

Task 1

Follow [ArgoCD GitOps demo](#) & send screenshots with Kubernetes resources from ArgoCD UI on your cluster

Task 2

Review deployment strategies examples at
<https://github.com/elisska/k8s-deployment-strategies>

Task 3

Setup GitHub Actions CI/CD workflow to implement single-service blue/green deployment to any cloud-hosted Kubernetes cluster (EKS, GKE, AKS):

- You can use any Dockerized application of your choice or take this one:
<https://github.com/elisska/github-actions-samples/tree/main/app>
- You can follow this logic of doing blue/green deployment:
<https://github.com/elisska/k8s-deployment-strategies/tree/master/blue-green/single-service>

Task 3 requirements:

- CI job should include running unit tests for an application and further build & push Docker image to registry
- CD job should deploy resources to K8S cluster and update resources definitions in Github repo.

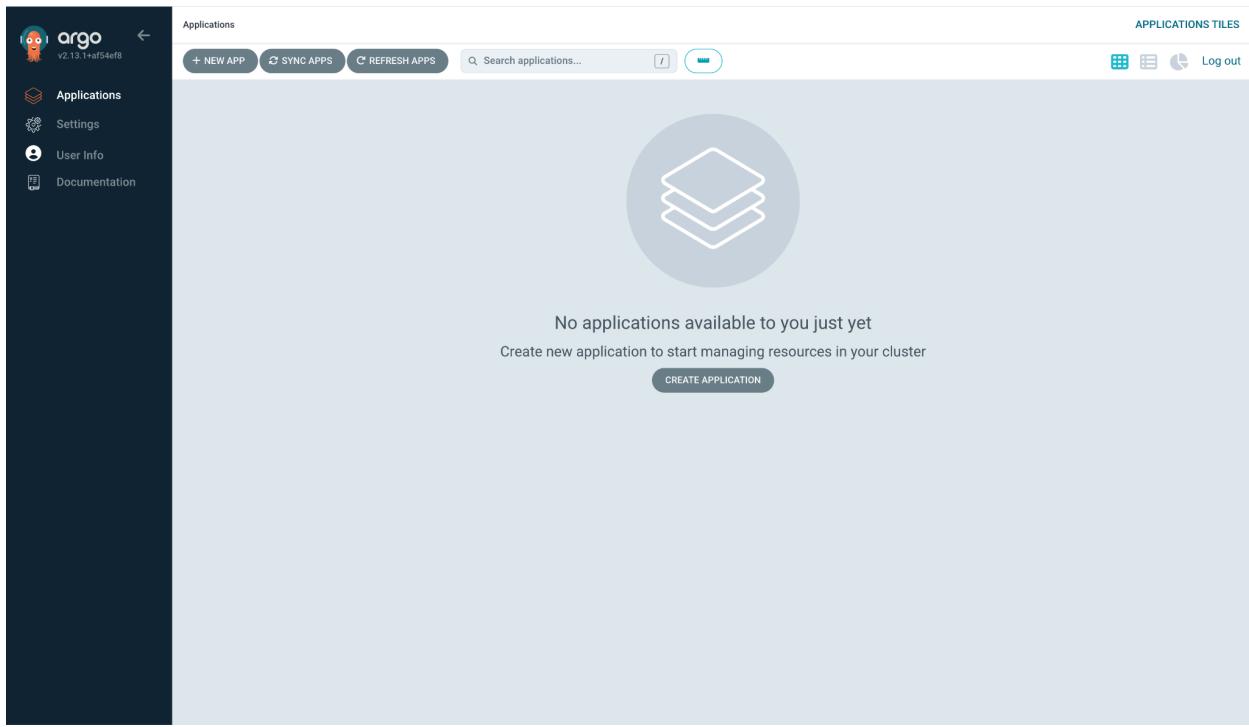
Task 4

Setup GitHub Actions CI/CD workflow to implement GitOps with ArgoCD deployment to any cloud-hosted Kubernetes cluster (EKS, GKE, AKS).

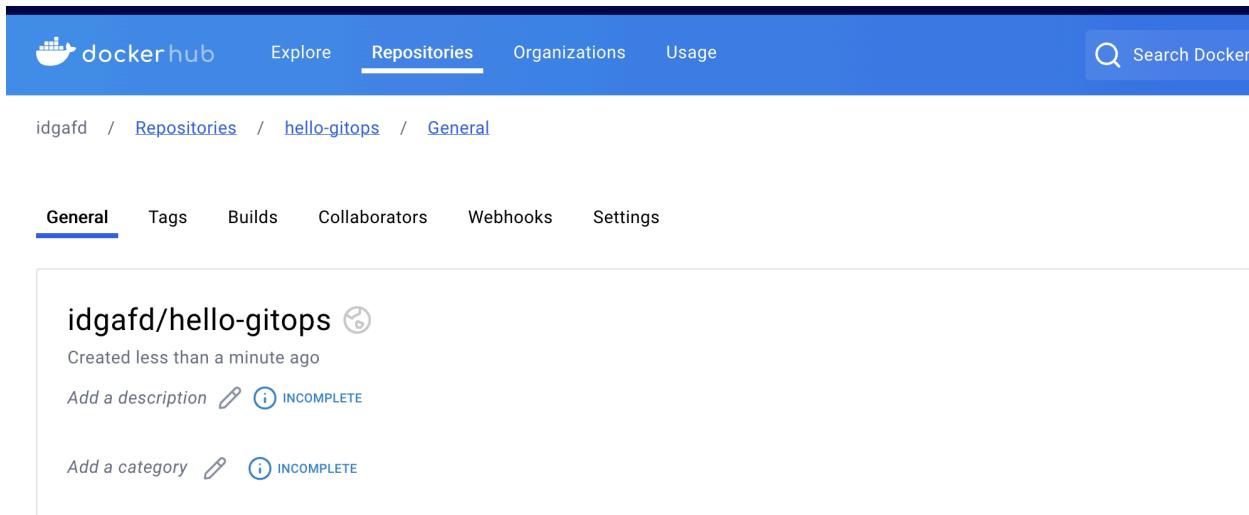
What to send: Github repo link with application code, k8s resources definitions, github actions workflow code.

Task 1

Default ArgoCD setup and login



New repository in DockerHub



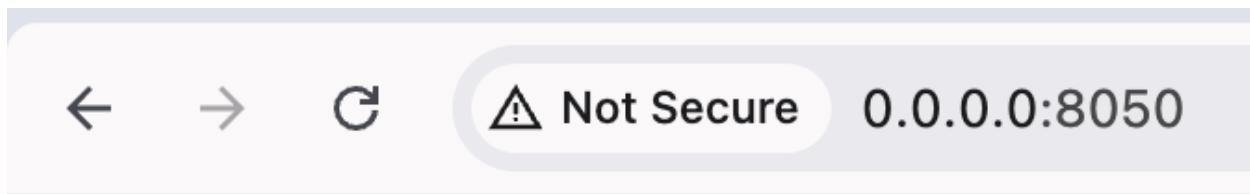
GitHub Actions secrets setup

The screenshot shows the GitHub Actions settings page for managing secrets. On the left, there's a sidebar with various repository settings like General, Access, Collaborators, and Code and automation. The main area is titled 'Actions secrets and variables'. It has two tabs: 'Secrets' (selected) and 'Variables'. Under 'Environment secrets', it says 'This environment has no secrets.' and has a 'Manage environment secrets' button. Under 'Repository secrets', there are two entries: 'DOCKER_PASSWORD' (last updated now) and 'DOCKER_USERNAME' (last updated 1 minute ago). A 'New repository secret' button is at the top right of this section. At the bottom of the sidebar, there are sections for Security, Integrations (GitHub Apps, Email notifications), Actions, Codespaces, and Dependabot.

Docker image build and test

```
anastasiamazur@bttrm-amazur-pro16 github-actions-samples % docker --version
Docker version 27.3.1, build ce12230
anastasiamazur@bttrm-amazur-pro16 github-actions-samples % docker build -t hello-gitops:latest ./app
[+] Building 8.2s (11/11) FINISHED                                            docker:desktop-linux
=> [internal] load build definition from Dockerfile                         0.0s
=> => transferring dockerfile: 267B                                         0.0s
=> [internal] load metadata for docker.io/library/python:3.7-slim          1.4s
=> [auth] library/python:pull token for registry-1.docker.io                0.0s
=> [internal] load .dockerignore                                           0.0s
=> => transferring context: 2B                                             0.0s
=> [1/5] FROM docker.io/library/python:3.7-slim@sha256:b53f496ca43e5af69  0.0s
=> => resolve docker.io/library/python:3.7-slim@sha256:b53f496ca43e5af69  0.0s
=> [internal] load build context                                           0.0s
=> => transferring context: 1.04kB                                         0.0s
=> CACHED [2/5] WORKDIR /app                                              0.0s
=> CACHED [3/5] RUN groupadd -r webservice && useradd --no-log-init -r -  0.0s
=> [4/5] COPY . .                                                       0.1s
=> [5/5] RUN pip install -r requirements.txt                            5.8s
=> exporting to image                                                 0.8s
=> => exporting layers                                                 0.5s
=> => exporting manifest sha256:73278971d276ca420bbbbf4dccc0b040d274d464 0.0s
=> => exporting config sha256:0b41151c350fed6a3bb1ee613fa06dc090c9afc540 0.0s
=> => exporting attestation manifest sha256:7c58d0b47ea98ec0144260b04b8b 0.0s
=> => exporting manifest list sha256:f1c4ad22a557e12d605da5289203a3f9fde 0.0s
=> => naming to docker.io/library/hello-gitops:latest                   0.0s
=> => unpacking to docker.io/library/hello-gitops:latest                  0.2s
anastasiamazur@bttrm-amazur-pro16 github-actions-samples % docker images
REPOSITORY           TAG      IMAGE ID      CREATED       SIZE
hello-gitops        latest   f1c4ad22a557  16 seconds ago  234MB
```

```
* Serving Flask app 'hello'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:8050
* Running on http://172.17.0.3:8050
Press CTRL+C to quit
172.17.0.1 - - [07/Dec/2024 15:12:09] "GET / HTTP/1.1" 200 -
```



Hello, Sasha!

Created App

hello-gitops

Project: default
Labels:
Status: Missing OutOfSync
Repository: https://github.com/idgafd/github-actions-samples
Target Rev.: HEAD
Path: k8s
Destination: in-cluster
Namespace: hello-gitops
Created At: 12/07/2024 17:30:16 (a few seconds ago)

SYNCHRONIZE CANCEL

Synchronizing application manifests from <https://github.com/idgafd/github-actions-samples>

Revision: HEAD

PRUNE DRY RUN APPLY ONLY FORCE

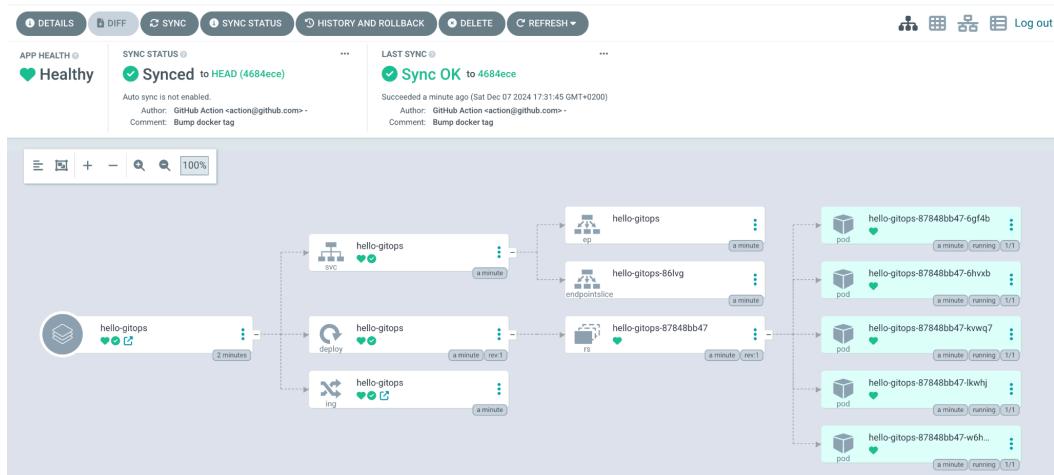
SYNC OPTIONS SKIP SCHEMA VALIDATION AUTO CREATE NAMESPACE
 PRUNE LAST APPLY OUT OF SYNC ONLY
 RESPECT IGNORE DIFFERENCES SERVER-SIDE APPLY

PRUNE PROPAGATION POLICY: foreground

REPLACE RETRY

SYNCHRONIZE RESOURCES: all / out of sync / none

/SERVICE/HELLO-GITOPS/HELLO-GITOPS
 APPS/DEPLOYMENT/HELLO-GITOPS/HELLO-GITOPS
 NETWORKING.K8S.IO/INGRESS/HELLO-GITOPS/HELLO-GITOPS



Change hello.py and rerun Sync

APP HEALTH ⓘ **Healthy**

SYNC STATUS ⓘ **Synced to HEAD (8fafd7a)**

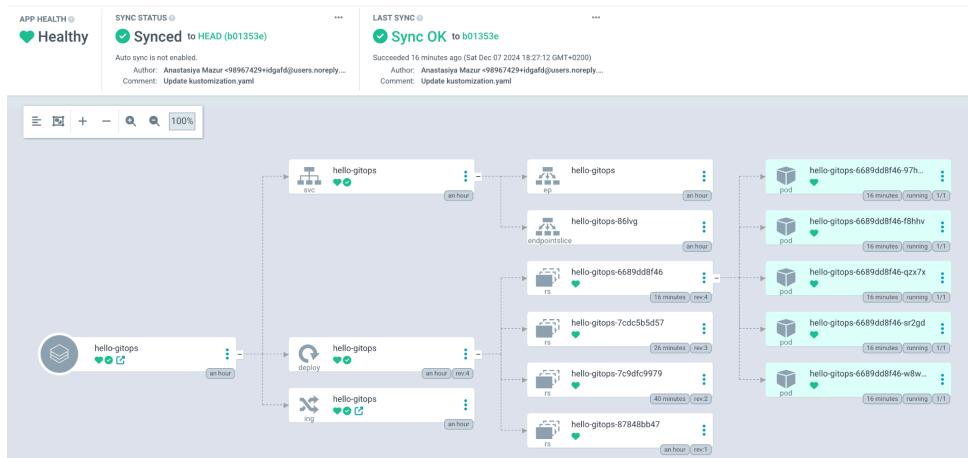
LAST SYNC ⓘ **Sync OK to 8fafd7a**

Auto sync is not enabled.

Author: Anastasiya Mazur <98967429+idgaf@users.noreply....
Comment: Update hello.py

Succeeded a few seconds ago (Sat Dec 07 2024 17:35:09 GMT+0200)
Author: Anastasiya Mazur <98967429+idgaf@users.noreply....
Comment: Update hello.py

ArgoCD UI



Check if the application has been successfully deployed

```
anastasiiamazur@bttrm-amazur-pro16 github-actions-samples % minikube status
minikube
type: Control Plane
host: Running
kublet: Running
apiserver: Running
kubeconfig: Configured

anastasiiamazur@bttrm-amazur-pro16 github-actions-samples % kubectl get all -n hello-gitops
NAME           READY   STATUS    RESTARTS   AGE
pod/hello-gitops-87848bb47-6gf4b   1/1    Running   0          7m33s
pod/hello-gitops-87848bb47-6hvxb   1/1    Running   0          7m33s
pod/hello-gitops-87848bb47-kwvq7   1/1    Running   0          7m33s
pod/hello-gitops-87848bb47-lkwj1   1/1    Running   0          7m33s
pod/hello-gitops-87848bb47-whm2   1/1    Running   0          7m33s

NAME            TYPE     CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
service/hello-gitops  ClusterIP  10.110.63.102  <none>        8080/TCP   7m33s

NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/hello-gitops  5/5     5           5          7m33s

NAME           DESIRED  CURRENT   READY   AGE
replicaset.apps/hello-gitops-87848bb47  5      5         5       7m33s
anastasiiamazur@bttrm-amazur-pro16 github-actions-samples % kubectl get pods -n hello-gitops
NAME           READY   STATUS    RESTARTS   AGE
hello-gitops-87848bb47-6gf4b   1/1    Running   0          7m53s
hello-gitops-87848bb47-6hvxb   1/1    Running   0          7m53s
hello-gitops-87848bb47-kwvq7   1/1    Running   0          7m53s
hello-gitops-87848bb47-lkwj1   1/1    Running   0          7m53s
hello-gitops-87848bb47-whm2   1/1    Running   0          7m53s
anastasiiamazur@bttrm-amazur-pro16 github-actions-samples % kubectl logs hello-gitops-87848bb47-6gf4b -n hello-gitops
* Serving Flask app "hello"
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:8080
* Running on http://10.244.0.20:8080
Press CTRL+C to quit
anastasiiamazur@bttrm-amazur-pro16 github-actions-samples % kubectl get svc -n hello-gitops
NAME           TYPE     CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
hello-gitops  ClusterIP  10.110.63.102  <none>        8080/TCP   8m59s
```

1. `minikube status`: check the status of Minikube
2. `kubectl get all -n hello-gitops`: check Kubernetes resources
3. `kubectl get pods -n hello-gitops`: check pods status in detail

4. `kubectl logs hello-gitops-87848bb47-6gf4b -n hello-gitops`: check logs of a specific pod
5. `kubectl get svc -n hello-gitops`: check access to the application through Minikube, we see a service of the ClusterIP type, so we will use portforward to access the web interface:

Access the application

```
|anastasiiamazur@bttrm-amazur-pro16 github-actions-samples % kubectl -n hello-gitops port-forward $(kubectl -n hello-gitops get po -o name | tail -n 1) 8111:8050
Forwarding from 127.0.0.1:8111 -> 8050
Forwarding from [::1]:8111 -> 8050
Handling connection for 8111
Handling connection for 8111
Handling connection for 8111
```

`kubectl -n hello-gitops port-forward $(kubectl -n hello-gitops get po -o name | tail -n 1) 8111:8050`

1. `kubectl -n hello-gitops`: indicates that we are running in the hello-gitops namespace
2. `kubectl -n hello-gitops get po -o name`: gets a list of all pods in the hello-gitops namespace (only names due to `-o name`)
3. `| tail -n 1`: selects the last sub in the list
4. `$(...)`: wrapper for executing a command within a command (in this case, the name of the last pod in the list is returned)
5. `port-forward`: forwards ports from the local host machine to the pod in Kubernetes (in this case, local port 8111 to port 8050 in the pod)

Result overview



Hello, Sasha!

But in the process of launching, I changed the inscription to “Hello, Anastasia!” and would like to see it...

Right way to change message step by step

1. Update code (`hello.py`)
2. Build and push Docker image
 - a. Note: using the `latest` tag may lead to Kubernetes using a cached image instead of pulling the updated one (and that was my mistake, then I remembered this tip from one of previous lectures and fixed that)

```
b. docker build -t idgafd/hello-gitops:2.0  
c. docker push idgafd/hello-gitops:2.0
```

[idgafd/hello-gitops](#)

Last pushed about 1 hour ago

Add a description INCOMPLETE

Add a category INCOMPLETE

Tags

This repository contains 2 tag(s).

Tag	OS	Type	Pulled	Pushed
2.0		Image	an hour ago	an hour ago
latest		Image	an hour ago	an hour ago

3. Update the kustomization.yaml file adding new image name and tag
4. Synchronize changes with ArgoCD (click Sync if settings are manual)

File changes

[github-actions-samples / k8s / kustomization.yaml](#) [github-actions-samples / app / hello.py](#)

```
idgafd Update kustomization.yaml
idgafd Update hello.py

Code Blame 20 lines (14 loc) · 377 Bytes Code 55% fast

Code Blame 11 lines (11 loc) · 223 Bytes

1 apiVersion: kustomize.config.k8s.io/v1beta1
2 kind: Kustomization
3 resources:
4 - deployment.yaml
5 - service.yaml
6 - ingress.yaml
7 namespace: hello-gitops
8 images:
9 - name: hello-gitops
10 newName: idgafd/hello-gitops
11 newTag: "2.0"
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

Result overview

```
[anastasiamazur@bttrm-amazur-pro16 ~ % curl localhost:8111
[Hello, Anastasiia!%]
```



Hello, Anastasiia!

Some screenshots from ArgoCD UI

APP HEALTH **SYNC STATUS** Synced to HEAD (b01353e) **LAST SYNC** Sync OK to b01353e

Auto sync is not enabled.
Author: Anastasiya Mazur <98967429+idgaf@users.noreply...
Comment: Update kustomization.yaml

Succeeded an hour ago (Sat Dec 07 2024 18:27:12 GMT+0200)
Author: Anastasiya Mazur <98967429+idgaf@users.noreply...
Comment: Update kustomization.yaml

hello-gitops

hello-gitops-6689dd8f46-w8wx4

SUMMARY **EVENTS** **LOGS**

KIND	Pod
NAME	hello-gitops-6689dd8f46-w8wx4
NAMESPACE	hello-gitops
CREATED AT	12/07/2024 18:27:18 (17 minutes ago)
IMAGES	idgaf/hello-gitops:2.0
STATE	Running
CONTAINER STATE	hello-gitops Container is running. It is started and ready.
HEALTH	Healthy
LINKS	

LIVE MANIFEST

```

1  apiVersion: v1
2  kind: Pod
3  metadata:
4    annotations:
5      kubectl.kubernetes.io/restartedAt: '2024-12-07T18:17:25+02:00'
6      creationTimestamp: '2024-12-07T16:27:18Z'
7      generateName: hello-gitops-6689dd8f46-
8    labels:
9      app: hello-gitops
10     pod-template-hash: 6689dd8f46

```

Hide Managed Fields **EDIT**

hello-gitops-6689dd8f46-w8wx4 ❤️

pod

SUMMARY **EVENTS** **LOGS**

REASON	MESSAGE	COUNT	FIRST OCCURRED	LAST OCCURRED
Pulled	Container image "idgafd/hello-gitops:2.0" already present on machine	1	17m ago Today at 6:27 PM	17m ago Today at 6:27 PM
Created	Created container hello-gitops	1	17m ago Today at 6:27 PM	17m ago Today at 6:27 PM
Started	Started container hello-gitops	1	17m ago Today at 6:27 PM	17m ago Today at 6:27 PM
Scheduled	Successfully assigned hello-gitops/hello-gitops-6689dd8f46-w8wx4 to minikube	1	17m ago Today at 6:27 PM	17m ago Today at 6:27 PM

hello-gitops-6689dd8f46-w8wx4 ❤️

SUMMARY **EVENTS** **LOGS**

containing

* Serving Flask app 'hello'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:8050
* Running on http://10.244.0.34:8050
Press CTRL+C to quit
127.0.0.1 - - [07/Dec/2024 16:28:08] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [07/Dec/2024 16:28:34] "GET / HTTP/1.1" 200 -

hello-gitops ❤️

deploy

SUMMARY **EVENTS** **LOGS**

containing

* Serving Flask app 'hello'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:8050
* Running on http://10.244.0.34:8050
Press CTRL+C to quit
* Serving Flask app 'hello'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:8050
* Running on http://10.244.0.34:8050
Press CTRL+C to quit
* Serving Flask app 'hello'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:8050
* Running on http://10.244.0.34:8050
Press CTRL+C to quit
* Serving Flask app 'hello'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:8050
* Running on http://10.244.0.34:8050
Press CTRL+C to quit
127.0.0.1 - - [07/Dec/2024 16:28:08] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [07/Dec/2024 16:28:34] "GET / HTTP/1.1" 200 -

Task 3

Google Cloud SDK installation

```
anastasiiamazur@bttrm-amazur-pro16 ~ % cd ~/Downloads/google-cloud-sdk  
anastasiiamazur@bttrm-amazur-pro16 google-cloud-sdk % ./install.sh  
  
Welcome to the Google Cloud CLI!  
  
To help improve the quality of this product, we collect anonymized usage data  
and anonymized stacktraces when crashes are encountered; additional information  
is available at <https://cloud.google.com/sdk/usage-statistics>. This data is  
handled in accordance with our privacy policy  
<https://cloud.google.com/terms/cloud-privacy-notice>. You may choose to opt in this  
collection now (by choosing 'Y' at the below prompt), or at any time in the  
future by running the following command:  
  
    gcloud config set disable_usage_reporting false  
  
Do you want to help improve the Google Cloud CLI (y/N)? y  
  
Your current Google Cloud CLI version is: 502.0.0  
The latest available version is: 502.0.0  
  
To install or remove components at your current SDK version [502.0.0], run:  
    $ gcloud components install COMPONENT_ID  
    $ gcloud components remove COMPONENT_ID  
  
To update your SDK installation to the latest version [502.0.0], run:  
    $ gcloud components update  
  
Modify profile to update your $PATH and enable shell command completion?  
  
Do you want to continue (Y/n)? y  
  
The Google Cloud SDK installer will now prompt you to update an rc file to bring  
the Google Cloud CLIs into your environment.  
  
Enter a path to an rc file to update, or leave blank to use  
[/Users/anastasiiamazur/.zshrc]:  
[/Users/anastasiiamazur/.zshrc] has been updated.  
  
==> Start a new shell for the changes to take effect.  
  
Google Cloud CLI works best with Python 3.11 and certain modules.  
Download and run Python 3.11 installer? (Y/n)? y  
  
Running Python 3.11 installer, you may be prompted for sudo password...  
Password:  
installer: Package name is Python  
installer: Installing at base path /  
installer: The install was successful.  
Setting up virtual environment  
Creating virtualenv...  
Installing modules...  
      5.5/5.5 MB 8.5 MB/s eta 0:00:00  
      89.7/89.7 kB 928.8 kB/s eta 0:00:00  
Installing build dependencies ... done  
Getting requirements to build wheel ... done  
Preparing metadata (pyproject.toml) ... done  
      58.4/58.4 kB 2.9 MB/s eta 0:00:00  
      11.2/11.2 MB 10.5 MB/s eta 0:00:00  
      167.3/167.3 kB 5.5 MB/s eta 0:00:00  
      178.7/178.7 kB 8.1 MB/s eta 0:00:00  
Building wheel for crcmod (pyproject.toml) ... done  
Virtual env enabled.  
  
For more information on how to get started, please visit:  
    https://cloud.google.com/sdk/docs/quickstarts  
  
anastasiiamazur@bttrm-amazur-pro16 google-cloud-sdk % gcloud --version  
zsh: command not found: gcloud  
anastasiiamazur@bttrm-amazur-pro16 google-cloud-sdk % nano ~/.zshrc  
anastasiiamazur@bttrm-amazur-pro16 google-cloud-sdk % source ~/.zshrc  
anastasiiamazur@bttrm-amazur-pro16 google-cloud-sdk % gcloud --version  
Google Cloud SDK 502.0.0  
bq 2.1.9  
core 2024.11.15  
gcloud-crc32c 1.0.0  
gsutil 5.31
```

Google Cloud login

```
anastasiiamazur@bttrm-amazur-pro16 ~ % gcloud auth login
```

Your browser has been opened to visit:

```
https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A8085%2F&scope=openid+https%3A%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2Fwww.googleapis.com%2Fauth%2Fsqlservice.logIn+https%3A%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&state=M4sj1c7fmMtbiXyAOb_mDzxMUV7b1E&access_type=offline&code_challenge=nm-Ua79LOu3KSqaIDboKH_-lVA6Xkyv3b1bBjhYrVRc&code_challenge_method=S256
```

You are now logged in as [nastiamazur.v@gmail.com].
Your current project is [None]. You can change this setting by running:
\$ gcloud config set project PROJECT_ID

Cluster creation

Cloud SDK > Documentation

Was this helpful?

You are now authenticated with the gcloud CLI!

[Send feedback](#)

The authentication flow has completed successfully. You may close this window, or check out the resources below.

The screenshot shows the 'Create an Autopilot cluster' page. At the top, there's a message about free trial status and credits, with an 'Activate' button. Below that, a back arrow and the title 'Create an Autopilot cluster'. To the right are 'SWITCH TO STANDARD CLUSTER' and 'LEARN' buttons. A warning message says to ensure the default GKE node service account has the 'Kubernetes Engine Default Node Service Account role' permissions. The main area is divided into sections:

- Cluster basics**: Set up basics for your cluster. Includes 'Fleet registration' (Manage multiple clusters together), 'Networking' (Define applications communication in the cluster), 'Advanced settings' (Review additional options), and 'Review and create' (Review all settings and create your cluster).
- Cluster basics details**: Describes creating an Autopilot cluster by specifying a name and region. It includes a 'Name' input field with 'autopilot-cluster-1' typed in, a 'Region' dropdown set to 'us-central1', and a note about cluster names starting with lowercase letters and being unique.
- Cluster tier**: Chooses a GKE cluster tier based on workload complexity. It highlights 'Run business critical workloads faster, safer, and easier at enterprise scale' and describes GKE Enterprise features like multi-cluster and multi-team support.
- Action buttons**: 'LEARN AND ENABLE' (disabled), 'CREATE' (blue button), 'CANCEL', and links to 'Equivalent REST' or 'COMMAND LINE'.

The screenshot shows the Google Cloud Kubernetes Engine interface. On the left, there's a sidebar with 'Learn about Enterprise', 'All Fleets', 'Resource Management' (with 'Overview' and 'Clusters' selected), and 'Workloads'. The main area is titled 'Kubernetes clusters' with tabs for 'OVERVIEW', 'OBSERVABILITY', and 'COST OPTIMIZATION'. A search bar says 'Filter Enter property name or value'. Below it is a table with columns: Status, Name (sorted by Name), Location, Tier, Number of nodes, Total vCPUs, Total memory, Notifications, and Labels. One row is visible: 'Status' (green checkmark), 'Name' (autopilot-cluster-1), 'Location' (us-central1), 'Tier' (Standard), 'Number of nodes' (0), 'Total vCPUs' (0), 'Total memory' (0 GB), 'Notifications' (none), and 'Labels' (empty).

Gcloud components install to check cluster locally

```
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % gcloud components install gke-gcloud-auth-plugin
```

Your current Google Cloud CLI version is: 502.0.0
Installing components from version: 502.0.0

These components will be installed.		
Name	Version	Size
gke-gcloud-auth-plugin (Platform Specific)	0.5.9	4.0 MiB

For the latest full release notes, please visit:
https://cloud.google.com/sdk/release_notes

Once started, canceling this operation may leave your SDK installation in an inconsistent state.

Do you want to continue (Y/n)? y

Performing in place update...

— Downloading: gke-gcloud-auth-plugin	—
— Downloading: gke-gcloud-auth-plugin (Platform Specific)	—
— Installing: gke-gcloud-auth-plugin	—
— Installing: gke-gcloud-auth-plugin (Platform Specific)	—

Performing post processing steps...done.

Update done!

```
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % gcloud components list
```

Your current Google Cloud CLI version is: 502.0.0
The latest available version is: 502.0.0

```
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % gcloud container clusters get-credentials autopilot-cluster-1 --region us-central1 --project snappy-spanner-444-103-i2
```

Fetching cluster endpoint and auth data.

WARNING: cluster autopilot-cluster-1 is not RUNNING. The kubernetes API may or may not be available. Check the cluster status for more information.
kubeconfig entry generated for autopilot-cluster-1.

Cluster is running

The screenshot shows the Google Cloud Kubernetes Engine interface. The table in the center has a new column 'Status' with a green checkmark next to 'autopilot-cluster-1'. All other columns remain the same: Name (sorted by Name), Location, Tier, Number of nodes, Total vCPUs, Total memory, Notifications, and Labels.

Local check

```
[anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % gcloud container clusters get-credentials autopilot-cluster-1 --region us-central1 --project snappy-spanner-444-103-i2
Fetching cluster endpoint and auth data.
kubeconfig entry generated for autopilot-cluster-1.
```

Cluster is empty until actual application run

```
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % kubectl get nodes  
No resources found  
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % kubectl get pods --all-namespaces
```

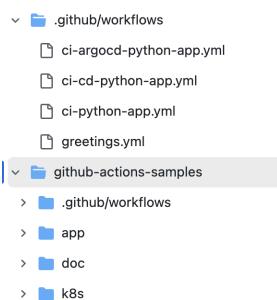
NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
gke-gmp-system	alertmanager-0	0/2	Pending	0	3m36s
gke-gmp-system	gmp-operator-84d74f7bdd-6sptm	0/1	Pending	0	3m36s
gke-gmp-system	rule-evaluator-6f659bc47f-sls49	0/2	Pending	0	3m36s
gke-managed-cim	kube-state-metrics-0	0/2	Pending	0	5m5s
kube-system	antrea-controller-horizontal-autoscaler-788c6c4c5b-skbbk	0/1	Pending	0	3m41s
kube-system	egress-nat-controller-76464f8896-r479x	0/1	Pending	0	3m27s
kube-system	event-exporter-gke-79cd469d79-r2m5g	0/2	Pending	0	4m59s
kube-system	konnectivity-agent-7d4dfcd799-2xbsn	0/2	Pending	0	3m6s
kube-system	konnectivity-agent-autoscaler-696cc5598c-lkvtt5	0/1	Pending	0	3m5s
kube-system	kube-dns-7d6869cf9d-c5b1v	0/5	Pending	0	5m18s
kube-system	kube-dns-autoscaler-6f896b6968-zkbqz	0/1	Pending	0	5m18s
kube-system	17-default-backend-78478f96f4-qlh85	0/1	Pending	0	3m2s
kube-system	metrics-server-v1.30.3-7fff7dc68d-7nrs8	0/1	Pending	0	3m29s
kube-system	metrics-server-v1.30.3-8987bd844-vtmh2	0/1	Pending	0	3m32s

Cluster detailed info via ID with GC login

```
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % gcloud container clusters describe autopilot-cluster-1 --region us-central1 --project snappy-spanner-444103-i2  
addonsConfig:  
  dnsConfig:  
    enabled: true  
  gcovPersistentDiskCsIDriverConfig:  
    enabled: true  
  gcpFilestoreCsIDriverConfig:  
    enabled: true  
  gcpFileCsIDriverConfig:  
    enabled: true  
  horizontalPodAutoscaling: {}  
  httpLoadBalancing: {}  
  kubernetesDashboard:  
    disabled: false  
  networkPolicyConfig:  
    disabled: true  
  parallelStoreCsIDriverConfig:  
    enabled: true  
  rawDiskConfig: {}  
  staticIamConfig:  
    enabled: true  
  authenticatorGroupsConfig: {}  
  autopilot:  
    enabled: true  
  autoScaling:  
    autoProvisioningNodePoolDefaults:  
      imageType: COS_CONTAINERD  
      management:  
        autoRepair: true  
        autoUpdate: true  
      oauthScopes:  
        - https://www.googleapis.com/auth/devstorage.read_only  
        - https://www.googleapis.com/auth/logging.write  
        - https://www.googleapis.com/auth/servicecontrol  
        - https://www.googleapis.com/auth/service.management.readonly  
        - https://www.googleapis.com/auth/trace.append  
      serviceAccount: default  
      upgradeSettings:  
        strategy: SURGE  
    autoscalingProfile: OPTIMIZE_UTILIZATION  
    enableNodeAutoprovisioning: true  
  resourceLimits:  
    - maximum: '1000000000'  
    resourceType: cpu
```

Project preparation

1. Moving files from forked repository into main course repository: moving workflows that will work in Actions to the root directory (the original ones are still stored in the original folder), add Dockerfile



2. Change main executable (to add some new tests at least)

```

1 import configparser
2 from flask import Flask
3
4 def get_message():
5     config = configparser.RawConfigParser()
6     config.read('config.properties')
7     if config.getboolean("features", "feature_1"):
8         return "Hello, Anastasia!"
9     else:
10        return "Hello, World!"
11
12 app = Flask(__name__)
13
14 @app.route("/")
15 def hello():
16     return get_message()
17
18 if __name__ == "__main__":
19     app.run(host='0.0.0.0', port=8050)

```

```

1 import pytest
2 from hello import app, get_message
3 from unittest.mock import patch
4
5
6 def test_get_message_anastasia():
7     with patch("hello.configparser.RawConfigParser.getboolean", return_value=True):
8         assert get_message() == "Hello, Anastasia!"
9
10 def test_get_message_world():
11     with patch("hello.configparser.RawConfigParser.getboolean", return_value=False):
12         assert get_message() == "Hello, World!"
13
14
15 @pytest.fixture
16 def client():
17     with app.test_client() as client:
18         yield client
19
20
21 def test_hello_endpoint_anastasia(client):
22     with patch("hello.configparser.RawConfigParser.getboolean", return_value=True):
23         response = client.get("/")
24         assert response.status_code == 200
25         assert response.data.decode() == "Hello, Anastasia!"
26
27 def test_hello_endpoint_world(client):
28     with patch("hello.configparser.RawConfigParser.getboolean", return_value=False):
29         response = client.get("/")
30         assert response.status_code == 200
31         assert response.data.decode() == "Hello, World!"
32
33

```

3. Fix working directory path and run CI test workflow

working-directory: ./github-actions-samples/app



4. Set all necessary secrets

Repository secrets		New repository secret
Name	Last updated	
DOCKER_PASSWORD	5 hours ago	
DOCKER_USERNAME	5 hours ago	
GCP_CREDENTIALS	5 hours ago	
GKE_PROJECT	5 hours ago	

5. Install the gke-gcloud-auth-plugin required for kubectl authentication

```

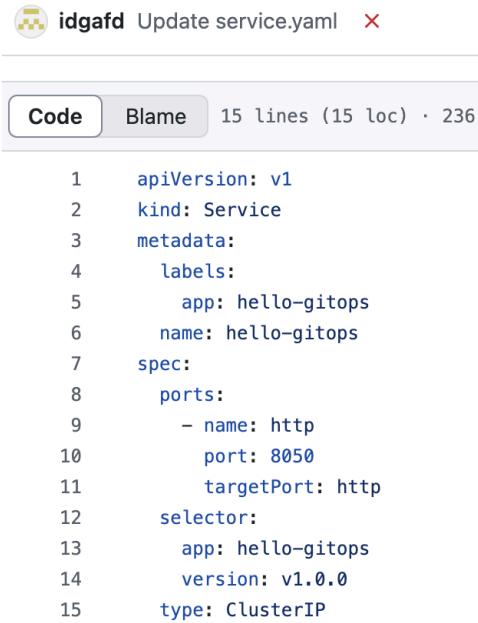
# - name: Set up Cloud SDK
#   uses: google-github-actions/setup-gcloud@v1
# Set up Cloud SDK and install the GKE Auth plugin
- name: Set up Cloud SDK and install gke-gcloud-auth-plugin
  uses: google-github-actions/setup-gcloud@v1
  with:
    project_id: '${{ secrets.GCP_PROJECT_ID }}' # Optional: Set project ID if needed
- run: |
  gcloud components install gke-gcloud-auth-plugin
  gcloud container clusters get-credentials "$GKE_CLUSTER" --region "$GKE_REGION"

```

6. Update GitHub Actions workflow to correctly reference the full path to the `k8s` directory

```
cd $GITHUB_WORKSPACE/github-actions-samples/k8s
```

7. Add a version label to a Kubernetes service for control, flexibility, and rollback options, for instance dynamically switch traffic between different versions without downtime



The screenshot shows a GitHub code editor interface. At the top, there's a title bar with a yellow icon, the text "idgafd", and "Update service.yaml". Below the title bar is a navigation bar with tabs: "Code" (which is selected), "Blame", and "15 lines (15 loc) · 236". The main area contains the following YAML code:

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    labels:
5      app: hello-gitops
6      name: hello-gitops
7  spec:
8    ports:
9      - name: http
10     port: 8050
11     targetPort: http
12   selector:
13     app: hello-gitops
14     version: v1.0.0
15   type: ClusterIP
```

8. Create Kubernetes namespace and service to match previous setting in manifest files

```
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % kubectl create ns hello-gitops
namespace/hello-gitops created
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % kubectl apply -f github-actions-samples/k8s/service.yaml -n hello-gitops
service/hello-gitops created
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % kubectl get svc -n hello-gitops
NAME        TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
hello-gitops  ClusterIP  34.118.229.242  <none>           8050/TCP    15s
```

Result overview

The CI/CD Action ended up with error code 1 at step step "*Create namespace & set in Kustomize*" with error description "*Both Blue and Green namespaces are present! Exiting*". The script correctly checks for the existence of two namespaces:

- `python-app-blue`
- `python-app-green`

I assumed that the error occurred because the script was designed to exit if both namespaces were present 😊

```

v ✘ Create namespace & set in Kustomize
1 ▼Run namespacePresentBlue='kubectl get ns | grep python-app-blue | wc -l'
2 namespacePresentBlue='kubectl get ns | grep python-app-blue | wc -l'
3 namespacePresentGreen='kubectl get ns | grep python-app-green | wc -l'
4 if [ $namespacePresentBlue -eq 0 ]
5 then
6     kubectl create ns python-app-blue
7     cd $GITHUB_WORKSPACE/github-actions-samples/k8s
8     kustomize edit set namespace python-app-blue
9     echo "NEW_NAMESPACE=python-app-blue" >> $GITHUB_ENV
10    echo "OLD_NAMESPACE=python-app-green" >> $GITHUB_ENV
11    elif [ $namespacePresentGreen -eq 0 ]
12    then
13        kubectl create ns python-app-green
14        cd $GITHUB_WORKSPACE/github-actions-samples/k8s
15        kustomize edit set namespace python-app-green
16        echo "NEW_NAMESPACE=python-app-green" >> $GITHUB_ENV
17        echo "OLD_NAMESPACE=python-app-blue" >> $GITHUB_ENV
18    else
19        echo "Both Blue and Green namespaces are present! Exiting"
20        exit 1
21    fi
22 shell: /usr/bin/bash -e ***0***  

23 env:
24     PROJECT_ID: ***
25     GKE_CLUSTER: autopilot-cluster-1
26     GKE_REGION: us-central1
27     CLOUDSDK_AUTH_CREDENTIAL_FILE_OVERRIDE: /home/runner/work/UCU_DevOps_Course/UCU_DevOps_Course/gha-creds-82a25e28fed8577d.json
28     GOOGLE_APPLICATION_CREDENTIALS: /home/runner/work/UCU_DevOps_Course/UCU_DevOps_Course/gha-creds-82a25e28fed8577d.json
29     GOOGLE_GHA_CREDS_PATH: /home/runner/work/UCU_DevOps_Course/UCU_DevOps_Course/gha-creds-82a25e28fed8577d.json
30     CLOUDSDK_CORE_PROJECT: ***
31     CLOUDSDK_PROJECT: ***
32     GCLOUD_PROJECT: ***
33     GCP_PROJECT: ***
34     GOOGLE_CLOUD_PROJECT: ***
35     CLOUDSDK_METRICS_ENVIRONMENT: github-actions-setup-gcloud
36     CLOUDSDK_METRICS_ENVIRONMENT_VERSION: 1.1.1
37 Both Blue and Green namespaces are present! Exiting
38 Error: Process completed with exit code 1.

```

The output of `kubectl get ns` confirms that: both `python-app-blue` and `python-app-green` namespaces exist and are active. These namespaces were likely created earlier during previous workflow runs or manually.

```
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % kubectl get ns
```

NAME	STATUS	AGE
default	Active	91m
gke-gmp-system	Active	89m
gke-managed-cim	Active	90m
gke-managed-filestorecsi	Active	90m
gke-managed-system	Active	89m
gmp-public	Active	89m
kube-node-lease	Active	91m
kube-public	Active	91m
kube-system	Active	91m
python-app-blue	Active	11m
python-app-green	Active	6m21s

Steps taken next to complete the CI/CD workflow

1. Modified the logic to determine the active namespace (the one currently serving traffic) using the `kubectl` command

```
echo "Both namespaces are present. Determining active namespace..."
activeNamespace=`kubectl get service hello-gitops -n hello-gitops -o=jsonpath='{.spec.selector.version}'`
```

2. Added a step to switch traffic to the new namespace

```
- name: Switch traffic to new namespace
  run: |
    echo "Switching traffic to $NEW_NAMESPACE..."
    kubectl patch service hello-gitops -n hello-gitops -p '{"spec":{"selector":{"version":"v2.0.0"}}}'
```

3. Ensured that unused namespaces are deleted to free up cluster resources

```
- name: Remove old namespace
  run: |
    echo "Removing old namespace: $OLD_NAMESPACE"
    kubectl delete ns $OLD_NAMESPACE --wait=true
```

4. Add update of Kubernetes resources (kustomization.yaml, deployment.yaml) to the new version and committed changes back to the GitHub repository using the ad-m/github-push-action@master action

```
- name: Commit files
  run: |
    git config --local user.email "action@github.com"
    git config --local user.name "GitHub Action"
    git commit -am "Bump docker tag"

- name: Push changes
  uses: ad-m/github-push-action@master
  with:
    github_token: ${{ secrets.GITHUB_TOKEN }}
```

5. Granted GitHub Workflows permission to write

Workflow permissions

Choose the default permissions granted to the GITHUB_TOKEN when running workflows in this repository. You can specify more granular permissions in the workflow using YAML. [Learn more about managing permissions](#).

Read and write permissions

Workflows have read and write permissions in the repository for all scopes.

Read repository contents and packages permissions

Workflows have read permissions in the repository for the contents and packages scopes only.

Choose whether GitHub Actions can create pull requests or submit approving pull request reviews.

Allow GitHub Actions to create and approve pull requests

[Save](#)

Parts of original code are commented but still present in the file, so the viewer can take a look at changes.

Result overview

 Update ci-cd-python-app.yml

CI/CD Workflow #32: Commit [a721bcb](#) pushed by idgadf

main

2 hours ago

1m 35s

...

Initial service version

```
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % kubectl get service hello-gitops -n hello-gitops -o=jsonpath='{.spec.selector.version}'  
v1.0.0%
```

And after the committed updates

```
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % kubectl get service hello-gitops -n hello-gitops -o=jsonpath='{.spec.selector.version}'  
v2.0.0%
```

Received new image tag

[UCU_DevOps_Course / github-actions-samples / k8s / kustomization.yaml](#)

```
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
resources:
- deployment.yaml
- service.yaml
- ingress.yaml
namespace: python-app-blue
images:
- name: hello-gitops
  newName: idgafid/hello-gitops
  newTag: a721bc3403ac9957f6367a92b90b295e4249e1
```

Check new namespace pods

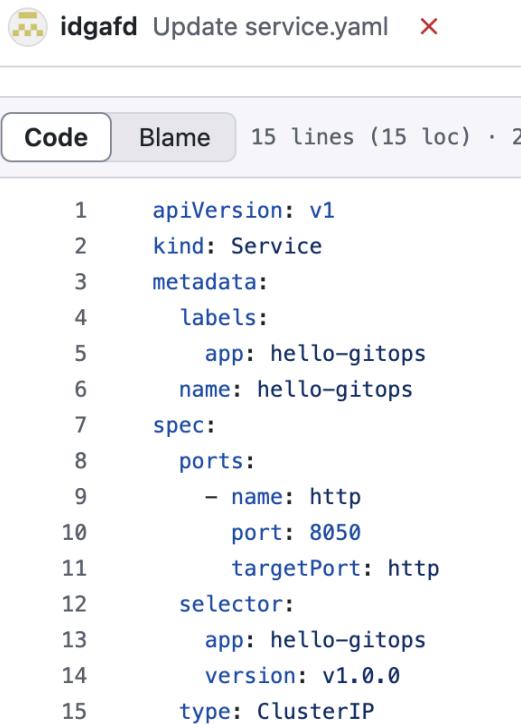
```
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % kubectl describe pods -n python-app-green
Name:           hello-gitops-79889c5546-425q2
Namespace:      python-app-green
Priority:      0
Service Account: default
Node:          gk3-autopilot-cluster-1-pool-2-edab5f82-tknq/10.128.0.5
Start Time:    Sun, 08 Dec 2024 13:25:40 +0200
Labels:        app=hello-gitops
Annotations:   pod-template-hash=79889c5546
               cloud.google.com/cluster_autoscaler_unhelpable_since: 2024-12-08T11:25:37+0000
               cloud.google.com/cluster_autoscaler_unhelpable_until: Inf
Status:        Terminating (lasts <invalid>)
Termination Grace Period: 30s
SeccompProfile: RuntimeDefault
IP:            10.28.0.87
IPs:          IP: 10.28.0.87
```

But I still wasn't entirely satisfied 😞 with the result because of **problems** such as:

- Hardcoded service version

```
- name: Switch traffic to new namespace
  run: |
    echo "Switching traffic to $NEW_NAMESPACE..."
    kubectl patch service hello-gitops -n hello-gitops -p '{"spec":{"selector":{"version":"v2.0.0"}}}'
```

- No automatic update of service version in service.yaml



The screenshot shows a GitHub commit interface. At the top, there's a small icon of a person with a checkmark, followed by the text "idgafd Update service.yaml" and a red "X" button. Below this is a navigation bar with tabs: "Code" (which is selected), "Blame", and "15 lines (15 loc) · 2". The main area contains the YAML configuration for a service:

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    labels:
5      app: hello-gitops
6      name: hello-gitops
7  spec:
8    ports:
9      - name: http
10     port: 8050
11     targetPort: http
12   selector:
13     app: hello-gitops
14     version: v1.0.0
15   type: ClusterIP
```

Therefore, I made the corresponding improvements in the code (the commented parts are previous versions):

```
# - name: Switch traffic to new namespace
#   run: |
#     echo "Switching traffic to $NEW_NAMESPACE..."
#     kubectl patch service hello-gitops -n hello-gitops -p '{"spec":{"selector":{"version":"v2.0.0"}}}'
```



```
- name: Switch traffic to new namespace
  run: |
    echo "Switching traffic to $NEW_NAMESPACE..."
```



```
# Get the current version from the service selector
currentVersion=$(kubectl get service hello-gitops -n hello-gitops -o=jsonpath='{.spec.selector.version}')
```



```
# Increment the patch version (Z) in X.Y.Z format
newVersion=$(echo $currentVersion | awk -F. '{$NF = $NF + 1; print $1"."$2"."$3}' | tr -d '[:space:]')
```



```
# Patch the service with the new version
echo "Updating service to use version $newVersion..."
kubectl patch service hello-gitops -n hello-gitops -p "{\"spec\":{\"selector\":{\"version\":\"$newVersion\"}}}"
```

```

# - name: Commit files
#   run: |
#     git config --local user.email "action@github.com"
#     git config --local user.name "GitHub Action"
#     git commit -am "Bump docker tag"

- name: Update service.yaml with the new version
  run: |
    echo "Fetching currently new version from service selector..."
    newVersion=$(kubectl get service hello-gitops -n hello-gitops -o=jsonpath='{.spec.selector.version}')
    echo "Updating service.yaml with version $newVersion..."
    sed -i "s/version: .*/version: $newVersion/" github-actions-samples/k8s/service.yaml
    echo "Updated service.yaml:"
    cat github-actions-samples/k8s/service.yaml

- name: Commit updated service.yaml
  run: |
    echo "Configuring Git for commit..."
    git config --local user.email "action@github.com"
    git config --local user.name "GitHub Action"

    echo "Checking for changes in service.yaml..."
    if git diff --exit-code github-actions-samples/k8s/service.yaml; then
      echo "No changes detected in service.yaml. Skipping commit."
    else
      echo "Changes detected in service.yaml. Committing changes..."
      git add github-actions-samples/k8s/service.yaml
      git commit -m "Update service.yaml to reflect version $newVersion"
    fi

```

Result overview

[Update ci-cd-python-app.yml](#) main 6 minutes ago ...
CI/CD Workflow #42: Commit 44802c9 pushed by idgafd

[anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % kubectl get service hello-gitops -n hello-gitops -o=jsonpath='{.spec.selector.version}' v2.0.5]

Yes... it took a bit of time 😊

Updated version in service.yaml

[UCU_DevOps_Course / github-actions-samples / k8s / service.yaml](#)

The GitHub code preview interface shows the updated content of the service.yaml file. The file is 15 lines long and 236 bytes in size. It includes annotations indicating a 55% faster write speed.

```

1  apiVersion: v1
2  kind: Service
3  metadata:
4    labels:
5      app: hello-gitops
6      name: hello-gitops
7  spec:
8    ports:
9      - name: http
10     port: 8050
11     targetPort: http
12   selector:
13     app: hello-gitops
14     version: v2.0.5
15   type: ClusterIP

```

Updated image tag in kustomization.yaml

[UCU_DevOps_Course / github-actions-samples / k8s / kustomization.yaml](#) ↗

actions-user Bump docker tag

Code Blame 11 lines (11 loc) · 262 Bytes Code 55% faster with GitHub

```
1 apiVersion: kustomize.config.k8s.io/v1beta1
2 kind: Kustomization
3 resources:
4   - deployment.yaml
5   - service.yaml
6   - ingress.yaml
7   namespace: python-app-green
8   images:
9     - name: hello-gitops
10    newName: idgafd/hello-gitops
11    newTag: e1b472d533b6e2f9b89f3732fd4be8bb42edf67e
```

Main logs info:

Deploy succeeded 6 minutes ago in 1m 22s

Search logs

Create namespace & set in Kustomize 3s

```
1 ► Run namespacePresentBlue=kubectl get ns | grep python-app-blue | wc -l'
50 namespace/python-app-blue created
```

Update Kubernetes resources 1s

```
1 ► Run cd $GITHUB_WORKSPACE/github-actions-samples/k8s
24 service/hello-gitops created
25 Warning: autopilot-default-resources-mutator:Autopilot updated Deployment python-app-blue/hello-gitops: defaulted unspecified 'cpu' resource for containers [hello-gitops]
(see http://g.co/gke/autopilot-defaults).
26 deployment.apps/hello-gitops created
27 ingress.networking.k8s.io/hello-gitops created
```

Switch traffic to new namespace 0s

```
1 ► Run echo "Switching traffic to $NEW_NAMESPACE..."
30 Switching traffic to python-app-blue...
31 Updating service to use version v2.0.5...
32 service/hello-gitops patched
```

Remove old namespace 37s

```
1 ► Run echo "Removing old namespace: $OLD_NAMESPACE"
21 Removing old namespace: python-app-green
22 namespace "python-app-green" deleted
```

Update service.yaml with the new version 0s

```
1 ► Run echo "Fetching currently new version from service selector..."
25 Fetching currently new version from service selector...
26 Updating service.yaml with version v2.0.5...
27 Updated service.yaml:
28   apiVersion: v1
29   kind: Service
30   metadata:
31     labels:
32       app: hello-gitops
33       name: hello-gitops
34   spec:
35     ports:
36       - name: http
37         port: 8080
38         targetPort: http
39     selector:
40       app: hello-gitops
41       version: v2.0.5
42     type: ClusterIP
```

```

Commit updated service.yaml
 1 ► Run echo "Configuring Git for commit..." 
 2 Configuring Git for commit...
 3 Checking for changes in service.yaml...
 4 diff --git a/github-actions-samples/k8s/service.yaml b/github-actions-samples/k8s/service.yaml
 5 index 71b4a06..33df86b 100644
 6 --- a/github-actions-samples/k8s/service.yaml
 7 +++ b/github-actions-samples/k8s/service.yaml
 8 @@ -11,5 +11,5 @@ spec:
 9     selector:
10         app: hello-gitops
11     -   version: v2.0.0
12     +   version: v2.0.5
13         type: ClusterIP
14 Changes detected in service.yaml. Committing changes...
15 [main 738b72a] Update service.yaml to reflect version
16 1 file changed, 1 insertion(+), 1 deletion(-)

Push changes
 1 ► Run ad-m/github-push-action@master
 2 Push to branch main
 3 To https://github.com/**/UCU_DevOps_Course.git
 4 44802c9..738b72a HEAD -> main

```

New challenge

At that moment I was absolutely happy and thought everything worked out, and then I identified a problem with namespace overrides and incorrect version assignment 😅

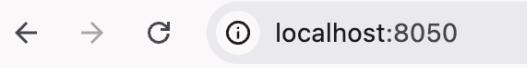
At the time of version 2.0.5, the only way to access the site was by changing the labels in the python-app-blue namespace to version 2.0.0, despite the fact that in the kustomisation.yaml and service.yaml files the versions and namespaces were different. Namespace hello-devops (initial) were running but not working with any versions.

```

anastasiiamazur@bttrm-amazur-pro16 ~ % kubectl get pods -n python-app-blue --show-labels
NAME        READY   STATUS    RESTARTS   AGE   LABELS
hello-gitops-54b54bf567-25fht  1/1   Running   0          33m   app=hello-gitops,pod-template-hash=54b54bf567
hello-gitops-54b54bf567-2xnf8  1/1   Running   0          33m   app=hello-gitops,pod-template-hash=54b54bf567
hello-gitops-54b54bf567-jwx88  1/1   Running   0          33m   app=hello-gitops,pod-template-hash=54b54bf567
hello-gitops-54b54bf567-qkzr   1/1   Running   0          33m   app=hello-gitops,pod-template-hash=54b54bf567
hello-gitops-54b54bf567-w99t7  1/1   Running   0          33m   app=hello-gitops,pod-template-hash=54b54bf567
anastasiiamazur@bttrm-amazur-pro16 ~ % kubectl describe svc hello-gitops -n python-app-blue
Name:           hello-gitops
Namespace:      python-app-blue
Labels:         app=hello-gitops
Annotations:   cloud.google.com/ingress.class: "nginx"
Selector:       app=hello-gitops,version=v2.0.0
Type:          ClusterIP
IP Family Policy: SingleStack
IP Families:   IPv4
IP:            34.118.230.64
IPs:           34.118.230.64
Port:          http  8050/TCP
TargetPort:    http/TCP
Endpoints:    
Session Affinity: None
Internal Traffic Policy: Cluster
Events:
  Type  Reason  Age   From      Message
  ----  ----   --   --        --
  Normal ADD     34m  sc-gateway-controller  python-app-blue/hello-gitops
  Normal DNSRecordProvisioningSucceeded 34m (x5 over 34m)  clouddns-controller  DNS records updated
anastasiiamazur@bttrm-amazur-pro16 ~ % kubectl label pods -l app=hello-gitops version=v2.0.0 --overwrite -n python-app-blue

pod/hello-gitops-54b54bf567-25fht labeled
pod/hello-gitops-54b54bf567-2xnf8 labeled
pod/hello-gitops-54b54bf567-jwx88 labeled
pod/hello-gitops-54b54bf567-qkzr labeled
pod/hello-gitops-54b54bf567-w99t7 labeled
anastasiiamazur@bttrm-amazur-pro16 ~ % kubectl port-forward svc/hello-gitops 8050:8050 -n python-app-blue
Forwarding from 127.0.0.1:8050 -> 8050
Forwarding from [::]:8050 -> 8050
Handling connection for 8050
^C
anastasiiamazur@bttrm-amazur-pro16 ~ % kubectl get pods -n python-app-blue --show-labels
NAME        READY   STATUS    RESTARTS   AGE   LABELS
hello-gitops-54b54bf567-25fht  1/1   Running   0          35m   app=hello-gitops,pod-template-hash=54b54bf567,version=v2.0.0
hello-gitops-54b54bf567-2xnf8  1/1   Running   0          35m   app=hello-gitops,pod-template-hash=54b54bf567,version=v2.0.0
hello-gitops-54b54bf567-jwx88  1/1   Running   0          35m   app=hello-gitops,pod-template-hash=54b54bf567,version=v2.0.0
hello-gitops-54b54bf567-qkzr   1/1   Running   0          35m   app=hello-gitops,pod-template-hash=54b54bf567,version=v2.0.0
hello-gitops-54b54bf567-w99t7  1/1   Running   0          35m   app=hello-gitops,pod-template-hash=54b54bf567,version=v2.0.0

```



Hello, Anastasiia!

UCU_DevOps_Course / github-actions-samples / k8s / service.yaml

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: hello-gitops
  name: hello-gitops
spec:
  ports:
  - name: http
    port: 8050
    targetPort: http
  selector:
    app: hello-gitops
    version: v2.0.5
  type: ClusterIP
```

```
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
resources:
- deployment.yaml
- service.yaml
- ingress.yaml
namespace: python-app-green
images:
- name: hello-gitops
  newName: idgafd/hello-gitops
  newTag: e1b472d533b6e2f9b89f3732fd4be8bb42edf67e
```

I needed to rewrite the approach to blue/green deployment without creating new namespaces, but with replacing the versions of the pods in the same namespace.

Result overview

Pods for blue and green branches are empty after deleting old namespaces.

```
anastasiiamazur@bttrm-amazur-pro16 ~ % kubectl get pods -n python-app-blue
kubectl get pods -n python-app-green
```

```
No resources found in python-app-blue namespace.
No resources found in python-app-green namespace.
anastasiiamazur@bttrm-amazur-pro16 ~ % kubectl get pods -n hello-gitops --show-labels
kubectl get svc -n hello-gitops
```

```
NAME          READY   STATUS    RESTARTS   AGE   LABELS
hello-gitops-774965b74-5qmm6  1/1    Running   0          2m34s  app=hello-gitops,pod-template-hash=774965b74
hello-gitops-774965b74-jncqr  0/1    Error     0          3m48s  app=hello-gitops,pod-template-hash=774965b74
hello-gitops-774965b74-lzspw  1/1    Running   0          3m48s  app=hello-gitops,pod-template-hash=774965b74
hello-gitops-774965b74-m24mq  1/1    Running   0          3m48s  app=hello-gitops,pod-template-hash=774965b74
hello-gitops-774965b74-phtrh  1/1    Running   0          3m48s  app=hello-gitops,pod-template-hash=774965b74
hello-gitops-774965b74-v99gz  1/1    Running   0          87s   app=hello-gitops,pod-template-hash=774965b74
NAME          TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
hello-gitops   ClusterIP  34.118.230.203  <none>           8050/TCP   5h37m
anastasiiamazur@bttrm-amazur-pro16 ~ % kubectl describe svc hello-gitops -n hello-gitops
```

Name:	hello-gitops
Namespace:	hello-gitops
Labels:	app=hello-gitops
Annotations:	cloud.google.com/neg: {"ingress":true}
Selector:	app=hello-gitops,version=v2.0.6
Type:	ClusterIP
IP Family Policy:	SingleStack
IP Families:	IPv4
IP:	34.118.230.203
IPs:	34.118.230.203
Port:	http 8050/TCP
TargetPort:	http/TCP
Endpoints:	
Session Affinity:	None
Internal Traffic Policy:	Cluster
Events:	

```
  Type  Reason  Age   From           Message
  ----  ----   --   --            --
Normal ADD     30m  sc-gateway-controller  hello-gitops/hello-gitops
anastasiiamazur@bttrm-amazur-pro16 ~ % kubectl get service hello-gitops -n hello-gitops -o=jsonpath='{.spec.selector.version}'
```

v2.0.6

Namespace did not changed, but version and newTag did

UCU_DevOps_Course / github-actions-samples / k8s / service.yaml

```
actions-user Update service.yaml to reflect version

Code Blame 15 lines (15 loc) · 236 Bytes Code 55% faster
1 apiVersion: v1
2 kind: Service
3 metadata:
4   labels:
5     app: hello-gitops
6     name: hello-gitops
7   spec:
8     ports:
9       - name: http
10      port: 8050
11      targetPort: http
12     selector:
13       app: hello-gitops
14       version: v2.0.6
15     type: ClusterIP
```

UCU_DevOps_Course / github-actions-samples / k8s / kustomization.yaml

```
idgafdf Update kustomization.yaml ×

Code Blame 11 lines (11 loc) · 258 Bytes Code 55% faster with Git
1 apiVersion: kustomize.config.k8s.io/v1beta1
2 kind: Kustomization
3 resources:
4   - deployment.yaml
5   - service.yaml
6   - ingress.yaml
7   namespace: hello-gitops
8   images:
9     - name: hello-gitops
10    newName: idgafdf/hello-gitops
11    newTag: elb472d533b6e2f9b89f3732fd4be8bb42edf67e
```

We need to apply the version label to access the web service

```
anastasiiamazur@bttrm-amazur-pro16 ~ % kubectl port-forward svc/hello-gitops 8050:8050 -n hello-gitops
```

```
^C
```

```
anastasiiamazur@bttrm-amazur-pro16 ~ % kubectl get pods -n hello-gitops --show-labels
```

NAME	READY	STATUS	RESTARTS	AGE	LABELS
hello-gitops-774965b74-5qmm6	1/1	Running	0	7m	app=hello-gitops,pod-template-hash=774965b74
hello-gitops-774965b74-1z spp	1/1	Running	0	8m14s	app=hello-gitops,pod-template-hash=774965b74
hello-gitops-774965b74-m24mq	1/1	Running	0	8m14s	app=hello-gitops,pod-template-hash=774965b74
hello-gitops-774965b74-phtrh	1/1	Running	0	8m14s	app=hello-gitops,pod-template-hash=774965b74
hello-gitops-774965b74-v99gz	1/1	Running	0	5m53s	app=hello-gitops,pod-template-hash=774965b74
anastasiiamazur@bttrm-amazur-pro16 ~ % kubectl label pods -l app=hello-gitops version=v2.0.6 --overwrite -n hello-gitops					
pod/hello-gitops-774965b74-5qmm6	labeled				
pod/hello-gitops-774965b74-1z spp	labeled				
pod/hello-gitops-774965b74-m24mq	labeled				
pod/hello-gitops-774965b74-phtrh	labeled				
pod/hello-gitops-774965b74-v99gz	labeled				

```
anastasiiamazur@bttrm-amazur-pro16 ~ % kubectl get pods -n hello-gitops --show-labels
NAME          READY   STATUS    RESTARTS   AGE   LABELS
hello-gitops-774965b74-5qmm6  1/1    Running   0          7m24s  app=hello-gitops,pod-template-hash=774965b74,version=v2.0.6
hello-gitops-774965b74-1z spp  1/1    Running   0          8m38s  app=hello-gitops,pod-template-hash=774965b74,version=v2.0.6
hello-gitops-774965b74-m24mq  1/1    Running   0          8m38s  app=hello-gitops,pod-template-hash=774965b74,version=v2.0.6
hello-gitops-774965b74-phtrh  1/1    Running   0          8m38s  app=hello-gitops,pod-template-hash=774965b74,version=v2.0.6
hello-gitops-774965b74-v99gz  1/1    Running   0          6m17s  app=hello-gitops,pod-template-hash=774965b74,version=v2.0.6
anastasiiamazur@bttrm-amazur-pro16 ~ % kubectl port-forward svc/hello-gitops 8050:8050 -n hello-gitops
```

```
Forwarding from 127.0.0.1:8050 -> 8050
Forwarding from [::1]:8050 -> 8050
Handling connection for 8050
Handling connection for 8050
Handling connection for 8050
Handling connection for 8050
```

Now I can access the web service via localhost to IP address (Port Forwarding)

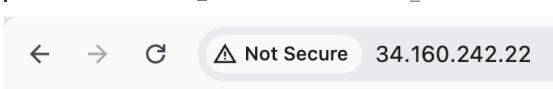


Hello, Anastasiia!

Or via Ingress ta Load Balancer

```
anastasiiamazur@bttrm-amazur-pro16 ~ % kubectl apply -f github-actions-samples/k8s/ingress.yaml -n hello-gitops
ingress.networking.k8s.io/hello-gitops unchanged
anastasiiamazur@bttrm-amazur-pro16 ~ % curl http://34.160.242.22
```

Hello, Anastasiia!



Hello, Anastasiia!

The classic Blue/Green approach uses two namespaces or two separate deployments for different versions, and the service switches traffic between them. My approach:

- Pros:
 - simple (not for me but overall 😊) and does not require extra resources
 - reduces complexity by using a single namespace
- Cons:
 - makes it difficult to isolate the new version for preliminary testing
 - removing the old version immediately after switching traffic can create problems in case of errors.

What can be done better

1. Simplify the strategy by updating service labels and selectors dynamically within the same namespace. Alternatively, maintain both blue and green namespaces to enable more traditional blue/green deployment benefits such as easier rollbacks, isolated testing, and high availability during transitions
2. Improve version increment logic by automating it in the CI/CD pipeline. This eliminates the need for hardcoding and ensures consistent versioning
3. Integrate tools like Prometheus and Grafana to enable centralized logging and monitoring
4. Clean up the ci-cd-python-app.yml code from unnecessary commented versions of events (but I would like to keep them at least for myself to come back and do a bug review 👍)

k8s resources definitions

```
anastasiamazur@bttrm-amazur-pro16 ~ % kubectl get pods -n hello-gitops --show-labels
NAME                  READY   STATUS    RESTARTS   AGE     LABELS
hello-gitops-774965b74-5qmm6  1/1    Running   0          31m    app=hello-gitops,pod-template-hash=774965b74,version=v2.0.6
hello-gitops-774965b74-1zsp  1/1    Running   0          33m    app=hello-gitops,pod-template-hash=774965b74,version=v2.0.6
hello-gitops-774965b74-m24mq  1/1    Running   0          33m    app=hello-gitops,pod-template-hash=774965b74,version=v2.0.6
hello-gitops-774965b74-phtrh  1/1    Running   0          33m    app=hello-gitops,pod-template-hash=774965b74,version=v2.0.6
hello-gitops-774965b74-v99gz  1/1    Running   0          30m    app=hello-gitops,pod-template-hash=774965b74,version=v2.0.6
anastasiamazur@bttrm-amazur-pro16 ~ % kubectl get svc -n hello-gitops
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
hello-gitops  ClusterIP  34.118.230.203  <none>        8050/TCP  6h6m
```

Github repo link: https://github.com/idgaf/UCU_DevOps_Course

- github actions workflow code is in the root directory .github/workflows
- app code is in folder github-actions-samples (where the initial workflows from forked repository are present, don't pay attention)

Task 4

Let's try to set up the last workflow for ArgoCD deployment 🎉

The screenshot shows a list of 150 workflow runs. Three specific runs are highlighted for the 'Update ci-cd-python-app.yml' workflow. Each run is associated with a green checkmark icon, indicating success. The runs occurred at different times: 4 minutes ago, 4 minutes ago, and 4 minutes ago. The status for all runs is 'Success'. The branch for each run is 'main'. The actor for each run is 'idgafid'. A search bar at the top right says 'Filter workflow runs'.

Run #	Event	Status	Branch	Actor
1	4 minutes ago	Success	main	idgafid
2	4 minutes ago	Success	main	idgafid
3	4 minutes ago	Success	main	idgafid

It was a bit easier... about 10 hours easier, to be honest 😊

Problem Overview

During the Blue/Green deployment process, Git conflicts arose in the CI/CD pipeline when attempting to push changes to the remote `main` branch. The conflicts occurred because the remote branch contained changes that were not present in the pipeline's local copy of the repository. This situation is common when multiple workflows or users modify the same branch simultaneously, leading to rejected pushes due to out-of-sync branches.

I introduced the following steps to handle potential local untracked or unstaged changes and ensure a clean pull from the remote branch in `ci-argocd-python-app.yml`:

1. Stashed local changes to avoid overwriting them during the pull operation
2. Pulled and rebased local changes onto the local branch, ensuring a linear Git history
3. Reapplied the stashed changes after the pull operation (if there were no stashed changes, the pipeline continues without interruption)

If changes were made, they are staged and committed. Otherwise, the step is skipped.

The updated branch is pushed to the remote repository.

```
- name: Stash any local changes
  run: |
    git stash --include-untracked

- name: Pull changes from remote
  run: |
    git config --local user.email "action@github.com"
    git config --local user.name "GitHub Action"
    git pull --rebase origin main

- name: Apply stashed changes (if any)
  run: |
    git stash pop || echo "No stashed changes to apply"

- name: Commit changes if any
  run: |
    git config --local user.email "action@github.com"
    git config --local user.name "GitHub Action"
    if git diff --quiet; then
      echo "No changes to commit."
    else
      git add .
      git commit -m "Bump docker tag"
    fi

- name: Push changes
  uses: ad-m/github-push-action@master
  with:
    github_token: ${{ secrets.GITHUB_TOKEN }}
```

As an alternative, a streamlined approach was implemented in `ci-cd-python-app.yml` to fetch and rebase the local branch against the remote:

```
- name: Pull latest changes
  run: |
    git config --local user.email "action@github.com"
    git config --local user.name "GitHub Action"
    git fetch origin
    git rebase origin/main || echo "Rebase not required"

- name: Push changes
  uses: ad-m/github-push-action@master
  with:
    github_token: ${{ secrets.GITHUB_TOKEN }}
```

This ensures the local branch is up-to-date with the remote without creating unnecessary merge commits:

```
git fetch origin
git rebase origin/main || echo "Rebase not required"
```

The updated branch is pushed after the rebase.

Both approaches successfully resolved the conflicts by ensuring that the local branch in the CI/CD pipeline was fully synchronized with the remote branch before pushing changes.

Github repo link: https://github.com/idgafd/UCU_DevOps_Course

- github actions workflow code is in the root directory `.github/workflows`
- app code is in folder `github-actions-samples` (where the initial workflows from forked repository are present, don't pay attention)