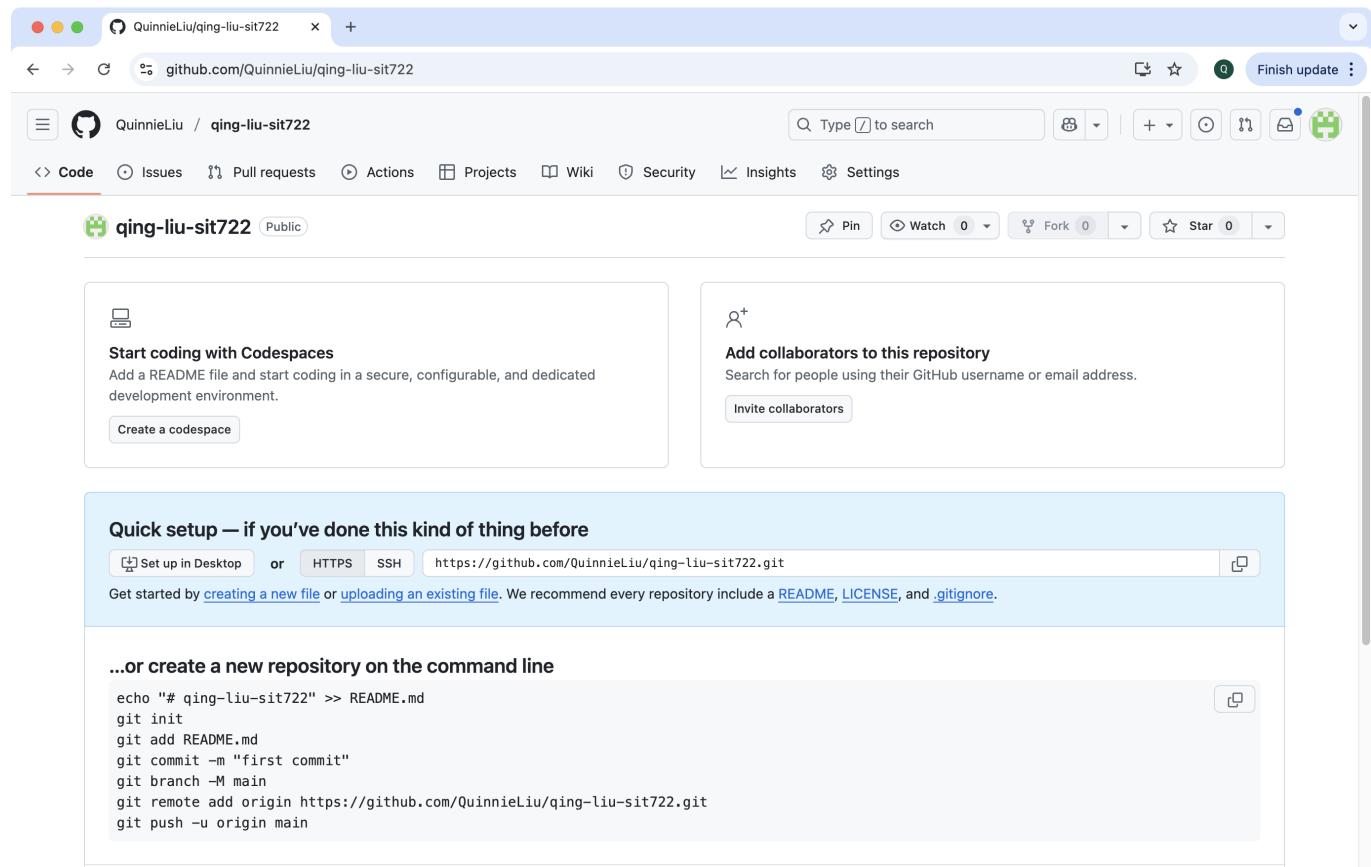


Detailed documentation

1 Create new Github repository



2 Create a local project folder

Code

qing-liu-sit722 Public

main 1 Branch 0 Tags

Go to file Add file Code

QuinnieLiu first commit 16960cc · now 1 Commit

README.md first commit now

README

qing-liu-sit722

```
git remote add origin https://github.com/QuinnieLiu/qing-liu-sit722.git
git push -u origin main
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
hint:
hint: Disable this message with "git config set advice.defaultBranchName false"
Initialized empty Git repository in /Users/qingliu/Documents/Trimester/2025_T2/722_software_deployment_and_operation/demo/qing-liu-sit722/.git/
[master (root-commit) 16960cc] first commit
  1 file changed, 1 insertion(+)
  create mode 100644 README.md
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 235 bytes | 235.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/QuinnieLiu/qing-liu-sit722.git
 * [new branch]  main -> main
branch 'main' set up to track 'origin/main'.
qingliu@allenbook qing-liu-sit722 %
```

© 2025 GitHub, Inc.

3 Push task resources to remote repo

Code

qing-liu-sit722 Public

main 1 Branch 0 Tags

Go to file Add file Code

QuinnieLiu Code from Task Resources d680022 · now 2 Commits

backend Code from Task Resources now

frontend Code from Task Resources now

.gitignore Code from Task Resources now

README.md first commit 6 minutes ago

docker-compose.yml Code from Task Resources now

README

qing-liu-sit722

About

Task10.2D

Readme

Activity

0 stars

0 watching

0 forks

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

my personal information

3.1 Create necessary Azure resources

```
RESOURCE_GROUP="rg-sit722-taskd"
LOCATION="australiaeast"
ACR_NAME="acrsit722taskd"
AKS_NAME="aks-sit722-taskd"
SERVICE_PRINCIPAL_NAME="sp-sit722-cicd"

az group create --name $RESOURCE_GROUP --location $LOCATION

az acr create --resource-group $RESOURCE_GROUP --name $ACR_NAME --sku
Basic --admin-enabled true

az aks create \
--resource-group $RESOURCE_GROUP \
--name $AKS_NAME \
--node-count 2 \
--node-vm-size Standard_B2s \
--enable-addons monitoring \
--generate-ssh-keys \
--attach-acr $ACR_NAME

az aks get-credentials --resource-group $RESOURCE_GROUP --name $AKS_NAME

az ad sp create-for-rbac \
--name $SERVICE_PRINCIPAL_NAME \
--role contributor \
--scopes /subscriptions/$(az account show --query id --output
tsv)/resourceGroups/$RESOURCE_GROUP \
--sdk-auth

az acr show --name $ACR_NAME --query loginServer --output tsv

az acr credential show --name $ACR_NAME
```

3.2 Add Github Actions

CI testing: <https://github.com/QuinnieLiu/qing-liu-sit722/blob/main/.github/workflows/ci-testing-branch.yaml>

CD Staging: <https://github.com/QuinnieLiu/qing-liu-sit722/blob/main/.github/workflows/cd-staging-deployment.yaml>

CD Production: <https://github.com/QuinnieLiu/qing-liu-sit722/blob/main/.github/workflows/cd-production-deployment.yaml>

3.3 Other code changes

Github Repo: <https://github.com/QuinnieLiu/qing-liu-sit722/tree/main>

k8s: <https://github.com/QuinnieLiu/qing-liu-sit722/tree/main/k8s>

4 CI Pipeline run on testing branch commits

4.1 Add a commit to testing branch

The screenshot shows a Mac OS X desktop environment. In the foreground, a terminal window is open with the command `git status`, displaying the current branch as `testing` and a list of staged changes. In the background, a Visual Studio Code window is open, showing the file `main.py`. The code implements a FastAPI application with RabbitMQ integration. A GitHub pull request is visible in the sidebar, showing a commit message: "Message (%e to commit on "testing")". The pull request has been merged into the `main/main` branch.

```
main.py
backend > product_service > app > main.py

101
102     # Global RabbitMQ connection and channel objects
103     rabbitmq_connection: Optional[aio_pika.Connection] = None
104     rabbitmq_channel: Optional[aio_pika.Channel] = None
105     rabbitmq_exchange: Optional[aio_pika.Exchange] = None
106
107     # --- FastAPI Application Setup ---
108     app = FastAPI(
109         title="Product Service API",
110         description="Manages products and stock for mini-eCommerce app, with Azure Storage integration.",
111         version="1.0.5",
112     )
113
114     # Enable CORS (for frontend dev/testing)
115     app.add_middleware(
116         CORSMiddleware,
117         allow_origins=["*"], # Use specific origins in production
118         allow_credentials=True,
119         allow_methods=["*"],
120         allow_headers=["*"],
121     )
122
123
124     # --- RabbitMQ Helper Functions ---
125     async def connect_to_rabbitmq():
126         """Establishes an asynchronous connection to RabbitMQ."""
127         global rabbitmq_connection, rabbitmq_channel, rabbitmq_exchange
128
129         rabbitmq_url = (
130             f"amqp://:{RABBITMQ_USER}:{RABBITMQ_PASS}@{RABBITMQ_HOST}:{RABBITMQ_PORT}/"
131         )
132         max_retries = 10
133         retry_delay_seconds = 5
134
135         for i in range(max_retries):
136             try:
137                 logger.info(
138                     f"Product Service: Attempting to connect to RabbitMQ (attempt {i+1}/{max_retries})..."
139                 )
140                 rabbitmq_connection = await aio_pika.connect_robust(rabbitmq_url)
141                 rabbitmq_channel = await rabbitmq_connection.channel()
142                 # Declare a direct exchange for events
143                 rabbitmq_exchange = await rabbitmq_channel.declare_exchange(
```

4.2 CI Pipeline triggered

The screenshot shows a GitHub Actions run details page for a workflow named "Increase product service version to trigger the pipeline #17". The workflow was triggered via push now by user QuinnyeLiu. The status is "In progress". The total duration is listed as "—" and there are no artifacts. The workflow file is "ci-testing-branch.yaml" with the "on: push" trigger. The run details show two steps: "Run Tests" (status: "In progress", duration: 13s) and "Build and Push Images to ACR" (status: "Not Started").

4.3 CI Pipeline complete

Increase product service version to trigger the pipeline #17

Triggered via push 5 minutes ago
Status: Success
Total duration: 4m 50s
Artifacts: -

ci-testing-branch.yaml
on: push

```

graph LR
    A[Run Tests] --> B[Build and Push Images...]
    
```

Re-run all jobs ⋮

5 CD Staging Pipeline run on completion of CI Pipeline

5.1 CD Staging Pipeline triggered

CD Pipeline - Staging Deployment #15

Triggered via workflow run 1 minute ago
Status: In progress
Total duration: -
Artifacts: -

cd-staging-deployment.yaml
on: workflow_run

```

graph LR
    A[Deploy Backend Service...] --> B[Deploy Frontend to Staging]
    
```

Cancel workflow ⋮

5.2 Staging environment created

Deploy Frontend to Staging
Started 2m ago

- > ✓ Checkout code 1s
- > ✓ Set staging environment name 0s
- > ✓ Azure Login 2s
- > ✓ Get AKS credentials 2s
- > ✓ Deploy Frontend 7s
- > ✓ Wait for frontend deployment 8s
- > ✓ Get staging environment info 4s
- > ✓ Run acceptance tests 25s
- > ○ Manual testing window 35s
 - 1 ► Run echo "🌐 Staging environment is ready for manual testing!"
 - 17 🌐 Staging environment is ready for manual testing!
 - 18 Namespace: staging-be1c9136
 - 19 You have 30 minutes to perform manual testing.
 - 20 The environment will be automatically cleaned up after this time.

aks-sit722-taskd | Workloads

Pod	Namespace	Status	Replicas	Last Seen	CPU %	Memory %
coredns-autoscaler	kube-system	Green	1/1	15 hours	0%	0%
konnectivity-agent	kube-system	Green	2/2	15 hours	0%	0%
konnectivity-agent-autoscaler	kube-system	Green	1/1	15 hours	0%	0%
metrics-server	kube-system	Green	2/2	15 hours	0%	0%
ama-logs-rs	kube-system	Green	1/1	15 hours	0%	0%
postgresql	production	Green	1/1	25 minutes	0%	0%
rabbitmq	production	Green	1/1	25 minutes	0%	0%
customer-service	production	Green	1/1	24 minutes	0%	0%
product-service	production	Green	1/1	24 minutes	0%	0%
order-service	production	Green	1/1	24 minutes	0%	0%
frontend	production	Green	1/1	22 minutes	0%	0%
postgresql	staging-be1c9136	Green	1/1	5 minutes	0%	0%
rabbitmq	staging-be1c9136	Green	1/1	5 minutes	0%	0%
customer-service	staging-be1c9136	Green	1/1	4 minutes	0%	0%
product-service	staging-be1c9136	Green	1/1	4 minutes	0%	0%
order-service	staging-be1c9136	Green	1/1	4 minutes	0%	0%
frontend	staging-be1c9136	Green	1/1	2 minutes	0%	0%

5.3 Visit Frontend in staging environment

The screenshot shows a browser window with three tabs: "CD Pipeline - Staging Deploy", "aks-sit722-taskd - Microsoft", and "E-commerce Application". The "E-commerce Application" tab is active, displaying the "Customer Management" section with fields for Email, Password, First Name, Last Name, Phone Number, and Shipping Address. To the right, the developer tools Network tab is open, showing a list of requests. One request is selected: "products/" with a status of 200 OK. The response headers include Access-Control-Allow-Credentials: true, Access-Control-Allow-Origin: *, Content-Length: 2, Content-Type: application/json, Date: Fri, 26 Sep 2025 02:12:09 GMT, and Server: uvicorn.

The staging deployment will be cleaned up in 30 minutes automatically or as soon as the CD pipeline is cancelled manually

6 Production deployment on closed pull request

6.1 Raise PR and merge

The screenshot shows a GitHub pull request page for a repository named "QuinnieLiu / qing-liu-sit722". The pull request is titled "Increase product service version to trigger the pipeline #9" and has been merged. The commit message is "Increase product service version to trigger the pipeline". The pull request has 1 commit, 2 checks, and 1 file changed. The merge commit message is "QuinnieLiu merged commit addf802 into main now". The pull request has been successfully merged and closed. A note says "You're all set — the testing branch can be safely deleted." Below the pull request, there is a comment input field with placeholder text "Add your comment here..." and a rich text editor toolbar.

6.2 CD Production Pipeline triggered on PR

The screenshot shows the GitHub Actions interface for a workflow named "Increase product service version to trigger the pipeline #6". The workflow is triggered via a pull request and is currently in progress. The status bar indicates "In progress". The workflow file is "cd-production-deployment.yaml" and it triggers on "pull_request". The workflow consists of three steps: "Deploy Backend Services" (49s), "Deploy Frontend to Production", and "Rollback on Deployment Failure".

6.3 CD Production Pipeline complete

The screenshot shows the GitHub Actions interface for the same workflow. The status bar now shows "Success" and the total duration is "2m 32s". All three steps have completed successfully: "Deploy Backend Services" (1m 5s), "Deploy Frontend to Production" (1m 21s), and "Rollback on Deployment Failure" (0s). The "cd-production-deployment.yaml" file is triggered on "pull_request".

6.3 Production environment created

aks-sit722-taskd | Workloads

Deployments

Name	Namespace	Ready	Age	CPU	Memory
coredns	kube-system	2/2	16 hours	Enable metrics	
coredns-autoscaler	kube-system	1/1	16 hours		
konnectivity-agent	kube-system	2/2	16 hours		
konnectivity-agent-autoscaler	kube-system	1/1	16 hours		
metrics-server	kube-system	2/2	16 hours		
ama-logs-rs	kube-system	1/1	16 hours		
postgresql	production	1/1	30 minutes		
rabbitmq	production	1/1	30 minutes		
customer-service	production	1/1	29 minutes		
product-service	production	1/1	29 minutes		
order-service	production	1/1	29 minutes		
frontend	production	1/1	27 minutes		

aks-sit722-taskd | Services and ingresses

Services

Name	Namespace	Status	Type	Cluster IP	External IP	Ports	Age
kubernetes	default	Ok	ClusterIP	10.0.0.1		443/TCP	16 hours
kube-dns	kube-system	Ok	ClusterIP	10.0.0.10		53/UDP,53/TCP	16 hours
metrics-server	kube-system	Ok	ClusterIP	10.0.217.111		443/TCP	16 hours
postgresql	production	Ok	ClusterIP	10.0.192.241		5432/TCP	31 minutes
rabbitmq	production	Ok	ClusterIP	10.0.33.50		5672/TCP,15672...	31 minutes
customer-service	production	Ok	LoadBalancer	10.0.168.170	20.167.18.41	8000:32327/TCP	30 minutes
product-service	production	Ok	LoadBalancer	10.0.183.56	4.200.111.148	8001:31318/TCP	30 minutes
order-service	production	Ok	LoadBalancer	10.0.189.63	4.254.121.134	8002:31761/TCP	30 minutes
frontend	production	Ok	LoadBalancer	10.0.133.139	4.237.179.230	80:32510/TCP	28 minutes

6.4 Visit Frontend in production environment

The screenshot shows a browser window with multiple tabs. The active tab is titled "E-commerce Application". The main content area displays the "Customer Management" page, which includes fields for "Email", "Password", "First Name", "Last Name", "Phone Number", and "Shipping Address". Above the form, there is a message: "Product Microservice, Order Microservice, Customer Microservice, RabbitMQ for Async Events, Persistent Storage & Azure Uploads." To the right of the main content, the browser's developer tools Network tab is open, showing network requests. The timeline at the top indicates a total duration of 200 ms, divided into 50 ms, 100 ms, 150 ms, and 200 ms segments. The table below lists five requests:

Name	Status	Type	Initiator	Size	Time
4.237.179.230	200	docu...	Other	14.9 ...	57 ms
main.js	200	script	(index):7	25.3...	58 ms
products/	200	fetch	main.js:47	0.2 kB	58 ms
customers/	200	fetch	main.js:249	0.2 kB	65 ms
orders/	200	fetch	main.js:411	0.2 kB	65 ms

At the bottom of the Network tab, it says "5 requests | 40.8 kB transferred | 39.7 kB resources | Finish: 168 ms".