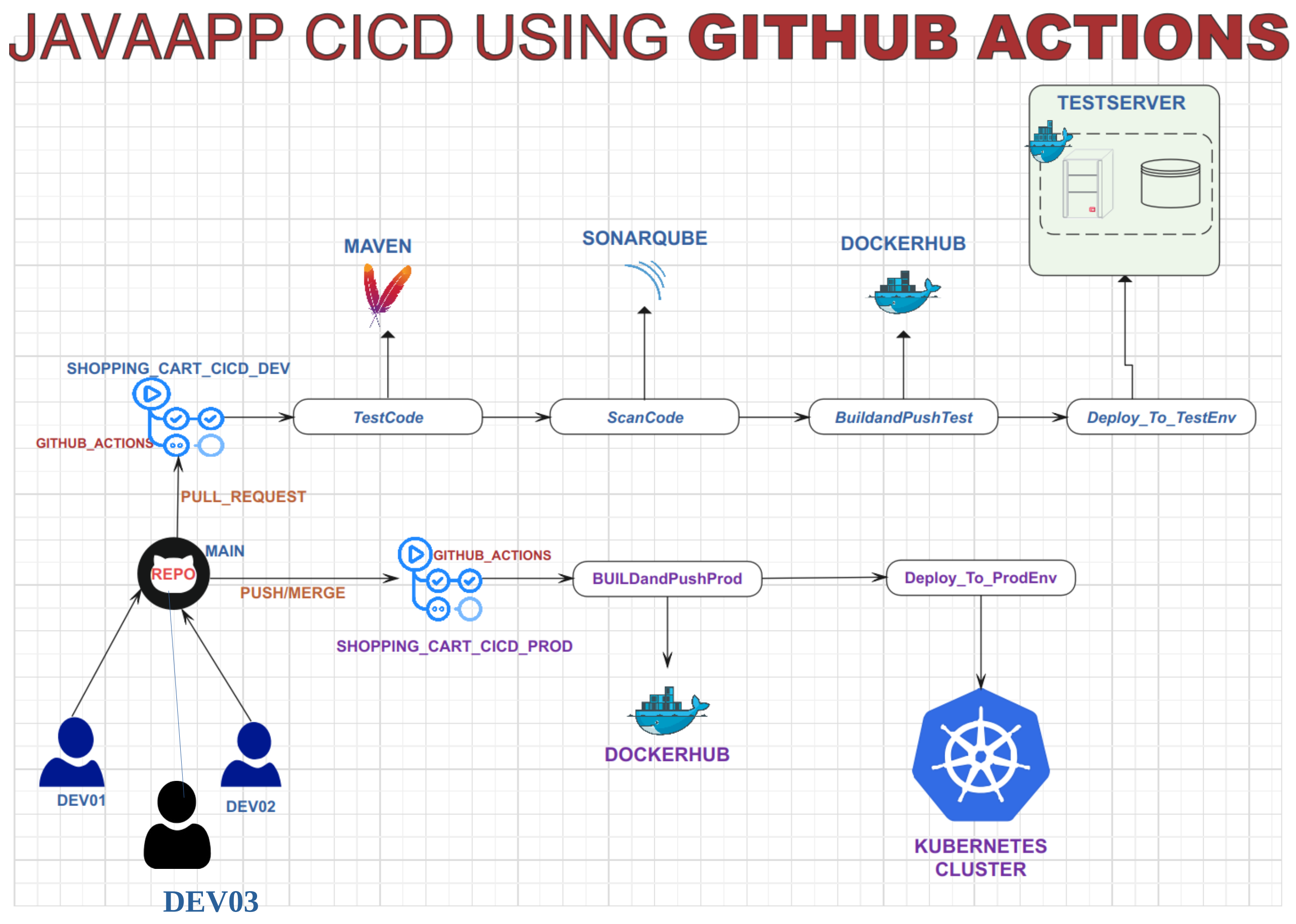
Project Title:JAVA APP CICD USING GITHUB ACTIONS

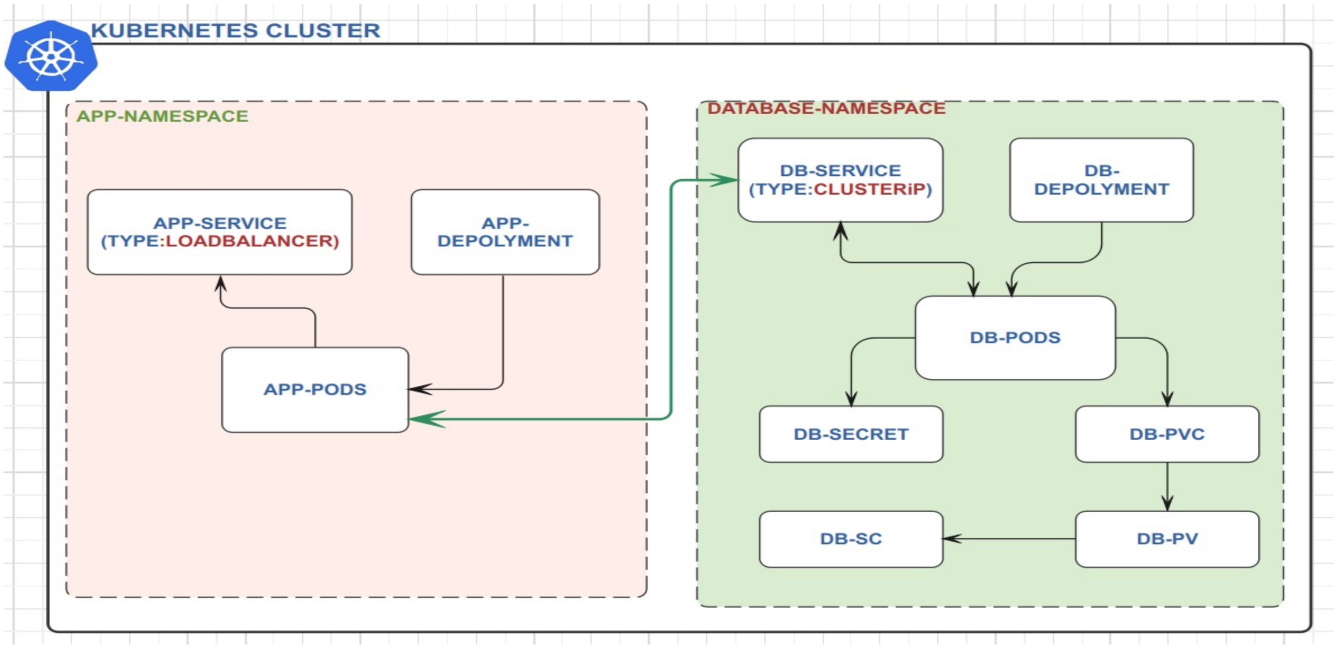
ProjectRepo:https://github.com/robudexIT/shopping_cart

Notes:The Java application use in this project is clone from this github repository: *https://github.com/shashirajraja/shopping-cart.git.*

Project Architecture:



Kubernetes Cluster Architecture:



Project Overview:

1. **Workflow Structure:** The project utilizes two GitHub workflows or pipelines: one dedicated to the **development/test** environment and the other to the **production** environment.

2. **Development/Test Pipeline:**

- This pipeline is triggered by **pull requests** to the **main** branch.
- It executes the following steps:
  1. Running tests using **Maven**.
  2. Performing additional code analysis using **SonarQube**.
  3. Building a **Docker** image for the test environment.
  4. Pushing the **Docker** image to **DockerHub**.
  5. Deploying the **Docker** image to the test environment.

3. **Production Pipeline:**

- This pipeline is triggered by **merges or pushes** to the **main** branch.
- It includes a **manual approval step**.
- Upon approval, the pipeline:
  1. Builds and pushes the production Docker image with **prod** tag.
  2. Deploys the production image to the **production environment**.

4. **Branch Protection**: Direct pushes to the main branch are prohibited to maintain code integrity and ensure changes go through the proper workflow.

5. **Team Structure:**

- The project team consists of three members:
  1. **Dev01:** Responsible for updating the code by initiating pull requests.
  2. **Dev02:**Authorized to review and merge code changes.
  3. **Dev03:** Responsible for granting **manual approval** in the CI/CD production pipeline.

6. **Test Environment:**

- Hosted on a single server running **Docker Engine**.
- Infrastructure provided by Digital Ocean.

7. **Production Environment:**

- Utilizes a Kubernetes cluster.
- Infrastructure provided by Digital Ocean.

Project Flow of Execution:

1. Create 3 github Users

robudex17 - Dev01:

1. **Role**: Initiates pull requests.
2. **Responsibilities**: Updating code and initiating pull requests for review

robudexIT - Dev02:

1. **Role**: Responsible for merging and pushing changes to the main branch.
2. **Responsibilities**: Reviews pull requests and merges approved changes into the main branch.

robudex2023 - Dev03:

1. **Role**: Responsible for **approving or rejecting** changes in the production workflow.
2. **Responsibilities**: Manually approves changes in the CI/CD production pipeline or requests adjustments if necessary.

Then I clone https://github.com/shashirajraja/shopping-cart.git java webapps project, create **DockerFile** and **Kubernetes manifiests** and push it to my **Dev02** users.

Current project tree. Shown below

```
├── databases
├── Dockerfile - this file will dockerirze the shopping-cart apps
├── kubernetes -  this folder contain my kubernetes manifest
├── LICENSE
├── pom.xml
├── README.md
├── src
└── WebContent
```

```
    Kubernetes Manifest
├── application
│   ├── shopping-cart-app-deployment.yaml
│   └── shopping-cart-app-service.yaml
├── database
│   ├── shopping-cart-db-deploment.yaml
│   ├── shopping-cart-db-service.yaml
│   ├── shopping-cart-pv.yaml
│   ├── shopping-cart-secrets.yaml
│   └── storage-class.yaml
│
```

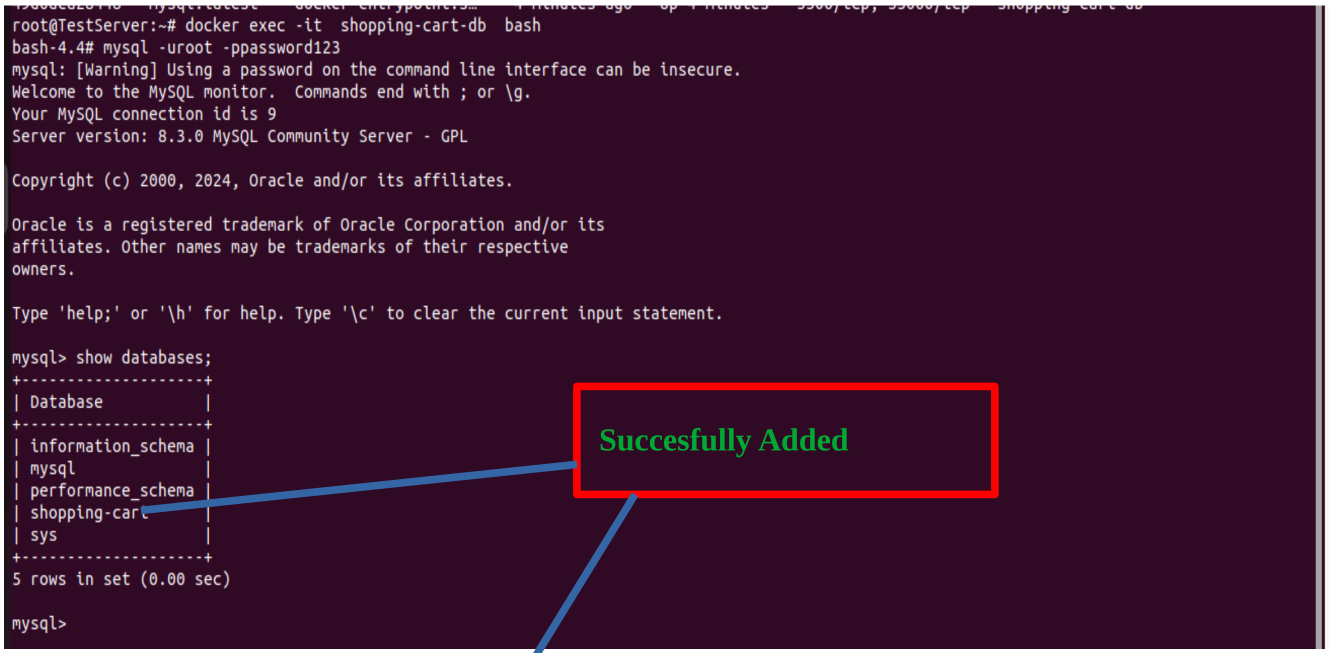
2.On the test server, I created the latest MySQL Docker container and restored the database of the project using these commands.

- cd /root
- git clone https://github.com/robudexIT/shopping_cart.git
- mkdir /root/mysqldb

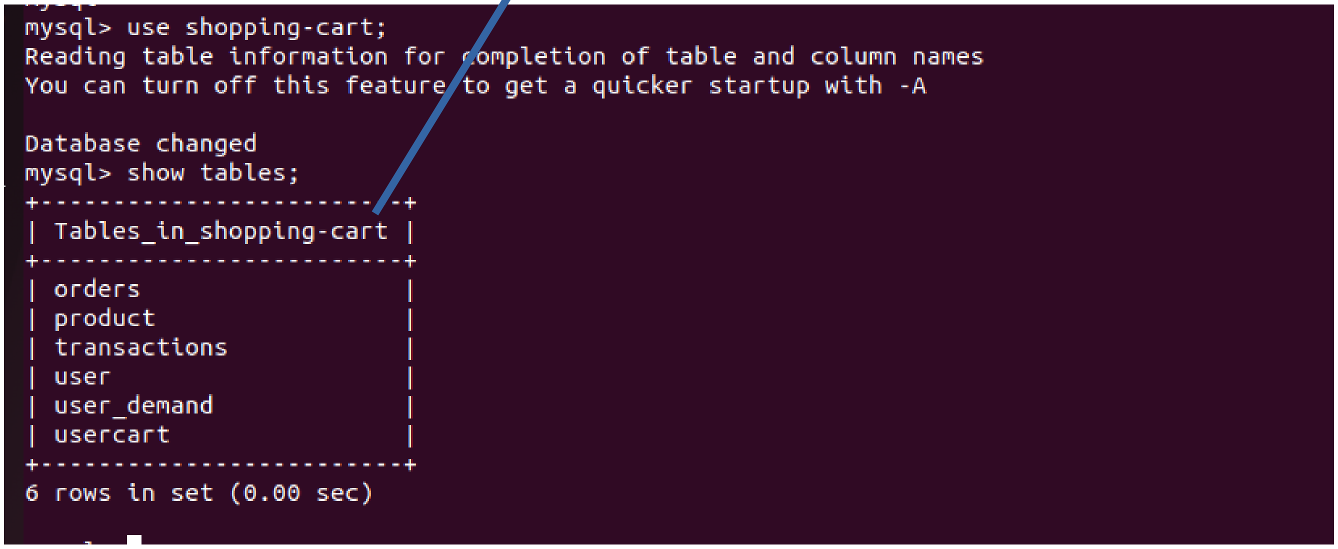
- docker run --name **shopping-cart-db** -v /root/mysqldb:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=p**assword123** -e MYSQL_DATABASE=**shopping-cart** -d **mysql:latest**
- docker exec -i shopping-cart-db sh -c 'exec mysql -uroot -ppassword123 shopping-cart' < /root/shopping_cart/databases/mysql_query.sql

Then check if database was properly restore by these commands:

- docker exec -it shopping-cart-db bash
- mysql -uroot -ppassword123
- show databases;
- use shopping-cart;
- show tables;



```
root@TestServer:~# docker exec -it  shopping-cart-db  bash
bash-4.4# mysql -uroot -ppassword123
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.3.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| shopping-cart      |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql>
```

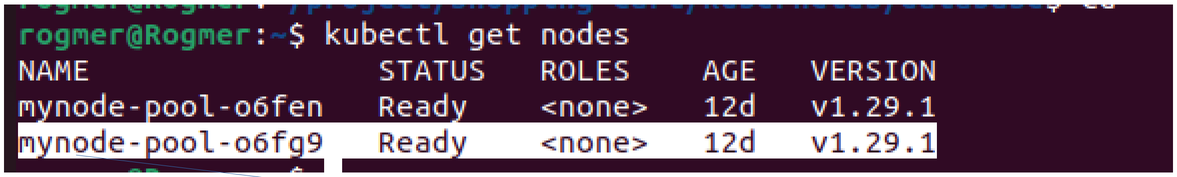


```
mysql> use shopping-cart;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-------------------------+
| Tables_in_shopping-cart |
+-------------------------+
| orders                  |
| product                 |
| transactions            |
| user                    |
| user_demand             |
| usercart                |
+-------------------------+
6 rows in set (0.00 sec)
```

3. On My Kubernetes Cluster I Create Mysql Deployment and Service

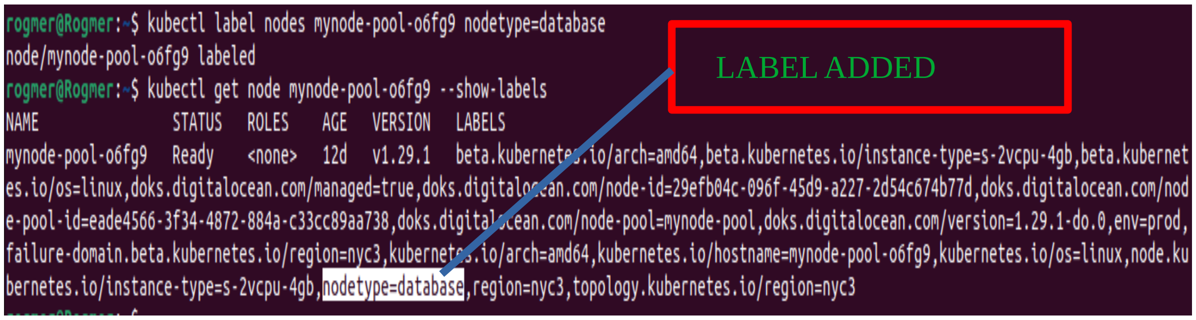
Note: I will utilize one of my Kubernetes nodes' storage in a **PersistentVolume**, ensuring that MySQL deploys on that node. This will be achieved by adding a **label** to the node and using this label in the deployment. Here are the commands:

- `kubectl get nodes`



```
rogmer@Rogmer:~$ kubectl get nodes
NAME                 STATUS   ROLES    AGE   VERSION
mynode-pool-o6fen    Ready    <none>   12d   v1.29.1
mynode-pool-o6fg9    Ready    <none>   12d   v1.29.1
```

- kubectl label nodes **mynode-pool-o6fg9** nodetype=database
- kubectl get node **mynode-pool-o6fg9** --show-labels



```
rogmer@Rogmer:~$ kubectl label nodes mynode-pool-o6fg9 nodetype=database
node/mynode-pool-o6fg9 labeled
rogmer@Rogmer:~$ kubectl get node mynode-pool-o6fg9 --show-labels
NAME                STATUS   ROLES    AGE   VERSION   LABELS
mynode-pool-o6fg9   Ready    <none>   12d   v1.29.1   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/instance-type=s-2vcpu-4gb,beta.kubernet
es.io/os=linux,doks.digitalocean.com/managed=true,doks.digitalocean.com/node-id=29efb04c-096f-45d9-a227-2d54c674b77d,doks.digitalocean.com/nod
e-pool-id=eade4566-3f34-4872-884a-c33cc89aa738,doks.digitalocean.com/node-pool=mynode-pool,doks.digitalocean.com/version=1.29.1-do.0,env=prod,
failure-domain.beta.kubernetes.io/region=nyc3,kubernetes.io/arch=amd64,kubernetes.io/hostname=mynode-pool-o6fg9,kubernetes.io/os=linux,node.ku
bernetes.io/instance-type=s-2vcpu-4gb,nodetype=database,region=nyc3,topology.kubernetes.io/region=nyc3
```

I like the separation between the **app** and the **database** so I created namespace

- kubectl create namespace **database-namespace**
- kubectl create namespace **app-namespace**

Clone git https://github.com/robudexIT/shopping_cart.git and Navigate (cd) to **shopping-cart/kubernetes/database** and run the command in order

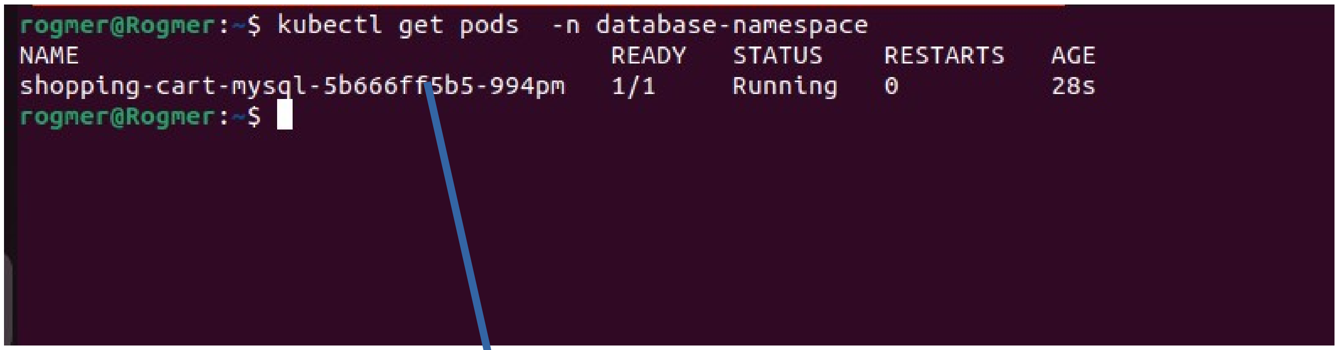
- kubectl apply -f storage-class.yaml -n **database-namespace**
- kubectl apply -f shopping-cart-pv.yaml -n **database-namespace**
- kubectl apply -f shopping-cart-pvc.yaml -n **database-namespace**
- kubectl apply -f shopping-cart-db-deploment.yaml -n **database-namespace**
- kubectl apply -f shopping-cart-db-service.yaml -n **database-namespace**

Verify Kubernetes Objects

- kubectl get namespace
- kubectl get sc -n **database-namespace**
- kubectl get pv -n **database-namespace**
- kubectl get pvc -n **database-namespace**
- kubectl get deployment -n **database-namespace**
- kubectl get service -n **database-namespace**
- kubectl get pods -n **database-namespace**

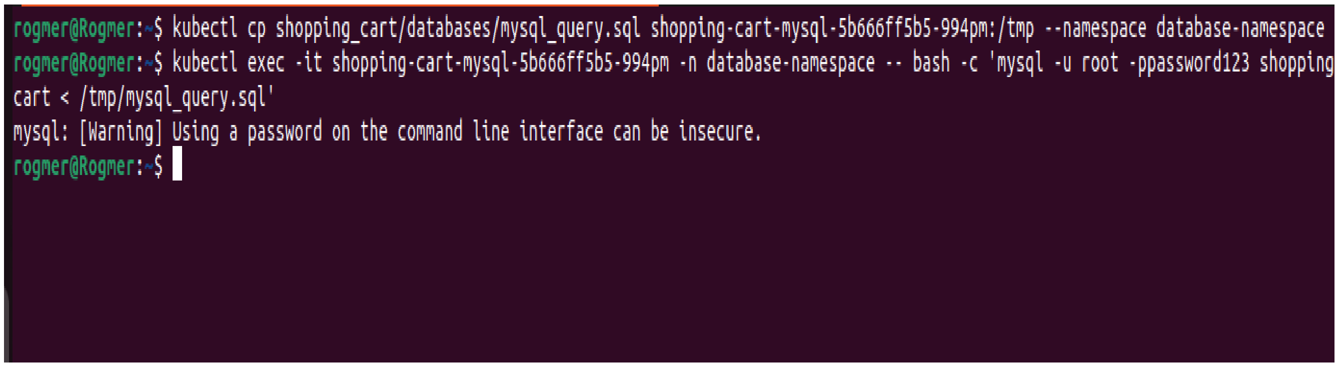
4. Restore shopping-cart database to shopping-cart-mysql pod with commands

- kubectl get pods -n **database-namespace**



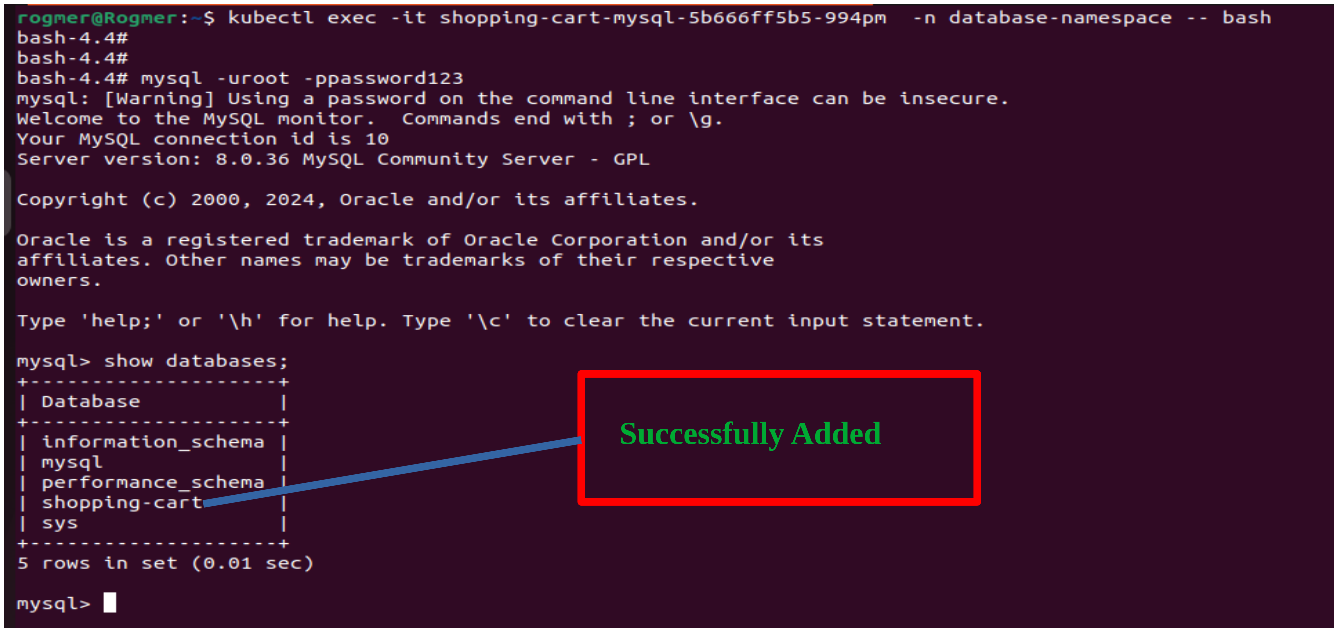
```
rogmer@Rogmer:~$ kubectl get pods  -n database-namespace
NAME                                READY   STATUS    RESTARTS   AGE
shopping-cart-mysql-5b666ff5b5-994pm   1/1     Running   0          28s
rogmer@Rogmer:~$ 
```

- kubectl cp shopping_cart/databases/mysql_query.sql **shopping-cart-mysql-5b666ff5b5-994pm**:/tmp --namespace **database-namespace**
- kubectl exec -it **shopping-cart-mysql-5b666ff5b5-994pm** -n database-namespace -- bash -c 'mysql -u root -ppassword123 shopping-cart < /tmp/mysql_query.sql'

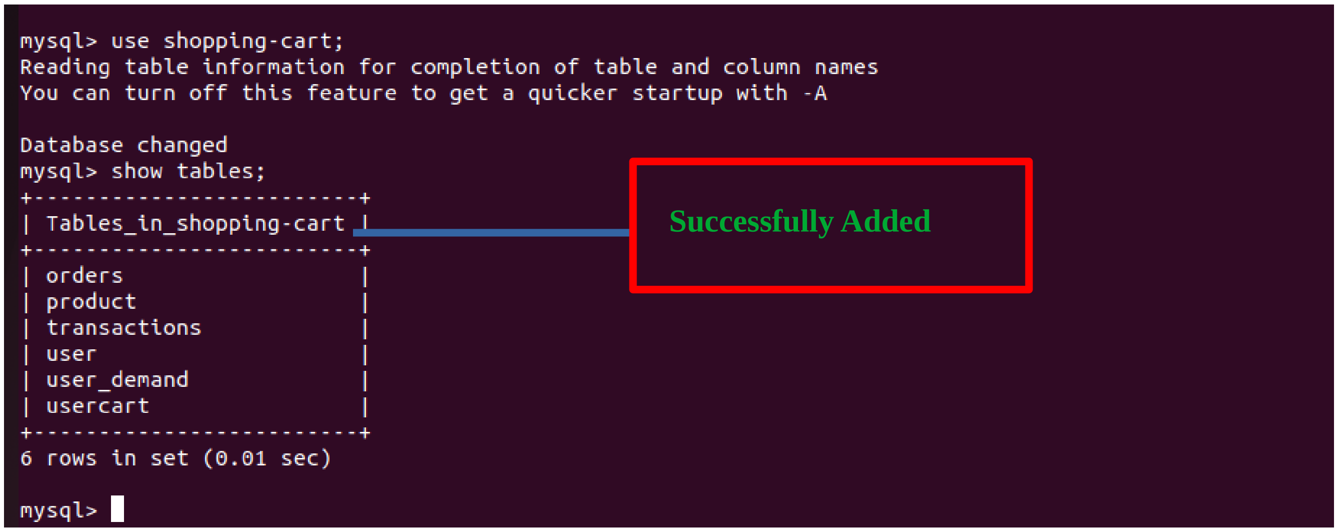


```
rogmer@Rogmer:~$ kubectl cp shopping_cart/databases/mysql_query.sql shopping-cart-mysql-5b666ff5b5-994pm:/tmp --namespace database-namespace
rogmer@Rogmer:~$ kubectl exec -it shopping-cart-mysql-5b666ff5b5-994pm -n database-namespace -- bash -c 'mysql -u root -ppassword123 shopping
cart < /tmp/mysql_query.sql'
mysql: [Warning] Using a password on the command line interface can be insecure.
rogmer@Rogmer:~$ 
```

- kubectl exec -it **shopping-cart-mysql-5b666ff5b5-994pm** -n database-namespace – bash
- mysql -uroot -ppassword123
- show databases;
- use shopping-cart;
- show tables;



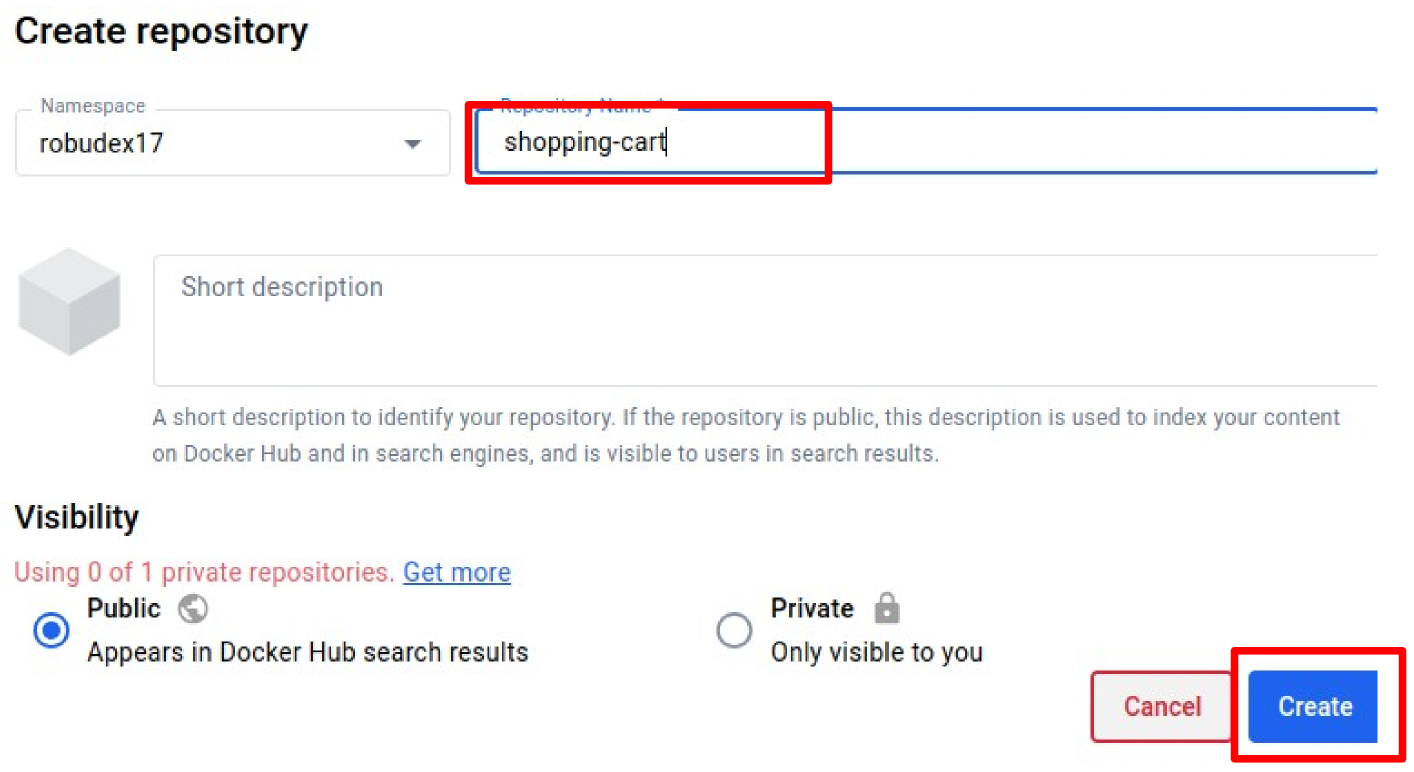
```
rogmer@Rogmer:~$ kubectl exec -it shopping-cart-mysql-5b666ff5b5-994pm  -n database-namespace -- bash
bash-4.4#
bash-4.4#
bash-4.4# mysql -uroot -ppassword123
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.36 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| shopping-cart      |
| sys                |
+--------------------+
5 rows in set (0.01 sec)

mysql> 
```



```
mysql> use shopping-cart;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+------------------------+
| Tables_in_shopping-cart |
+------------------------+
| orders                 |
| product                |
| transactions           |
| user                   |
| user_demand            |
| usercart               |
+------------------------+
6 rows in set (0.01 sec)

mysql> 
```

5. Now that the MySQL databases are installed on the **test server** and in the **Kubernetes cluster**, it's time to set up the **DockerHub repository**, **SonarQube (organization**, **project**, **quality gates** and **token**).

DockerHub:



Create repository

Namespace: robudex17

Repository Name: shopping-cart

Short description

A short description to identify your repository. If the repository is public, this description is used to index your content on Docker Hub and in search engines, and is visible to users in search results.

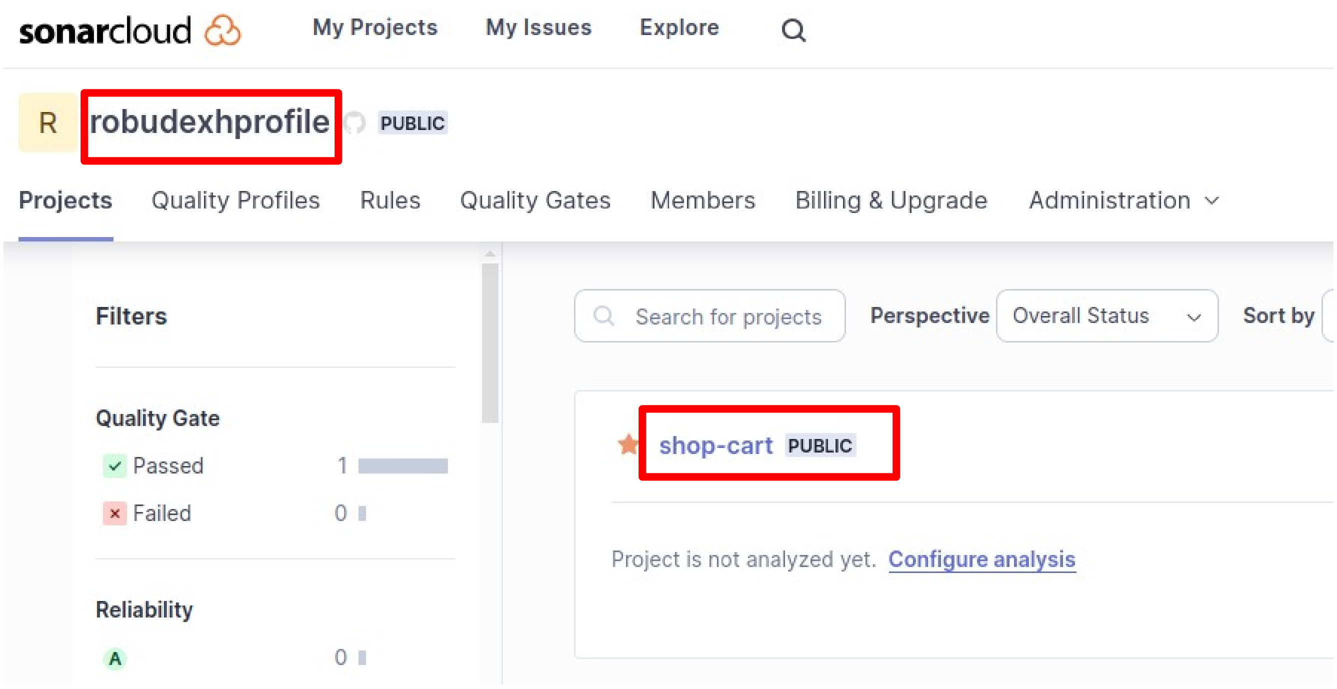
Visibility

Using 0 of 1 private repositories. Get more

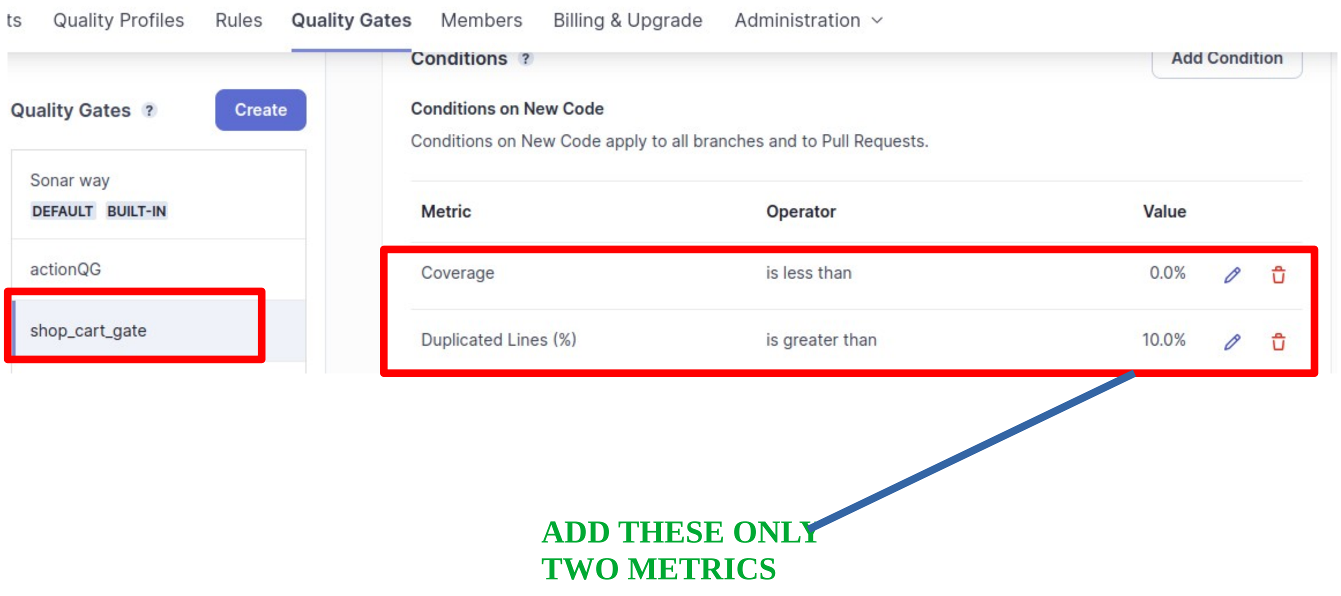
[✓ Public] Appears in Docker Hub search results

Private Only visible to you

Sonarqube Organization and Project



Quality Gates have been added with the name "**shop_cart_gate**". I have configured it to be less restrictive setting **coverage to 0.0%** and **duplicated lines to 10.0%**. Please note that this configuration is not recommended and is solely for demonstration purposes.

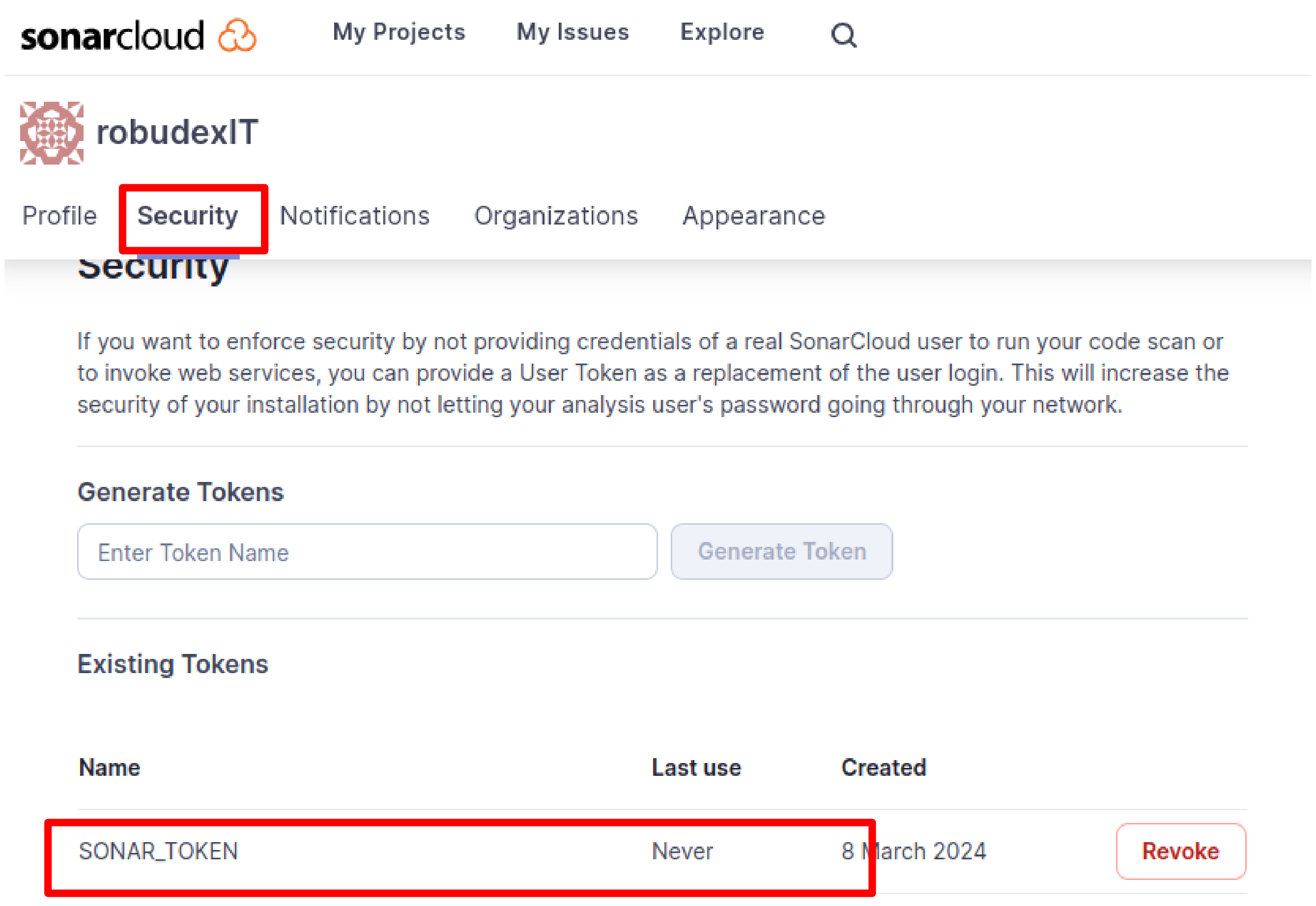


Conditions on New Code

Conditions on New Code apply to all branches and to Pull Requests.

Metric	Operator	Value
Coverage	is less than	0.0%
Duplicated Lines (%)	is greater than	10.0%

Add Token


sonarcloud

My Projects My Issues Explore

robudexIT

Profile Security Notifications Organizations Appearance

Security

If you want to enforce security by not providing credentials of a real SonarCloud user to run your code scan or to invoke web services, you can provide a User Token as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.

Generate Tokens

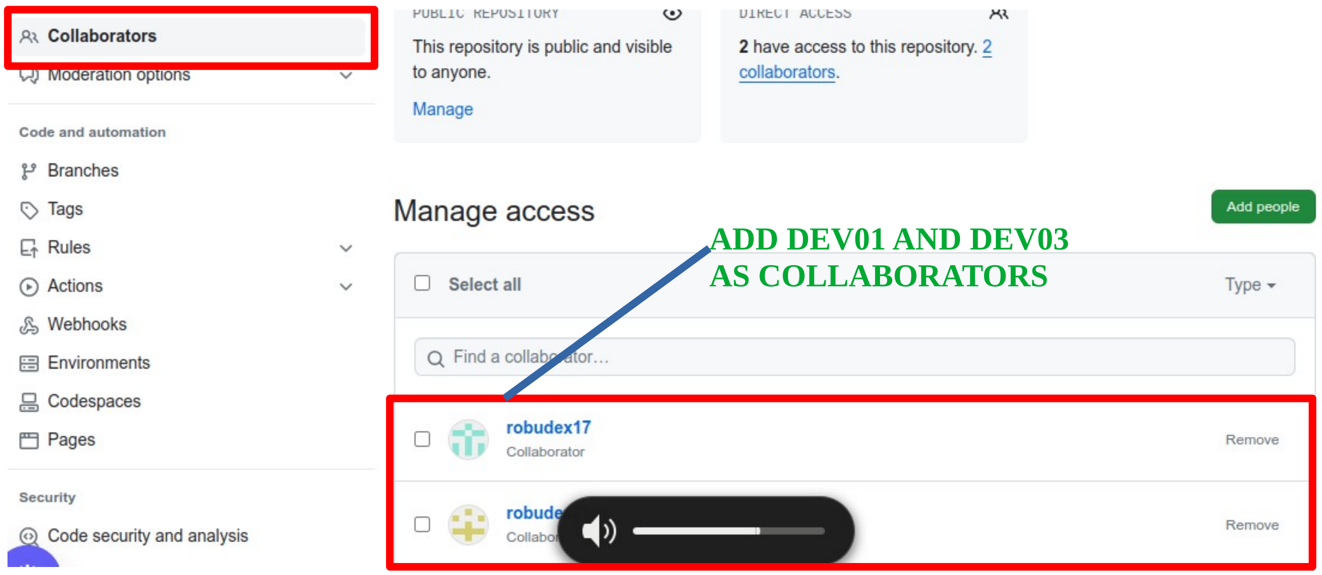
Enter Token Name | Generate Token

Existing Tokens

Name	Last use	Created	
SONAR_TOKEN	Never	8 March 2024	Revoke

6. Configure **Repo Settings**, Add **Production Environment**, **Secrets** And **Github Workflows/Pipelines**

Add **Dev01** and **Dev03** as Collaborators:

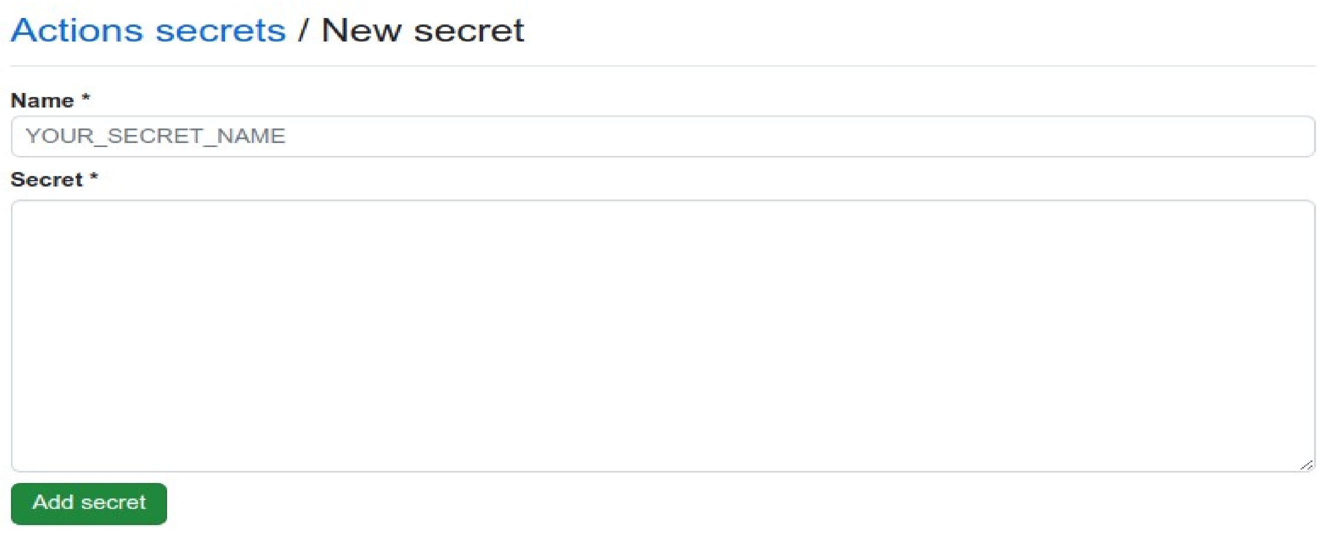

Collaborators
Moderation options
Code and automation
Branches
Tags
Rules
Actions
Webhooks
Environments
Codespaces
Pages
Security
Code security and analysis

PUBLIC REPOSITORY
This repository is public and visible to anyone.
Manage

DIRECT ACCESS
2 have access to this repository. 2 collaborators.

Manage access
Add people
Select all
Type
Find a collaborator…
robudex17
Collaborator
Remove
robude
Collabo
Remove

Next I add the secrets and production environment. (Settings->
Security>Secrets and Variables->Actions)

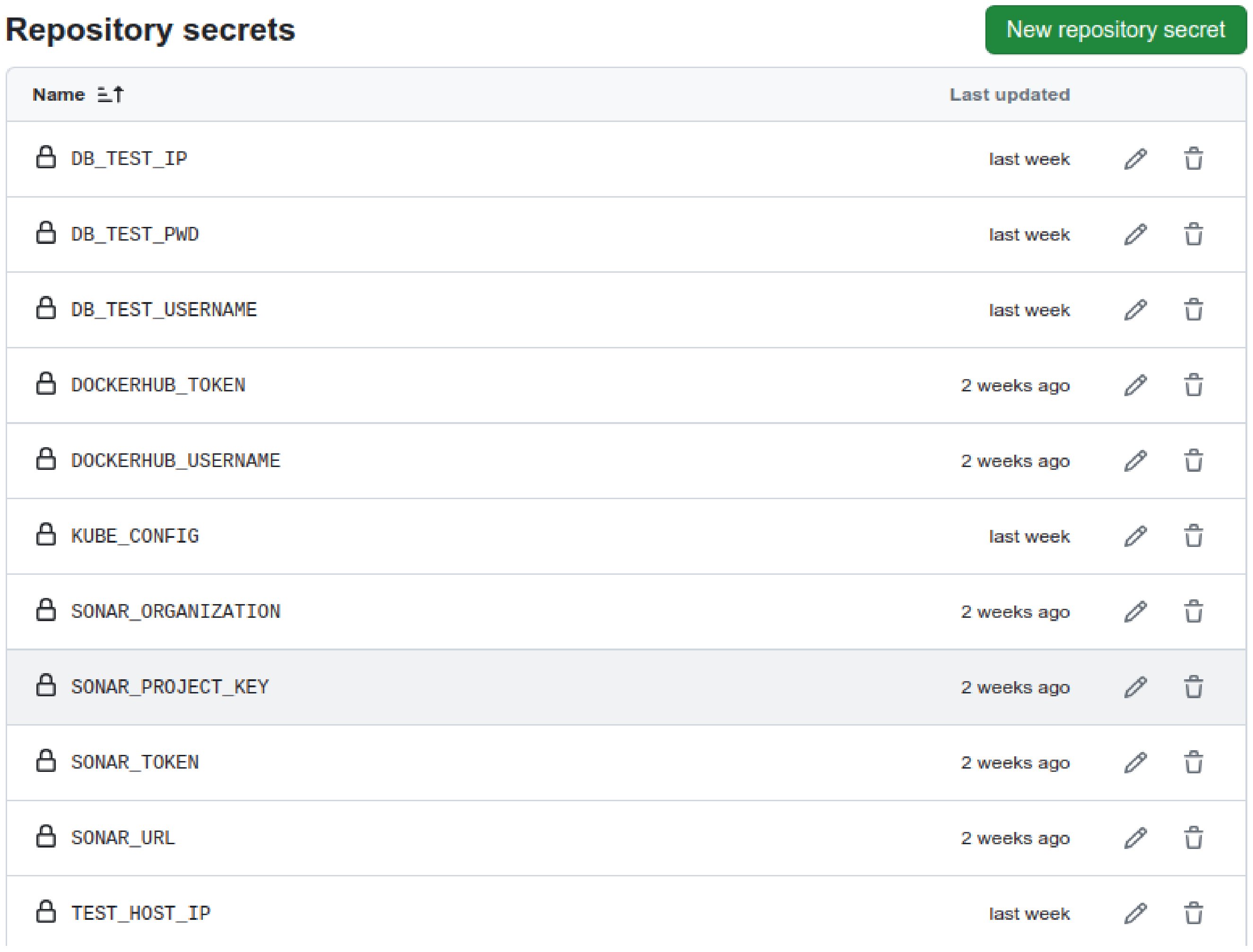

Actions secrets / New secret

Name *

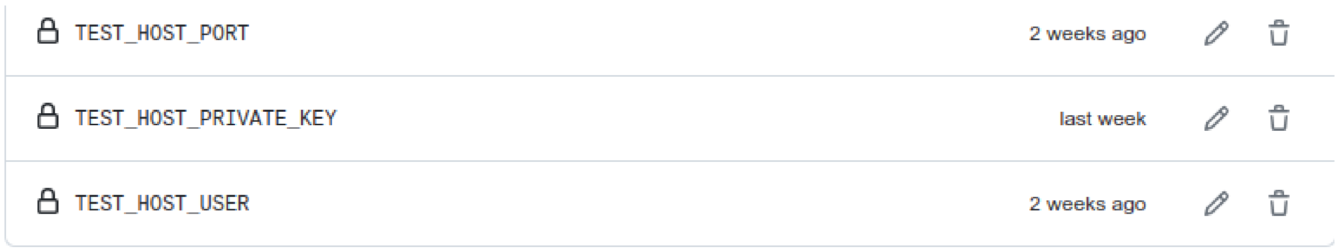
YOUR_SECRET_NAME

Secret *

These are secrets that I added

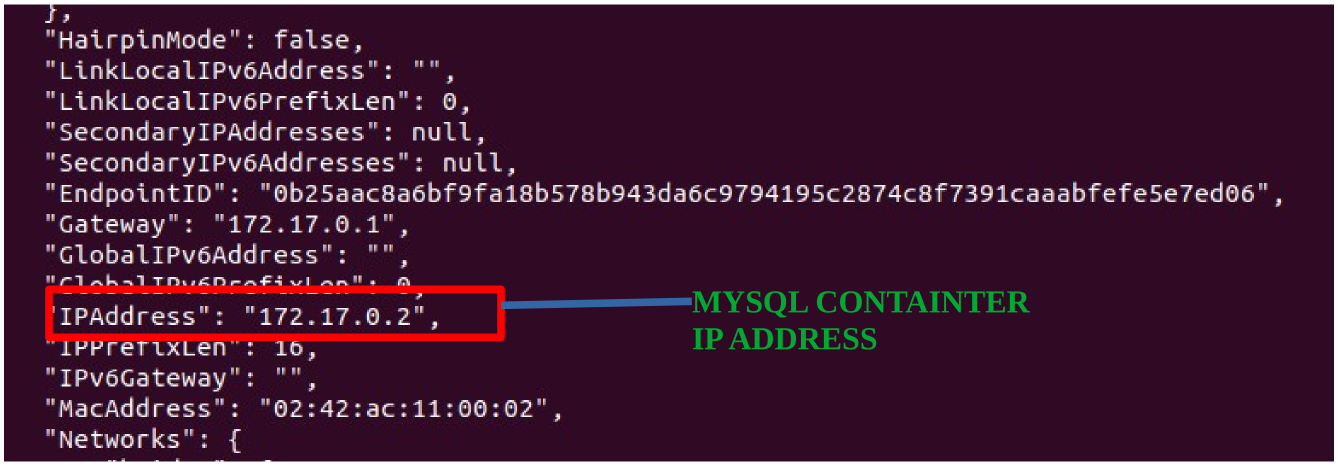

Repository secrets

Name	Last updated
DB_TEST_IP	last week
DB_TEST_PWD	last week
DB_TEST_USERNAME	last week
DOCKERHUB_TOKEN	2 weeks ago
DOCKERHUB_USERNAME	2 weeks ago
KUBE_CONFIG	last week
SONAR_ORGANIZATION	2 weeks ago
SONAR_PROJECT_KEY	2 weeks ago
SONAR_TOKEN	2 weeks ago
SONAR_URL	2 weeks ago
TEST_HOST_IP	last week



TEST_HOST_PORT	2 weeks ago	
TEST_HOST_PRIVATE_KEY	last week	
TEST_HOST_USER	2 weeks ago	

DB_TEST_IP - mysql docker container **IP Address** running on **Test Server**.
You can get docker containter ip address by (**docker inspect <container-name>/container-id**)



```
},
"HairpinMode": false,
"LinkLocalIPv6Address": "",
"LinkLocalIPv6PrefixLen": 0,
"SecondaryIPAddresses": null,
"SecondaryIPv6Addresses": null,
"EndpointID": "0b25aac8a6bf9fa18b578b943da6c9794195c2874c8f7391caaabfefe5e7ed06",
"Gateway": "172.17.0.1",
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"IPAddress": "172.17.0.2",
"IPPrefixLen": 16,
"IPv6Gateway": "",
"MacAddress": "02:42:ac:11:00:02",
"Networks": {
```

DB_TEST_PWD - mysql docker container password running on Test Server

DB_TEST_USERNAME - mysql docker container root user running on Test Server

DOCKERHUB_USERNAME – dockerhub username

DOCKERHUB_TOKEN – dockerhub password

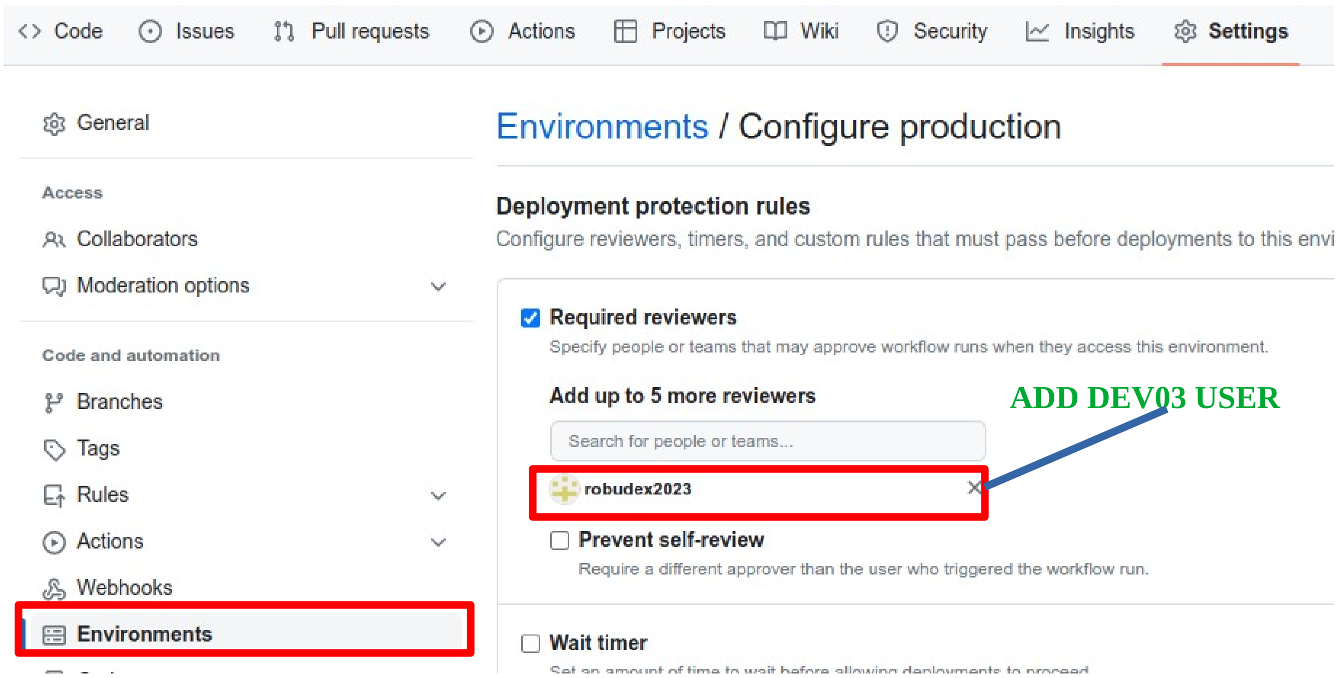
KUBE_CONFIG – kubernetes cluster configuration you can get this in **.kube/config**

SONAR_ORGANIZATION – your sonarqube organization on this project
SONAR_PROJECT_KEY – your sonarqube project key
SONAR_TOKEN - your sonarqube token
SONAR_URL – sonarqube url (https://sonarcloud.io)

TEST_HOST_IP – IP address of the Test Server
TEST_HOST_PORT – Test Server SSH port

TEST_HOST_PRIVATE_KEY - Test Server Private Key
You can generate key using (**ssh-keygen**).
TEST_HOST_USER – Test Server User (root)

Create **Production** Environment and Add **Dev03** as reviewer. Production pipeline will not run without the **approval** of **Dev03**



Environments / Configure production

Deployment protection rules

Configure reviewers, timers, and custom rules that must pass before deployments to this envi

☑ Required reviewers

Specify people or teams that may approve workflow runs when they access this environment.

Add up to 5 more reviewers

Search for people or teams...

robudex2023

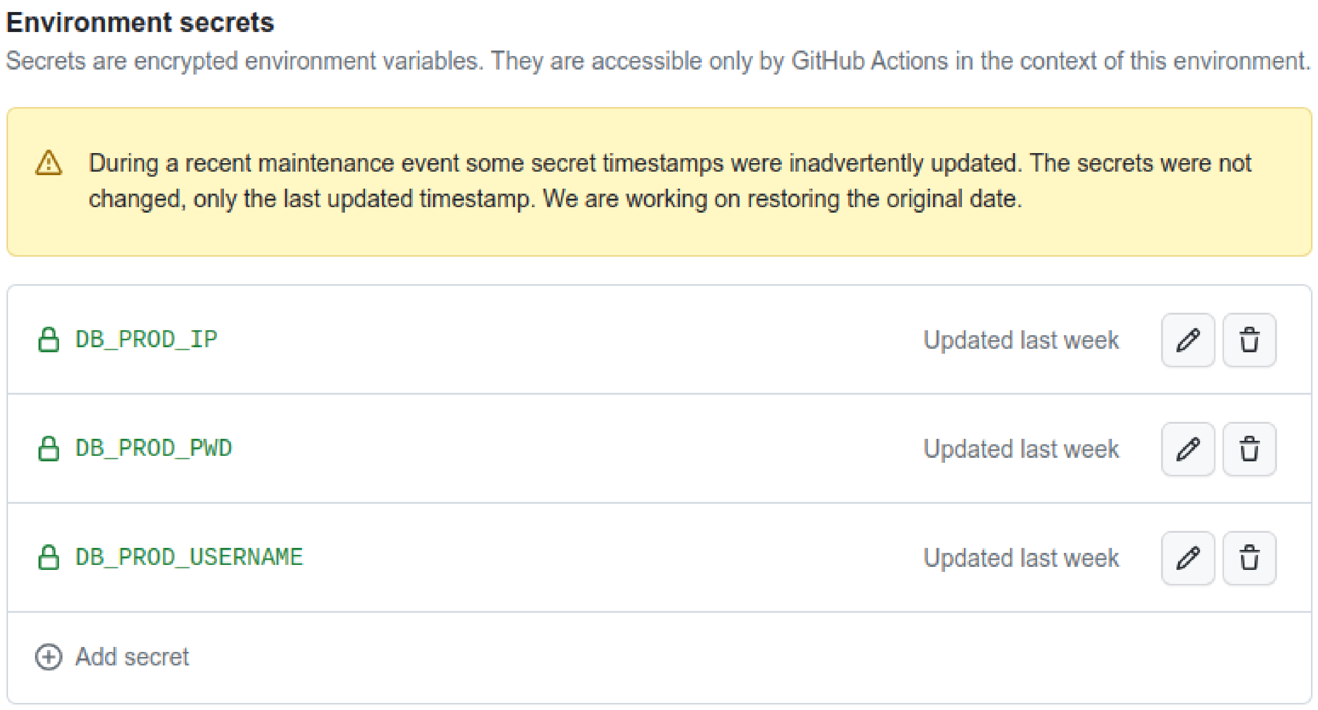
☐ Prevent self-review

Require a different approver than the user who triggered the workflow run.

☐ Wait timer

Set an amount of time to wait before allowing deployments to proceed

Production Environment Secrets



Environment secrets

Secrets are encrypted environment variables. They are accessible only by GitHub Actions in the context of this environment.

⚠ During a recent maintenance event some secret timestamps were inadvertently updated. The secrets were not changed, only the last updated timestamp. We are working on restoring the original date.

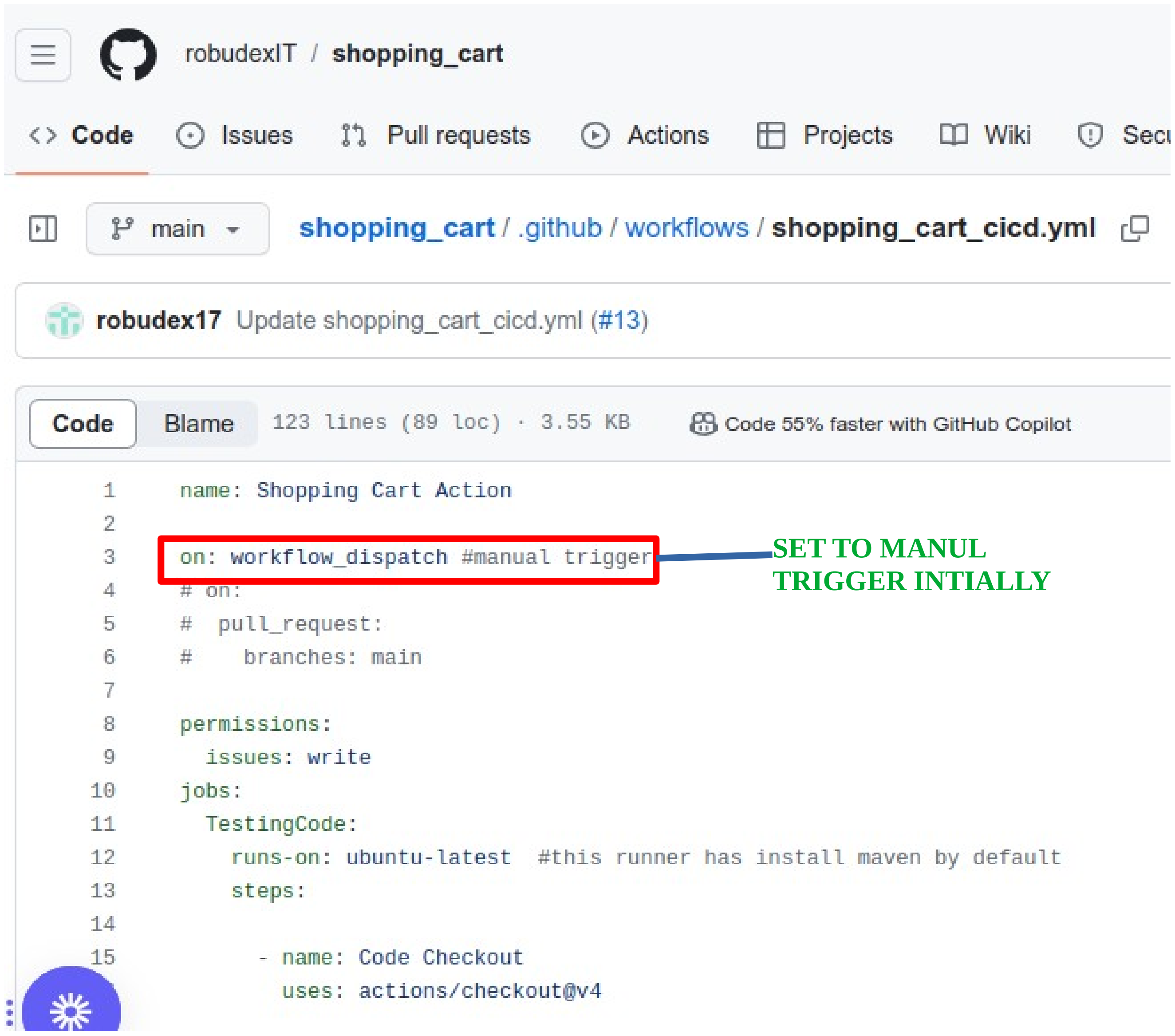
DB_PROD_IP	Updated last week
DB_PROD_PWD	Updated last week
DB_PROD_USERNAME	Updated last week

DB_PROD_IP - mysql Pod IP address in kubernets cluster
DB_PROD_USERNAME - mysql Pod username in kubernetes cluster
DB_PROD_PWD- mysql Pod password in kubernetes cluster

Creating Test/Dev Pipelines:

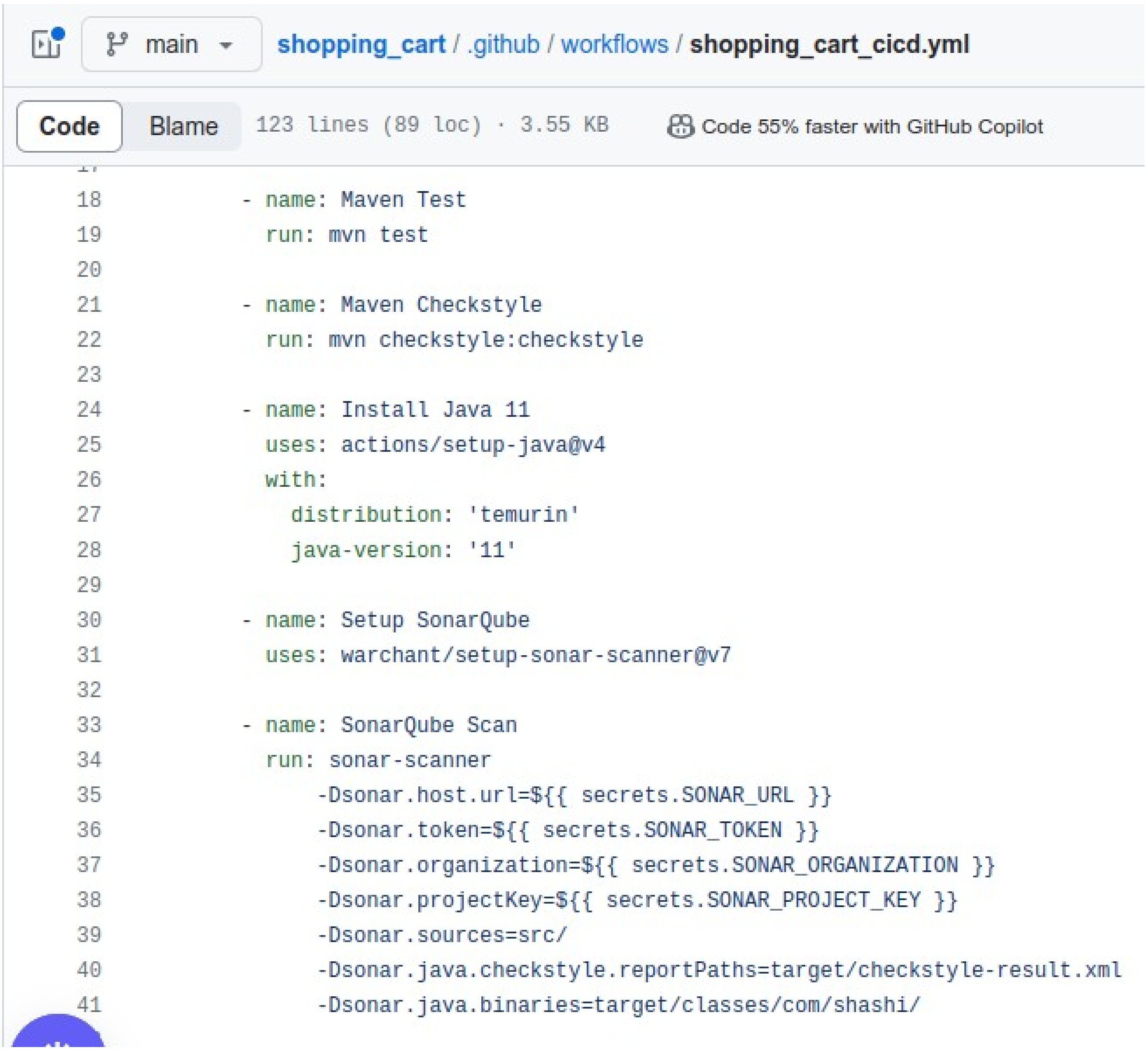
- Now that All secrets and environment are setup its time to create Pipelines

On My **Dev02** shopping_cart repo, I created new blank workflow name it **shopping_cart_cicd.yml**


robudexIT / shopping_cart
Code
Issues
Pull requests
Actions
Projects
Wiki
Secu
main
shopping_cart / .github / workflows / shopping_cart_cicd.yml
robudex17 Update shopping_cart_cicd.yml (#13)
Code
Blame
123 lines (89 loc) · 3.55 KB
Code 55% faster with GitHub Copilot

```yaml
name: Shopping Cart Action

on: workflow_dispatch #manual trigger
# on:
#  pull_request:
#    branches: main

permissions:
  issues: write
jobs:
  TestingCode:
    runs-on: ubuntu-latest  #this runner has install maven by default
    steps:

      - name: Code Checkout
        uses: actions/checkout@v4
```

SET TO MANUL TRIGGER INTIALLY



main

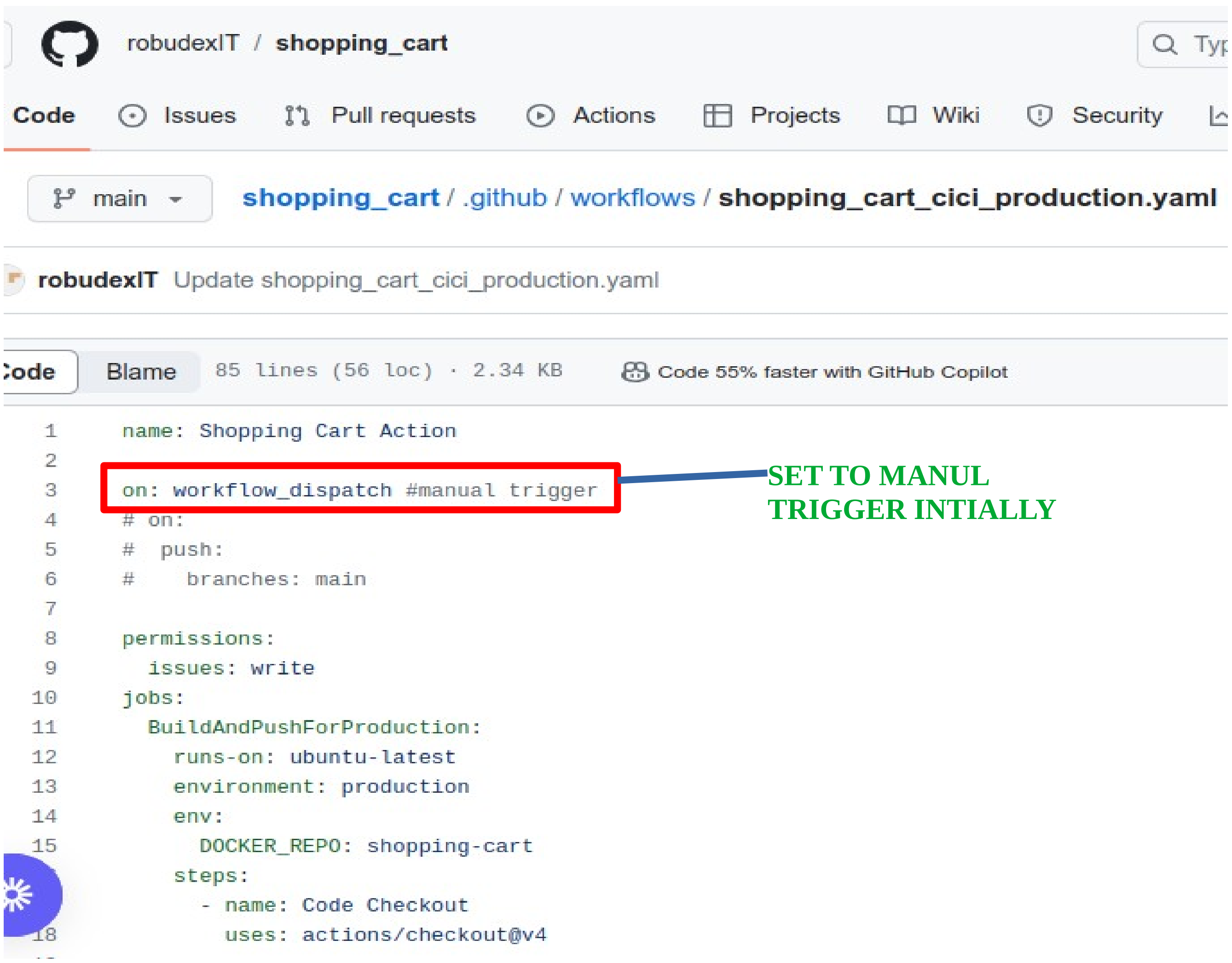
shopping_cart / .github / workflows / shopping_cart_cicd.yml

Code Blame 123 lines (89 loc) · 3.55 KB Code 55% faster with GitHub Copilot

```yaml
      - name: Maven Test
        run: mvn test

      - name: Maven Checkstyle
        run: mvn checkstyle:checkstyle

      - name: Install Java 11
        uses: actions/setup-java@v4
        with:
          distribution: 'temurin'
          java-version: '11'

      - name: Setup SonarQube
        uses: warchant/setup-sonar-scanner@v7

      - name: SonarQube Scan
        run: sonar-scanner
            -Dsonar.host.url=${{ secrets.SONAR_URL }}
            -Dsonar.token=${{ secrets.SONAR_TOKEN }}
            -Dsonar.organization=${{ secrets.SONAR_ORGANIZATION }}
            -Dsonar.projectKey=${{ secrets.SONAR_PROJECT_KEY }}
            -Dsonar.sources=src/
            -Dsonar.java.checkstyle.reportPaths=target/checkstyle-result.xml
            -Dsonar.java.binaries=target/classes/com/shashi/
      - name: SonarQube Quality Gate Check
        id: sonarqube-quality-gate-check
        uses: sonarsource/sonarqube-quality-gate-action@master
        timeout-minutes: 5
        env:
          SONAR_TOKEN: ${{ secrets.SONAR_TOKEN }}
          SONAR_HOST_URL: ${{ secrets.SONAR_URLL }}

  BuildAndPushForTesting:
    runs-on: ubuntu-latest
    needs: TestingCode
    env:
      DOCKER_REPO: shopping-cart
    steps:
      - name: Code Checkout
        uses: actions/checkout@v4

      - name: Update application.properties file
        env:
           DB_TEST_IP: ${{ secrets.DB_TEST_IP }}
           DB_TEST_USERNAME: ${{ secrets.DB_TEST_USERNAME }}
           DB_TEST_PWD: ${{ secrets.DB_TEST_PWD }}
        run: |
          sed -i "s|db.connectionString =.*|db.connectionString = jdbc:mysql://$DB_TEST_IP:3306/shopping-cart|"  src/application.properties
          sed -i "s|db.username =.*|db.username = $DB_TEST_USERNAME|" src/application.properties
          sed -i "s|db.password =.*|db.password = $DB_TEST_PWD|" src/application.properties

      - name: Login to Docker Hub
        uses: docker/login-action@v3
        with:
          username: ${{ secrets.DOCKERHUB_USERNAME }}
          password: ${{ secrets.DOCKERHUB_TOKEN }}

      - name: Build and Push docker image to Dockerhub
        uses: docker/build-push-action@v5
        with:
          context: ./
          push: true
          tags: ${{ secrets.DOCKERHUB_USERNAME }}/${{env.DOCKER_REPO}}:test
           # - ${{ secrets.DOCKERHUB_USERNAME }}/${{ DOCKER_REPO }}:${{GITHUB_RUN_NUMBER}}
```

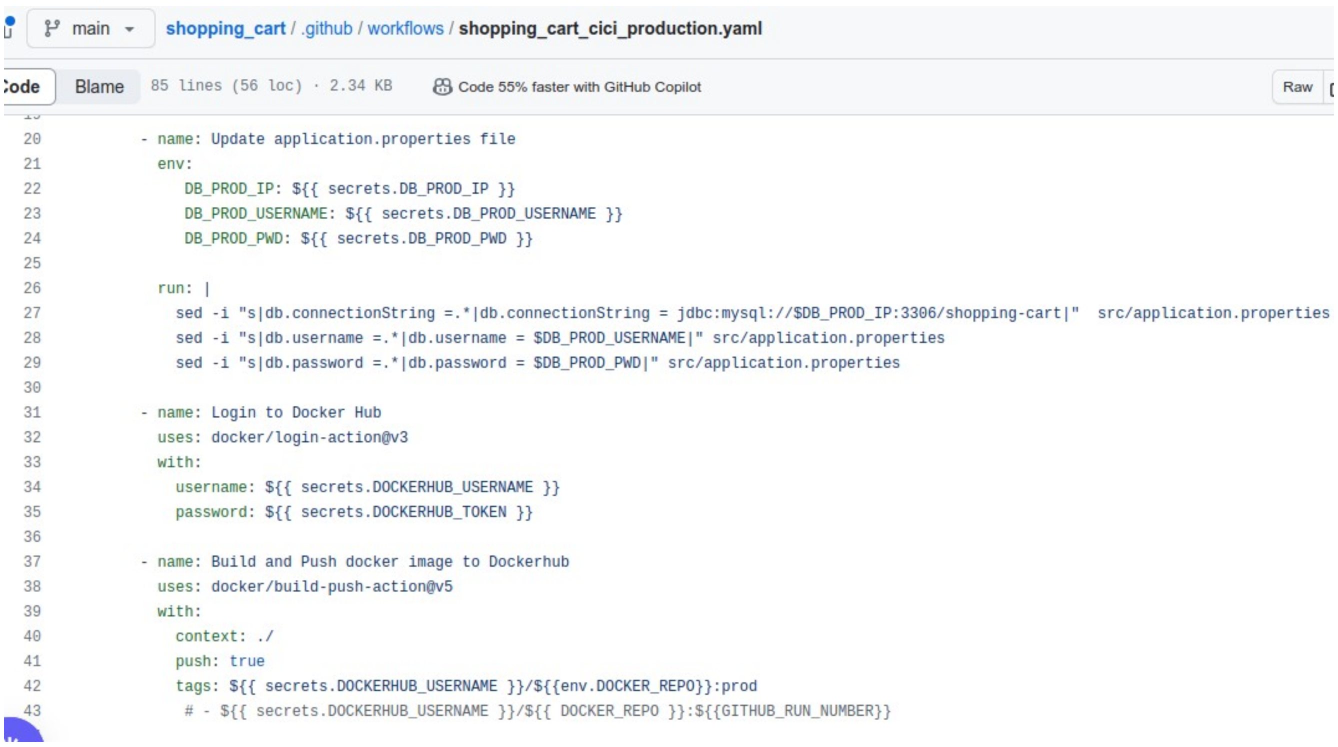
```yaml
DeployToTestEnv:
  runs-on: ubuntu-latest
  needs: BuildAndPushForTesting
  env:
    DOCKER_USER: ${{ secrets.DOCKERHUB_USERNAME }}
    DOCKER_REPO: shopping-cart
    APP: shopping-cart-app
    APP_PORT: 8081

  steps:
    - name: Execute SSH commmands on remote server
      uses: JimCronqvist/action-ssh@master
      with:
        hosts: ${{ secrets.TEST_HOST_USER }}@${{secrets.TEST_HOST_IP}}
        privateKey: ${{ secrets.TEST_HOST_PRIVATE_KEY }}
        command: |
          docker stop ${{env.APP}}
          sleep 2
          docker pull ${{env.DOCKER_USER }}/${{env.DOCKER_REPO}}:test
          sleep 2
          docker run --name ${{env.APP}} -d --rm -p ${{env.APP_PORT}}:8080 ${{env.DOCKER_USER }}/${{env.DOCKER_REPO}}:test
```

Creating Production Pipelines:On My **Dev02** shopping_cart repo, I created new blank workflow name it **shopping_cart_cicd_production.yml**


robudexIT / shopping_cart
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
main
shopping_cart / .github / workflows / shopping_cart_cici_production.yaml
robudexIT Update shopping_cart_cici_production.yaml
Code
Blame
85 lines (56 loc) · 2.34 KB
Code 55% faster with GitHub Copilot

```yaml
name: Shopping Cart Action

on: workflow_dispatch #manual trigger
# on:
#   push:
#     branches: main

permissions:
  issues: write
jobs:
  BuildAndPushForProduction:
    runs-on: ubuntu-latest
    environment: production
    env:
      DOCKER_REPO: shopping-cart
    steps:
      - name: Code Checkout
        uses: actions/checkout@v4
```

SET TO MANUL TRIGGER INTIALLY

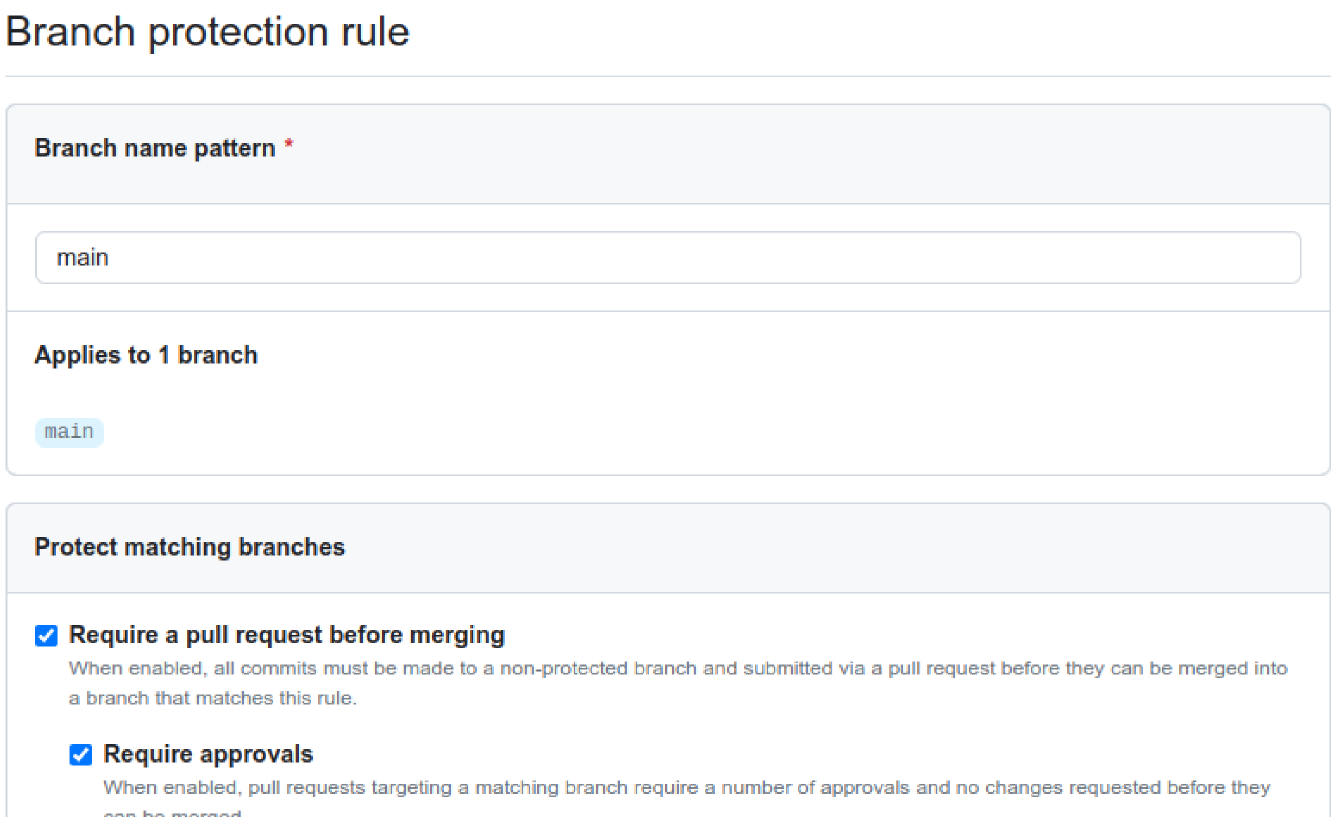

main
shopping_cart / .github / workflows / shopping_cart_cici_production.yaml
Code
Blame
85 lines (56 loc) · 2.34 KB
Code 55% faster with GitHub Copilot
Raw

```yaml
      - name: Update application.properties file
        env:
           DB_PROD_IP: ${{ secrets.DB_PROD_IP }}
           DB_PROD_USERNAME: ${{ secrets.DB_PROD_USERNAME }}
           DB_PROD_PWD: ${{ secrets.DB_PROD_PWD }}

        run: |
          sed -i "s|db.connectionString =.*|db.connectionString = jdbc:mysql://$DB_PROD_IP:3306/shopping-cart|"  src/application.properties
          sed -i "s|db.username =.*|db.username = $DB_PROD_USERNAME|" src/application.properties
          sed -i "s|db.password =.*|db.password = $DB_PROD_PWD|" src/application.properties

      - name: Login to Docker Hub
        uses: docker/login-action@v3
        with:
          username: ${{ secrets.DOCKERHUB_USERNAME }}
          password: ${{ secrets.DOCKERHUB_TOKEN }}

      - name: Build and Push docker image to Dockerhub
        uses: docker/build-push-action@v5
        with:
          context: ./
          push: true
          tags: ${{ secrets.DOCKERHUB_USERNAME }}/${{env.DOCKER_REPO}}:prod
           # - ${{ secrets.DOCKERHUB_USERNAME }}/${{ DOCKER_REPO }}:${{GITHUB_RUN_NUMBER}}
```

```yaml
DeployToKubernetes:
  runs-on: ubuntu-latest
  needs: BuildAndPushForProduction
  steps:
    - name: Code Checkout
      uses: actions/checkout@v4
    - name: Setup Kubernetes Configuration
      uses: tale/kubectl-action@v1
      with:
        base64-kube-config: ${{ secrets.KUBE_CONFIG }}
    - name: Check for existing deployment
      run: |
        if kubectl get deployment shopping-cart-java-app -n app-namespace >/dev/null 2>&1; then
          echo "Deployment exists, rolling out update"
          kubectl rollout restart deployment shopping-cart-java-app -n app-namespace
        else
          echo "Deployment not found, applying new resources"
          kubectl apply -f kubernetes/application/shopping-cart-app-deployment.yaml -n app-namespace
        fi
```

As you notice, I tempoary set the two pipelines to manual trigger (**workflow_dispatch**) to avoid running the pipeline because it is not ready yet.

Create Branch Rules for main branch. **Settings -> Branches**
Check the Option Seen below:



Branch protection rule

Branch name pattern *

main

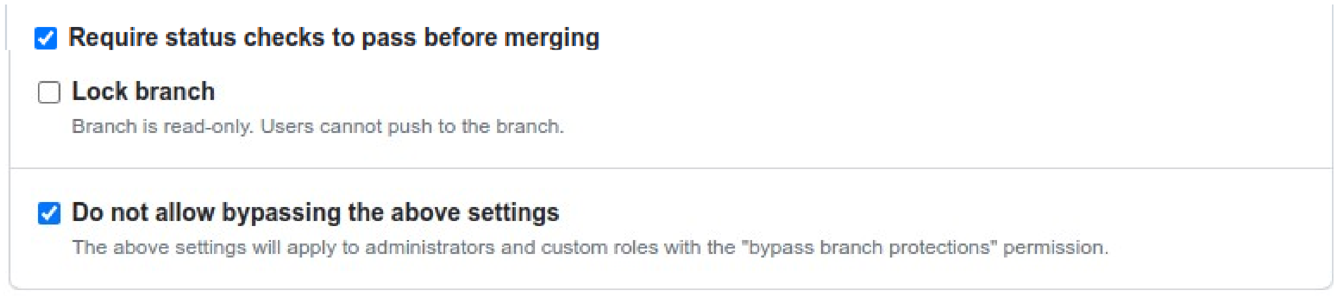
Applies to 1 branch

`main`

Protect matching branches

[✓] **Require a pull request before merging**
When enabled, all commits must be made to a non-protected branch and submitted via a pull request before they can be merged into a branch that matches this rule.

[✓] **Require approvals**
When enabled, pull requests targeting a matching branch require a number of approvals and no changes requested before they

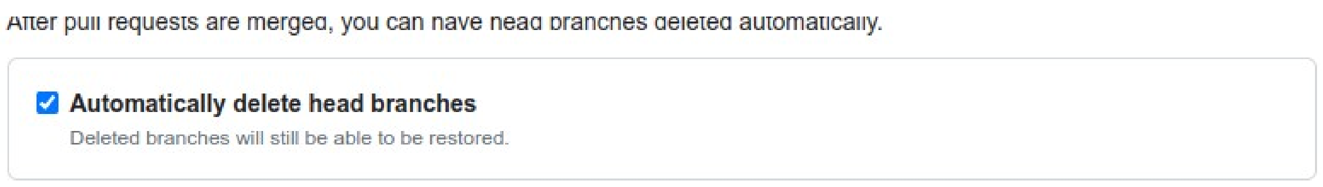


[✓] **Require status checks to pass before merging**

[ ] **Lock branch**
Branch is read-only. Users cannot push to the branch.

[✓] **Do not allow bypassing the above settings**
The above settings will apply to administrators and custom roles with the "bypass branch protections" permission.

On Settings-> General-> Pull Requests



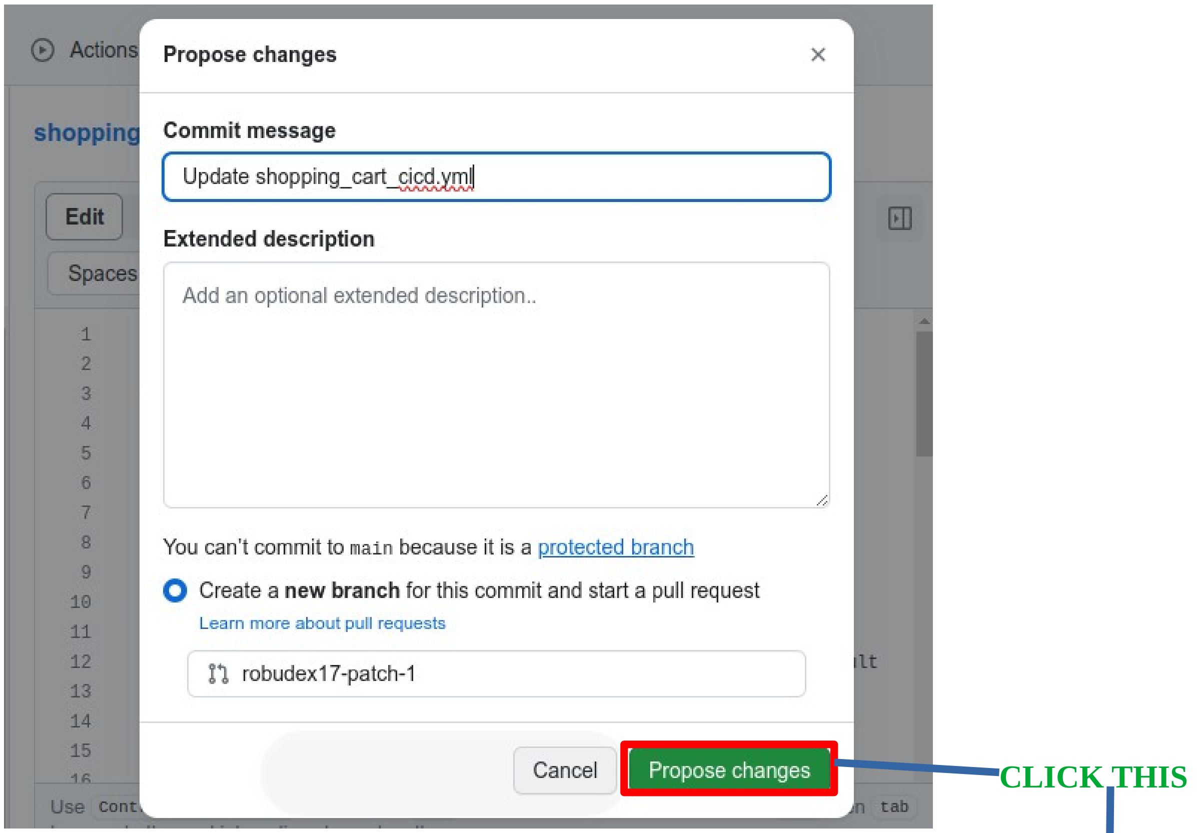
After pull requests are merged, you can have head branches deleted automatically.

[✓] **Automatically delete head branches**
Deleted branches will still be able to be restored.

7. Now All are Set, Its time to test.

On **Dev01**-> I Update **shopping_cart_cicd.yml** from workflow_dispatch to pull_request

As you can see. It cannot directly push to main to make some changes. Click Propose changes, Description and create pull request



Update shopping_cart_cicd.yml

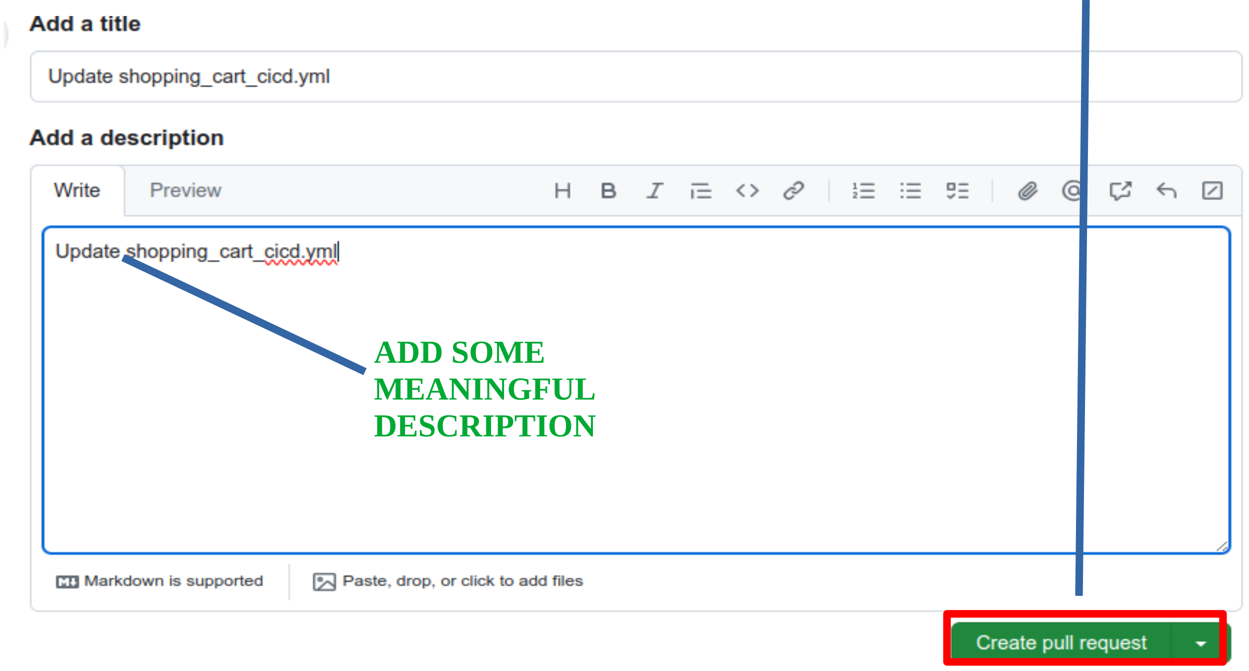
Add an optional extended description..

You can't commit to `main` because it is a protected branch

Create a **new branch** for this commit and start a pull request

Learn more about pull requests

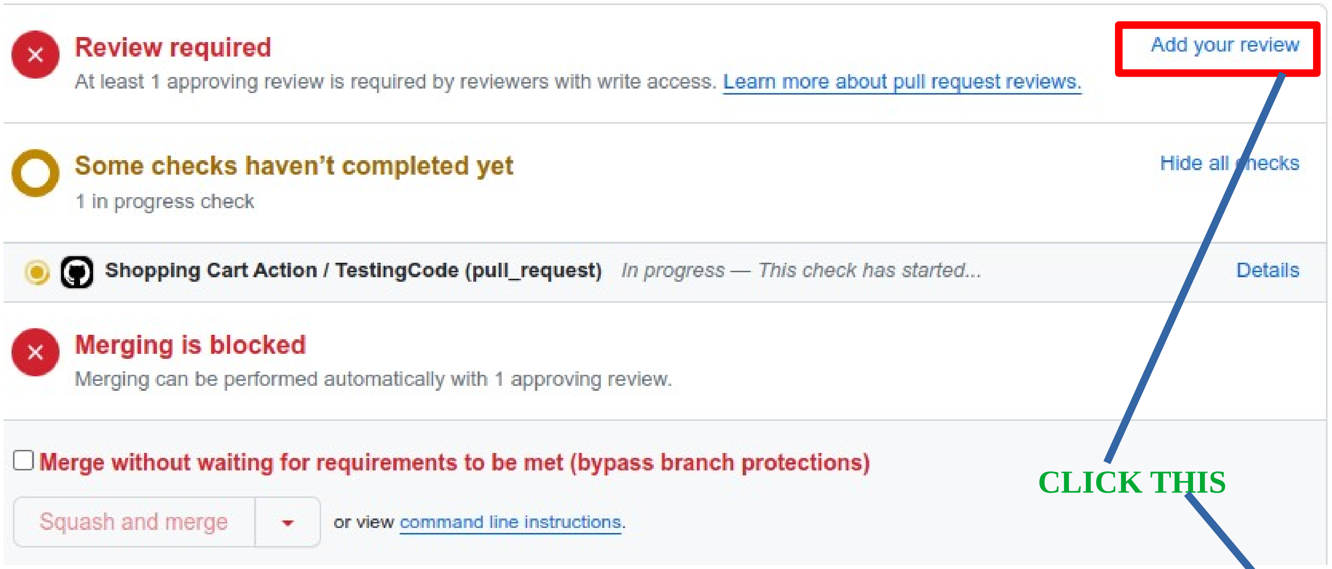
robudex17-patch-1



Update shopping_cart_cicd.yml

Update shopping_cart_cicd.yml

Goto Dev02 Account and check the Pull Request and Approved it



Review required
At least 1 approving review is required by reviewers with write access. Learn more about pull request reviews.

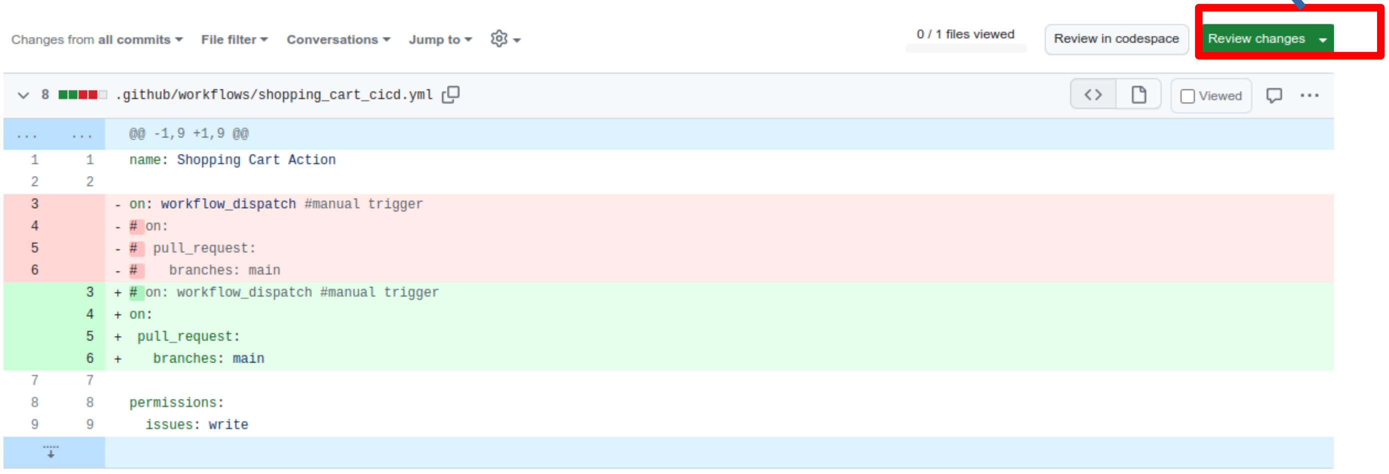
Some checks haven't completed yet
1 in progress check

Shopping Cart Action / TestingCode (pull_request) *In progress — This check has started...* Details

Merging is blocked
Merging can be performed automatically with 1 approving review.

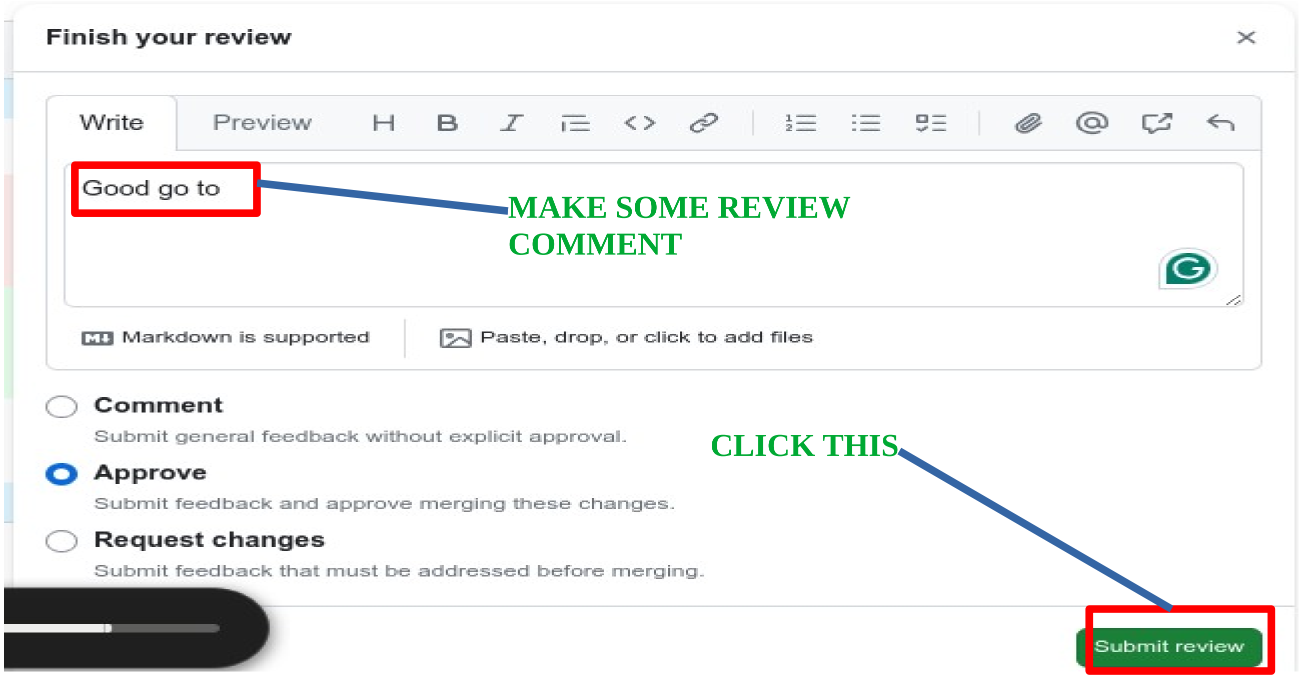
☐ Merge without waiting for requirements to be met (bypass branch protections)

Squash and merge ▾ or view command line instructions.



.github/workflows/shopping_cart_cicd.yml

```
@@ -1,9 +1,9 @@
  name: Shopping Cart Action

- on: workflow_dispatch #manual trigger
- # on:
- #   pull_request:
- #     branches: main
+ # on: workflow_dispatch #manual trigger
+ on:
+   pull_request:
+     branches: main

  permissions:
    issues: write
```



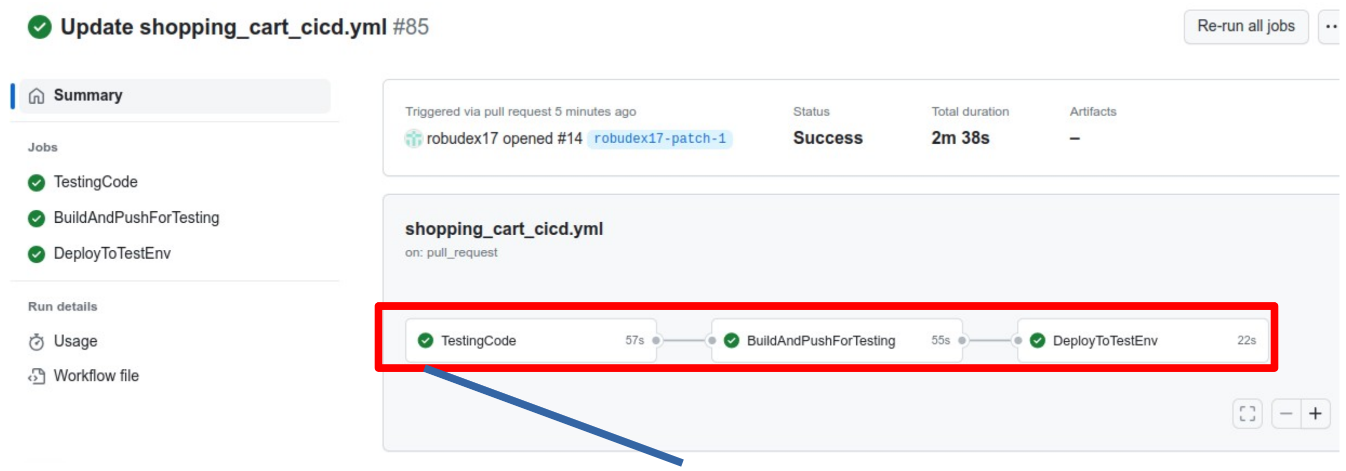
Good go to

( ) Comment
Submit general feedback without explicit approval.

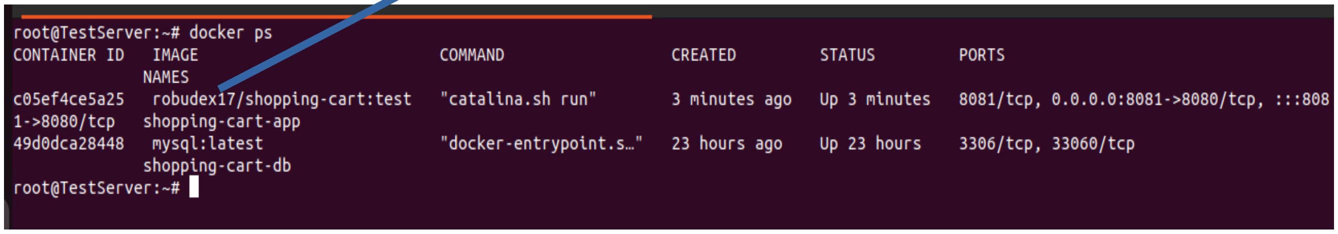
[✓ Approve]
Submit feedback and approve merging these changes.

( ) Request changes
Submit feedback that must be addressed before merging.

Goto Actions and see that the **shopping_cart_cicd** execute successfully.

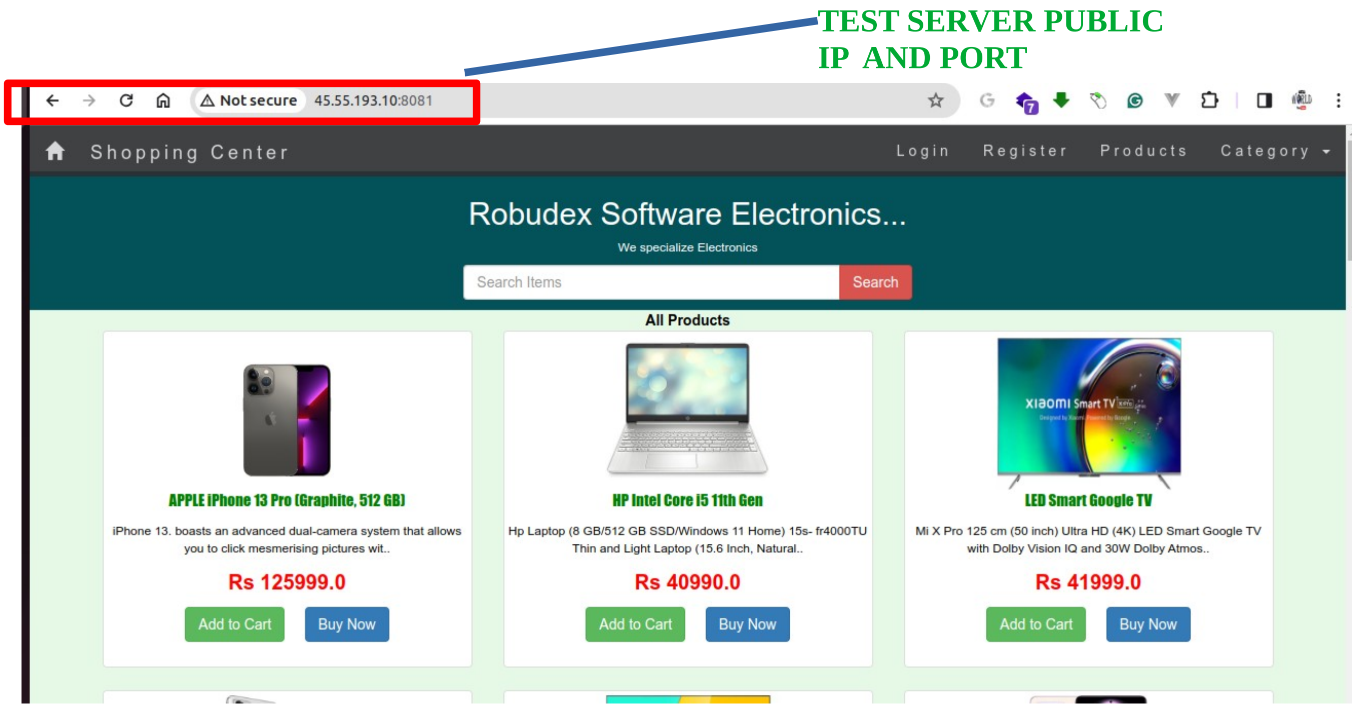


Java Webapp is successfully install in **Test Server**

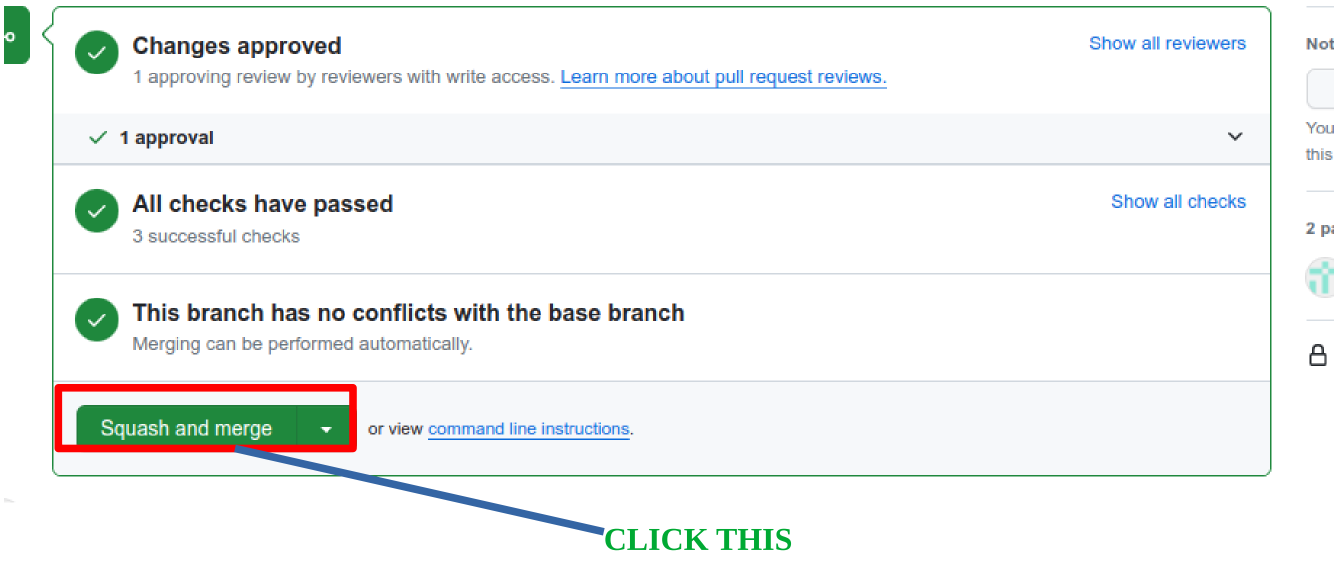


```
root@TestServer:~# docker ps
CONTAINER ID   IMAGE                            COMMAND                  CREATED         STATUS          PORTS
               NAMES
c05ef4ce5a25   robudex17/shopping-cart:test     "catalina.sh run"        3 minutes ago   Up 3 minutes    8081/tcp, 0.0.0.0:8081->8080/tcp, :::808
1->8080/tcp    shopping-cart-app
49d0dca28448   mysql:latest                     "docker-entrypoint.s…"   23 hours ago    Up 23 hours     3306/tcp, 33060/tcp
               shopping-cart-db
root@TestServer:~# 
```

Paste Test Server Public ip and 8081 port


TEST SERVER PUBLIC IP AND PORT
Not secure 45.55.193.10:8081
Shopping Center
Login
Register
Products
Category
Robudex Software Electronics...
We specialize Electronics
Search Items
Search
All Products
APPLE iPhone 13 Pro (Graphite, 512 GB)
iPhone 13. boasts an advanced dual-camera system that allows you to click mesmerising pictures wit..
Rs 125999.0
Add to Cart
Buy Now
HP Intel Core i5 11th Gen
Hp Laptop (8 GB/512 GB SSD/Windows 11 Home) 15s- fr4000TU Thin and Light Laptop (15.6 Inch, Natural..
Rs 40990.0
Add to Cart
Buy Now
LED Smart Google TV
Mi X Pro 125 cm (50 inch) Ultra HD (4K) LED Smart Google TV with Dolby Vision IQ and 30W Dolby Atmos..
Rs 41999.0
Add to Cart
Buy Now

After ensuring the successful deployment on the Test Env, it's time to merge the changes into the **main branch**. To do so, **Dev02** should navigate to the **Pull Requests** section, select the relevant pull request, and opt for the "**Squash and Merge**" option. This action condenses all commits from the feature branch into a single commit before merging them into the main branch.


Changes approved

1 approving review by reviewers with write access. Learn more about pull request reviews.

1 approval

All checks have passed

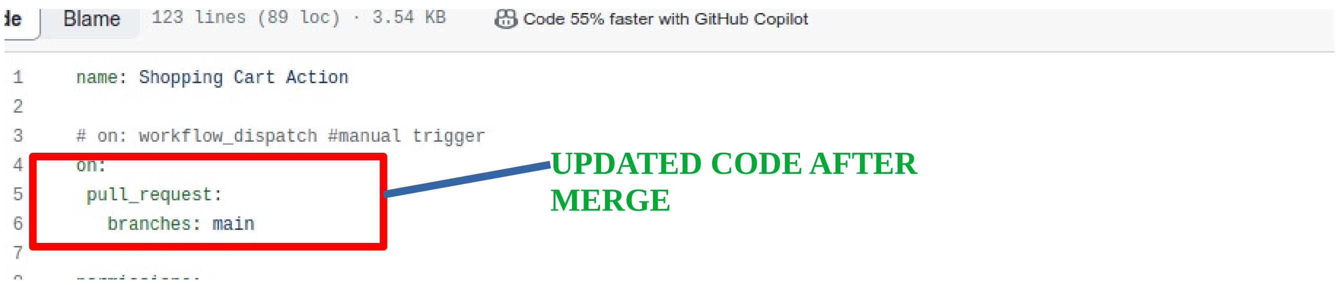
3 successful checks

This branch has no conflicts with the base branch

Merging can be performed automatically.

or view command line instructions.

Notice that the change .github/workflows/shopping_cart_cicd.yml has been merge to main branch

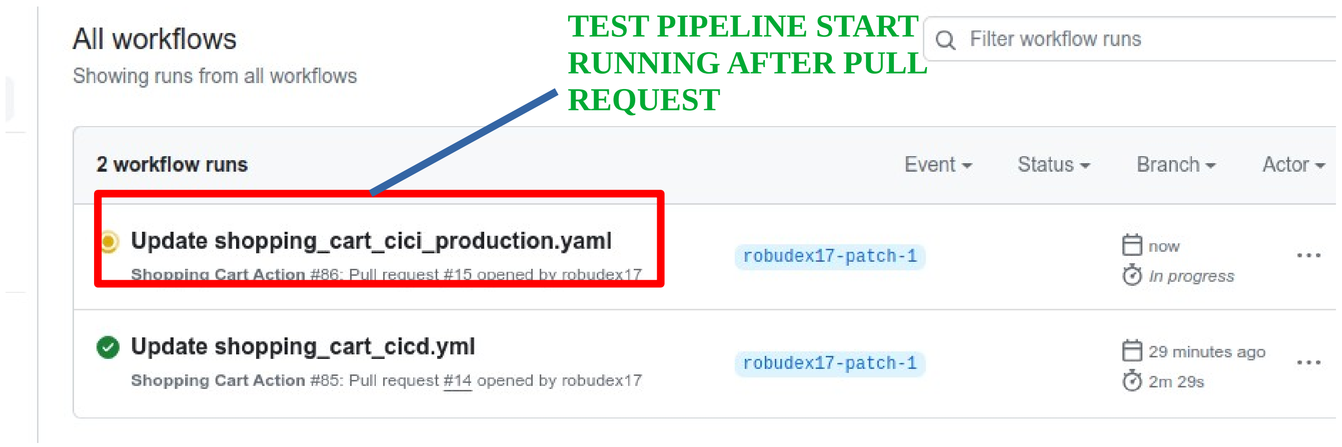

```
name: Shopping Cart Action

# on: workflow_dispatch #manual trigger
on:
 pull_request:
   branches: main
```

Blame
123 lines (89 loc) · 3.54 KB
Code 55% faster with GitHub Copilot
UPDATED CODE AFTER MERGE

Now Its time to update the **.github/workflows/shopping_cart_cici_production.yaml** as well from **workflow_dispatch** to **push**. On **Dev01** change the file and create pull request. Then approve the **Pull request** on the **Dev02** Account.

Note that the **shopping_cart_cicd** pipeline executes because another pull request has been initiated. Please wait for the pipeline execution to finish before merging it into the main branch.

Now because there is a merge or a push in the main branch production pipeline execute



All workflows

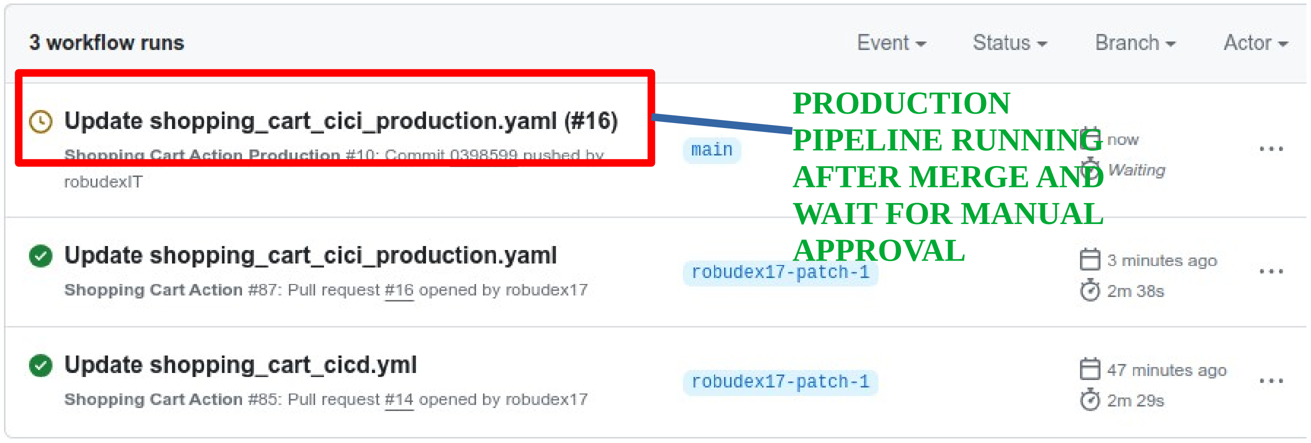
Showing runs from all workflows

Filter workflow runs

2 workflow runs — Event ▾ Status ▾ Branch ▾ Actor ▾

Update shopping_cart_cici_production.yaml
Shopping Cart Action #86: Pull request #15 opened by robudex17 — robudex17-patch-1 — now — In progress

Update shopping_cart_cicd.yml
Shopping Cart Action #85: Pull request #14 opened by robudex17 — robudex17-patch-1 — 29 minutes ago — 2m 29s


PRODUCTION PIPELINE RUNNING AFTER MERGE AND WAIT FOR MANUAL APPROVAL

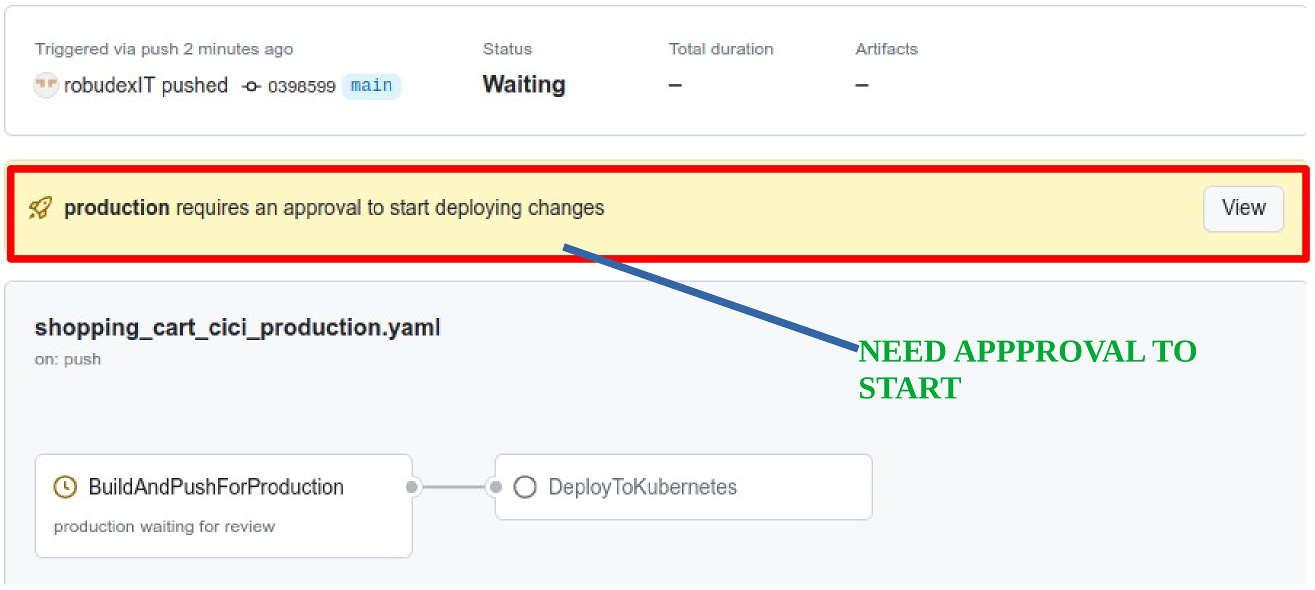
3 workflow runs — Event ▾ Status ▾ Branch ▾ Actor ▾

Update shopping_cart_cici_production.yaml (#16)
Shopping Cart Action Production #10: Commit 0398599 pushed by robudexIT — main — now — Waiting

Update shopping_cart_cici_production.yaml
Shopping Cart Action #87: Pull request #16 opened by robudex17 — robudex17-patch-1 — 3 minutes ago — 2m 38s

Update shopping_cart_cicd.yml
Shopping Cart Action #85: Pull request #14 opened by robudex17 — robudex17-patch-1 — 47 minutes ago — 2m 29s

Click the workflow and notice it requires approval



Triggered via push 2 minutes ago — robudexIT pushed 0398599 main

Status: Waiting

Total duration: –

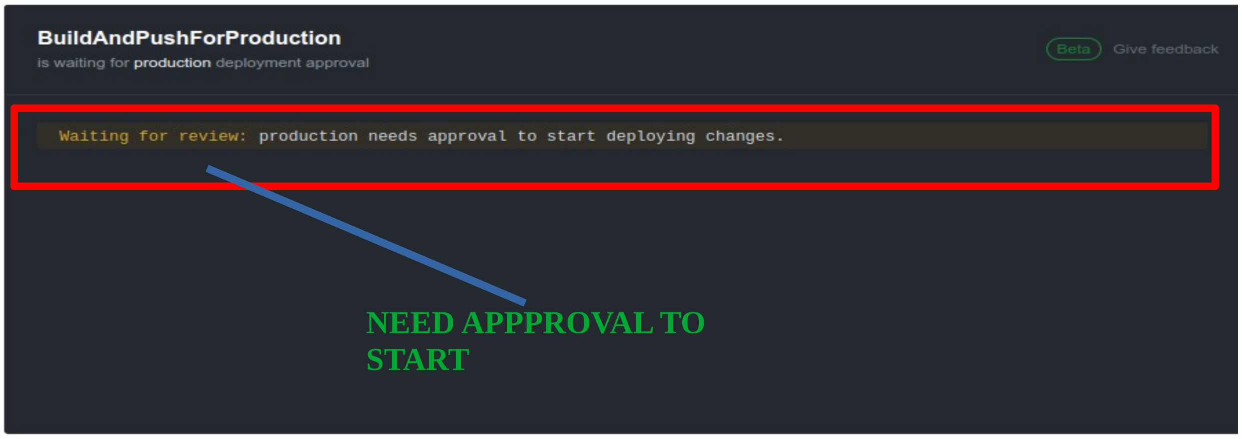
Artifacts: –

production requires an approval to start deploying changes — View

shopping_cart_cici_production.yaml
on: push

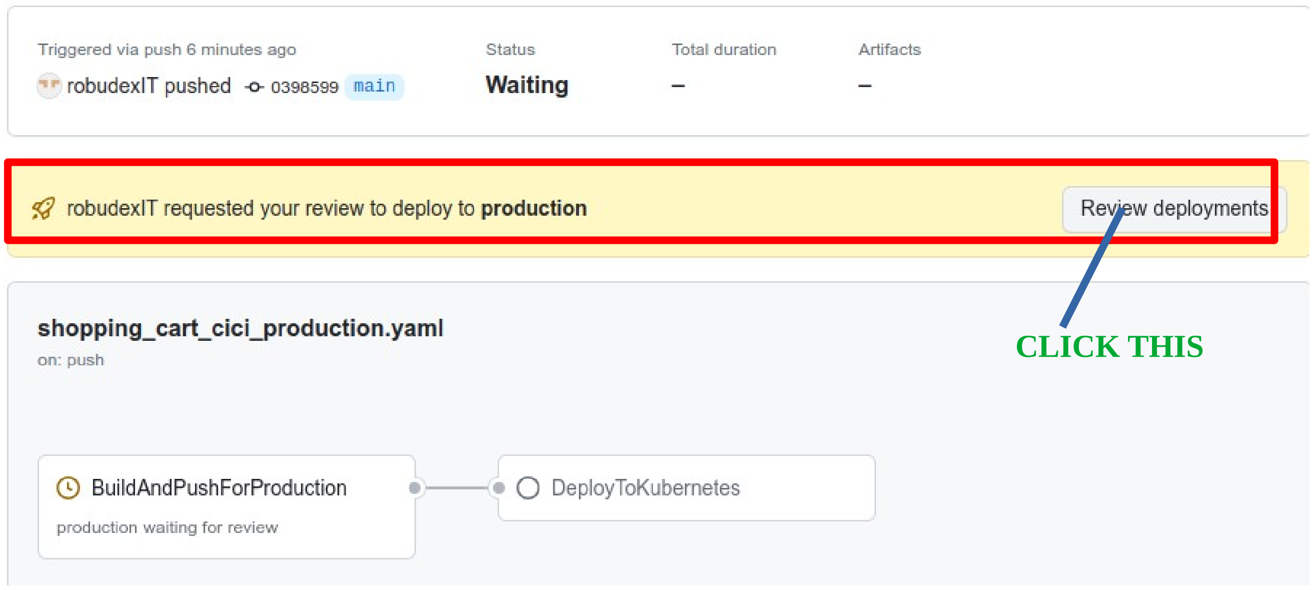
BuildAndPushForProduction — production waiting for review

DeployToKubernetes

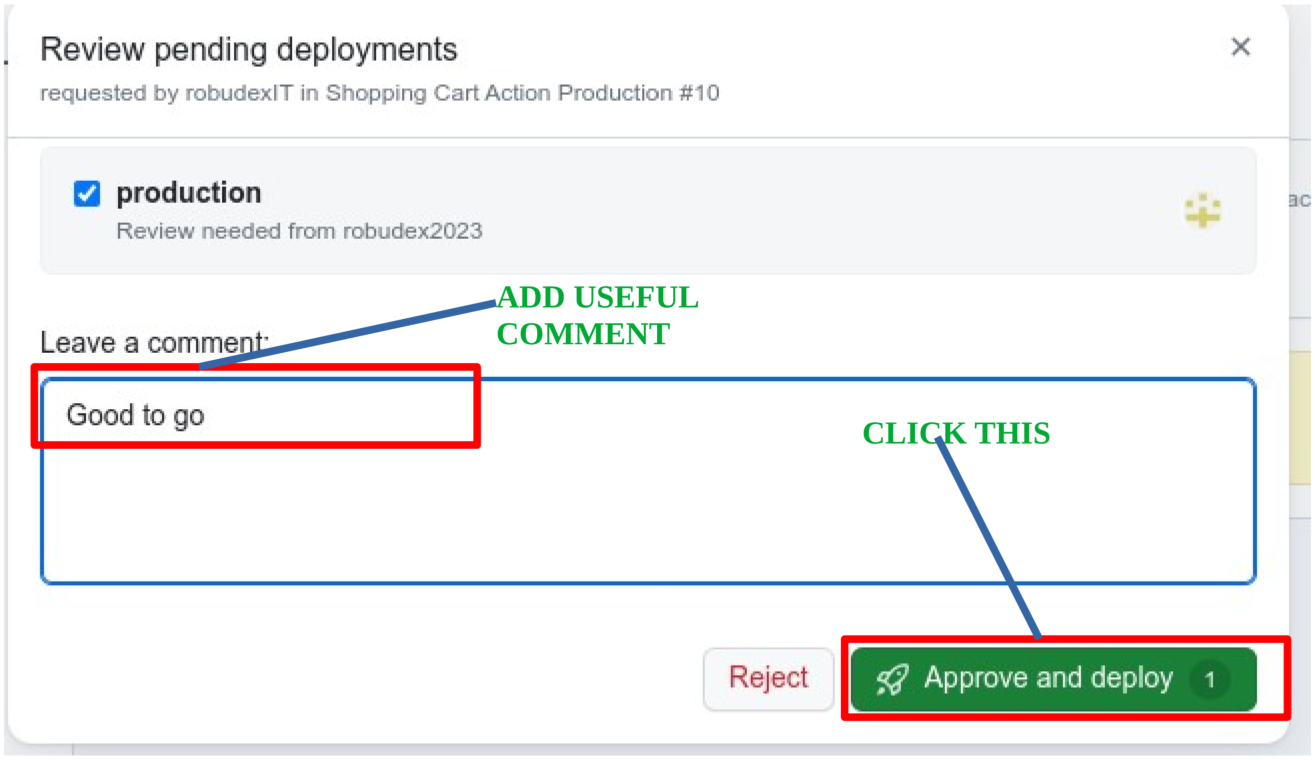

BuildAndPushForProduction
is waiting for production deployment approval
Beta
Give feedback
NEED APPPROVAL TO START

```
Waiting for review: production needs approval to start deploying changes.
```

Given that I've configured **Dev03** to solely **approve or reject** the pipeline execution, it won't initiate or conclude without **Dev03's approval**. Please proceed to Dev03 and provide approval.


Triggered via push 6 minutes ago
robudexIT pushed 0398599 main
Status
Waiting
Total duration
–
Artifacts
–
Review deployments
CLICK THIS
shopping_cart_cici_production.yaml
on: push
BuildAndPushForProduction
production waiting for review
DeployToKubernetes

robudexIT requested your review to deploy to **production**


Review pending deployments

requested by robudexIT in Shopping Cart Action Production #10

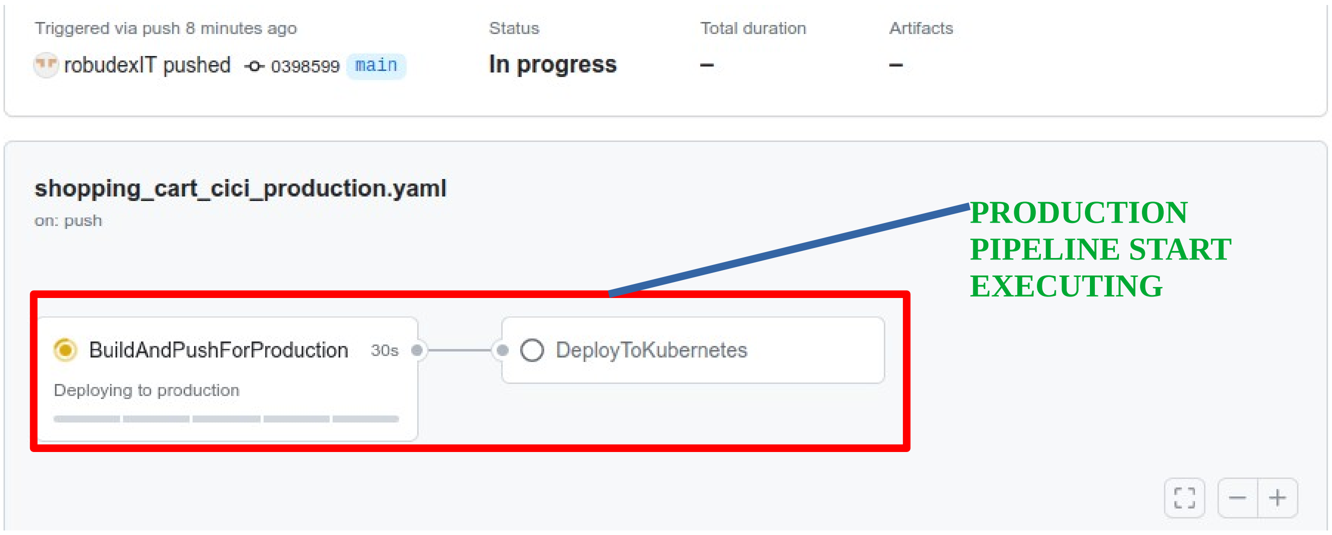
production

Review needed from robudex2023

Leave a comment:

Good to go

Production pipeline Start executing...


Triggered via push 8 minutes ago

robudexIT pushed 0398599 main

Status: In progress

Total duration: –

Artifacts: –

shopping_cart_cici_production.yaml

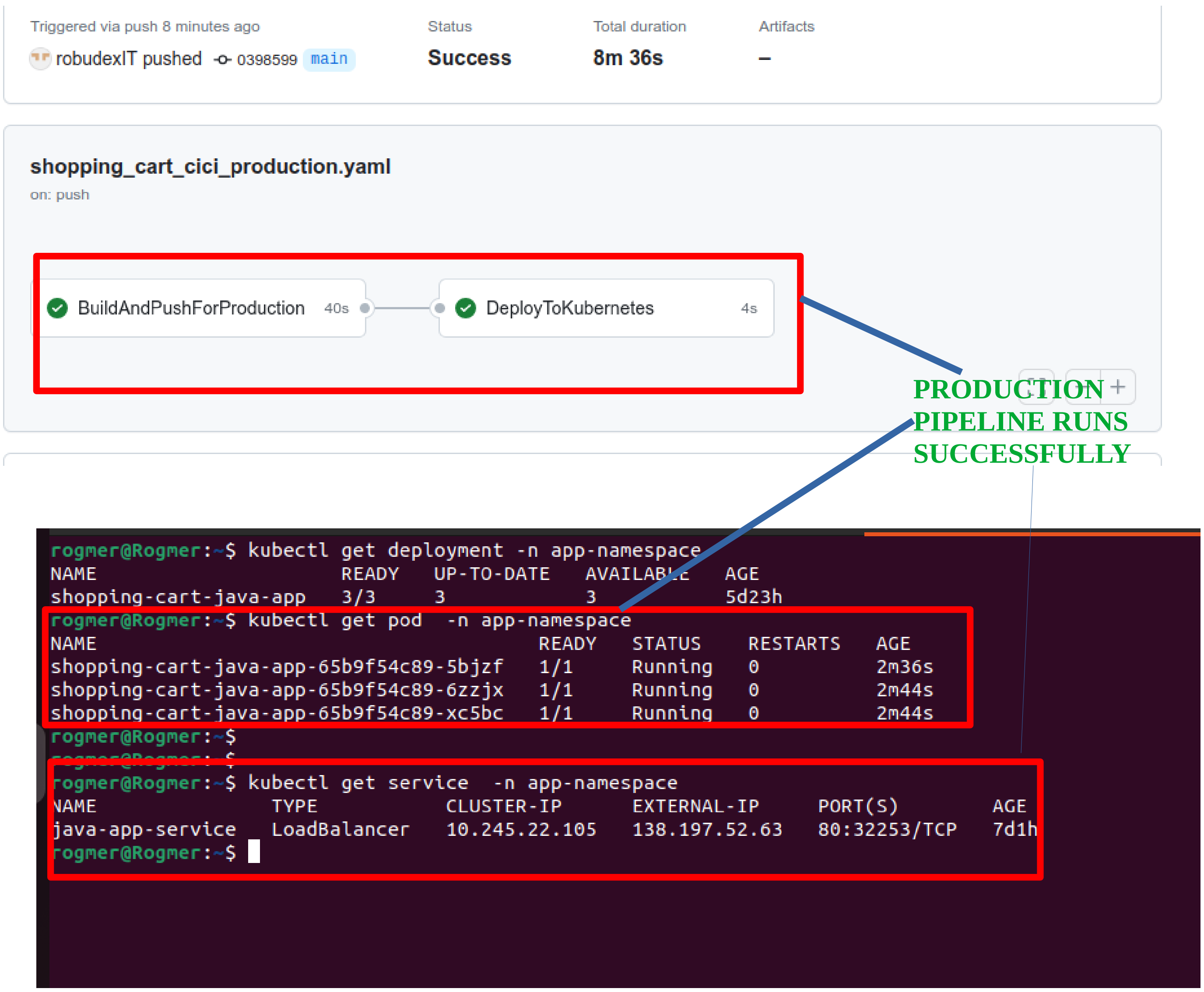
on: push

BuildAndPushForProduction 30s

Deploying to production

DeployToKubernetes

Production pipeline Done.. Check Kubernetes cluster


Triggered via push 8 minutes ago

robudexIT pushed 0398599 main

Status: Success

Total duration: 8m 36s

Artifacts: –

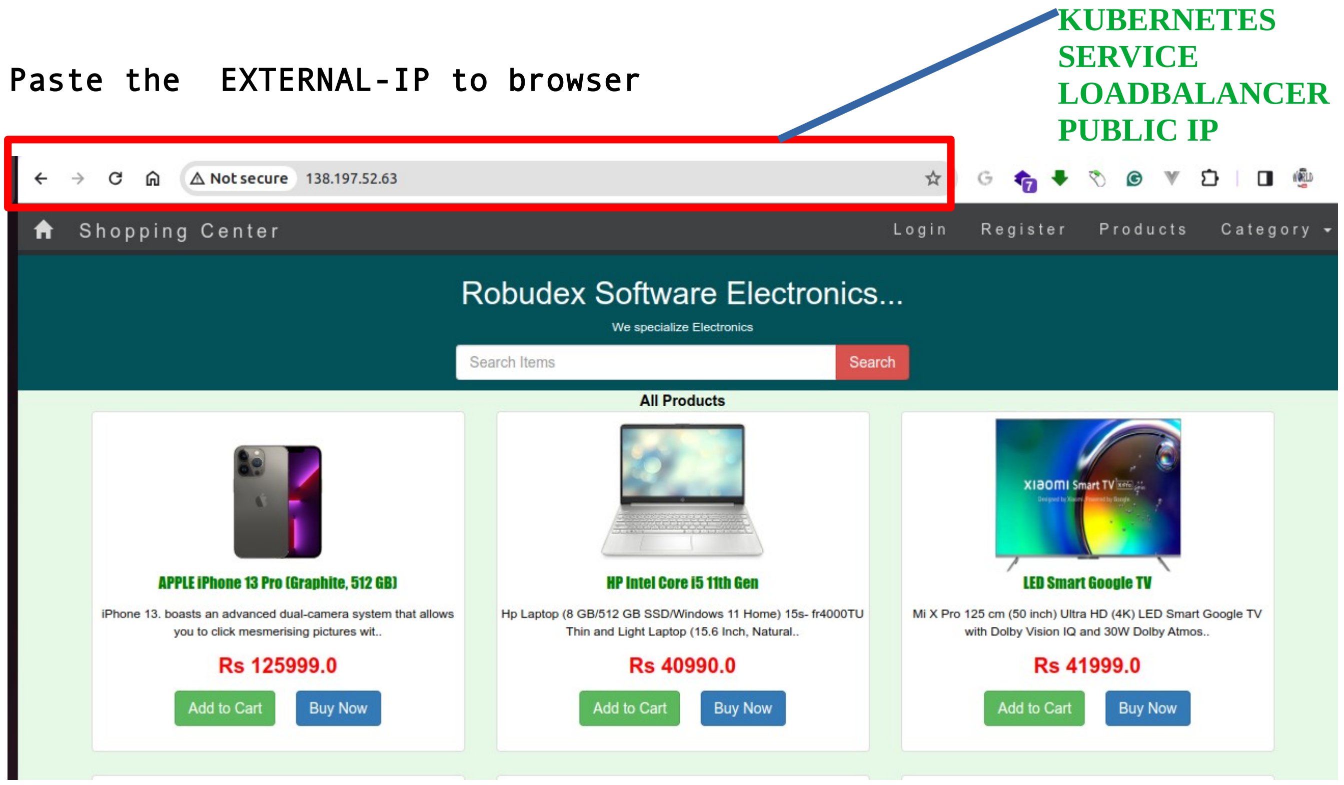
shopping_cart_cici_production.yaml

on: push

BuildAndPushForProduction 40s → DeployToKubernetes 4s

```
rogmer@Rogmer:~$ kubectl get deployment -n app-namespace
NAME                      READY   UP-TO-DATE   AVAILABLE   AGE
shopping-cart-java-app    3/3     3            3           5d23h
rogmer@Rogmer:~$ kubectl get pod  -n app-namespace
NAME                                     READY   STATUS    RESTARTS   AGE
shopping-cart-java-app-65b9f54c89-5bjzf  1/1     Running   0          2m36s
shopping-cart-java-app-65b9f54c89-6zzjx  1/1     Running   0          2m44s
shopping-cart-java-app-65b9f54c89-xc5bc  1/1     Running   0          2m44s
rogmer@Rogmer:~$
rogmer@Rogmer:~$
rogmer@Rogmer:~$ kubectl get service  -n app-namespace
NAME               TYPE           CLUSTER-IP      EXTERNAL-IP     PORT(S)        AGE
java-app-service   LoadBalancer   10.245.22.105   138.197.52.63   80:32253/TCP   7d1h
rogmer@Rogmer:~$
```

Paste the EXTERNAL-IP to browser


KUBERNETES SERVICE LOADBALANCER PUBLIC IP
Not secure
138.197.52.63
Shopping Center
Login
Register
Products
Category
Robudex Software Electronics...
We specialize Electronics
Search Items
Search
All Products
APPLE iPhone 13 Pro (Graphite, 512 GB)
iPhone 13. boasts an advanced dual-camera system that allows you to click mesmerising pictures wit..
Rs 125999.0
Add to Cart
Buy Now
HP Intel Core i5 11th Gen
Hp Laptop (8 GB/512 GB SSD/Windows 11 Home) 15s- fr4000TU Thin and Light Laptop (15.6 Inch, Natural..
Rs 40990.0
Add to Cart
Buy Now
LED Smart Google TV
Mi X Pro 125 cm (50 inch) Ultra HD (4K) LED Smart Google TV with Dolby Vision IQ and 30W Dolby Atmos..
Rs 41999.0
Add to Cart
Buy Now

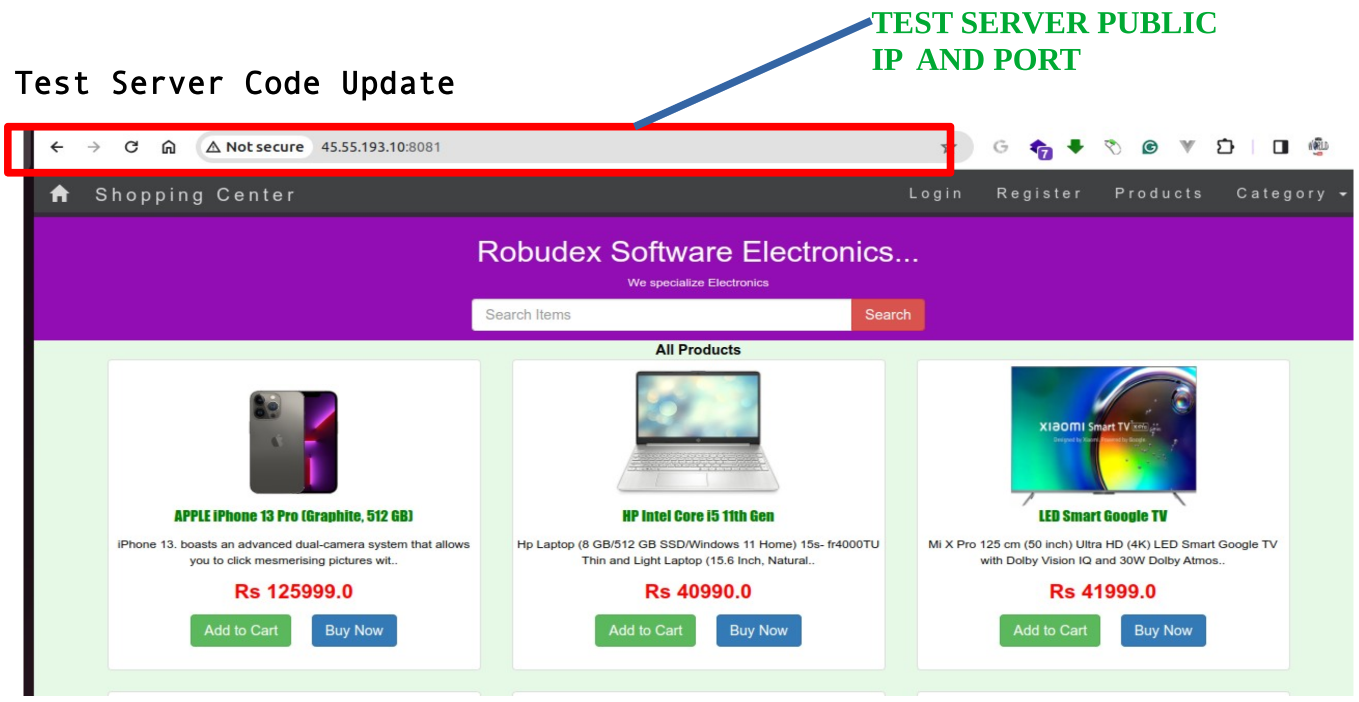
The production environment in the Kubernetes cluster is operational. Now, it's time to implement some code changes, create a pull request, merge it into the main branch, and ultimately, approve the production pipeline.

The flow will start in..

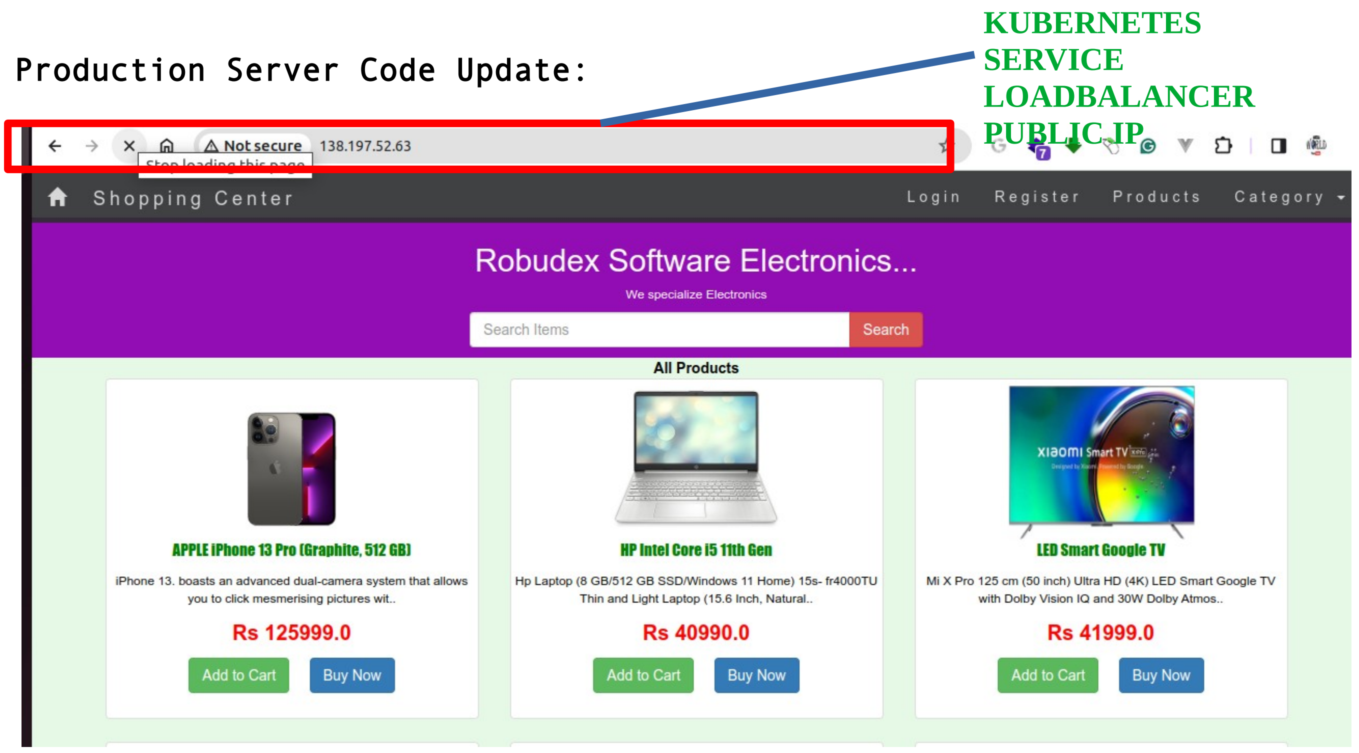
Dev01 --> make code change then make pull request
Dev02 --> Approve the pull request and then merge to main
Dev03 --> Approve the production pipeline execution.

Since I am not a **java developer** I will change only the **header bg color** to purple in **WebContent/header.jsp** file.

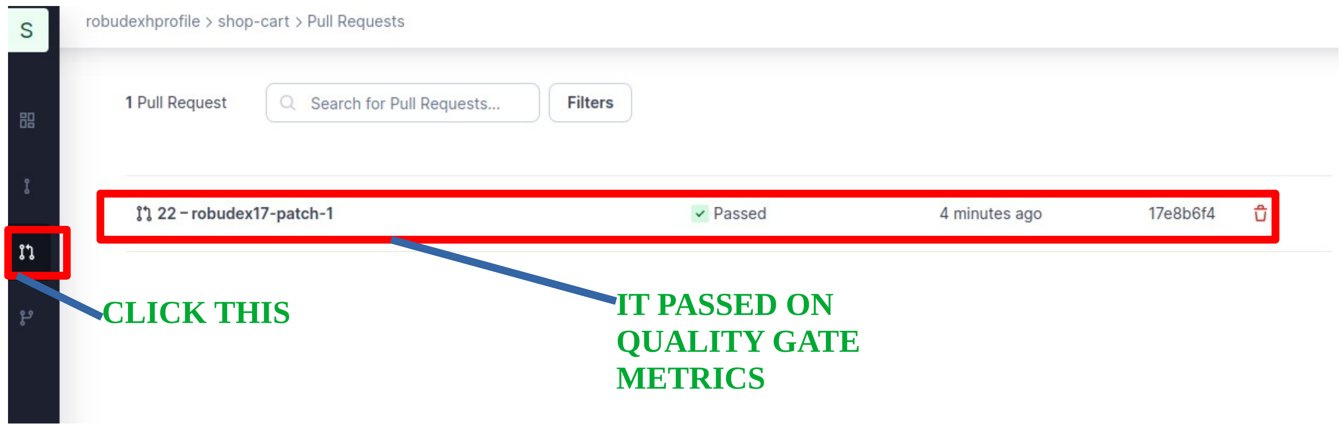
Test Server Code Update


TEST SERVER PUBLIC IP AND PORT
Not secure 45.55.193.10:8081
Shopping Center
Login
Register
Products
Category
Robudex Software Electronics...
We specialize Electronics
Search Items
Search
All Products
APPLE iPhone 13 Pro (Graphite, 512 GB)
iPhone 13. boasts an advanced dual-camera system that allows you to click mesmerising pictures wit..
Rs 125999.0
Add to Cart
Buy Now
HP Intel Core i5 11th Gen
Hp Laptop (8 GB/512 GB SSD/Windows 11 Home) 15s- fr4000TU Thin and Light Laptop (15.6 Inch, Natural..
Rs 40990.0
Add to Cart
Buy Now
LED Smart Google TV
Mi X Pro 125 cm (50 inch) Ultra HD (4K) LED Smart Google TV with Dolby Vision IQ and 30W Dolby Atmos..
Rs 41999.0
Add to Cart
Buy Now

Production Server Code Update:


KUBERNETES SERVICE LOADBALANCER PUBLIC IP
Not secure 138.197.52.63
Stop loading this page
Shopping Center
Login
Register
Products
Category
Robudex Software Electronics...
We specialize Electronics
Search Items
Search
All Products
APPLE iPhone 13 Pro (Graphite, 512 GB)
iPhone 13. boasts an advanced dual-camera system that allows you to click mesmerising pictures wit..
Rs 125999.0
Add to Cart
Buy Now
HP Intel Core i5 11th Gen
Hp Laptop (8 GB/512 GB SSD/Windows 11 Home) 15s- fr4000TU Thin and Light Laptop (15.6 Inch, Natural..
Rs 40990.0
Add to Cart
Buy Now
LED Smart Google TV
Mi X Pro 125 cm (50 inch) Ultra HD (4K) LED Smart Google TV with Dolby Vision IQ and 30W Dolby Atmos..
Rs 41999.0
Add to Cart
Buy Now

Now Let check our **sonarqube acccount**. Under Your **Organization**, Click the **shop-cart** project, Click the **pull_request**. And see that result is passed.

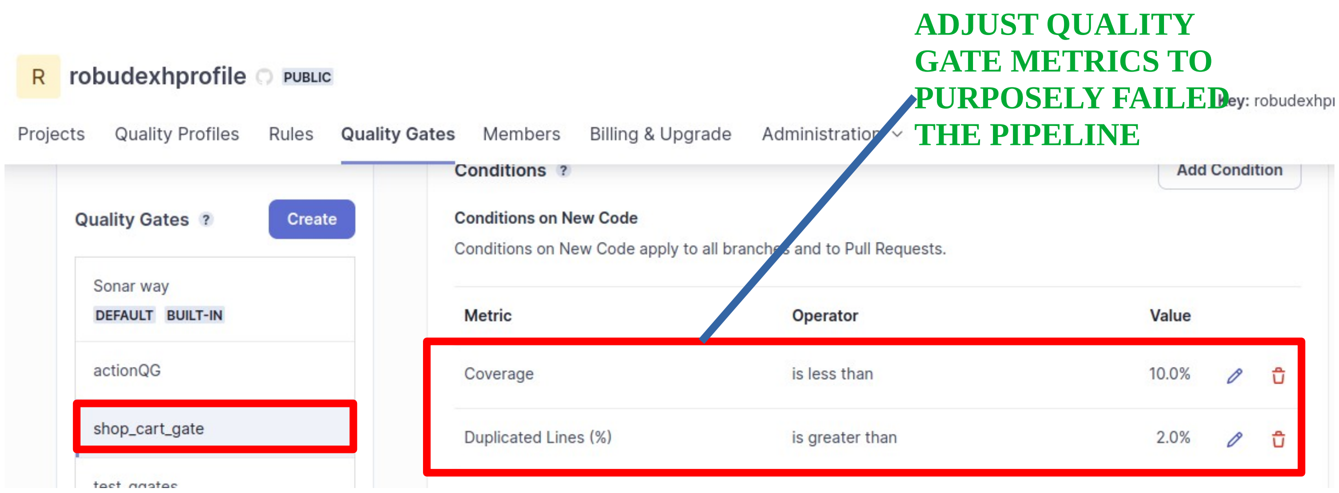

robudexhprofile > shop-cart > Pull Requests

1 Pull Request

22 – robudex17-patch-1 ✓ Passed 4 minutes ago 17e8b6f4

Search for Pull Requests...
Filters
CLICK THIS
IT PASSED ON QUALITY GATE METRICS

Lets Adust the **Quality Gate** And **Re-run** the Test Pipeline. (Note that we expect failure execution this time)

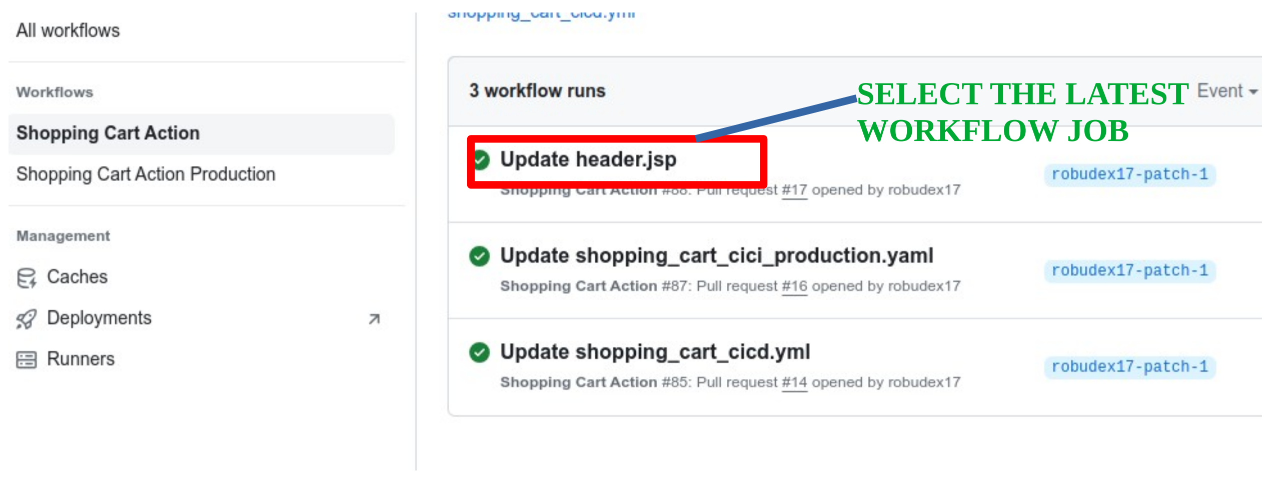

robudexhprofile PUBLIC

Conditions on New Code

Conditions on New Code apply to all branches and to Pull Requests.

Metric	Operator	Value
Coverage	is less than	10.0%
Duplicated Lines (%)	is greater than	2.0%

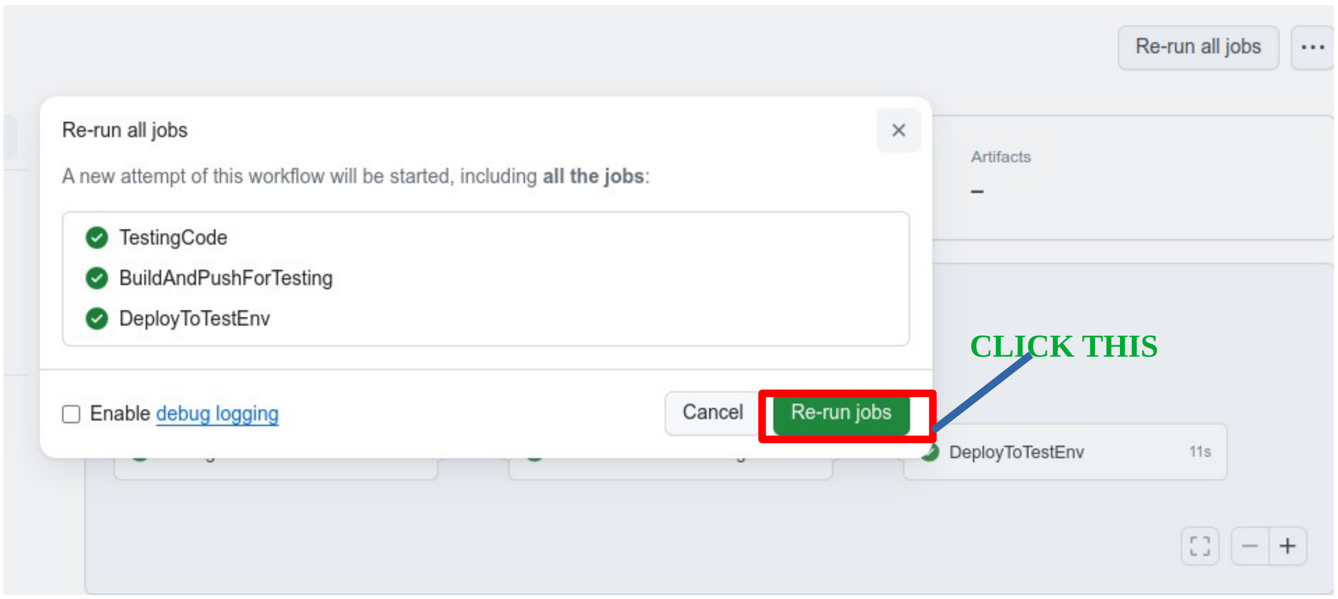
Click the Latest workfows of TEST Pipeline and Re-rull all jobs


3 workflow runs

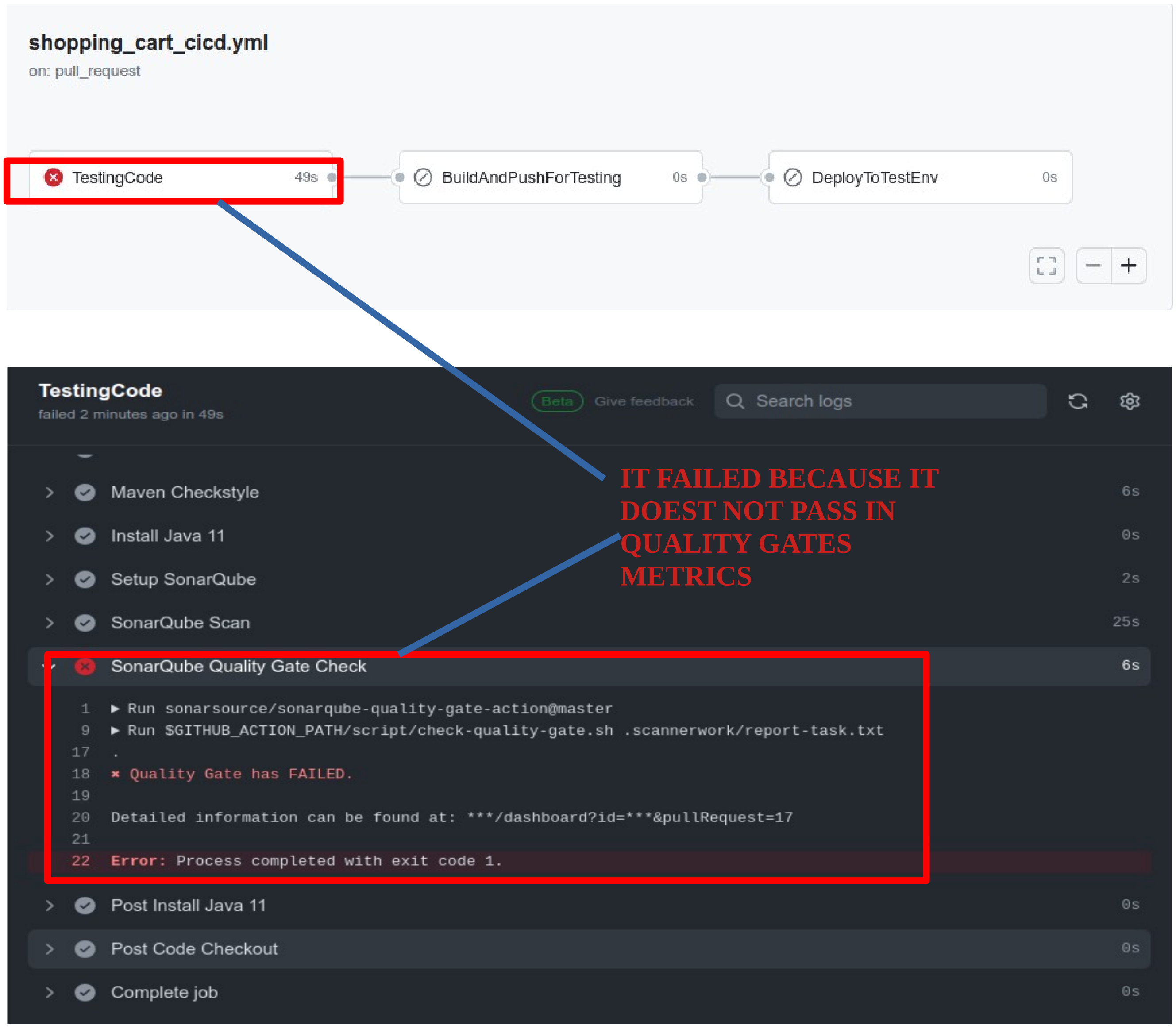
Update header.jsp robudex17-patch-1
Shopping Cart Action #88: Pull request #17 opened by robudex17

Update shopping_cart_cici_production.yaml robudex17-patch-1
Shopping Cart Action #87: Pull request #16 opened by robudex17

Update shopping_cart_cicd.yml robudex17-patch-1
Shopping Cart Action #85: Pull request #14 opened by robudex17

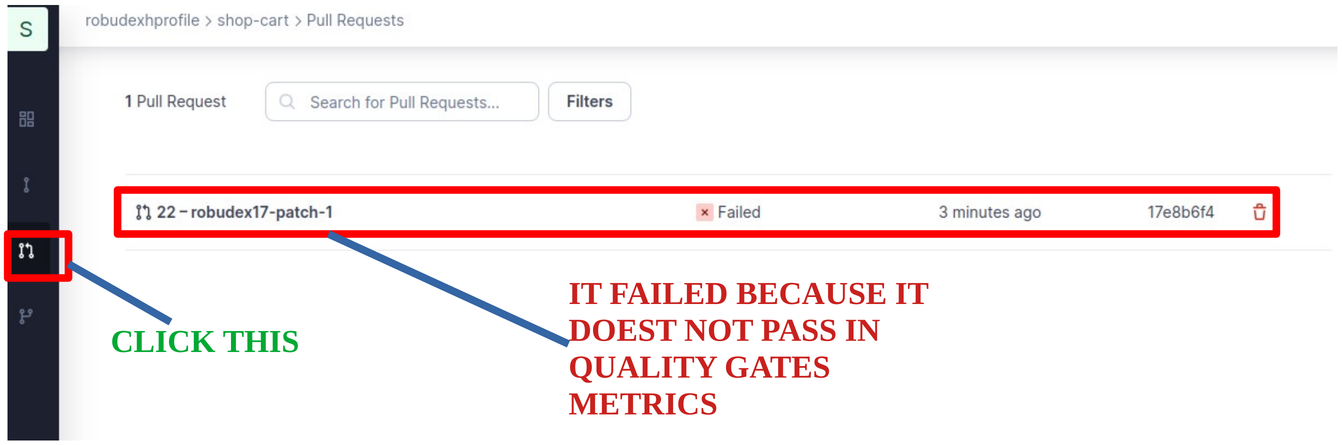


And indeed the pipeline fails.



```
▶ Run sonarsource/sonarqube-quality-gate-action@master
▶ Run $GITHUB_ACTION_PATH/script/check-quality-gate.sh .scannerwork/report-task.txt
.
✖ Quality Gate has FAILED.

Detailed information can be found at: ***/dashboard?id=***&pullRequest=17

Error: Process completed with exit code 1.
```

On SonarQube:


robudexhprofile > shop-cart > Pull Requests

1 Pull Request

22 – robudex17-patch-1	Failed	3 minutes ago	17e8b6f4

Search for Pull Requests...
Filters
CLICK THIS
IT FAILED BECAUSE IT DOEST NOT PASS IN QUALITY GATES METRICS

Lets back the sonar_cart_gate Metrics to less restrictive again ang re-run the Test Pipeline. To make it passed again.

THAT CONCLUDE THE DOCUMENTAION...THANK YOU