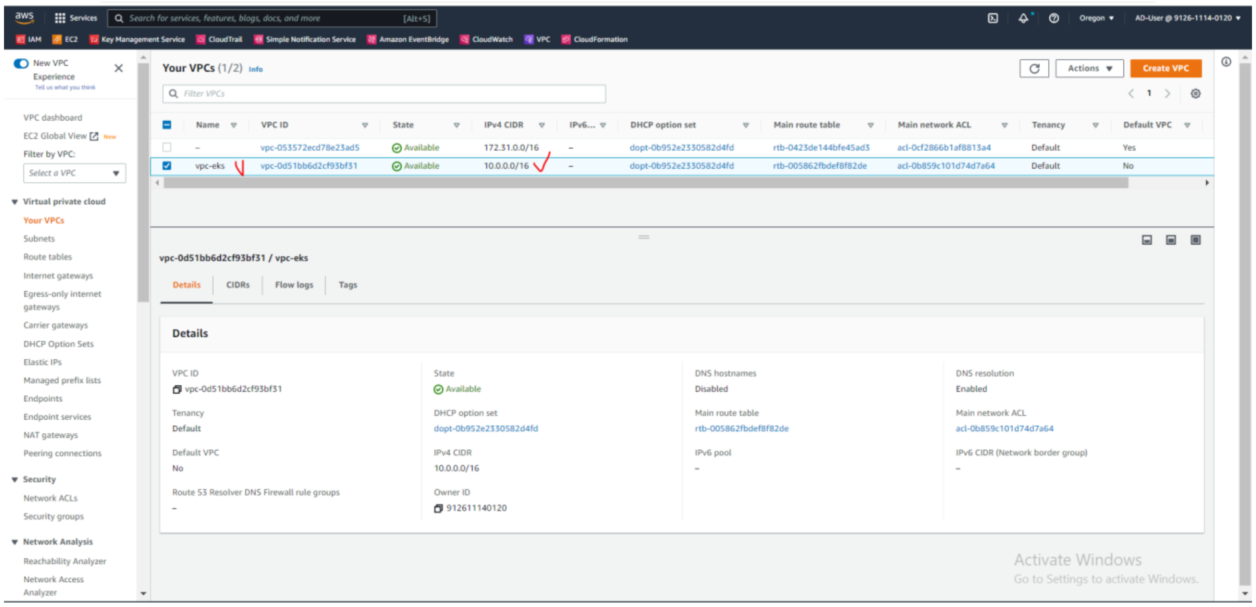
1. Creating your own VPC:



Your VPCs (1/2)

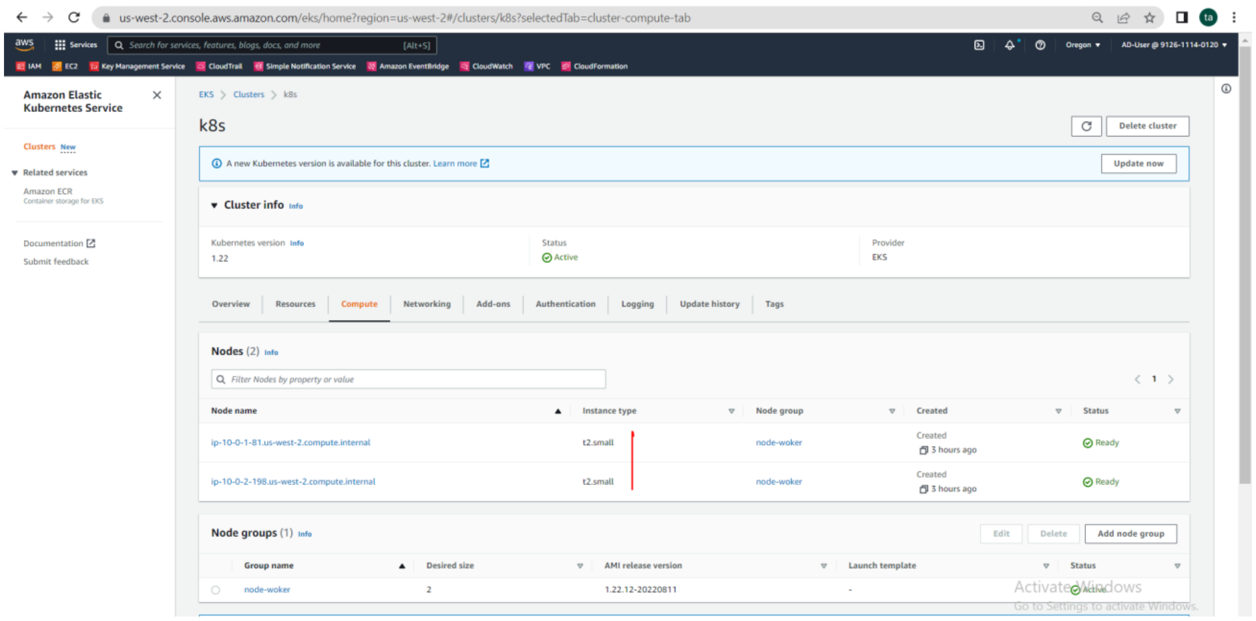
Name	VPC ID	State	IPv4 CIDR	IPv6...	DHCP option set	Main route table	Main network ACL	Tenancy	Default VPC
–	vpc-053572ecd78e23ad5	Available	172.31.0.0/16	–	dopt-0b952e2330582d4fd	rtb-0423de144bfe45ad3	acl-0cf2866b1af8813a4	Default	Yes
vpc-eks	vpc-0d51bb6d2cf93bf31	Available	10.0.0.0/16	–	dopt-0b952e2330582d4fd	rtb-005862fbdef8f82de	acl-0b859c101d74d7a64	Default	No

vpc-0d51bb6d2cf93bf31 / vpc-eks

Details

VPC ID: vpc-0d51bb6d2cf93bf31	State: Available	DNS hostnames: Disabled	DNS resolution: Enabled
Tenancy: Default	DHCP option set: dopt-0b952e2330582d4fd	Main route table: rtb-005862fbdef8f82de	Main network ACL: acl-0b859c101d74d7a64
Default VPC: No	IPv4 CIDR: 10.0.0.0/16	IPv6 pool: –	IPv6 CIDR (Network border group): –
Route 53 Resolver DNS Firewall rule groups: –	Owner ID: 912611140120		

2. Creating an EKS cluster 2 worker nodes on the public subnet



us-west-2.console.aws.amazon.com/eks/home?region=us-west-2#/clusters/k8s?selectedTab=cluster-compute-tab

k8s

A new Kubernetes version is available for this cluster. Learn more

Cluster info

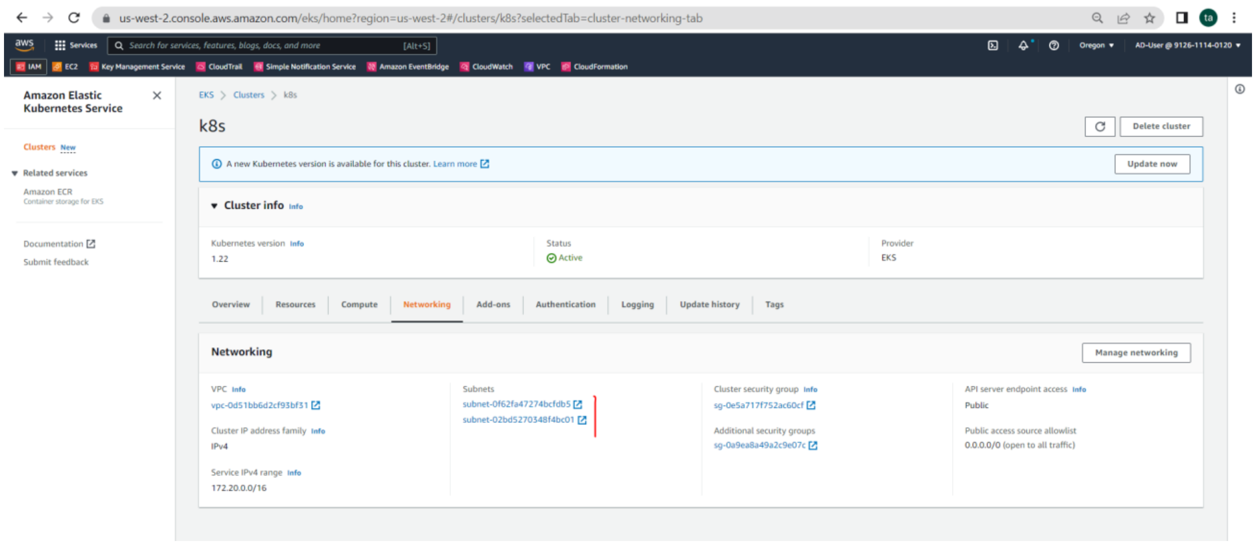
Kubernetes version: 1.22
Status: Active
Provider: EKS

Nodes (2)

Node name	Instance type	Node group	Created	Status
ip-10-0-1-81.us-west-2.compute.internal	t2.small	node-woker	Created 3 hours ago	Ready
ip-10-0-2-198.us-west-2.compute.internal	t2.small	node-woker	Created 3 hours ago	Ready

Node groups (1)

Group name	Desired size	AMI release version	Launch template	Status
node-woker	2	1.22.12-20220811	-	Active



k8s

A new Kubernetes version is available for this cluster. Learn more

Cluster info Info

Kubernetes version	Status	Provider
1.22	Active	EKS

Networking

VPC Info
vpc-0d51bb6d2cf93bf31

Cluster IP address family Info
IPv4

Service IPv4 range Info
172.20.0.0/16

Subnets
subnet-0f62fa47274bcfdb5
subnet-02bd5270348f4bc01

Cluster security group Info
sg-0e5a717f752ac60cf

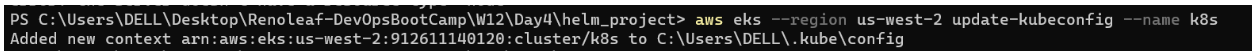
Additional security groups
sg-0a9ea8a49a2c9e07c

API server endpoint access Info
Public

Public access source allowlist
0.0.0.0/0 (open to all traffic)

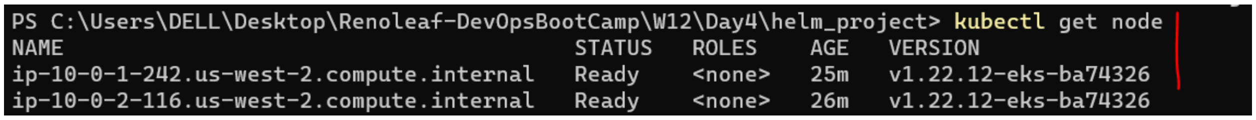
3. Creating chart mysql by Helm:

- Step 1: Add configure context with cluster:



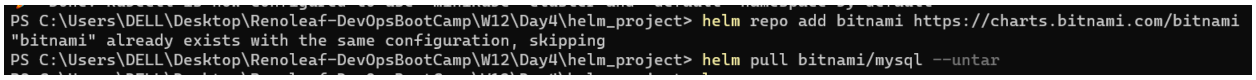
```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project> aws eks --region us-west-2 update-kubeconfig --name k8s
Added new context arn:aws:eks:us-west-2:912611140120:cluster/k8s to C:\Users\DELL\.kube\config
```

Check get node



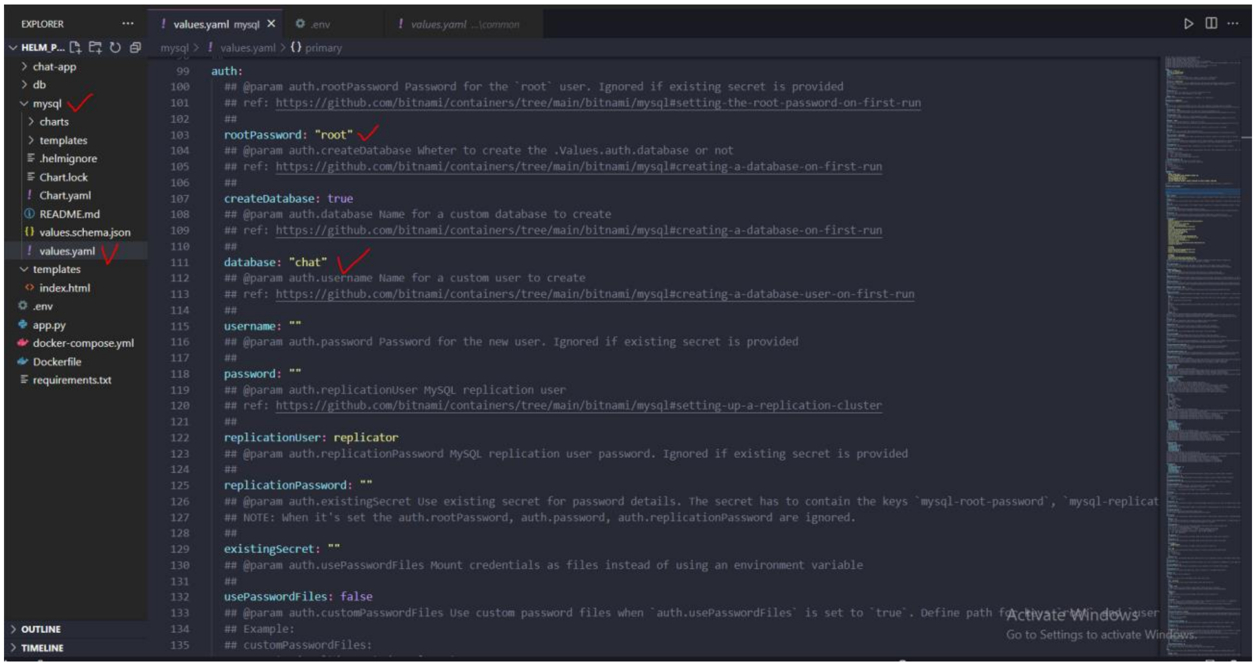
```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project> kubectl get node
NAME                                       STATUS   ROLES    AGE   VERSION
ip-10-0-1-242.us-west-2.compute.internal   Ready    <none>   25m   v1.22.12-eks-ba74326
ip-10-0-2-116.us-west-2.compute.internal   Ready    <none>   26m   v1.22.12-eks-ba74326
```

- Step 2: add bitnami repo and pull mysql chart to create mysql



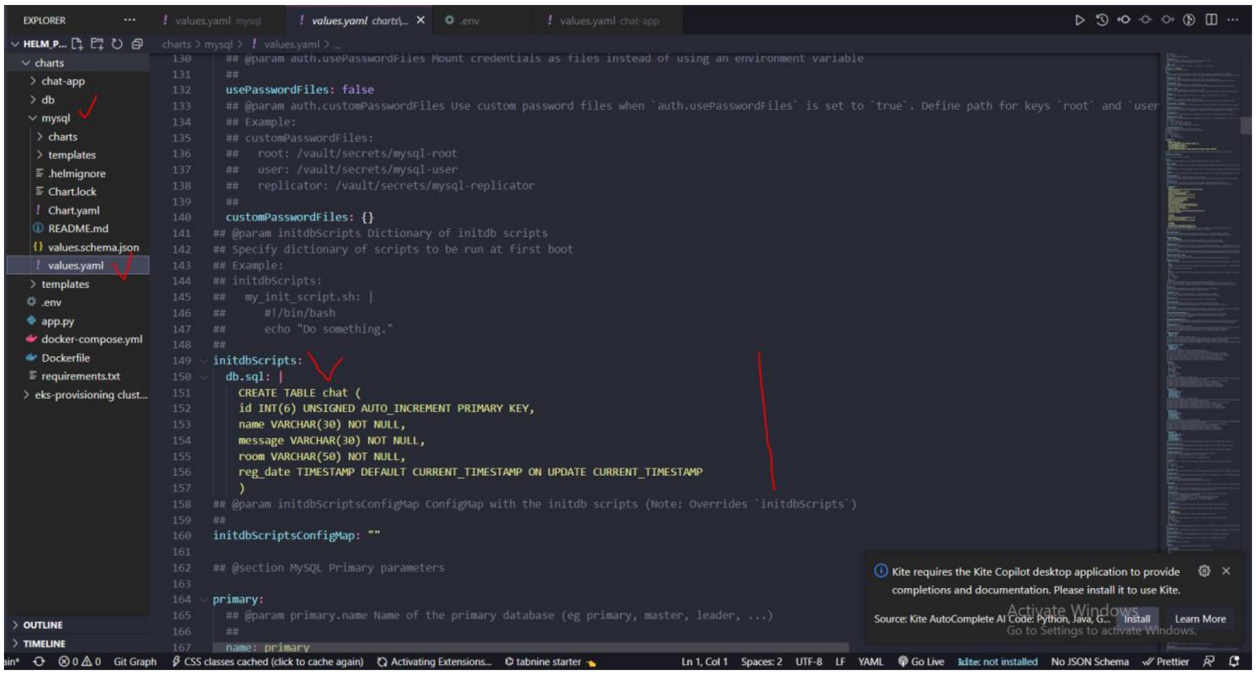
```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project> helm repo add bitnami https://charts.bitnami.com/bitnami
"bitnami" already exists with the same configuration, skipping
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project> helm pull bitnami/mysql --untar
```

- Step 3: configure file values.yaml (same with .env file)



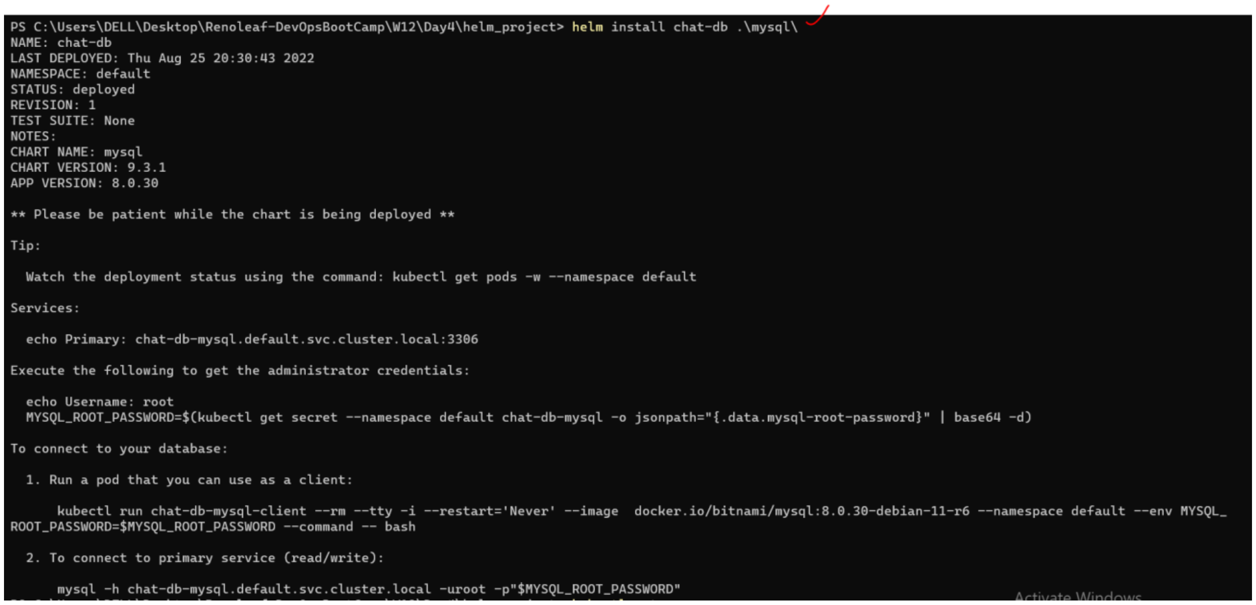
```
auth:
  ## @param auth.rootPassword Password for the `root` user. Ignored if existing secret is provided
  ## ref: https://github.com/bitnami/containers/tree/main/bitnami/mysql#setting-the-root-password-on-first-run
  ##
  rootPassword: "root"
  ## @param auth.createDatabase Wheter to create the .Values.auth.database or not
  ## ref: https://github.com/bitnami/containers/tree/main/bitnami/mysql#creating-a-database-on-first-run
  ##
  createDatabase: true
  ## @param auth.database Name for a custom database to create
  ## ref: https://github.com/bitnami/containers/tree/main/bitnami/mysql#creating-a-database-on-first-run
  ##
  database: "chat"
  ## @param auth.username Name for a custom user to create
  ## ref: https://github.com/bitnami/containers/tree/main/bitnami/mysql#creating-a-database-user-on-first-run
  ##
  username: ""
  ## @param auth.password Password for the new user. Ignored if existing secret is provided
  ##
  password: ""
  ## @param auth.replicationUser MySQL replication user
  ## ref: https://github.com/bitnami/containers/tree/main/bitnami/mysql#setting-up-a-replication-cluster
  ##
  replicationUser: replicator
  ## @param auth.replicationPassword MySQL replication user password. Ignored if existing secret is provided
  ##
  replicationPassword: ""
  ## @param auth.existingSecret Use existing secret for password details. The secret has to contain the keys `mysql-root-password`, `mysql-replicat
  ## NOTE: When it's set the auth.rootPassword, auth.password, auth.replicationPassword are ignored.
  ##
  existingSecret: ""
  ## @param auth.usePasswordFiles Mount credentials as files instead of using an environment variable
  ##
  usePasswordFiles: false
  ## @param auth.customPasswordFiles Use custom password files when `auth.usePasswordFiles` is set to `true`. Define path fo[illegible] user
  ## Example:
  ## customPasswordFiles:
```

Configure initscript:



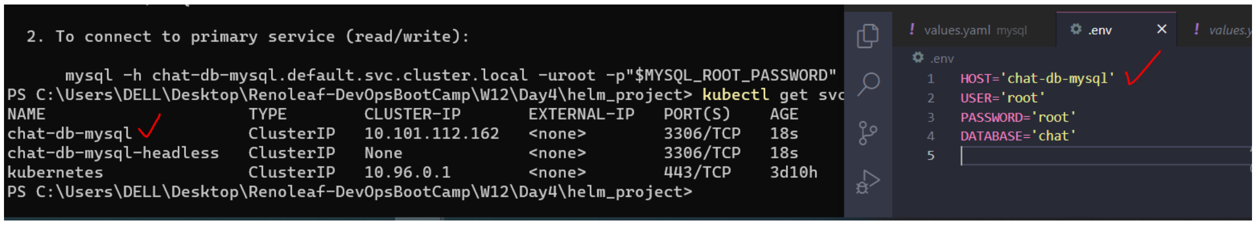
```
  ## @param auth.usePasswordFiles Mount credentials as files instead of using an environment variable
  ##
  usePasswordFiles: false
  ## @param auth.customPasswordFiles Use custom password files when `auth.usePasswordFiles` is set to `true`. Define path for keys `root` and `user
  ## Example:
  ## customPasswordFiles:
  ##   root: /vault/secrets/mysql-root
  ##   user: /vault/secrets/mysql-user
  ##   replicator: /vault/secrets/mysql-replicator
  ##
  customPasswordFiles: {}
## @param initdbScripts Dictionary of initdb scripts
## Specify dictionary of scripts to be run at first boot
## Example:
## initdbScripts:
##   my_init_script.sh: |
##      #!/bin/bash
##      echo "Do something."
##
initdbScripts:
  db.sql: |
    CREATE TABLE chat (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(30) NOT NULL,
    message VARCHAR(30) NOT NULL,
    room VARCHAR(50) NOT NULL,
    reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
    )
## @param initdbScriptsConfigMap ConfigMap with the initdb scripts (Note: Overrides `initdbScripts`)
##
initdbScriptsConfigMap: ""

## @section MySQL Primary parameters

primary:
  ## @param primary.name Name of the primary database (eg primary, master, leader, ...)
  ##
  name: primary
```

- Step 4: Install the chart mysql db by helm



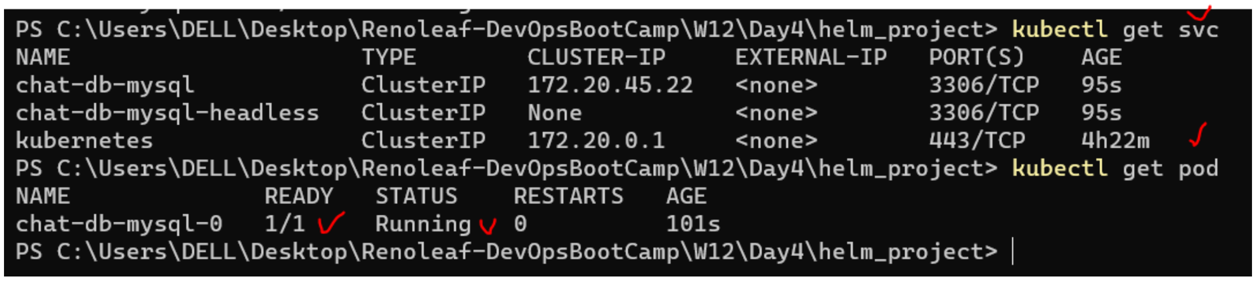
```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project> helm install chat-db .\mysql\
NAME: chat-db
LAST DEPLOYED: Thu Aug 25 20:30:43 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
CHART NAME: mysql
CHART VERSION: 9.3.1
APP VERSION: 8.0.30

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

Tip:

  Watch the deployment status using the command: kubectl get pods -w --namespace default

Services:

  echo Primary: chat-db-mysql.default.svc.cluster.local:3306

Execute the following to get the administrator credentials:

  echo Username: root
  MYSQL_ROOT_PASSWORD=$(kubectl get secret --namespace default chat-db-mysql -o jsonpath="{.data.mysql-root-password}" | base64 -d)

To connect to your database:

  1. Run a pod that you can use as a client:

      kubectl run chat-db-mysql-client --rm --tty -i --restart='Never' --image  docker.io/bitnami/mysql:8.0.30-debian-11-r6 --namespace default --env MYSQL_
ROOT_PASSWORD=$MYSQL_ROOT_PASSWORD --command -- bash

  2. To connect to primary service (read/write):

      mysql -h chat-db-mysql.default.svc.cluster.local -uroot -p"$MYSQL_ROOT_PASSWORD"
```

Get service name, database name (form values.yaml file) and pass to .env file:



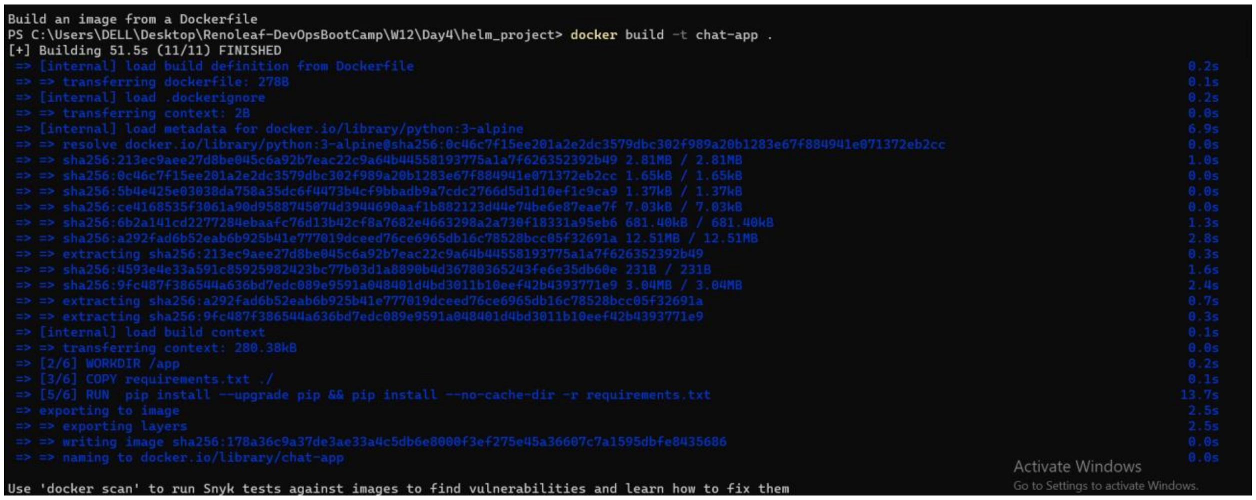
```
  2. To connect to primary service (read/write):

      mysql -h chat-db-mysql.default.svc.cluster.local -uroot -p"$MYSQL_ROOT_PASSWORD"
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project> kubectl get svc
NAME                     TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
chat-db-mysql            ClusterIP   10.101.112.162   <none>        3306/TCP   18s
chat-db-mysql-headless   ClusterIP   None             <none>        3306/TCP   18s
kubernetes               ClusterIP   10.96.0.1        <none>        443/TCP    3d10h
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project>
```

```
HOST='chat-db-mysql'
USER='root'
PASSWORD='root'
DATABASE='chat'
```



```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project> kubectl get svc
NAME                     TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)    AGE
chat-db-mysql            ClusterIP   172.20.45.22   <none>        3306/TCP   95s
chat-db-mysql-headless   ClusterIP   None           <none>        3306/TCP   95s
kubernetes               ClusterIP   172.20.0.1     <none>        443/TCP    4h22m
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project> kubectl get pod
NAME              READY   STATUS    RESTARTS   AGE
chat-db-mysql-0   1/1     Running   0          101s
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project>
```

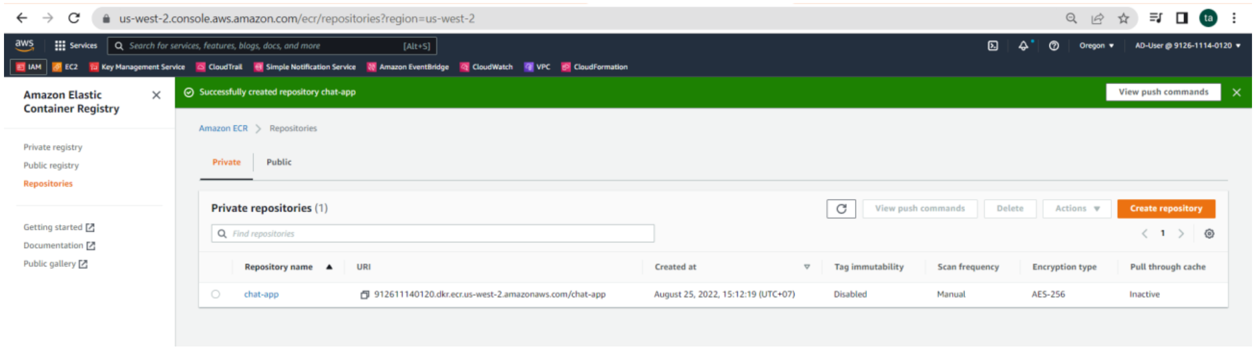
4. Packaging ChatApp application into a Docker image.



```
Build an image from a Dockerfile
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project> docker build -t chat-app .
[+] Building 51.5s (11/11) FINISHED
 => [internal] load build definition from Dockerfile                                                                     0.2s
 => => transferring dockerfile: 278B                                                                                     0.1s
 => [internal] load .dockerignore                                                                                        0.2s
 => => transferring context: 2B                                                                                          0.0s
 => [internal] load metadata for docker.io/library/python:3-alpine                                                       6.9s
 => => resolve docker.io/library/python:3-alpine@sha256:0c46c7f15ee201a2e2dc3579dbc302f989a20b1283e67f884941e071372eb2cc 0.0s
 => => sha256:213ec9aee27d8be045c6a92b7eac22c9a64b44558193775a1a7f626352392b49 2.81MB / 2.81MB                            1.0s
 => => sha256:0c46c7f15ee201a2e2dc3579dbc302f989a20b1283e67f884941e071372eb2cc 1.65kB / 1.65kB                           0.0s
 => => sha256:5b4e425e03038da758a35dc6f4473b4cf9bbadb9a7cdc2766d5d1d10ef1c9ca9 1.37kB / 1.37kB                           0.0s
 => => sha256:ce4168535f3061a90d9588745074d3944690aaf1b882123d44e74be6e87eae7f 7.03kB / 7.03kB                           0.0s
 => => sha256:6b2a141cd2277284ebaafc76d13b42cf8a7682e4663298a2a730f18331a95eb6 681.40kB / 681.40kB                       1.3s
 => => sha256:a292fad6b52eab6b925b41e777019dceed76ce6965db16c78528bcc05f32691a 12.51MB / 12.51MB                         2.8s
 => => extracting sha256:213ec9aee27d8be045c6a92b7eac22c9a64b44558193775a1a7f626352392b49                                0.3s
 => => sha256:4593e4e33a591c85925982423bc77b03d1a8890b4d367803652 43fe6e35db60e 231B / 231B                              1.6s
 => => sha256:9fc487f386544a636bd7edc089e9591a048401d4bd3011b10eef42b4393771e9 3.04MB / 3.04MB                           2.4s
 => => extracting sha256:a292fad6b52eab6b925b41e777019dceed76ce6965db16c78528bcc05f32691a                                0.7s
 => => extracting sha256:9fc487f386544a636bd7edc089e9591a048401d4bd3011b10eef42b4393771e9                                0.3s
 => [internal] load build context                                                                                        0.1s
 => => transferring context: 280.38kB                                                                                    0.0s
 => [2/6] WORKDIR /app                                                                                                   0.2s
 => [3/6] COPY requirements.txt ./                                                                                       0.1s
 => [5/6] RUN  pip install --upgrade pip && pip install --no-cache-dir -r requirements.txt                              13.7s
 => exporting to image                                                                                                   2.5s
 => => exporting layers                                                                                                  2.5s
 => => writing image sha256:178a36c9a37de3ae33a4c5db6e8000f3ef275e45a36607c7a1595dbfe8435686                             0.0s
 => => naming to docker.io/library/chat-app                                                                              0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

5. Uploading the Docker image to the container registry ECR.

- step 1: Create chat-app in ECR



us-west-2.console.aws.amazon.com/ecr/repositories?region=us-west-2

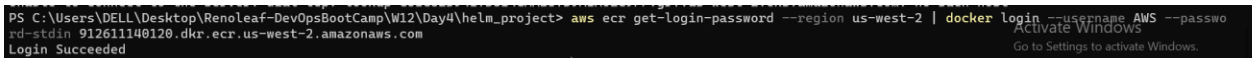
Successfully created repository chat-app

Amazon ECR > Repositories

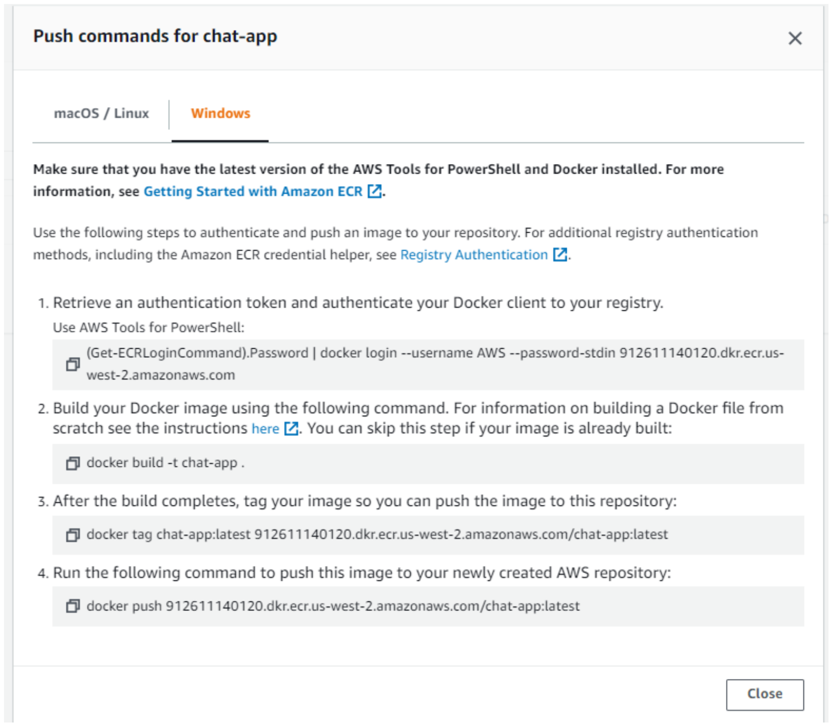
Private repositories (1)

	Repository name	URI	Created at	Tag immutability	Scan frequency	Encryption type	Pull through cache
○	chat-app	912611140120.dkr.ecr.us-west-2.amazonaws.com/chat-app	August 25, 2022, 15:12:19 (UTC+07)	Disabled	Manual	AES-256	Inactive

- Step 2: log in to push image



```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project> aws ecr get-login-password --region us-west-2 | docker login --username AWS --passwo
rd-stdin 912611140120.dkr.ecr.us-west-2.amazonaws.com
Login Succeeded
```



Push commands for chat-app

macOS / Linux | **Windows**

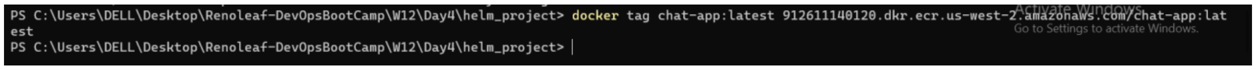
Make sure that you have the latest version of the AWS Tools for PowerShell and Docker installed. For more information, see Getting Started with Amazon ECR.

Use the following steps to authenticate and push an image to your repository. For additional registry authentication methods, including the Amazon ECR credential helper, see Registry Authentication.

1. Retrieve an authentication token and authenticate your Docker client to your registry.
   Use AWS Tools for PowerShell:
   ```
   (Get-ECRLoginCommand).Password | docker login --username AWS --password-stdin 912611140120.dkr.ecr.us-west-2.amazonaws.com
   ```
2. Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions here. You can skip this step if your image is already built:
   ```
   docker build -t chat-app .
   ```
3. After the build completes, tag your image so you can push the image to this repository:
   ```
   docker tag chat-app:latest 912611140120.dkr.ecr.us-west-2.amazonaws.com/chat-app:latest
   ```
4. Run the following command to push this image to your newly created AWS repository:
   ```
   docker push 912611140120.dkr.ecr.us-west-2.amazonaws.com/chat-app:latest
   ```

Close

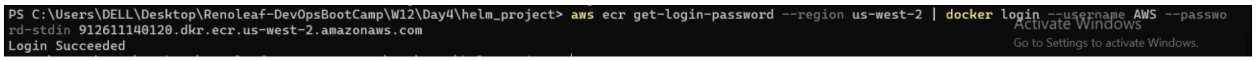
- Step 3: Give tag name



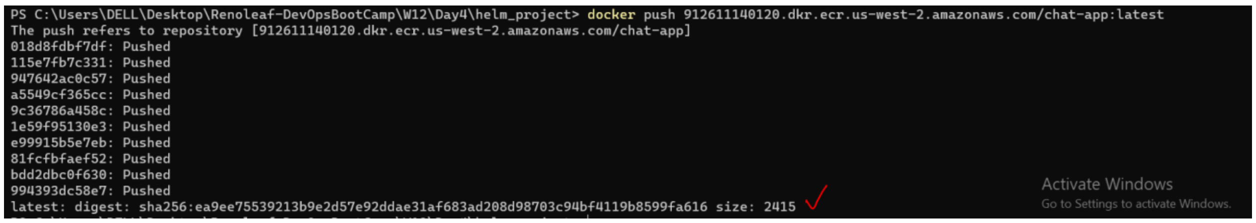
```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project> docker tag chat-app:latest 912611140120.dkr.ecr.us-west-2.amazonaws.com/chat-app:lat
est
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project> |
```

- Step 4: Push image to ECR

Log in to push image to ECR

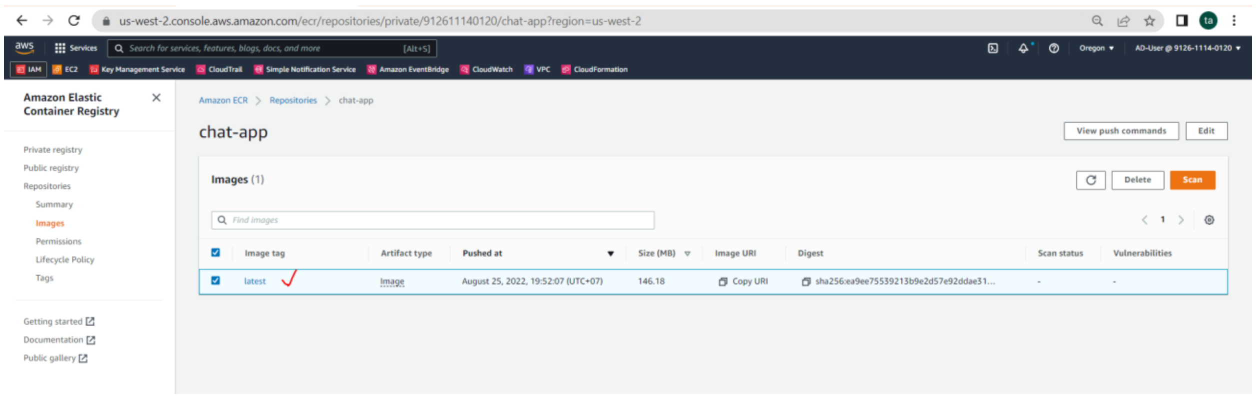


```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project> aws ecr get-login-password --region us-west-2 | docker login --username AWS --passwo
rd-stdin 912611140120.dkr.ecr.us-west-2.amazonaws.com
Login Succeeded
```



```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project> docker push 912611140120.dkr.ecr.us-west-2.amazonaws.com/chat-app:latest
The push refers to repository [912611140120.dkr.ecr.us-west-2.amazonaws.com/chat-app]
018d8fdbf7df: Pushed
115e7fb7c331: Pushed
947642ac0c57: Pushed
a5549cf365cc: Pushed
9c36786a458c: Pushed
1e59f95130e3: Pushed
e99915b5e7eb: Pushed
81fcfbfaef52: Pushed
bdd2dbc0f630: Pushed
994393dc58e7: Pushed
latest: digest: sha256:ea9ee75539213b9e2d57e92ddae31af683ad208d98703c94bf4119b8599fa616 size: 2415
```

- Step 5: Check image in ECR



us-west-2.console.aws.amazon.com/ecr/repositories/private/912611140120/chat-app?region=us-west-2

Amazon ECR > Repositories > chat-app

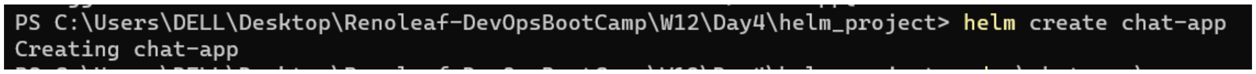
chat-app

Images (1)

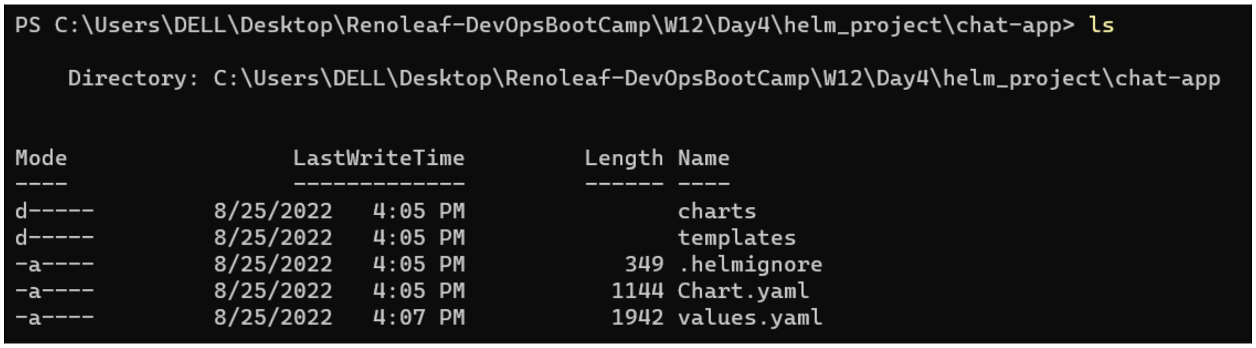
☑	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Scan status	Vulnerabilities
☑	latest	Image	August 25, 2022, 19:52:07 (UTC+07)	146.18	Copy URI	sha256:ea9ee75539213b9e2d57e92ddae31...	-	-

6. Creating chart chat-app by Helm:

- step 1: create chart chat-app

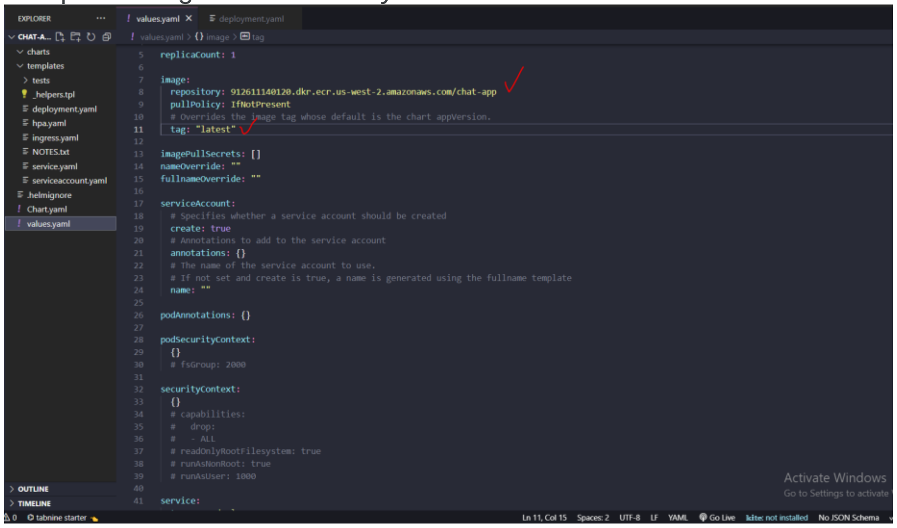


```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project> helm create chat-app
Creating chat-app
```

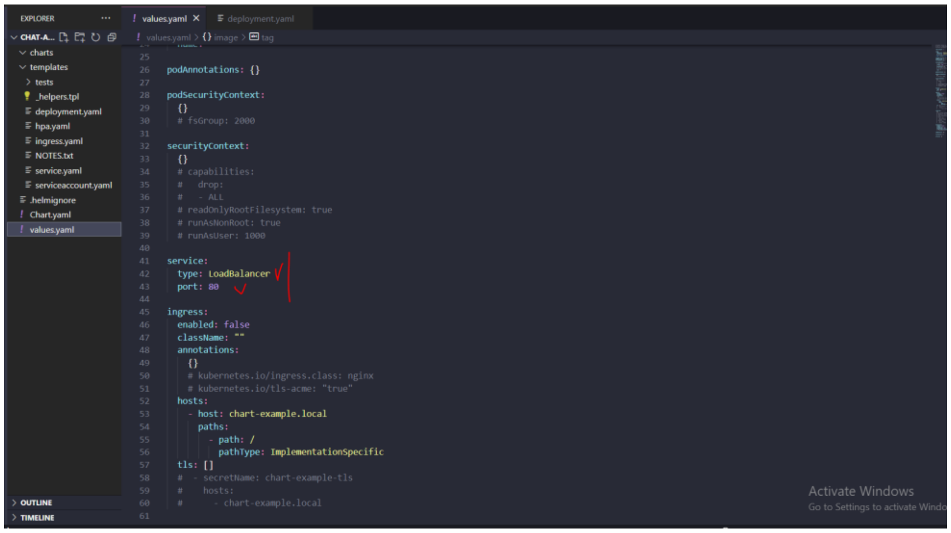


```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project\chat-app> ls

    Directory: C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project\chat-app


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         8/25/2022   4:05 PM                charts
d-----         8/25/2022   4:05 PM                templates
-a----         8/25/2022   4:05 PM            349 .helmignore
-a----         8/25/2022   4:05 PM           1144 Chart.yaml
-a----         8/25/2022   4:07 PM           1942 values.yaml
```

- step 2: configure file values.yaml

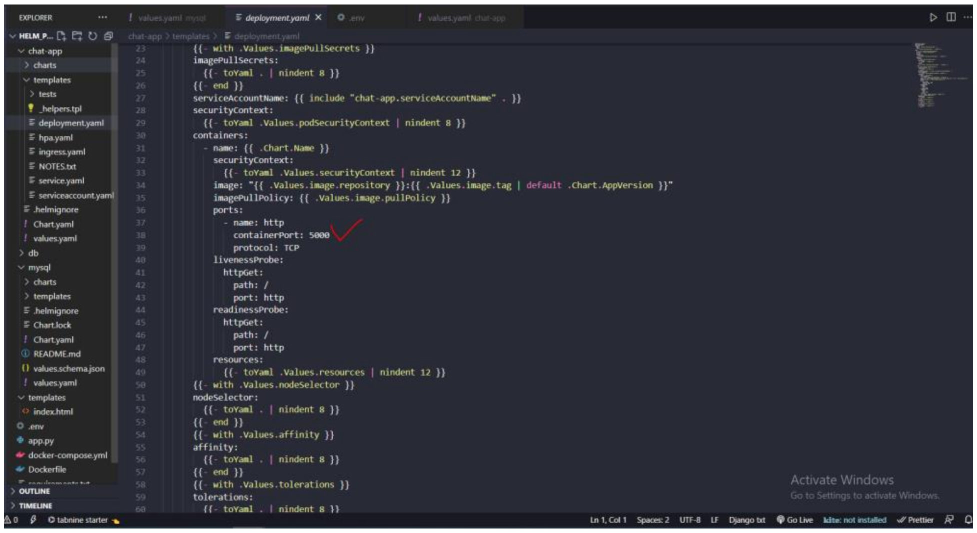


```
replicaCount: 1

image:
  repository: 912611140120.dkr.ecr.us-west-2.amazonaws.com/chat-app ✓
  pullPolicy: IfNotPresent
  # Overrides the image tag whose default is the chart appVersion.
  tag: "latest" ✓

imagePullSecrets: []
nameOverride: ""
fullnameOverride: ""

serviceAccount:
  # Specifies whether a service account should be created
  create: true
  # Annotations to add to the service account
  annotations: {}
  # The name of the service account to use.
  # If not set and create is true, a name is generated using the fullname template
  name: ""

podAnnotations: {}

podSecurityContext:
  {}
  # fsGroup: 2000

securityContext:
  {}
  # capabilities:
  #   drop:
  #   - ALL
  # readOnlyRootFilesystem: true
  # runAsNonRoot: true
  # runAsUser: 1000

service:
```



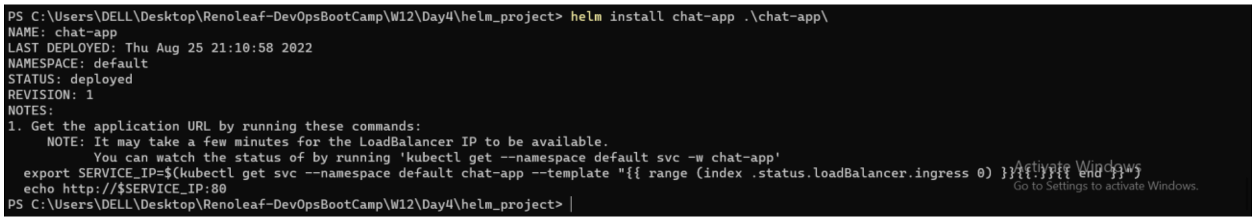
```

podAnnotations: {}

podSecurityContext:
  {}
  # fsGroup: 2000

securityContext:
  {}
  # capabilities:
  #   drop:
  #   - ALL
  # readOnlyRootFilesystem: true
  # runAsNonRoot: true
  # runAsUser: 1000

service:
  type: LoadBalancer ✓
  port: 80 ✓

ingress:
  enabled: false
  className: ""
  annotations:
    {}
    # kubernetes.io/ingress.class: nginx
    # kubernetes.io/tls-acme: "true"
  hosts:
    - host: chart-example.local
      paths:
        - path: /
          pathType: ImplementationSpecific
  tls: []
  #  - secretName: chart-example-tls
  #    hosts:
  #      - chart-example.local

```

Configure file deployment.yaml:

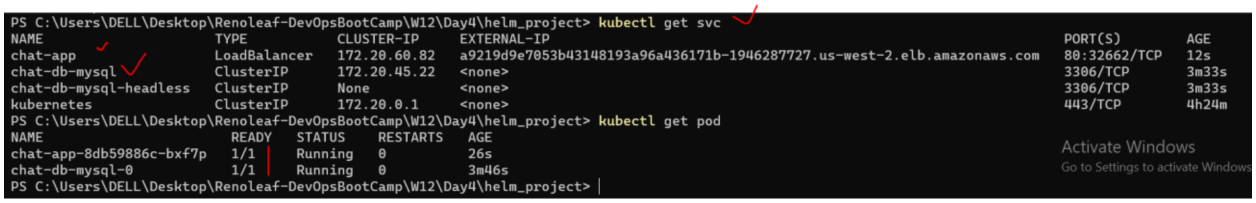


```yaml
      {{- with .Values.imagePullSecrets }}
      imagePullSecrets:
        {{- toYaml . | nindent 8 }}
      {{- end }}
      serviceAccountName: {{ include "chat-app.serviceAccountName" . }}
      securityContext:
        {{- toYaml .Values.podSecurityContext | nindent 8 }}
      containers:
        - name: {{ .Chart.Name }}
          securityContext:
            {{- toYaml .Values.securityContext | nindent 12 }}
          image: "{{ .Values.image.repository }}:{{ .Values.image.tag | default .Chart.AppVersion }}"
          imagePullPolicy: {{ .Values.image.pullPolicy }}
          ports:
            - name: http
              containerPort: 5000
              protocol: TCP
          livenessProbe:
            httpGet:
              path: /
              port: http
          readinessProbe:
            httpGet:
              path: /
              port: http
          resources:
            {{- toYaml .Values.resources | nindent 12 }}
      {{- with .Values.nodeSelector }}
      nodeSelector:
        {{- toYaml . | nindent 8 }}
      {{- end }}
      {{- with .Values.affinity }}
      affinity:
        {{- toYaml . | nindent 8 }}
      {{- end }}
      {{- with .Values.tolerations }}
      tolerations:
        {{- toYaml . | nindent 8 }}
```

- step 2: install chart chat-app

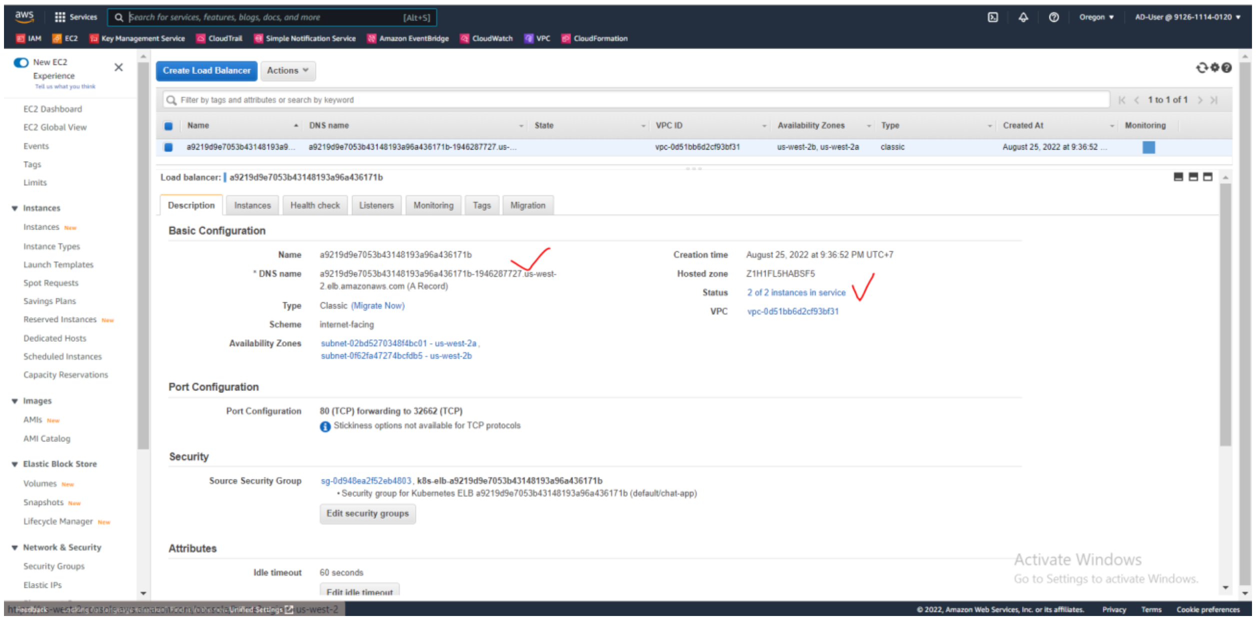


```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project> helm install chat-app .\chat-app\
NAME: chat-app
LAST DEPLOYED: Thu Aug 25 21:10:58 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
1. Get the application URL by running these commands:
     NOTE: It may take a few minutes for the LoadBalancer IP to be available.
           You can watch the status of by running 'kubectl get --namespace default svc -w chat-app'
  export SERVICE_IP=$(kubectl get svc --namespace default chat-app --template "{{ range (index .status.loadBalancer.ingress 0) }}{{.}}{{ end }}")
  echo http://$SERVICE_IP:80
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project>
```



```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project> kubectl get svc
NAME                   TYPE           CLUSTER-IP     EXTERNAL-IP                                                                PORT(S)        AGE
chat-app               LoadBalancer   172.20.60.82   a9219d9e7053b43148193a96a436171b-1946287727.us-west-2.elb.amazonaws.com   80:32662/TCP   12s
chat-db-mysql          ClusterIP      172.20.45.22   <none>                                                                     3306/TCP       3m33s
chat-db-mysql-headless ClusterIP      None           <none>                                                                     3306/TCP       3m33s
kubernetes             ClusterIP      172.20.0.1     <none>                                                                     443/TCP        4h24m
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project> kubectl get pod
NAME                       READY   STATUS    RESTARTS   AGE
chat-app-8db59886c-bxf7p   1/1     Running   0          26s
chat-db-mysql-0            1/1     Running   0          3m46s
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project>
```

Check load balancer



Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring
a9219d9e7053b43148193a9...	a9219d9e7053b43148193a96a436171b-1946287727.us-...		vpc-0d51bb6d2cf93bf31	us-west-2b, us-west-2a	classic	August 25, 2022 at 9:36:52	

Load balancer: a9219d9e7053b43148193a96a436171b

Basic Configuration

Name	a9219d9e7053b43148193a96a436171b	Creation time	August 25, 2022 at 9:36:52 PM UTC+7
* DNS name	a9219d9e7053b43148193a96a436171b-1946287727.us-west-2.elb.amazonaws.com (A Record)	Hosted zone	Z1H1FL5HABSF5
Type	Classic (Migrate Now)	Status	2 of 2 instances in service
Scheme	internet-facing	VPC	vpc-0d51bb6d2cf93bf31
Availability Zones	subnet-02bd5270348f4bc01 - us-west-2a, subnet-0f62fa47274bcfdb5 - us-west-2b		

Port Configuration

Port Configuration: 80 (TCP) forwarding to 32662 (TCP)
Stickiness options not available for TCP protocols

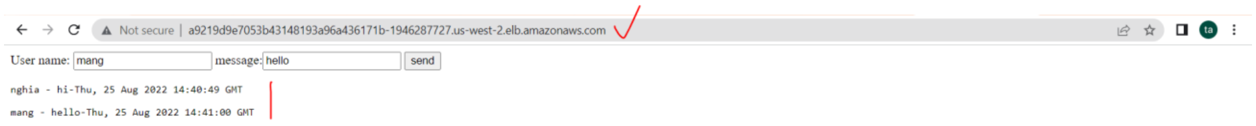
Security

Source Security Group: sg-0d948ea2f52eb4803, k8s-elb-a9219d9e7053b43148193a96a436171b
• Security group for Kubernetes ELB a9219d9e7053b43148193a96a436171b (default/chat-app)

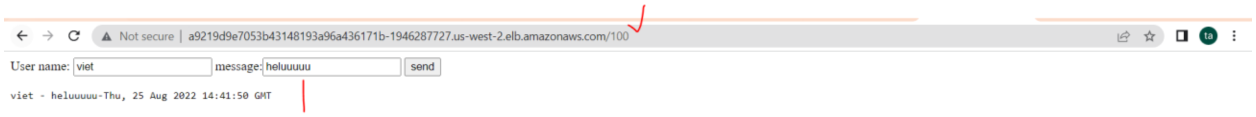
Attributes

Idle timeout: 60 seconds

Result:


Not secure | a9219d9e7053b43148193a96a436171b-1946287727.us-west-2.elb.amazonaws.com
User name:
message:
send

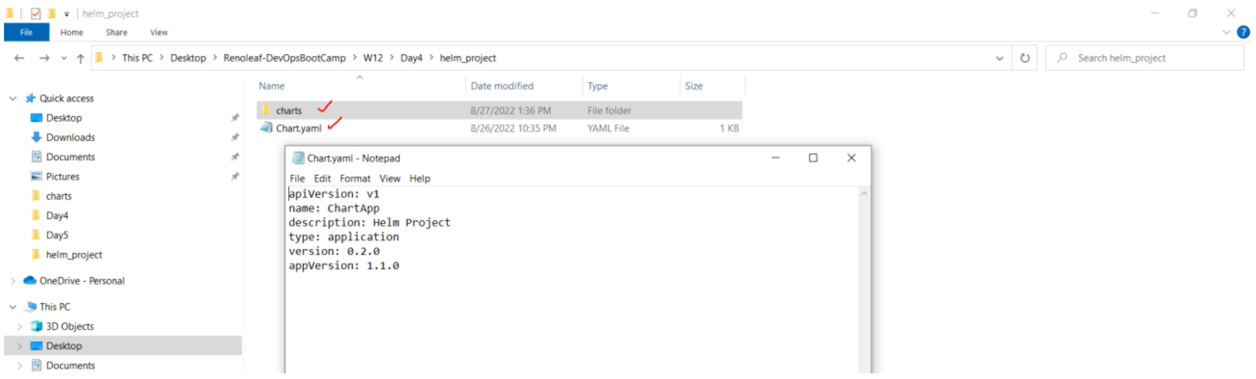
```
nghia - hi-Thu, 25 Aug 2022 14:40:49 GMT
mang - hello-Thu, 25 Aug 2022 14:41:00 GMT
```


Not secure | a9219d9e7053b43148193a96a436171b-1946287727.us-west-2.elb.amazonaws.com/100
User name:
message:
send

```
viet - heluuuuu-Thu, 25 Aug 2022 14:41:50 GMT
```

- 9. Packaging chart and pushing to the repository

Step 1:


This PC > Desktop > Renoleaf-DevOpsBootCamp > W12 > Day4 > helm_project
charts 8/27/2022 1:36 PM File folder
Chart.yaml 8/26/2022 10:35 PM YAML File 1 KB
Chart.yaml - Notepad

```
apiVersion: v1
name: ChartApp
description: Helm Project
type: application
version: 0.2.0
appVersion: 1.1.0
```

- Make folder chart and move all the project file to chart folder

- create Chart.yaml with content below:

apiVersion: v1

name: ChartApp

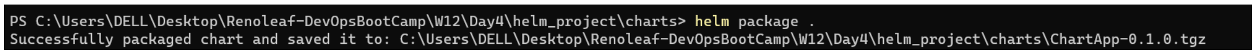
description: Helm Project

type: application

version: 0.2.0

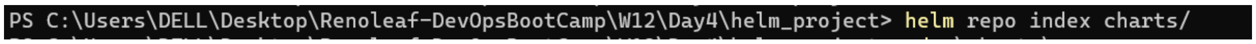
appVersion: 1.1.0

- Step 2: Go to folder chart and then package chart



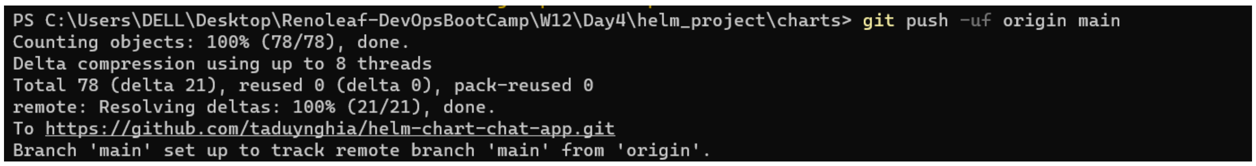
```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project\charts> helm package .
Successfully packaged chart and saved it to: C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project\charts\ChartApp-0.1.0.tgz
```

- Step 3: create index.yaml file

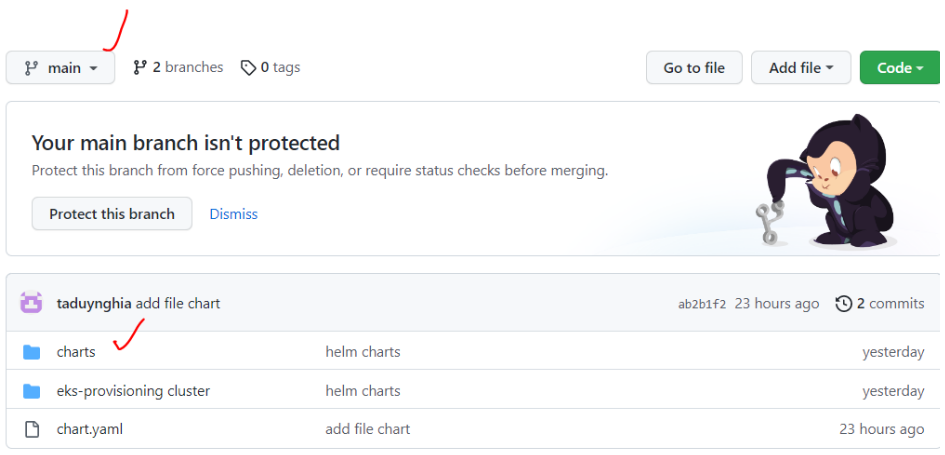


```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project> helm repo index charts/
```

- Step 4: Push project to git hub



```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project\charts> git push -uf origin main
Counting objects: 100% (78/78), done.
Delta compression using up to 8 threads
Total 78 (delta 21), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (21/21), done.
To https://github.com/taduynghia/helm-chart-chat-app.git
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

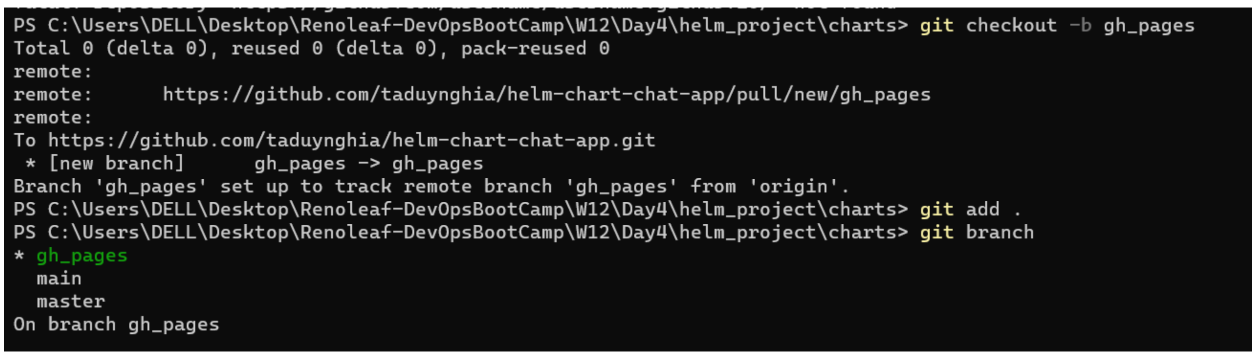


Your main branch isn't protected

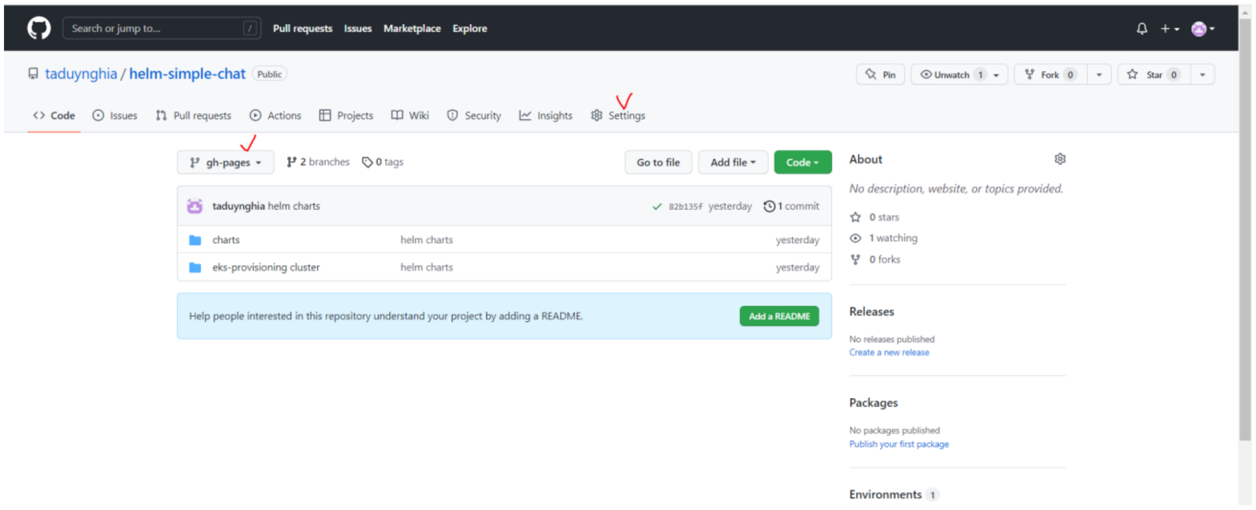
Protect this branch from force pushing, deletion, or require status checks before merging.

taduynghia add file chart		ab2b1f2 23 hours ago 2 commits
charts	helm charts	yesterday
eks-provisioning cluster	helm charts	yesterday
chart.yaml	add file chart	23 hours ago

- Step 5: Create branch gh_pages to get url config artifacthub.io



```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project\charts> git checkout -b gh_pages
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote:      https://github.com/taduynghia/helm-chart-chat-app/pull/new/gh_pages
remote:
To https://github.com/taduynghia/helm-chart-chat-app.git
 * [new branch]      gh_pages -> gh_pages
Branch 'gh_pages' set up to track remote branch 'gh_pages' from 'origin'.
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project\charts> git add .
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W12\Day4\helm_project\charts> git branch
* gh_pages
  main
  master
On branch gh_pages
```



taduynghia helm charts ✓ 82b135f yesterday 1 commit

charts	helm charts	yesterday
eks-provisioning cluster	helm charts	yesterday

Help people interested in this repository understand your project by adding a README. Add a README

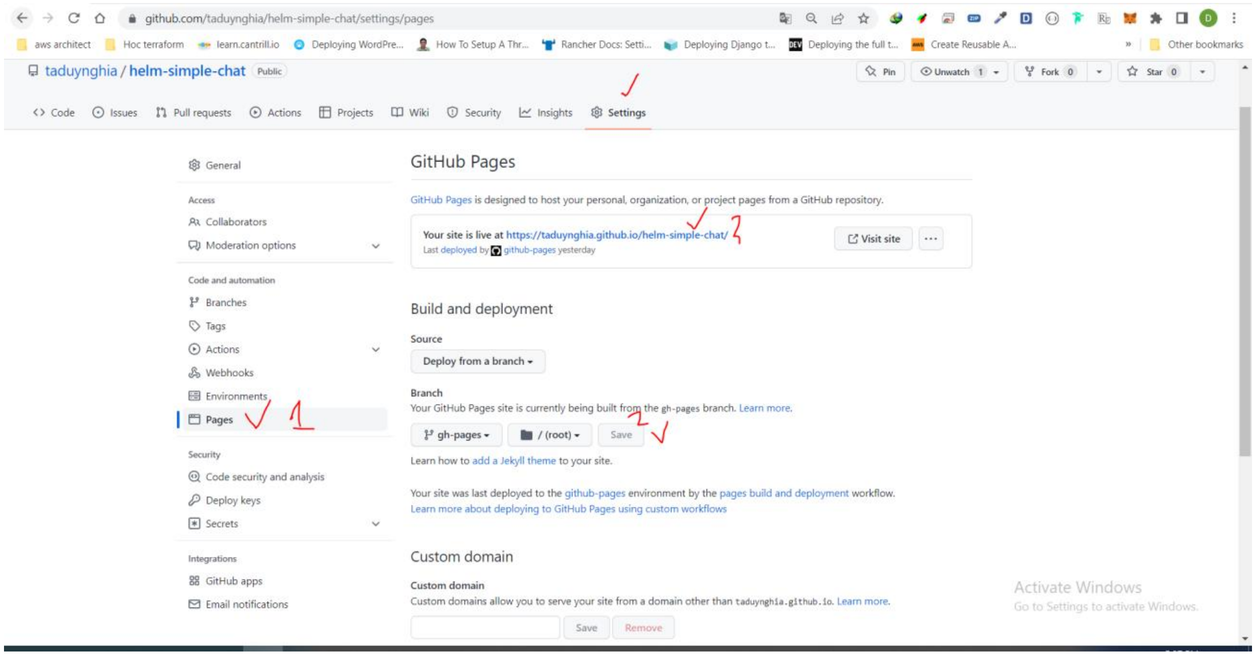
No description, website, or topics provided.

0 stars
1 watching
0 forks

Releases
No releases published
Create a new release

Packages
No packages published
Publish your first package

Environments 1



GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at https://taduynghia.github.io/helm-simple-chat/ Visit site
Last deployed by github-pages yesterday

Build and deployment

Source
Deploy from a branch

Branch
Your GitHub Pages site is currently being built from the gh-pages branch. Learn more.
gh-pages / (root) Save

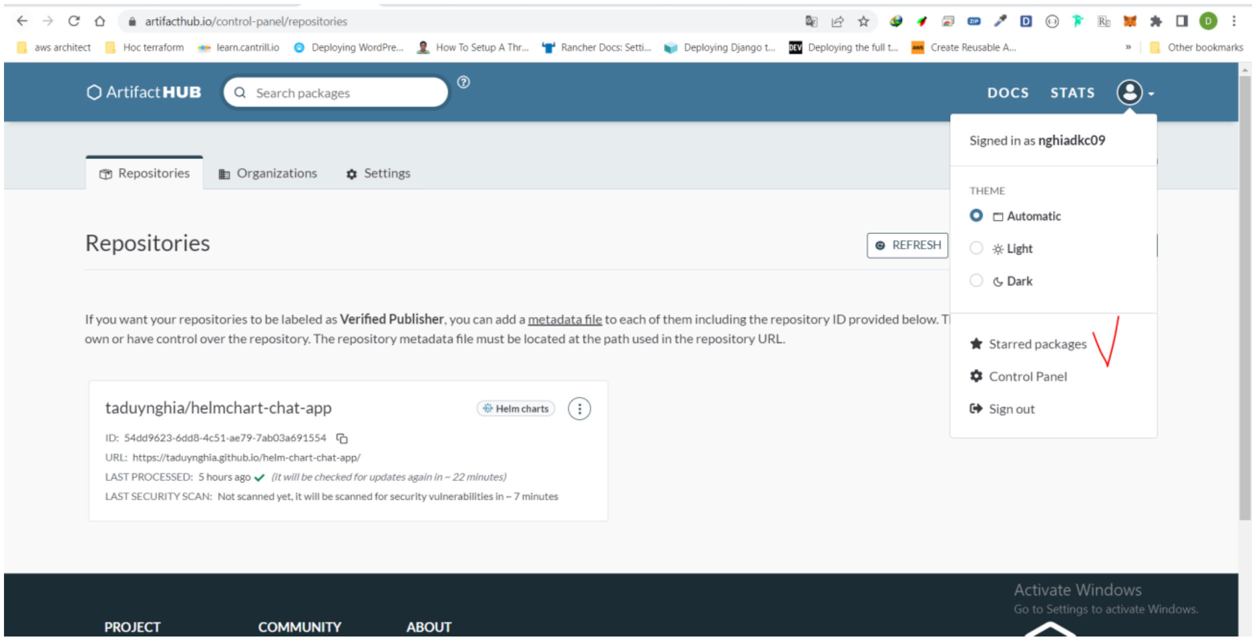
Learn how to add a Jekyll theme to your site.

Your site was last deployed to the github-pages environment by the pages build and deployment workflow.
Learn more about deploying to GitHub Pages using custom workflows

Custom domain

Custom domain
Custom domains allow you to serve your site from a domain other than taduynghia.github.io. Learn more.
Save Remove

- step 6: config aritfactoryhub.io

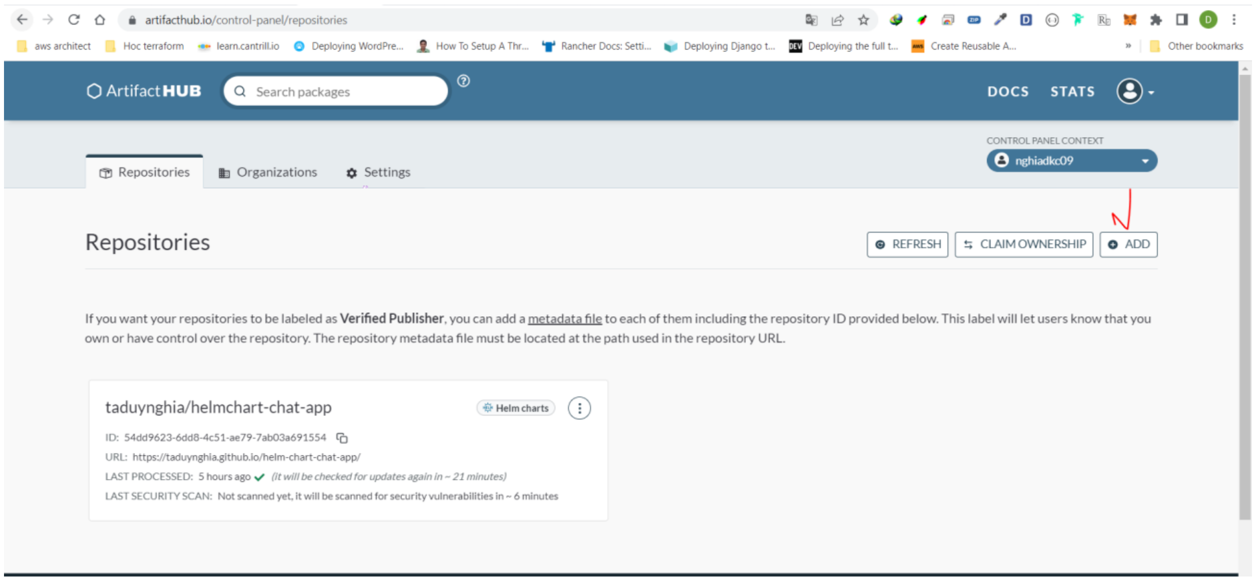


Repositories

If you want your repositories to be labeled as **Verified Publisher**, you can add a metadata file to each of them including the repository ID provided below. T own or have control over the repository. The repository metadata file must be located at the path used in the repository URL.

taduynghia/helmchart-chat-app

ID: 54dd9623-6dd8-4c51-ae79-7ab03a691554
URL: https://taduynghia.github.io/helm-chart-chat-app/
LAST PROCESSED: 5 hours ago ✓ *(it will be checked for updates again in ~ 22 minutes)*
LAST SECURITY SCAN: Not scanned yet, it will be scanned for security vulnerabilities in ~ 7 minutes

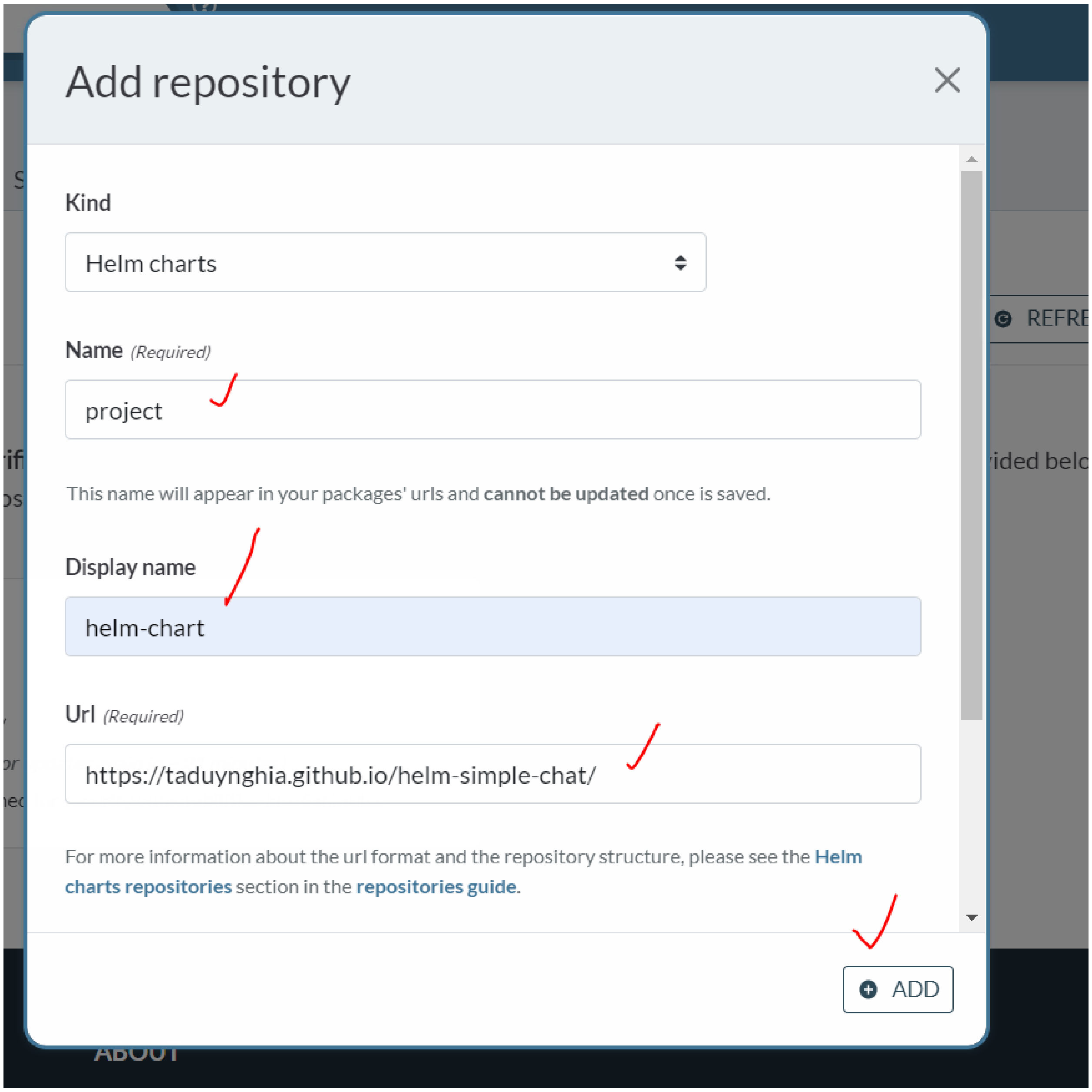


Repositories

If you want your repositories to be labeled as **Verified Publisher**, you can add a metadata file to each of them including the repository ID provided below. This label will let users know that you own or have control over the repository. The repository metadata file must be located at the path used in the repository URL.

taduynghia/helmchart-chat-app

ID: 54dd9623-6dd8-4c51-ae79-7ab03a691554
URL: https://taduynghia.github.io/helm-chart-chat-app/
LAST PROCESSED: 5 hours ago ✓ *(it will be checked for updates again in ~ 21 minutes)*
LAST SECURITY SCAN: Not scanned yet, it will be scanned for security vulnerabilities in ~ 6 minutes



Add repository

Kind

Helm charts

Name *(Required)*

project

This name will appear in your packages' urls and **cannot be updated** once is saved.

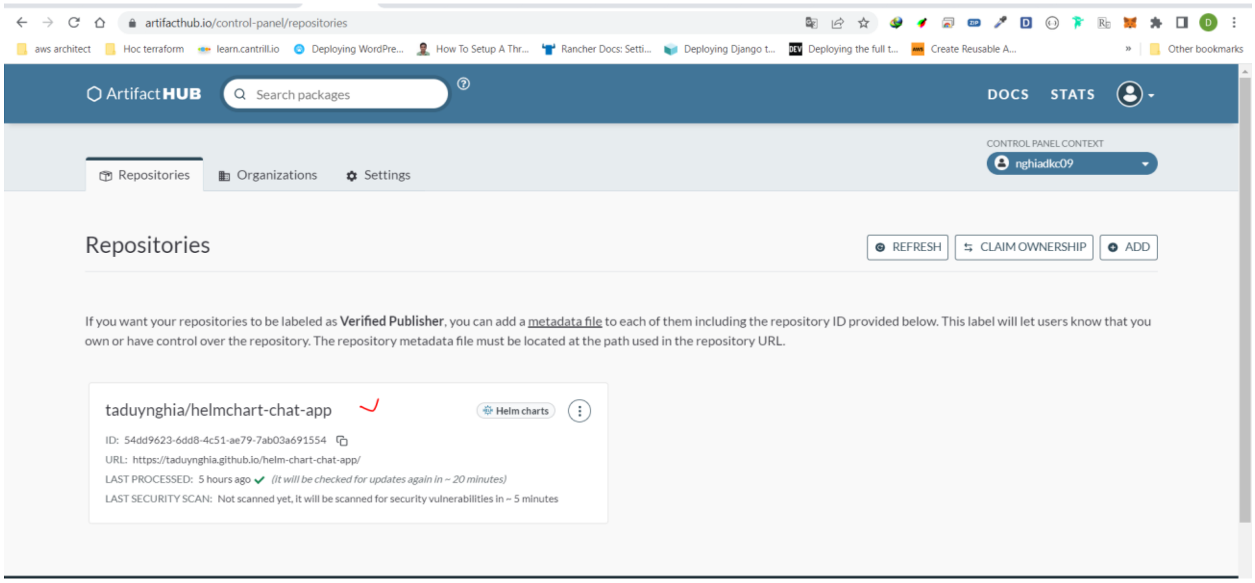
Display name

helm-chart

Url *(Required)*

https://taduynghia.github.io/helm-simple-chat/

For more information about the url format and the repository structure, please see the **Helm charts repositories** section in the **repositories guide**.



artifacthub.io/control-panel/repositories

Repositories

If you want your repositories to be labeled as **Verified Publisher**, you can add a metadata file to each of them including the repository ID provided below. This label will let users know that you own or have control over the repository. The repository metadata file must be located at the path used in the repository URL.

taduynghia/helmchart-chat-app

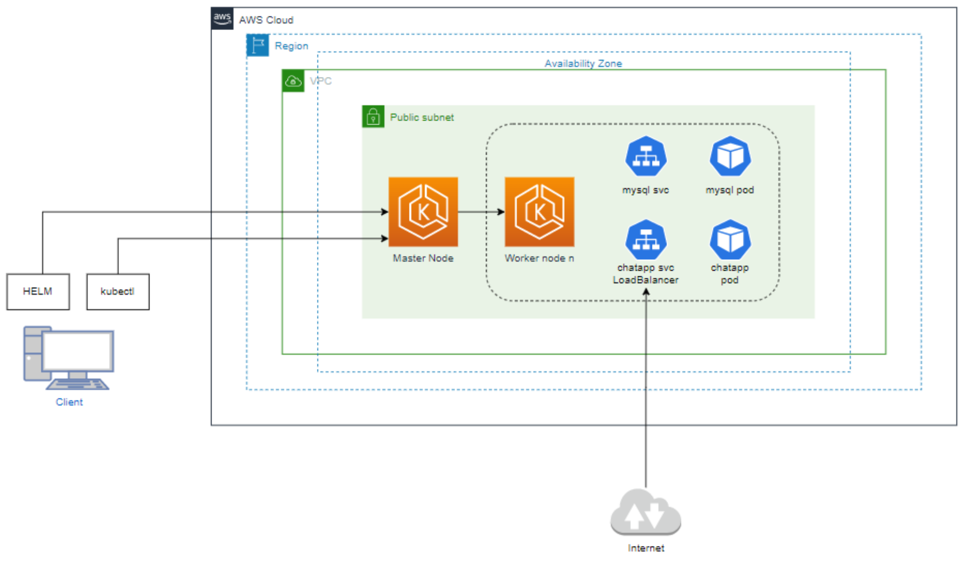
Helm charts

ID: 54dd9623-6dd8-4c51-ae79-7ab03a691554

URL: https://taduynghia.github.io/helm-chart-chat-app/

LAST PROCESSED: 5 hours ago ✓ *(it will be checked for updates again in ~ 20 minutes)*

LAST SECURITY SCAN: Not scanned yet, it will be scanned for security vulnerabilities in ~ 5 minutes


AWS Cloud
Region
Availability Zone
VPC
Public subnet
Master Node
Worker node n
mysql svc
mysql pod
chatapp svc LoadBalancer
chatapp pod
HELM
kubectl
Client
Internet