```

پاد مستر دیتابیس:

```

PS D:\uni\Cloud\Final project\order-tracking-app\order-tracking-app> kubectl describe pod master-postgres-dcd487bbf-z2xqj
Name:          master-postgres-dcd487bbf-z2xqj
Namespace:     default
Priority:       0
Service Account: default
Node:          minikube/192.168.49.2
Start Time:    Sun, 12 Jan 2025 22:16:36 +0330
Labels:        app=master-postgres
               pod-template-hash=dcd487bbf
Annotations:   <none>
Status:        Running
IP:            10.244.0.74
IPs:
  IP:          10.244.0.74
Controlled By: ReplicaSet/master-postgres-dcd487bbf
Containers:
  master-postgres:
    Container ID:  docker://0afd21b23b86f773514e35edfc9298a97f7f8b1fe07f3d988c125cf3bd4a9084
    Image:         postgres:14-alpine
    Image ID:      docker://sha256:4ee606c1eb50c792ad22ed9b8c1d93b8d76f106194fe9e6d37b40efda9dd20ba
    Port:          5432/TCP
    Host Port:     0/TCP
    Args:
      postgres
      -c
      wal_level=replica
      -c
      hot_standby=on
      -c
      max_wal_senders=10
      -c
      max_replication_slots=10
      -c
      hot_standby_feedback=on
    State:          Running
      Started:      Sun, 12 Jan 2025 22:16:37 +0330
    Ready:          True
    Restart Count:  0
    Environment Variables from:
      master-postgres-conf  ConfigMap  Optional: false
    Environment:          <none>
    Mounts:
      /docker-entrypoint-initdb.d from init-sql (rw)
      /var/lib/postgresql/data from postgresdb (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-bkz4k (ro)
Conditions:
  Type                               Status
  PodReadyToStartContainers          True
  Initialized                        True
  Ready                              True
  ContainersReady                    True
  PodScheduled                       True
Volumes:
  init-sql:
    Type:          ConfigMap (a volume populated by a ConfigMap)
    Name:          master-postgres-init-sql
    Optional:      false
  postgresdb:
    Type:          PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:     master-postgres-volume-claim
    ReadOnly:      false
  kube-api-access-bkz4k:
    Type:          Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:    kube-root-ca.crt
    ConfigMapOptional: <nil>

```

پاد اسلیو دیتابیس:

```

PS D:\uni\Cloud\Final project\order-tracking-app\order-tracking-app> kubectl describe pod slave-postgres-66ff4cb77-jzzqp
Name: slave-postgres-66ff4cb77-jzzqp
Namespace: default
Priority: 0
Service Account: default
Node: minikube/192.168.49.2
Start Time: Sun, 12 Jan 2025 22:16:36 +0330
Labels: app=slave-postgres
        pod-template-hash=66ff4cb77
Annotations: <none>
Status: Running
IP: 10.244.0.75
IPs:
  IP: 10.244.0.75
Controlled By: ReplicaSet/slave-postgres-66ff4cb77
Containers:
  slave-postgres:
    Container ID: docker://3bbf169988e766941eb489447c629b4fab6323479fea7f61f8597ee16ea5bf2e
    Image: postgres:14-alpine
    Image ID: docker://sha256:4ee606c1eb50c792ad22ed9b8c1d93b8d76f106194fe9e6d37b40efda9dd20ba
    Port: 5432/TCP
    Host Port: 0/TCP
    Command:
      bash
      -c
    Args:
      until pg_basebackup --pgdata=/var/lib/postgresql/data -R --slot=replication_slot --host=master-svc --port=5432
      do
        echo 'Waiting for primary to connect...'
        sleep 1s
      done
      echo 'Backup done, starting replica...'
      chown -R postgres:postgres /var/lib/postgresql/data
      chmod 0700 /var/lib/postgresql/data
      su - postgres -c "postgres -D /var/lib/postgresql/data"
    State: Running
      Started: Sun, 12 Jan 2025 22:16:37 +0330
    Ready: True
    Restart Count: 0
    Environment Variables from:
      slave-postgres-conf ConfigMap Optional: false
    Environment: <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-f66cn (ro)
Conditions:
  Type                               Status
  PodReadyToStartContainers          True
  Initialized                         True
  Ready                              True
  ContainersReady                    True
  PodScheduled                       True
Volumes:
  postgresdb:
    Type: PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName: slave-postgres-volume-claim
    ReadOnly: false
  kube-api-access-f66cn:
    Type: Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName: kube-root-ca.crt
    ConfigMapOptional: <nil>
    DownwardAPI: true

```

بالا بودز اپلیکیشن:

POST /submit

Parameters

Cancel

Name	Description
orderDTO * required object (query)	<pre>{ "userId": 1234, "productName": "test for report", "count": 9999, "address": "University" }</pre>

Execute Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/submit?userId=1234&productName=test%20for%20report&count=9999&address=University' \
  -H 'accept: */*' \
  -d ''
```

Request URL

http://localhost:8080/submit?userId=1234&productName=test%20for%20report&count=9999&address=University

Server response

Code	Details
200	<p>Response body</p> <pre>1</pre> <p>Response headers</p> <pre>connection: keep-alive content-type: application/json date: Sun, 12 Jan 2025 18:53:38 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre>
















پایین آمدن پروژه:

```
PS D:\uni\Cloud\Final project\order-tracking-app\order-tracking-app> helm uninstall project
release "project" uninstalled
```

فاز ۳:

بخش اول:

کانتینر ها

<input type="checkbox"/>	 gitlab	-	Running (2/2)	45 minutes ago	  
<input type="checkbox"/>	 gitlab d7c905e564ed 	gitlab/gitlab-ce:17.5.5-ce.0	Running	8000:8000 	45 minutes ago   
<input type="checkbox"/>	 runner 602f20e579b2 	gitlab/gitlab-runner:alpine	Running	45 minutes ago	  

Runner

Runners

[New instance runner](#)

All 1 Instance 1 Group 0 Project 0

🕒 Search or filter results... 🔍 Created date ▾ ⬇

Online 1 Offline 0 Stale 0

<input type="checkbox"/>	Status	Runner configuration	Owner
<input type="checkbox"/>	Online Idle	#1 (3maJQoEu) Instance 🕒 Last contact: just now 📍 172.18.0.1 🔄 0 📅 Created by Administrator 16 minutes ago test-runner	Administrator ✎ ✖

فرایند اتصال gitlab به runner

```
docker-desktop:/# gitlab-runner register \
--url http://localhost:8080 \
--token glrt-t1_3maJQoEuxEqYxMVv65xt \
--docker-volumes /var/run/docker.sock:/var/run/docker
.sock \
--docker-network-mode 'host'
Runtime platform arch=arm64 os=linux pid=58 revision=3153ccc6 version=17.7.0
Running in system-mode.

Enter the GitLab instance URL (for example, https://gitlab.com/):
http://localhost:8080
Enter the registration token:
glrt-t1_3maJQoEuxEqYxMVv65xt
Verifying runner... is valid runner=t1_3maJQo
Enter a name for the runner. This is stored only in the local config.toml file:
[docker-desktop]: test-runner
Enter an executor: custom, docker+machine, kubernetes, shell, ssh, parallels, virtualbox, docker, docker-windows, docker-autoscaler, instance:
docker
Enter the default Docker image (for example, ruby:2.7):
gitlab/gitlab-runner:alpine
Runner registered successfully. Feel free to start it, but if it's running already the config should be automatically reloaded!

Configuration (with the authentication token) was saved in "/etc/gitlab-runner/config.toml"
docker-desktop:/#
```

محتوای docker compose

```
services:
  gitlab:
    image: gitlab/gitlab-ce:17.5.5-ce.0
    container_name: gitlab
    restart: always
    platform: linux/amd64
    environment:
      GITLAB_ROOT_EMAIL: "admin@cloud.com"
      GITLAB_ROOT_PASSWORD: "Abcd@0123456789"
      GITLAB_OMNIBUS_CONFIG: |
        external_url 'http://localhost:8000'
        nginx['listen_port'] = 8000
        nginx['redirect_http_to_https'] = false
        nginx['ssl_certificate'] = "..."
        nginx['ssl_certificate_key'] = "..."
        prometheus_monitoring['enable'] = false
        alertmanager['enable'] = false
        puma['worker_processes'] = 0
        sidekiq['max_concurrency'] = 10
    ports:
      - '8000:8000'
    volumes:
      - gitlab_config:/etc/gitlab
      - gitlab_logs:/var/log/gitlab
      - gitlab_data:/var/opt/gitlab
    shm_size: '4gb'
    privileged: true
```

```
services:
  gitlab:
    image: gitlab/gitlab-runner:alpine
    shm_size: '4gb'
    privileged: true
    sidekiq['max_concurrency'] = 10
    ports:
      - '8000:8000'
    volumes:
      - gitlab_config:/etc/gitlab
      - gitlab_logs:/var/log/gitlab
      - gitlab_data:/var/opt/gitlab

  gitlab-runner:
    image: gitlab/gitlab-runner:alpine
    container_name: gitlab-runner
    network_mode: 'host'
    depends_on:
      - gitlab
    restart: always
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
      - runner-c:/etc/gitlab-runner

volumes:
  gitlab_config:
  gitlab_logs:
  gitlab_data:
  runner-c:
```

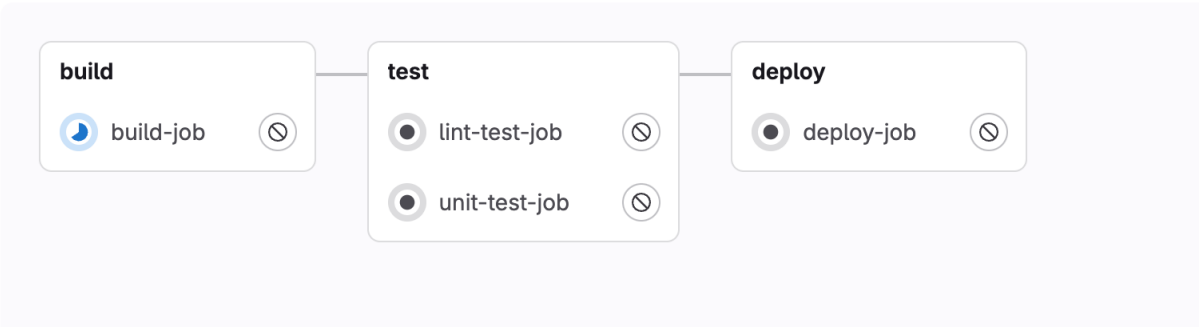
Update .gitlab-ci.yml file

Running Administrator created pipeline for commit `b4a65702` just now

For `main`

latest 4 jobs In progress, queued for 13 seconds

Pipeline Jobs 4 Tests 0



Update .gitlab-ci.yml file

Running Administrator created pipeline for commit `b4a65702` just now


For `main`

latest 4 jobs In progress, queued for 13 seconds

Pipeline **Jobs 4** Tests 0

Status	Job	Stage	Coverage
Created	#4: deploy-job <code>main</code> <code>b4a65702</code>	deploy	
Created	#3: lint-test-job <code>main</code> <code>b4a65702</code>	test	
Created	#2: unit-test-job <code>main</code> <code>b4a65702</code>	test	
Running 00:00:30	#1: build-job <code>main</code> <code>b4a65702</code>	build	

نمایش کد های پروژه در گیت لب

 +

🔍 Search or go to...

Project

cloud-final

Pinned

Issues 0

Merge requests 0

Manage

Plan

Code

Merge requests 0

Repository

Branches

Commits

Tags

Repository graph

Compare revisions

Snippets

Build

Secure

Deploy

Operate

Monitor

localhost:8000/root/cloud-final/-/tree/main

Administrator / cloud-final / Repository

main cloud-final +

History Find file Edit Code

debug .gitlab-ci.yml file add docker image
Administrator authored 3 minutes ago

Name	Last commit	Last update
.idea	Initial commit	3 hours ago
order-tracking-app	Initial commit	3 hours ago
src	Initial commit	3 hours ago
.DS_Store	Initial commit	3 hours ago
.gitlab-ci.yml	debug .gitlab-ci.yml file add docker image	3 minutes ago
Dockerfile	Initial commit	3 hours ago
README.md	Initial commit	21 hours ago
docker-compose.yaml	Initial commit	3 hours ago
pom.xml	Initial commit	3 hours ago
values.yaml	Initial commit	3 hours ago
values.yaml.save	Initial commit	3 hours ago

README.md

cloud-final

Getting started

نمایش اتصال رانر و اجرای pipeline

Administrator / cloud-final / Jobs / #91

🔍 Search or go to...

Project

cloud-final

Pinned

Issues

Merge requests

Manage

Plan

Code

Build

Pipelines

Jobs

Pipeline editor

Pipeline schedules

Artifacts

Secure

Deploy

Operate

Monitor

Analyze

Settings

Help

Admin

Search visible log output

🔍

📄

↶

⬆

⬇

⬅

344.4s

#5 ...

#7 [stage-1 1/4] FROM docker.io/library/openjdk:17-jdk-slim-buster@sha256:2e7658fb62d1c6f319ff9870614deaaf8e06038dd41eec3d1ecdcfabd186234fd

#7 sha256:6a2ff3da472d3e74705427c6b8ac620684ee8c6d130060777a70dd139cd89254 36.70MB / 186.53MB 1344.8s

#7 ...

#5 [builder 1/6] FROM docker.io/library/maven:3.8.7-eclipse-temurin-17@sha256:76789e7bf6713b7fe617b0e72ccf1e0cc23425bc41610c878f13a9b2ffdd2127d

#5 sha256:31c8eedb11250dec332adf8491b833000e8d820fe2f467b5afe963e3f6499a39 56.62MB / 191.27MB 1349.4s

#5 sha256:31c8eedb11250dec332adf8491b833000e8d820fe2f467b5afe963e3f6499a39 56.62MB / 191.27MB 1354.5s

#5 ...

#7 [stage-1 1/4] FROM docker.io/library/openjdk:17-jdk-slim-buster@sha256:2e7658fb62d1c6f319ff9870614deaaf8e06038dd41eec3d1ecdcfabd186234fd

#7 sha256:6a2ff3da472d3e74705427c6b8ac620684ee8c6d130060777a70dd139cd89254 37.75MB / 186.53MB 1354.9s

#7 ...

#5 [builder 1/6] FROM docker.io/library/maven:3.8.7-eclipse-temurin-17@sha256:76789e7bf6713b7fe617b0e72ccf1e0cc23425bc41610c878f13a9b2ffdd2127d

#5 sha256:31c8eedb11250dec332adf8491b833000e8d820fe2f467b5afe963e3f6499a39 57.67MB / 191.27MB 1359.5s

#5 sha256:31c8eedb11250dec332adf8491b833000e8d820fe2f467b5afe963e3f6499a39 58.72MB / 191.27MB 1364.6s

#5 ...

#7 [stage-1 1/4] FROM docker.io/library/openjdk:17-jdk-slim-buster@sha256:2e7658fb62d1c6f319ff9870614deaaf8e06038dd41eec3d1ecdcfabd186234fd

#7 sha256:6a2ff3da472d3e74705427c6b8ac620684ee8c6d130060777a70dd139cd89254 38.80MB / 186.53MB 1364.9s

#7 ...

#5 [builder 1/6] FROM docker.io/library/maven:3.8.7-eclipse-temurin-17@sha256:76789e7bf6713b7fe617b0e72ccf1e0cc23425bc41610c878f13a9b2ffdd2127d

...

Elapsed time: 23 minutes 13 seconds

Queued: 2 seconds

Timeout: 1h (from project)

Runner: #1 (3maJQoEu)

Commit e76863ea

Update .gitlab-ci.yml file

Pipeline #25 Running for main

build

Related jobs

→ build

Running

Update .gitlab-ci.yml file

#25 main e76863ea

latest

فایل gitlab-ci

این فایل شامل دو stage است که یکی برای build و دیگری برای push کردن تغییرات است. قبل از push ورود به رجیستری انجام میشود.

```
variables:
  CUSTOM_REGISTRY: "registry.gitlab.com"
  DOCKER_CLIENT_TIMEOUT: 300
  COMPOSE_HTTP_TIMEOUT: 300

stages:
  - build
  - push

build:
  stage: build
  image: docker:latest
  script:
    - unset DOCKER_HOST
    - echo "Building Docker image..."
    - docker build -t "$CUSTOM_REGISTRY/$CI_PROJECT_NAMESPACE/$CI_PROJECT_NAME:${CI_COMMIT_SHA:0:8}" .
    - echo "build completed"
  only:
    - main

push:
  stage: push
  image: docker:latest
  script:
    - unset DOCKER_HOST
    - echo "Logging into GitLab registry..."
    - echo "$CI_REGISTRY_PASSWORD" | docker login $CI_REGISTRY -u $CI_REGISTRY_USER --password-stdin
    - echo "Pushing Docker image..."
    - docker push "$CUSTOM_REGISTRY/$CI_PROJECT_NAMESPACE/$CI_PROJECT_NAME:${CI_COMMIT_SHA:0:8}"
```