

Project: Operationalizing a Co-working Space Service

April 1 2024

Dahi Nemutlu



Contents

Introduction	3
Create an EKS Cluster	3
Configure a Database for the Service	5
Set up Continuous Integration with AWS ECR & CodeBuild.....	9
Deploy the Application	14
Load Test and Monitoring.....	17

Introduction

The Coworking Space Service is a set of APIs that enables users to request one-time tokens and administrators to authorize access to a coworking space. This service follows a microservice pattern and the APIs are split into distinct services that can be deployed and managed independently of one another.

This project aims to build a pipeline to deploy an API in Kubernetes that provides business analysts basic analytics data on user activity in the service.

Create an EKS Cluster

Create an Amazon EKS cluster by using the AWS Management Console:

EKS > Clusters > demo-eks-cluster

demo-eks-cluster

Delete cluster

New versions are available for 3 add-ons.

▼ Cluster info

Info

Status

Active

Kubernetes version

1.29

Info

Support type

Standard support until March 23, 2025

Provider

EKS

<

Overview

Resources

Compute

Networking

Add-ons

Access

Observabil

>

Details

API server endpoint

https://8F080ED78689B3CC0390B1F3A0B09BFA.sk1.us-west-1.eks.amazonaws.com

Certificate authority

LS0tLS1CRUdJTjBDRVJU
SUZJQ0FUR50tLS0tCK1J
SURCVENDQWUyZ0F3S

OpenID Connect provider URL

https://oidc.eks.us-west-1.amazonaws.com/id/8F080ED78689B3CC0390B1F3A0B09BFA

Cluster IAM role ARN

arn:aws:iam::265681628035:role/eksClusterRole

Created

March 31, 2024, 11:08 (UTC+03:00)

Cluster ARN

arn:aws:eks:us-west-1:265681628035:cluster/demo-eks-cluster

Platform version

Info




eks.5

3


Create a managed node group:

EKS > Clusters > demo-eks-cluster > Node groups > demo-node-group

demo-node-group








  

Node group configuration [Info](#)

Kubernetes version 1.29	AMI type Info AL2_x86_64	Status  Active
AMI release version Info 1.29.0-20240315	Instance types t3.medium	Disk size 20 GiB

< **Details** | Nodes | Health issues 0 | Kubernetes labels | Update config | Kubernetes >

Details

Node group ARN  arn:aws:eks:us-west-1:265681628035:nodegroup/demo-eks-cluster/demo-node-group/b6c74caf-515f-36cb-ba06-3dd0569e20bb	Autoscaling group name eks-demo-node-group-b6c74caf-515f-36cb-ba06-3dd0569e20bb 	Capacity type On-Demand	Subnets subnet-0eeee28a7bc52aa6a  subnet-0474e6f6e987dd5b4 
Created  6 hours ago	Node IAM role ARN  arn:aws:iam::265681628035:role/AmazonEKSNodeRole 	Desired size 2 nodes	Configure remote access to nodes off
		Minimum size 2 nodes	
		Maximum size 2 nodes	

Map the IAM entity/user to the aws-auth ConfigMap to allow access to the cluster:

```
[cloudshell-user@ip-10-2-39-106 ~]$ eksctl create iamidentitymapping \
> --cluster demo-eks-cluster \
> --region us-west-1 \
> --arn arn:aws:iam::265681628035:user/dahi \
> --group system:masters \
> --no-duplicate-arns \
> --username dahi
2024-03-29 10:27:31 [i] checking arn arn:aws:iam::265681628035:user/dahi against entries
in the auth ConfigMap
2024-03-29 10:27:31 [i] adding identity "arn:aws:iam::265681628035:user/dahi" to auth
ConfigMap
```

Configure a Database for the Service

Set up a Postgres database using a Helm Chart:

```
dahi@DESKTOP-ATVVKNF:~$ helm repo add bitnami https://charts.bitnami.com/bitnami
"bitnami" has been added to your repositories
dahi@DESKTOP-ATVVKNF:~$ helm install my-release bitnami/postgresql
NAME: my-release
LAST DEPLOYED: Sun Mar 31 11:23:28 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
CHART NAME: postgresql
CHART VERSION: 15.1.4
APP VERSION: 16.2.0

** Please be patient while the chart is being deployed **

PostgreSQL can be accessed via port 5432 on the following DNS names from within your cluster:

    my-release-postgresql.default.svc.cluster.local - Read/Write connection

To get the password for "postgres" run:

    export POSTGRES_PASSWORD=$(kubectl get secret --namespace default my-release-postgresql -o jsonpath="{.data.postgres-password}" | base64 -d)

To connect to your database run the following command:

    kubectl run my-release-postgresql-client --rm --tty -i --restart='Never' --namespace default --image docker.io/bitnami/postgresql:16.2.0-debian-12-r10 --env="PGPASSWORD=$POSTGRES_PASSWORD" \
        --command -- psql --host my-release-postgresql -U postgres -d postgres -p 5432

    > NOTE: If you access the container using bash, make sure that you execute
    "/opt/bitnami/scripts/postgresql/entrypoint.sh /bin/bash" in order to avoid the error
    "psql: local user with ID 1001} does not exist"

To connect to your database from outside the cluster execute the following commands:

    kubectl port-forward --namespace default svc/my-release-postgresql 5432:5432 &
    PGPASSWORD="$POSTGRES_PASSWORD" psql --host 127.0.0.1 -U postgres -d postgres -p 5432

WARNING: The configured password will be ignored on new installation in case when previous PostgreSQL release was deleted through the helm command. In that case, old PVC will have an old password, and setting it through helm won't take effect. Deleting persistent volumes (PVs) will solve the issue.
```

```
WARNING: There are "resources" sections in the chart not set. Using "resourcesPreset" is
not recommended for production. For production installations, please set the following
values according to your workload needs:
- primary.resources
- readReplicas.resources
+info https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
```

Note: If the PersistentVolumeClaim for Postgres is stuck in 'Pending' status with the below event message after the deployment:

```
dahi@DESKTOP-ATVVKNF:~$ kubectl describe pvc data-my-release-postgresql-0
Name:          data-my-release-postgresql-0
Namespace:     default
StorageClass:  gp2
Status:        Pending
.
.
.
Events:
  Type      Reason             Message
  ----      -
Normal     WaitForFirstConsumer 11m          persistentvolume-controller
              waiting for first consumer to be created before binding
Normal     ExternalProvisioning 103s (x43 over 11m) persistentvolume-controller
              Waiting for a volume to be created either by the external
provisioner 'ebs.csi.aws.com' or manually by the system administrator. If volume creation
is delayed, please verify that the provisioner is running and correctly registered.
```

[These steps](#) can be followed to solve the issue.

Test Database Connection:

```
dahi@DESKTOP-ATVVKNF:~$ export POSTGRES_PASSWORD=$(kubectl get secret --namespace default
my-release-postgresql -o jsonpath="{.data.postgres-password}" | base64 -d)
dahi@DESKTOP-ATVVKNF:~$ echo $POSTGRES_PASSWORD
I2nCIqCbei
dahi@DESKTOP-ATVVKNF:~$ kubectl exec -it my-release-postgresql-0 -- bash
I have no name!@my-release-postgresql-0:/ $ psql -U postgres -d postgres
Password for user postgres:
psql (16.2)
Type "help" for help.

postgres=# \l

              List of databases
  Name      | Owner   | Encoding | Locale Provider | Collate | Ctype   | ICU Locale | ICU Rules | Access privileges
-----+-----+-----+-----+-----+-----+-----+-----+-----+
 postgres   | postgres | UTF8     | libc             | en_US.UTF-8 | en_US.UTF-8 |             |           |
 template0  | postgres | UTF8     | libc             | en_US.UTF-8 | en_US.UTF-8 |             |           | =c/postgres      +
            |          |          |                  |          |          |             |           | postgres=Ctc/postgres
 template1  | postgres | UTF8     | libc             | en_US.UTF-8 | en_US.UTF-8 |             |           | =c/postgres      +
            |          |          |                  |          |          |             |           | postgres=Ctc/postgres
(3 rows)
```

Check Postgres service:

```
dahi@DESKTOP-ATVVKNF:~$ kubectl get svc
NAME                                TYPE                CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
kubernetes                         ClusterIP           10.100.0.1    <none>         443/TCP    54m
my-release-postgresql              ClusterIP           10.100.69.0   <none>         5432/TCP    44m
my-release-postgresql-hl           ClusterIP           None          <none>         5432/TCP    44m

dahi@DESKTOP-ATVVKNF:~$ kubectl describe svc my-release-postgresql
Name:                                my-release-postgresql
Namespace:                          default
Labels:                             app.kubernetes.io/component=primary
                                     app.kubernetes.io/instance=my-release
                                     app.kubernetes.io/managed-by=Helm
                                     app.kubernetes.io/name=postgresql
                                     app.kubernetes.io/version=16.2.0
                                     helm.sh/chart=postgresql-15.1.4
Annotations:                         meta.helm.sh/release-name: my-release
                                     meta.helm.sh/release-namespace: default
Selector:                           app.kubernetes.io/component=primary,app.kubernetes.io/instance=my-
release,app.kubernetes.io/name=postgresql
Type:                                ClusterIP
IP Family Policy:                    SingleStack
IP Families:                         IPv4
IP:                                  10.100.69.0
IPs:                                 10.100.69.0
Port:                                tcp-postgresql 5432/TCP
TargetPort:                          tcp-postgresql/TCP
Endpoints:                           172.31.16.219:5432
Session Affinity:                    None
Events:                              <none>
```

Connecting Via Port Forwarding:

```
dahi@DESKTOP-ATVVKNF:~$ kubectl port-forward service/my-release-postgresql 5433:5432 &
[1] 37408
dahi@DESKTOP-ATVVKNF:~$ Forwarding from 127.0.0.1:5433 -> 5432
Forwarding from [::1]:5433 -> 5432
```

Run Seed Files:

```
dahi@DESKTOP-ATVVKNF:~/microservices-aws-k8s-project/db$ kubectl get pods
NAME                                READY    STATUS    RESTARTS    AGE
my-release-postgresql-0             1/1      Running   0            121m

dahi@DESKTOP-ATVVKNF:~/microservices-aws-k8s-project/db$ kubectl cp 1_create_tables.sql
my-release-postgresql-0:/tmp
dahi@DESKTOP-ATVVKNF:~/microservices-aws-k8s-project/db$ kubectl cp 2_seed_users.sql my-
release-postgresql-0:/tmp
dahi@DESKTOP-ATVVKNF:~/microservices-aws-k8s-project/db$ kubectl cp 3_seed_tokens.sql my-
release-postgresql-0:/tmp
dahi@DESKTOP-ATVVKNF:~/microservices-aws-k8s-project/db$ kubectl exec -it my-release-
postgresql-0 -- bash
I have no name!@my-release-postgresql-0:/$ cd /tmp
```

```
I have no name!@my-release-postgresql-0:/tmp$ psql -U postgres -d postgres -f
1_create_tables.sql
Password for user postgres:
CREATE TABLE
CREATE INDEX
CREATE TABLE
CREATE INDEX
I have no name!@my-release-postgresql-0:/tmp$ psql -U postgres -d postgres -f
2_seed_users.sql
Password for user postgres:
INSERT 0 1
INSERT 0 1
INSERT 0 1
.
.
.
I have no name!@my-release-postgresql-0:/tmp$ psql -U postgres -d postgres -f
3_seed_tokens.sql
Password for user postgres:
INSERT 0 1
INSERT 0 1
INSERT 0 1
.
.
.
```

Checking the tables:

```
dahi@DESKTOP-ATVVKNF:~$ psql --host 127.0.0.1 -U postgres -d postgres -p 5433
Password for user postgres:
psql (14.11 (Ubuntu 14.11-0ubuntu0.22.04.1), server 16.2)
WARNING: psql major version 14, server major version 16.
         Some psql features might not work.
Type "help" for help.

postgres=# select * from users;
 id | first_name | last_name | joined_at | is_active
-----+-----+-----+-----+-----
  1 | Cristopher | Alexander | 2023-01-20 03:23:39.757813 | t
  2 | Dakota     | Gardner  | 2023-02-02 16:23:39.75783  | t
  3 | Mattie     | Moyer    | 2023-01-31 10:23:39.757836 | t
  4 | Darien     | Gamble   | 2023-02-13 05:23:39.75784  | t
  5 | Luca       | Abbott   | 2023-02-11 22:23:39.757844 | t

postgres=# select * from tokens;
 id | user_id | token | created_at | used_at
-----+-----+-----+-----+-----
  1 |    500 | 902912 | 2023-02-10 13:32:58.224409 | 
  2 |    173 | 940181 | 2023-02-08 08:37:58.224425 | 2023-02-08 08:29:58.224425
  3 |    167 | 542454 | 2023-02-11 13:08:58.224435 | 2023-02-11 13:02:58.224435
  4 |    332 | 478128 | 2023-02-09 16:20:58.224443 | 
  5 |    382 | 356931 | 2023-02-09 21:11:58.224448 | 
```

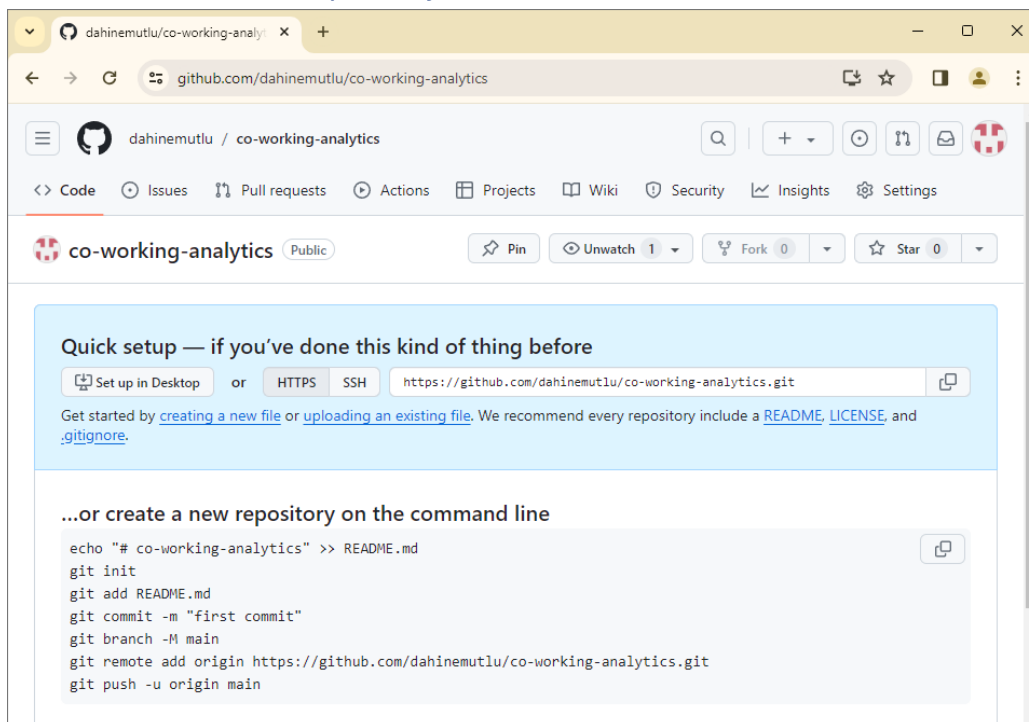

Running the application locally:

```
dahi@DESKTOP-ATVVKNF:~/microservices-aws-k8s-project/analytics/app$ export
DB_USERNAME=postgres
export POSTGRES_PASSWORD=I2nCIqCbei
export DB_PASSWORD=${POSTGRES_PASSWORD}
export DB_HOST=127.0.0.1
export DB_PORT=5433
export DB_NAME=postgres
dahi@DESKTOP-ATVVKNF:~/microservices-aws-k8s-project/analytics/app$ python3 app.py
* Serving Flask app 'config'
* 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://127.0.0.1:5153
* Running on http://172.19.161.153:5153
Press CTRL+C to quit

dahi@DESKTOP-ATVVKNF:~$ curl http://127.0.0.1:5153/api/reports/daily_usage
{"2023-02-07":40,"2023-02-08":202,"2023-02-09":179,"2023-02-10":158,"2023-02-
11":146,"2023-02-12":176,"2023-02-13":196,"2023-02-14":142}
dahi@DESKTOP-ATVVKNF:~$ curl http://127.0.0.1:5153/api/reports/user_visits
{"1":{"joined_at":"2023-01-20 03:23:39.757813","visits":6},"2":{"joined_at":"2023-02-02
16:23:39.757830","visits":5},"3":{"joined_at":"2023-01-31 10:23:39.757836",...
```

Set up Continuous Integration with AWS ECR & CodeBuild

Create a new GitHub repository:



Create a new AWS ECR repository:

[Amazon ECR](#) > [Private registry](#) > [Repositories](#) > [Create repository](#)

Create repository

General settings

Visibility settings [Info](#)
Choose the visibility setting for the repository.

☒ **Private**
Access is managed by IAM and repository policy permissions.

☐ **Public**
Publicly visible and accessible for image pulls.

Repository name
Provide a concise name. A developer should be able to identify the repository contents by the name.

265681628035.dkr.ecr.us-west-1.amazonaws.com/

13 out of 256 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, hyphens, underscores, periods and forward slashes.

Tag immutability [Info](#)
Enable tag immutability to prevent image tags from being overwritten by subsequent image pushes using the same tag. Disable tag immutability to allow image tags to be overwritten.

☒ **Enabled**

Once a repository is created, the visibility setting of the repository can't be changed.

Set up an AWS CodeBuild pipeline that has access to pull from this GitHub repository:

[Developer Tools](#) > [CodeBuild](#) > [Build projects](#) > [Create build project](#)

Create build project

Project configuration

Project name

A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and _.

Additional configuration
Description, Build badge, Concurrent build limit, tags

Source [Add source](#)

Source 1 - Primary

Source provider

Repository
☒ **Repository in my GitHub account** ☐ **Public repository**

GitHub repository

https://github.com/<user-name>/<repository-name>

Environment variables

Name	Value	Type	
AWS_DEFAULT_REGION	us-west-1	Plaintext ▼	Remove
AWS_ACCOUNT_ID	265681628035	Plaintext ▼	Remove
IMAGE_REPO_NAME	ecr-repo-demo	Plaintext ▼	Remove

Webhook - *optional* [Info](#)

☒ Rebuild every time a code change is pushed to this repository

Build type

☒ **Single build**
Triggers single build

☐ **Batch build**
Triggers multiple builds as single execution

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

Remove filter group

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

▼

PUSH ✕ PULL_REQUEST_MERGED ✕

Buildspec

Build specifications

☐ **Insert build commands**
Store build commands as build project configuration

☒ **Use a buildspec file**
Store build commands in a YAML-formatted buildspec file

Buildspec name - *optional*

By default, CodeBuild looks for a file named buildspec.yml in the source code root directory. If your buildspec file uses a different name or location, enter its path from the source root here (for example, buildspec-two.yml or configuration/buildspec.yml).

buildspec.yml

Modify the IAM role so that it can have permissions to push images into ECR:

[IAM](#) > [Roles](#) > [codebuild-demo-service-role](#) > Create policy

Step 1
Specify permissions

Step 2
Review and create

Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor Visual **JSON** Actions ▾ □

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Action": [  
7         "ecr:*"  
8       ],  
9       "Resource": [  
10        "*" ]  
11     }  
12   ]  
13 }  
14
```

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

[+ Add new statement](#)

Push a commit to the GitHub repository:

```
dnemu@DESKTOP-ATVVKNF MINGW64 /c/analytics  
$ git init  
Initialized empty Git repository in C:/analytics/.git/  
  
dnemu@DESKTOP-ATVVKNF MINGW64 /c/analytics (master)  
$ git add .  
  
dnemu@DESKTOP-ATVVKNF MINGW64 /c/analytics (master)  
$ git commit -m "first commit"  
[master (root-commit) ee8a833] first commit  
7 files changed, 137 insertions(+)  
create mode 100644 Dockerfile  
create mode 100644 app/__init__.py  
create mode 100644 app/__pycache__/config.cpython-310.pyc  
create mode 100644 app/app.py  
create mode 100644 app/config.py  
create mode 100644 app/requirements.txt  
create mode 100644 buildspect.yml  
  
dnemu@DESKTOP-ATVVKNF MINGW64 /c/analytics (master)  
$ git remote add origin https://github.com/dahinemutlu/co-working-analytics.git  
  
dnemu@DESKTOP-ATVVKNF MINGW64 /c/analytics (master)  
$ git branch -M main  
  
dnemu@DESKTOP-ATVVKNF MINGW64 /c/analytics (main)  
$ git push -u origin main
```

```
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (11/11), 2.58 KiB | 2.58 MiB/s, done.
Total 11 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/dahinemutlu/co-working-analytics.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

We should see in CodeBuild, a new build has been created. Once this build is succeeded, we should see the new image in the ECR that is pushed by CodeBuild automatically:

Developer Tools > CodeBuild > Build projects > codebuild-demo

codebuild-demo

Actions ▼ Create trigger Edit Debug build Start build with overrides

Start build

Build history Batch history Project details Build triggers Metrics

Build history

< 1 > ⚙

<input type="checkbox"/>	Build run	Status	Build number	Source version	Submitter	Duration
<input type="checkbox"/>	codebuild-demo:2903c82f-ff3f-401d-9571-f02f6ffca92f	✔ Succeeded	1	eefa833a6514865d787f871fec2fa1d7b4e2f41e	GitHub-Hookshot/7211029	40 seconds

Amazon ECR > Private registry > Repositories > ecr-repo-demo >
sha256:879376d161b7a3fdec5bb3a7f1538cdf3ef76fcf970aced2c071082630b95986

Image

Details

Image tags

1

URI

265681628035.dkr.ecr.us-west-1.amazonaws.com/ecr-repo-demo:1

Digest

sha256:879376d161b7a3fdec5bb3a7f1538cdf3ef76fcf970aced2c071082630b95986

General information

Artifact type

Image

Repository

ecr-repo-demo

Pushed at

March 29, 2024, 22:11:14 (UTC+03)

Size (MB)

67.17

Deploy the Application

Apply YAML configurations:

```
dahi@DESKTOP-ATVVKNF:~/microservices-aws-k8s-project/deployment$ kubectl apply -f analytics-configmap.yaml
configmap/db-env created
dahi@DESKTOP-ATVVKNF:~/microservices-aws-k8s-project/deployment$ kubectl apply -f analytics-secret.yaml
secret/db-secret created
dahi@DESKTOP-ATVVKNF:~/microservices-aws-k8s-project/deployment$ kubectl apply -f analytics.yaml
service/analytics created
deployment.apps/analytics created
```

Verify the deployment:

```
dahi@DESKTOP-ATVVKNF:~$ kubectl get svc
NAME                                TYPE                CLUSTER-IP          EXTERNAL-IP
PORT(S)          AGE
analytics         LoadBalancer       10.100.156.114
aff6d714e6073412c8ffac13469e5dcf-220302131.us-west-1.elb.amazonaws.com 5153:31959/TCP
46s
kubernetes        ClusterIP           10.100.0.1          <none>
443/TCP           12h
```

```

my-release-postgresql      ClusterIP      10.100.69.0      <none>
5432/TCP                   12h
my-release-postgresql-hl   ClusterIP      None              <none>
5432/TCP                   12h
dahi@DESKTOP-ATVVKNF:~$ kubectl describe deployment analytics
Name:                      analytics
Namespace:                 default
CreationTimestamp:         Sun, 31 Mar 2024 23:56:33 +0300
Labels:                    name=analytics
Annotations:               deployment.kubernetes.io/revision: 1
Selector:                  service=analytics
Replicas:                  1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:              RollingUpdate
MinReadySeconds:           0
RollingUpdateStrategy:     25% max unavailable, 25% max surge
Pod Template:
  Labels:  service=analytics
  Containers:
    analytics:
      Image:   265681628035.dkr.ecr.us-west-1.amazonaws.com/ecr-repo-demo:1
      Port:    <none>
      Host Port: <none>
      Liveness: http-get http://:5153/health_check delay=5s timeout=2s period=10s
#success=1 #failure=3
      Readiness: http-get http://:5153/readiness_check delay=5s timeout=5s period=10s
#success=1 #failure=3
      Environment:
        DB_NAME:      <set to the key 'DB_NAME' of config map 'db-env'>      Optional:
false
        DB_HOST:      <set to the key 'DB_HOST' of config map 'db-env'>      Optional:
false
        DB_PORT:      <set to the key 'DB_PORT' of config map 'db-env'>      Optional:
false
        DB_USERNAME:  <set to the key 'DB_USERNAME' of config map 'db-env'>  Optional:
false
        DB_PASSWORD:  <set to the key 'DB_PASSWORD' in secret 'db-secret'>  Optional:
false
      Mounts:          <none>
      Volumes:          <none>
Conditions:
  Type           Status  Reason
  ----           -
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets: <none>
NewReplicaSet:  analytics-65ccbb6fc8 (1/1 replicas created)
Events:
  Type    Reason             Age   From                  Message
  ----    -
  Normal  ScalingReplicaSet  101s  deployment-controller  Scaled up replica set
analytics-65ccbb6fc8 to 1

```

Verify the application:

```
dahi@DESKTOP-ATVVKNF:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
analytics-65ccbb6fc8-cw89z         1/1     Running   0           2m47s
my-release-postgresql-0            1/1     Running   0           12h
dahi@DESKTOP-ATVVKNF:~$ kubectl logs analytics-65ccbb6fc8-cw89z
* Serving Flask app 'config'
* 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://127.0.0.1:5153
* Running on http://172.31.22.240:5153
Press CTRL+C to quit
172.31.23.68 - - [31/Mar/2024 20:56:43] "GET /health_check HTTP/1.1" 200 -
172.31.23.68 - - [31/Mar/2024 20:56:43] "GET /readiness_check HTTP/1.1" 200 -
172.31.23.68 - - [31/Mar/2024 20:56:53] "GET /health_check HTTP/1.1" 200 -
172.31.23.68 - - [31/Mar/2024 20:56:53] "GET /readiness_check HTTP/1.1" 200 -
172.31.23.68 - - [31/Mar/2024 20:57:03] "GET /health_check HTTP/1.1" 200 -
172.31.23.68 - - [31/Mar/2024 20:57:03] "GET /readiness_check HTTP/1.1" 200 -
[2024-03-31 20:57:08,269] INFO in app: {'2023-02-12': 176, '2023-02-13': 196, '2023-02-
09': 179, '2023-02-11': 146, '2023-02-14': 142, '2023-02-07': 40, '2023-02-10': 158,
'2023-02-08': 202}
dahi@DESKTOP-ATVVKNF:~$ kubectl port-forward service/analytics 5152:5153 &
[1] 7243
dahi@DESKTOP-ATVVKNF:~$ Forwarding from 127.0.0.1:5152 -> 5153
Forwarding from [::1]:5152 -> 5153
dahi@DESKTOP-ATVVKNF:~$ curl http://127.0.0.1:5152/api/reports/daily_usage
{"2023-02-07":40,"2023-02-08":202,"2023-02-09":179,"2023-02-10":158,"2023-02-
11":146,"2023-02-12":176,"2023-02-13":196,"2023-02-14":142}
dahi@DESKTOP-ATVVKNF:~$ curl http://127.0.0.1:5152/api/reports/user_visits
{"1":{"joined_at":"2023-01-20 03:23:39.757813","visits":6},"2":{"joined_at":"2023-02-02
16:23:39.757830","visits":5},"3":{"joined_at":"2023-01-31
10:23:39.757836","visits":5},...
```


Load Test and Monitoring

Attach the necessary policy (CloudWatchAgentServerPolicy) to the IAM role for your worker nodes to send metrics and logs to CloudWatch:

[IAM](#) > [Roles](#) > AmazonEKSNodeRole

AmazonEKSNodeRole [Info](#)

Delete

Amazon EKS - Node role

[Permissions](#) | [Trust relationships](#) | [Tags](#) | [Access Advisor](#) | [Revoke sessions](#)

Permissions policies (5) [Info](#)

↻

Simulate [↗](#)

Remove

Add permissions ▼

You can attach up to 10 managed policies.

Filter by Type

All types ▼


< 1 > ⚙

<input type="checkbox"/>	Policy name ↗	Type
<input type="checkbox"/>	+ AmazonEC2ContainerRegistryReadOnly	AWS managed
<input type="checkbox"/>	+ AmazonEKS_CNI_Policy	AWS managed
<input type="checkbox"/>	+ AmazonEKSTaskNodePolicy	AWS managed
<input type="checkbox"/>	+ AmazonEMRReadOnlyAccessPolicy_v2	AWS managed
<input type="checkbox"/>	+ CloudWatchAgentServerPolicy	AWS managed

Add Amazon CloudWatch Observability add-on to the EKS cluster:

[EKS](#) > [Clusters](#) > demo-eks-cluster

demo-eks-cluster



Delete cluster

► **Cluster info** [Info](#)

< | Overview | Resources | Compute | Networking | **Add-ons** | Access | Observabil | >

Add-ons (5) [Info](#)

View details

Edit

Remove


Get more add-ons

Any category ▼

Any status ▼

5 matches


< 1 >



Amazon CloudWatch Observability


Install CloudWatch Agent and enable Container Insights within your cluster.

Category
observability

Status
 Active

Version
v1.4.0-eksbuild.1

IAM role for service account (IRSA)
Not set



Perform an API load testing:

HomeWorkspacesAPI Network

Search Postman

My Workspace

NewImport

Co-working Space AnalyRunner

Collections

Environments

History

Co-working Space Analytics

GET Daily Usage

GET User Visits

Run order

Deselect AllSelect AllReset

GET Daily Usage

GET User Visits

FunctionalPerformance

Test how your APIs perform under load

Simulate real-world traffic from your local machine and observe the performance of your APIs. Learn more about [performance testing](#)

Set up your performance test

Load profileVirtual usersTest duration

Peak5010 mins

50 VUs

010 mins

Simulate a fixed load of 10 users for 2 minutes. Then steadily increase the load to 50 users over the next 2 minutes and hold for 2 minutes, and then steadily decrease the load from 50 to 10 over the next 2 minutes, and maintain a fixed load of 10 users for 2 minutes.

Base load

10

Data file

FEATURE TRIAL

Select file

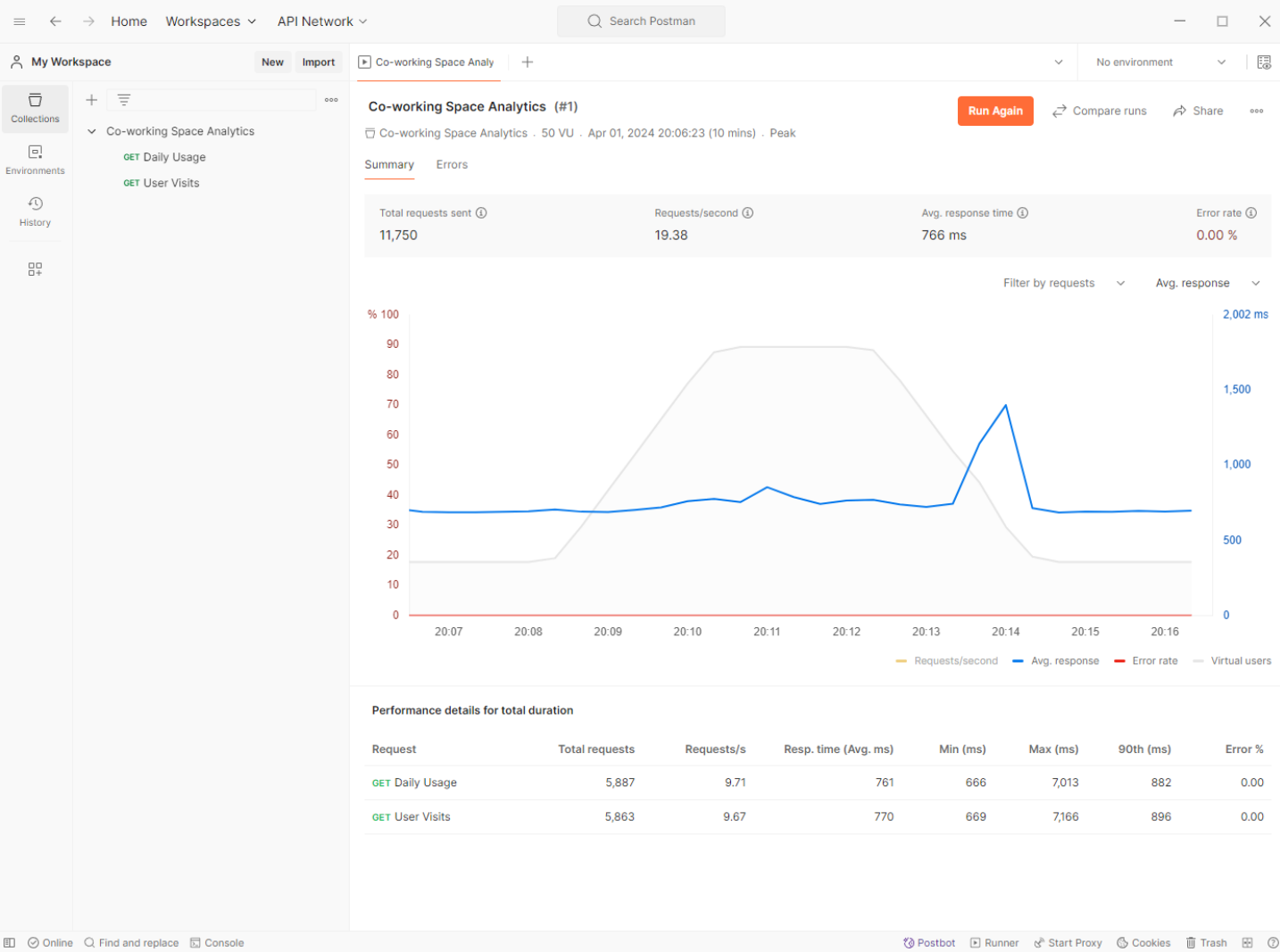
Run

OnlineFind and replaceConsole

PostbotRunnerStart ProxyCookiesTrash

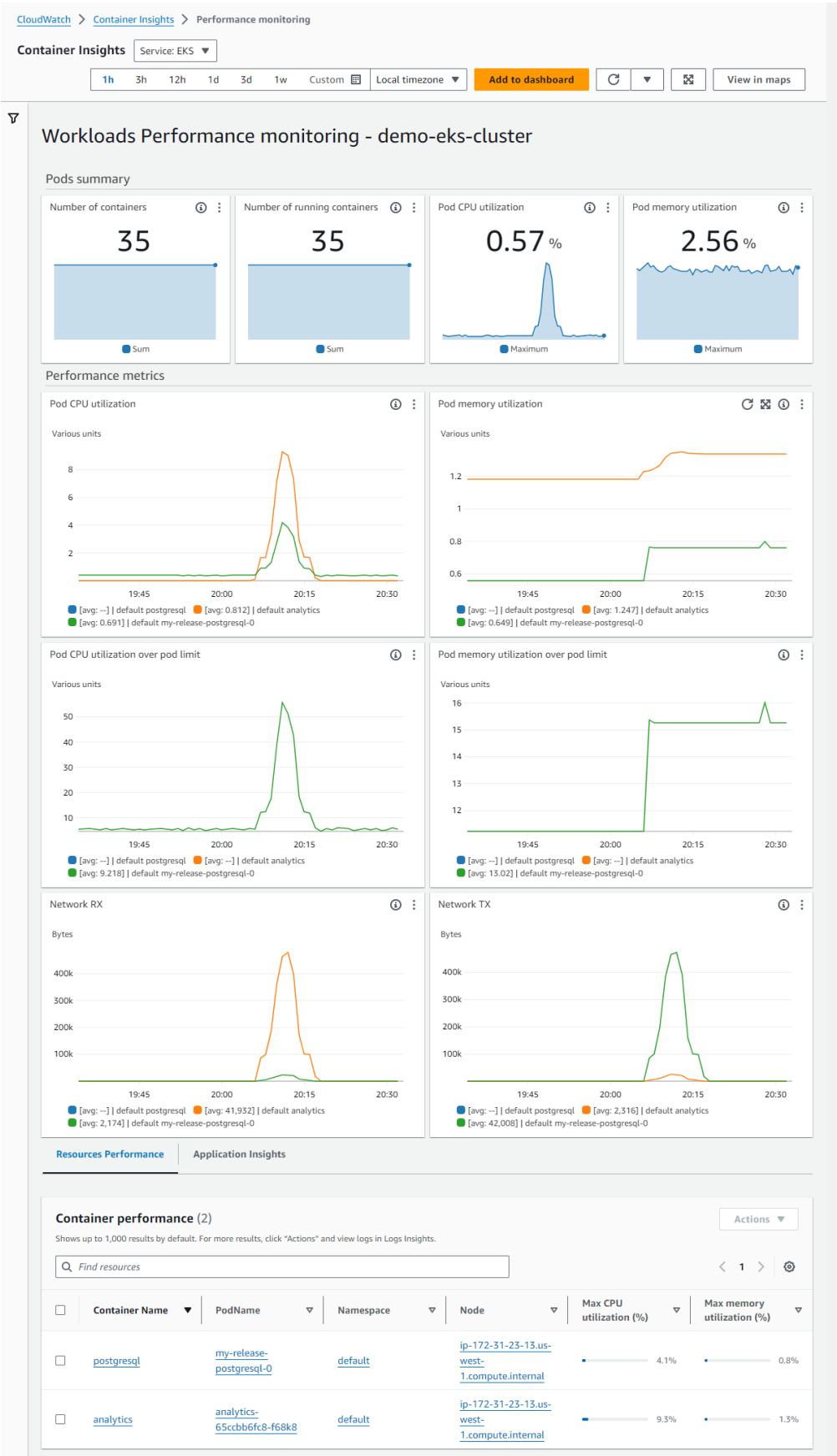
19

Results:



Both APIs were tested for 10 minutes with concurrent requests from 50 virtual users. All requests (~11k) were processed without any error and the average response time for both APIs is under 1 second.

Check AWS CloudWatch container insights:







The maximum CPU utilization was 9.3% and the maximum memory utilization was 1.3% by the analytics application pod. According to the performance test and CloudWatch metrics, we see that the system works smoothly under such a load and does not consume a lot of resources. If it is expected that there will be no more load than this, lower equipped AWS instances (such as t3.small) may be preferred to reduce the cost.



Check AWS CloudWatch logs:


[CloudWatch](#) > [Log groups](#) > [/aws/containerinsights/demo-eks-cluster/application](#) > All events


Log events


 Actions  Start tailing 


You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#) 

 INFO in app 

1m 1h 1h 

Local timezone 

Display 



Message

```
{
  "time": "2024-04-01T16:46:15.968198924Z",
  "stream": "stderr",
  "_p": "F",
  "log": "[2024-04-01 16:46:15,967] INFO in app: {'2023-02-12': 176, '2023-02-13': 196, '2023-02-09': 179, '2023-02-11': 146, '2023-02-14': 142, '2023-02-07': 40, '2023-02-10': 158, '2023-02-08': 202}",
  "kubernetes": {
    "pod_name": "analytics-65ccbb6fc8-f68k8",
    "namespace_name": "default",
    "pod_id": "958abb7c-2059-476b-9c7b-7001c74e9555",
    "host": "ip-172-31-23-13.us-west-1.compute.internal",
    "container_name": "analytics",
    "docker_id": "9141db8061bf13f0780a0f9ac003e32f07534bdb7f1f00f3829dac92fdf98576",
    "container_hash": "265681628035.dkr.ecr.us-west-1.amazonaws.com/ecr-repo-demo@sha256:879376d161b7a3fdec5bb3a7f1538cdf3ef76fcf970aced2c071082630b95986",
    "container_image": "265681628035.dkr.ecr.us-west-1.amazonaws.com/ecr-repo-demo:1"
  }
}
```

```
{
  "time": "2024-04-01T16:46:45.968445333Z",
  "stream": "stderr",
  "_p": "F",
  "log": "[2024-04-01 16:46:45,968] INFO in app: {'2023-02-12': 176, '2023-02-13': 196, '2023-02-09': 179, '2023-02-11': 146, '2023-02-14': 142, '2023-02-07': 40, '2023-02-10': 158, '2023-02-08': 202}",
  "kubernetes": {
    "pod_name": "analytics-65ccbb6fc8-f68k8",
    "namespace_name": "default",
    "pod_id": "958abb7c-2059-476b-9c7b-7001c74e9555",
    "host": "ip-172-31-23-13.us-west-1.compute.internal",
    "container_name": "analytics",
    "docker_id": "9141db8061bf13f0780a0f9ac003e32f07534bdb7f1f00f3829dac92fdf98576",
    "container_hash": "265681628035.dkr.ecr.us-west-1.amazonaws.com/ecr-repo-demo@sha256:879376d161b7a3fdec5bb3a7f1538cdf3ef76fcf970aced2c071082630b95986",
    "container_image": "265681628035.dkr.ecr.us-west-1.amazonaws.com/ecr-repo-demo:1"
  }
}
```