

DevOps Case Study

Serhat Düzen

Task Plan:

- Dockerfile create,
- Kubernetes files configuration,
- CI pipeline,
- CD pipeline,
- Examine environment

Dockerfile Create

- Dockerfile was created according to link. (

<https://github.com/blackred59/weatherForecastSampleApp/blob/main/Dockerfile>)

-

20 lines (17 sloc) | 527 Bytes

```
1 FROM mcr.microsoft.com/dotnet/aspnet:5.0 AS base
2 WORKDIR /app
3
4 FROM mcr.microsoft.com/dotnet/sdk:5.0 AS build
5 WORKDIR /src
6 COPY ["sample-app.csproj", "./"]
7 RUN dotnet restore "sample-app.csproj"
8 COPY . .
9 WORKDIR "/src/."
10 RUN dotnet build "sample-app.csproj" -c Release -o /src/build
11
12 FROM build AS publish
13 RUN dotnet publish "sample-app.csproj" -c Release -o /src/publish
14
15 FROM base AS final
16 WORKDIR /src
17 COPY --from=publish /src/publish .
18 EXPOSE 5000
19 ENV ASPNETCORE_URLS http://*:5000
20 ENTRYPOINT ["dotnet", "sample-app.dll"]
```

CI Pipeline

- I prefer Github actions. Because it is free for me.
- If the main branch is updated, it start.
- [https://github.com/blackred59/
weatherForecastSampleApp/blob/main/.github/
workflows/docker-image2.yml](https://github.com/blackred59/weatherForecastSampleApp/blob/main/.github/workflows/docker-image2.yml)



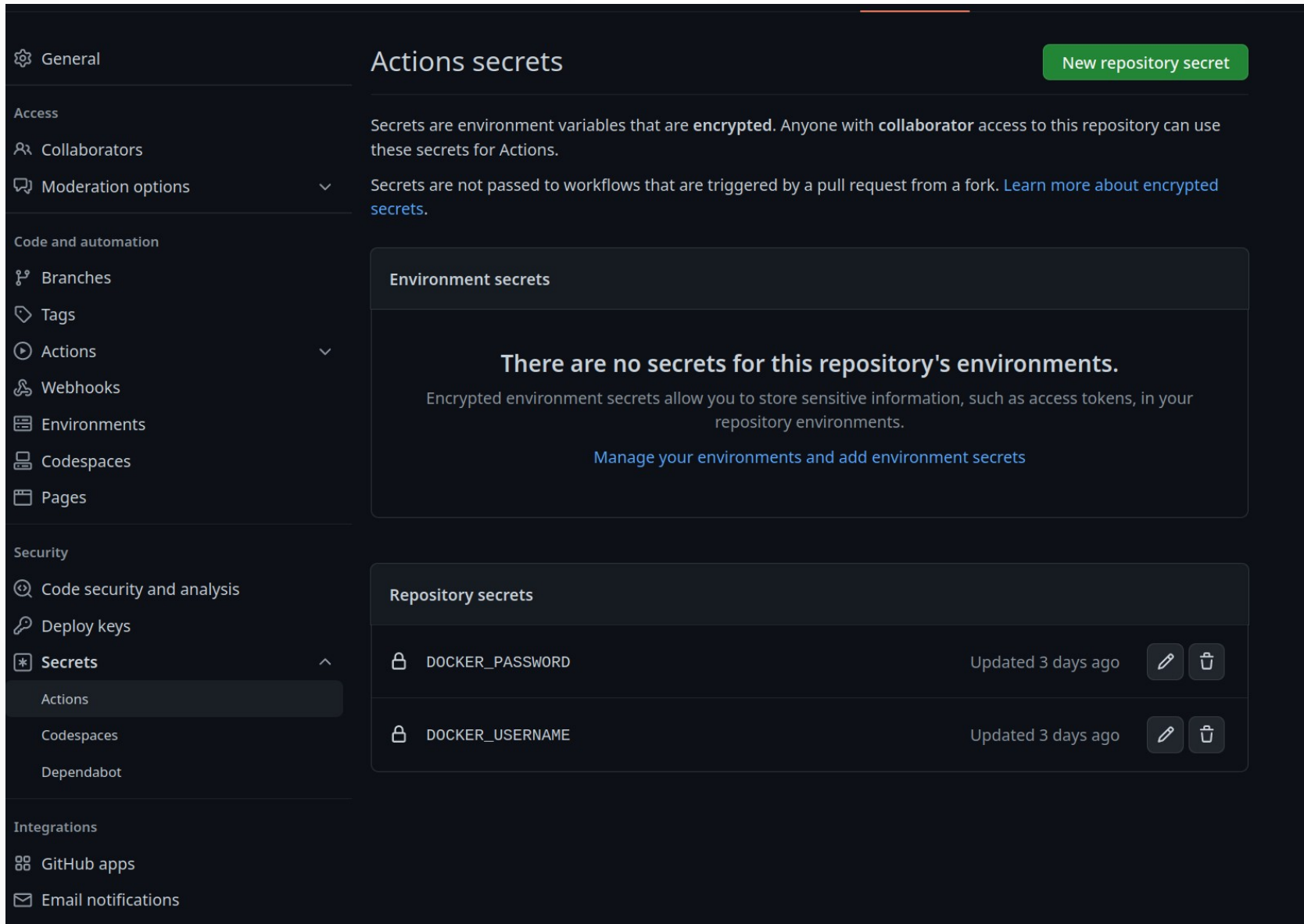
blackred59 Update docker-image2.yml ✓

1 contributor

37 lines (30 sloc) 1006 Bytes

```
1  name: Docker Image CI
2
3  on:
4    push:
5      branches: [ "main" ]
6    pull_request:
7      branches: [ "main" ]
8
9
10 jobs:
11   push_to_registry:
12     name: Push Docker image to Docker Hub
13     runs-on: ubuntu-latest
14     steps:
15       - name: Check out the repo
16         uses: actions/checkout@v3
17
18       - name: Log in to Docker Hub
19         uses: docker/login-action@f054a8b539a109f9f41c372932f1ae047eff08c9
20         with:
21           username: ${ secrets.DOCKER_USERNAME }
22           password: ${ secrets.DOCKER_PASSWORD }
23
24       - name: Extract metadata (tags, labels) for Docker
25         id: meta
26         uses: docker/metadata-action@98669ae865ea3cfffbcbaa878cf57c20bbf1c6c38
27         with:
28           images: blackred59/sample-app
29
30       - name: Build and push Docker image
31         uses: docker/build-push-action@ad44023a93711e3deb337508980b4b5e9bc5dc
32         with:
33           context: .
34           push: true
35           tags: ${ steps.meta.outputs.tags }
36           labels: ${ steps.meta.outputs.labels }
```


Docker Repo Credentials Configure



The screenshot shows the GitHub Actions secrets configuration page for a repository. The left sidebar contains navigation links for General, Access, Code and automation, Security, and Integrations. The 'Secrets' link under Security is highlighted. The main content area is titled 'Actions secrets' and includes a 'New repository secret' button. It explains that secrets are encrypted environment variables and provides instructions on how they are used. Below this, there are two sections: 'Environment secrets' and 'Repository secrets'. The 'Environment secrets' section states that there are no secrets for this repository's environments and provides a link to manage environments. The 'Repository secrets' section lists two secrets: 'DOCKER_PASSWORD' and 'DOCKER_USERNAME', both updated 3 days ago, with edit and delete icons.

General

Access

- Collaborators
- Moderation options

Code and automation

- Branches
- Tags
- Actions
- Webhooks
- Environments
- Codespaces
- Pages

Security

- Code security and analysis
- Deploy keys
- Secrets**
 - Actions
 - Codespaces
 - Dependabot

Integrations

- GitHub apps
- Email notifications

Actions secrets

[New repository secret](#)

Secrets are environment variables that are **encrypted**. Anyone with **collaborator** access to this repository can use these secrets for Actions.

Secrets are not passed to workflows that are triggered by a pull request from a fork. [Learn more about encrypted secrets.](#)







Environment secrets

There are no secrets for this repository's environments.

Encrypted environment secrets allow you to store sensitive information, such as access tokens, in your repository environments.

[Manage your environments and add environment secrets](#)

Repository secrets

 DOCKER_PASSWORD	Updated 3 days ago		
 DOCKER_USERNAME	Updated 3 days ago		



Add a short description for this repository

The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results.

[Update](#)

blackred59 / sample-app

Description

This repository does not have a description

Last pushed: 4 minutes ago

Docker commands

[Public View](#)

To push a new tag to this repository,

```
docker push blackred59/sample-app:tagname
```

Tags and scans

VULNERABILITY SCANNING - DISABLED [Enable](#)

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
main		Image	---	4 minutes ago

[See all](#)
[Go to Advanced Image Management](#)

Automated Builds


Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions.

[Upgrade](#)
[Learn more](#)

CD Pipeline with ArgoCD

- ArgoCD is a trend Gitops tool. I create CD pipeline with ArgoCD. My kubernetes configurations (<https://github.com/blackred59/weatherForecastSampleApp/tree/main/Kubernetes>)
- First I installed ArgoCD.

 **Argo CD**
v2.5.2+148d8da

[Applications](#)

[Settings](#)

[User Info](#)

[Documentation](#)

[←](#)


Applications


[+ NEW APP](#)

[↻ SYNC APPS](#)

[↻ REFRESH APPS](#)

[/](#)





No applications yet

Create new application to start managing resources in your cluster

[CREATE APPLICATION](#)

Project Creation

Google Translate

Azure-AytTech

Azure-Portal

GitHub

Docker Hub

Argo CD

v2.5.2+148d8da

Applications

Settings

User Info

Documentation

←

Applications

+ NEW APP

CREATE

CANCEL

GENERAL

Application Name

sampleapp

Project Name

default

SYNC POLICY

Automatic

☐ PRUNE RESOURCES

☐ SELF HEAL

☐ SET DELETION FINALIZER

SYNC OPTIONS

☐ SKIP SCHEMA VALIDATION

☐ PRUNE LAST

☐ RESPECT IGNORE DIFFERENCES

☐ AUTO-CREATE NAMESPACE

☐ APPLY OUT OF SYNC ONLY

☐ SERVER-SIDE APPLY

PRUNE PROPAGATION POLICY: foreground

☐ REPLACE

☐ RETRY

SOURCE

Repository URL

https://github.com/blackred59/weatherForecastSampleApp.git

GIT

Revision

main

Branches

Path

./Kubernetes/

DESTINATION

Cluster URL

https://kubernetes.default.svc

URL

Namespace

default

Sync



Argo CD

v2.5.2+148d8da

Applications

Settings

User Info

Documentation



FILTERS

☐ ★ Favorites Only

SYNC STATUS

☐ ○ Unknown 0

☐ ✓ Synced 1

☐ ⬆ OutOfSync 0

Applications

+ NEW APP

↻ SYNC APPS

↻ REFRESH APPS

🔍 Search applications



argocd/sampleapp



Project: default

Labels:

Status: ○ Progressing ✓ Synced

Repository: https://github.com/blackred59/weatherFor...

Target Re... main

Path: ./Kubernetes/

Destinati... in-cluster

Namespa... default

↻ SYNC

↻ REFRESH

✕ DELETE

Deployment was successful.

← → ↺ https://127.0.0.1:8080/applications/argocd/sampleapp?view=tree&resource= ☆

Google Translate Azure-AyitiTech Azure-Portal GitHub Docker Hub

Argo CD v2.5.2+148d8da

Applications Settings User Info Documentation

APPLICATIONS / sampleapp

APP DETAILS APP DIFF SYNC SYNC STATUS HISTORY AND ROLLBACK DELETE REFRESH

APP HEALTH **Progressing**

CURRENT SYNC STATUS **Synced** To main (ff5134d) MORE

LAST SYNC RESULT **Sync OK** To ff5134d MORE

Succeeded a few seconds ago (Mon Nov 21 2022 19:27:07 GMT+0300)
Author: blackred59 <117301119+blackred59@users.noreply.github.com> - Update deployment.yaml
Comment:

sampleapp a few seconds

rsweatherforecastdemo SVC a few seconds

rsweatherforecastdemo rsweatherforecastdemo-cdv66 endpointslice a few seconds

rsweatherforecastdemo rsweatherforecastdemo-5f6f... rs a few seconds rev:1

rsweatherforecastdemo-5f6f... pod a few seconds containercreating 0/1

rsweatherforecastdemo-5f6f... pod a few seconds containercreating 0/1

FILTERS

NAME

KINDS

SYNC STATUS

HEALTH STATUS

Synced 2

OutOfSync 0

Healthy 1

Access the service

- Service is running on 30007 NodePort.
- <http://127.0.0.1:30007/weatherforecast> is brings us api.



127.0.0.1:30007/weatherforecast

[Google Translate](#) [Azure-AytiTech](#) [Azure-Portal](#) [GitHub](#) [Docker Hub](#)

JSON Raw Data Headers

Save Copy Collapse All Expand All

```
▼ 0:
  date:      "2022-11-22T16:49:41.0785415+00:00"
  temperatureC: -3
  temperatureF: 27
  summary:    "Cool"
▼ 1:
  date:      "2022-11-23T16:49:41.0823455+00:00"
  temperatureC: 41
  temperatureF: 105
  summary:    "Bracing"
▼ 2:
  date:      "2022-11-24T16:49:41.0823495+00:00"
  temperatureC: -11
  temperatureF: 13
  summary:    "Bracing"
▼ 3:
  date:      "2022-11-25T16:49:41.0823498+00:00"
  temperatureC: -19
  temperatureF: -2
  summary:    "Cool"
▼ 4:
  date:      "2022-11-26T16:49:41.08235+00:00"
  temperatureC: 39
  temperatureF: 102
  summary:    "Warm"
```