Final Project 最终项目

Skills Network

Objectives 目标

In this lab, you will:
在这个实验中，你将：

- Build and deploy a simple Guestbook application
  构建并部署一个简单的留言簿应用程序
- Autoscale the Guestbook application using Horizontal Pod Autoscaler
  使用水平 Pod 自动缩放器对留言簿应用程序进行自动缩放
- Perform Rolling Updates and Rollbacks
  执行滚动更新和回滚

Project Overview 项目概述

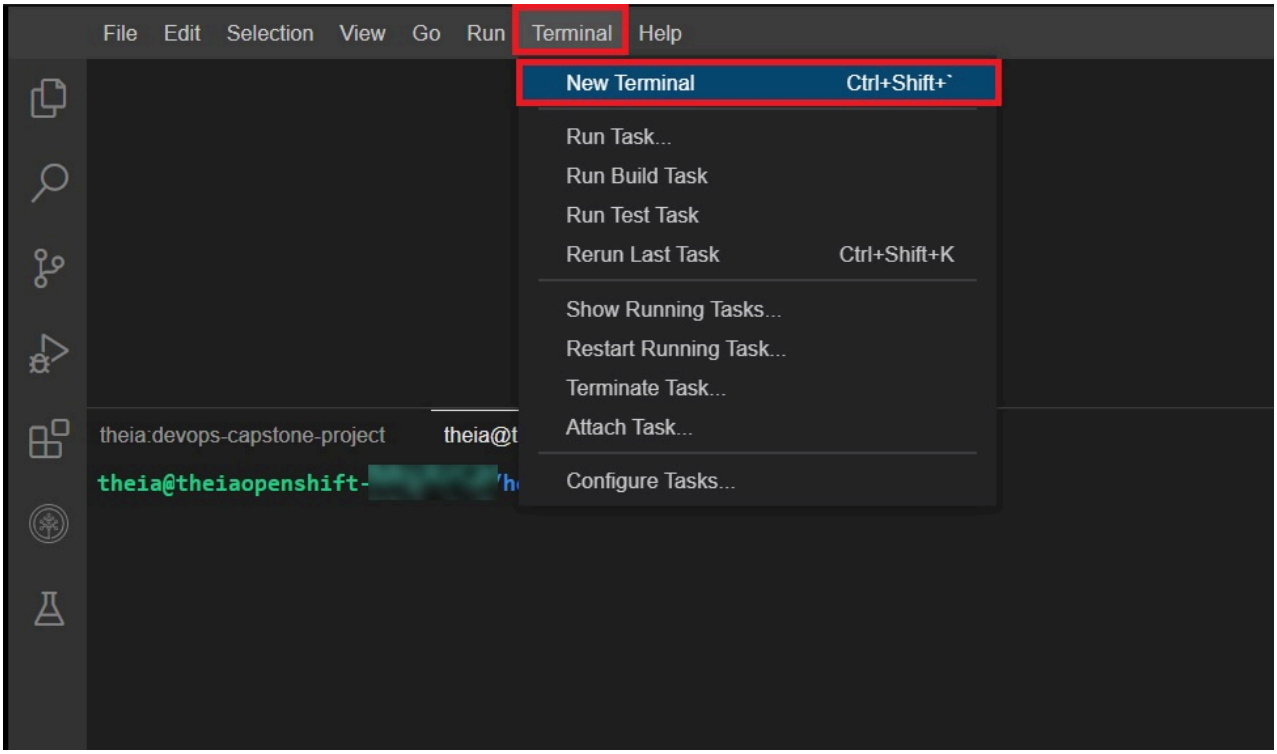
Guestbook application 留言簿应用程序

Guestbook is a simple web application that we will build and deploy with Docker and Kubernetes. The application consists of a web front end which will have a text input where you can enter any text and submit. For all of these we will create Kubernetes Deployments and Pods. Then we will apply Horizontal Pod Scaling to the Guestbook application and finally work on Rolling Updates and Rollbacks.
留言簿是一个简单的 Web 应用程序，我们将使用 Docker 和 Kubernetes 来构建和部署。该应用程序由一个 Web 前端组成，其中包含一个文本输入框，您可以输入任何文本并提交。对于所有这些，我们将创建 Kubernetes 部署和 Pod。然后我们将对留言簿应用程序应用水平 Pod 扩展，最后我们将处理滚动更新和回滚。

Verify the environment and command line tools
验证环境和命令行工具

1. If a terminal is not already open, open a terminal window by using the menu in the editor: `Terminal > New Terminal`.
   如果终端尚未打开，请通过编辑器中的菜单打开一个终端窗口： `Terminal > New Terminal` 。

   Note: Please wait for some time for the terminal prompt to appear.
   注意：请等待一段时间，以便终端提示符出现。



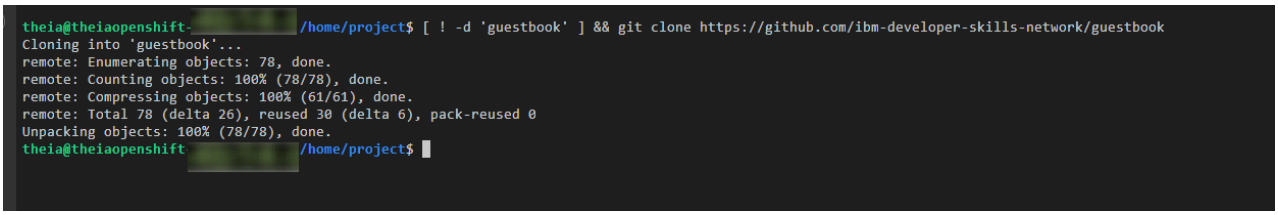
2. Change to your project folder.
   切换到你的项目文件夹。

   Note: If you are already on the `/home/project` folder, please skip this step.
   注意：如果您已经在 `/home/project` 文件夹中，请跳过此步骤。

```
cd /home/project
```

3. Clone the git repository that contains the artifacts needed for this lab.
克隆包含本实验所需工件的 git 仓库。

```
[ ! -d 'guestbook' ] && git clone https://github.com/ibm-developer-skills-network/guestbook
```



4. Change to the directory for this lab.
切换到本实验的目录。

```
cd guestbook
```

5. List the contents of this directory to see the artifacts for this lab.
列出此目录的内容，以查看本实验的工件。

```
ls
```

Build the guestbook app
构建留言簿应用程序

To begin, we will build and deploy the web front end for the guestbook app.
首先，我们将构建并部署留言簿应用的 Web 前端。

1. Change to the `v1/guestbook` directory.
切换到 `v1/guestbook` 目录。

```
cd v1/guestbook
```

2. Dockerfile incorporates a more advanced strategy called multi-stage builds, so feel free to read more about that here.
Dockerfile 采用了更高级的策略，称为多阶段构建，您可以在此处了解更多信息。

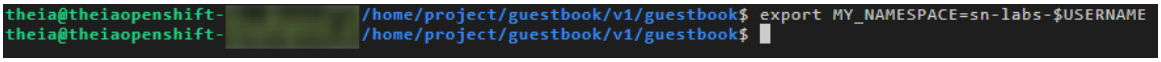
Complete the Dockerfile with the necessary Docker commands to build and push your image. The path to this file is `guestbook/v1/guestbook/Dockerfile`.
使用必要的 Docker 命令完成 Dockerfile，以构建和推送您的镜像。该文件的路径是 `guestbook/v1/guestbook/Dockerfile` 。

▼ Hint! 提示!
The FROM instruction initializes a new build stage and specifies the base image that subsequent instructions will build upon.
FROM 指令初始化一个新的构建阶段，并指定后续指令将基于的基础镜像。
The COPY command enables us to copy files to our image.
The ADD command is used to copy files/directories into a Docker image.
The RUN instruction executes commands.
COPY 命令使我们能够将文件复制到镜像中。ADD 命令用于将文件/目录复制到 Docker 镜像中。RUN 指令执行命令。
The EXPOSE instruction exposes a particular port with a specified protocol inside a Docker Container.
EXPOSE 指令在 Docker 容器中暴露指定协议的特定端口。
The CMD instruction provides a default for executing a container, or in other words, an executable that should run in your container.
CMD 指令为执行容器提供默认设置，换句话说，即为应在您的容器中运行的执行文件提供默认设置。

Take a screenshot of the completed dockerfile and save it as **Dockerfile.png**.
将完成的 Dockerfile 截图并保存为 Dockerfile.png。

3. Export your namespace as an environment variable so that it can be used in subsequent commands.
将您的命名空间导出为环境变量，以便在后续命令中使用。

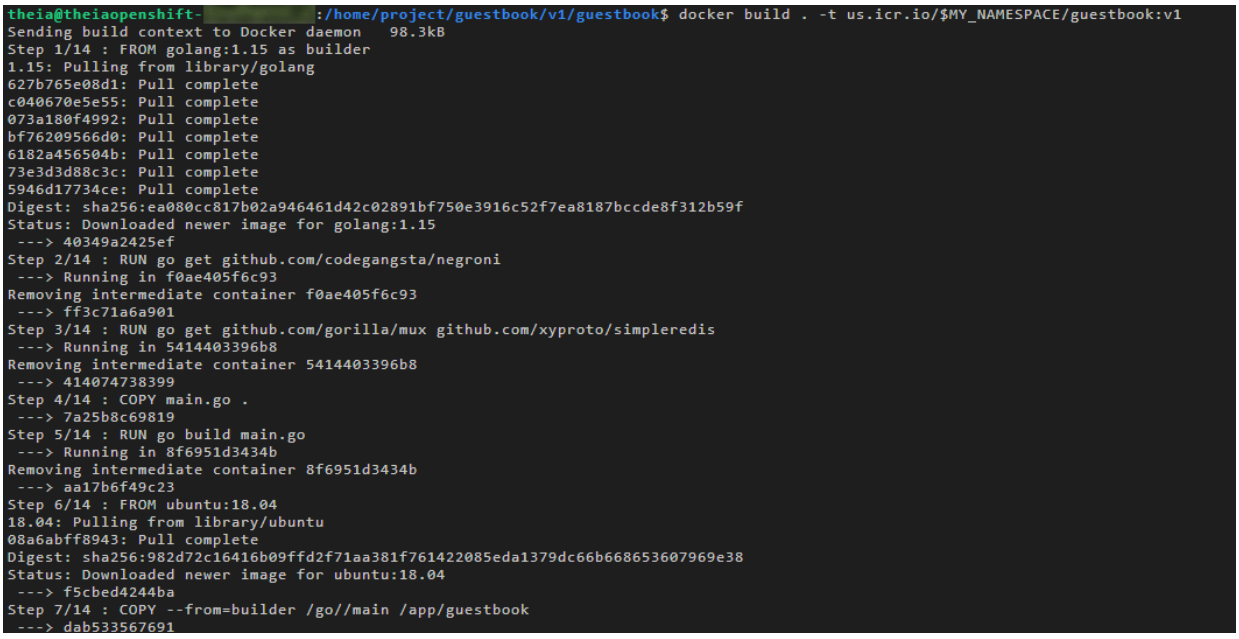
```
export MY_NAMESPACE=sn-labs-$USERNAME
```



4. Build the guestbook app using the Docker Build command.
使用 Docker Build 命令构建 guestbook 应用程序。

▼ Hint! 提示!

```
docker build . -t us.icr.io/$MY_NAMESPACE/guestbook:v1
```



5. Push the image to IBM Cloud Container Registry.
将图像推送到 IBM Cloud 容器注册中心。

▼ Hint! 提示!

```
docker push us.icr.io/$MY_NAMESPACE/guestbook:v1
```

```
theia@theiaopenshift-                :/home/project/guestbook/v1/guestbook$ docker push us.icr.io/$MY_NAMESPACE/guestbook:v1
The push refers to repository [us.icr.io/sn-labs-          /guestbook]
0791a8c653f4: Pushed
fd382c0f22b7: Pushed
aeb42d7f3c54: Pushed
54deed58a6a4: Pushed
5dddb4b5996f: Pushed
95c443da13bf: Pushed
v1: digest: sha256:459dc3814b3b0d59aee57f684db436c69abfa52a492fbed84e75c6c42188fe40 size: 1570
theia@theiaopenshift-                :/home/project/guestbook/v1/guestbook$
```

Note: If you have tried this lab earlier, there might be a possibility that the previous session is still persistent. In such a case, you will see a **'Layer already Exists'** message instead of the **'Pushed'** message in the above output. We recommend you to proceed with the next steps of the lab.
注意：如果您之前已经尝试过这个实验，那么之前会话可能仍然存在。在这种情况下，您在上面的输出中看到的将是"层已存在"消息，而不是"已推送"消息。我们建议您继续进行实验的下一步。

6. Verify that the image was pushed successfully.
验证图像是否已成功推送。

```
ibmcloud cr images
```

```
theia@theiaopenshift-lavanyar: /home/project/guestbook/v1/guestbook ×

theia@theiaopenshift-          /home/project/guestbook/v1/guestbook$ ibmcloud cr images
Listing images...

Repository                                        Tag      Digest         Namespace          Created          Size     Security status
us.icr.io/sn-labs-          /guestbook            v1       61e06d36213c   sn-labs-           21 minutes ago   32 MB    -
us.icr.io/sn-labsassets/instructions-splitter     latest   2af122cfe4ee   sn-labsassets      2 years ago      21 MB    -
us.icr.io/sn-labsassets/pgadmin-theia             latest   0adf67ad81a3   sn-labsassets      2 years ago      101 MB   -
us.icr.io/sn-labsassets/phpmyadmin                latest   b66c30786353   sn-labsassets      2 years ago      163 MB   -
us.icr.io/sn-labsassets/tts-standalone            latest   56ce85280714   sn-labsassets      1 week ago       3.7 GB   -

OK
theia@theiaopenshift-          /home/project/guestbook/v1/guestbook$
```

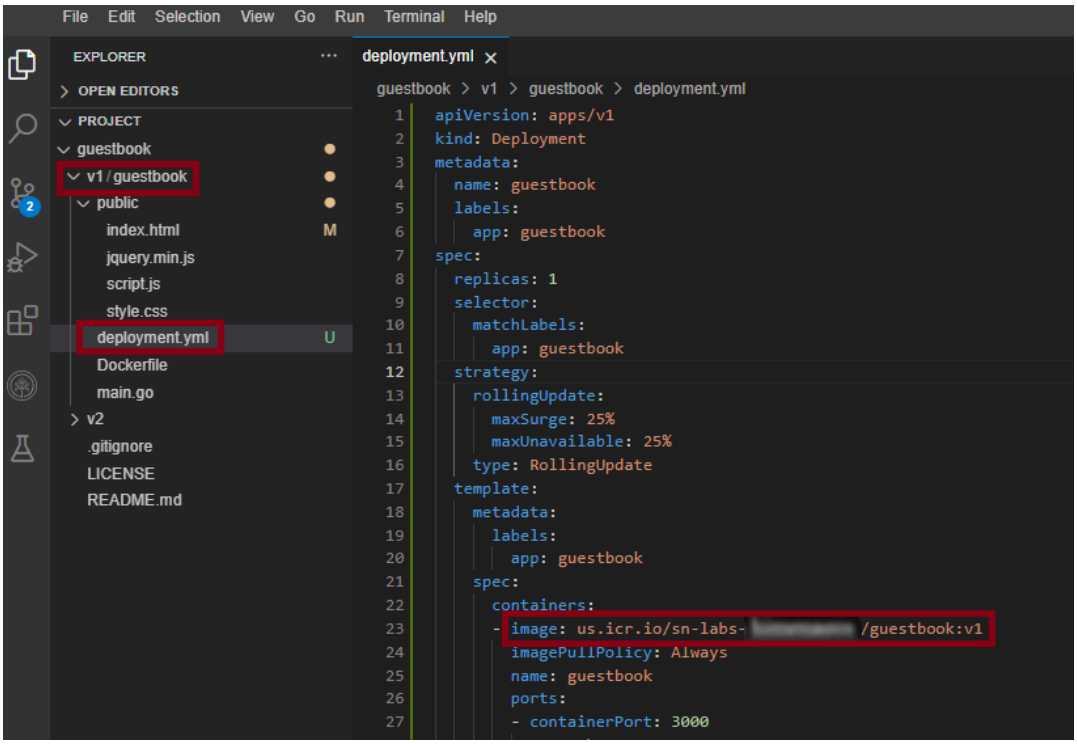
Take a screenshot of the output of Step 6 and save it as a .jpg or .png with the filename `crimages.png`. You will be prompted to upload the screenshot in the Peer Assignment.
对步骤 6 的输出进行截图，并将其保存为.jpg 或.png 格式，文件名使用 `crimages.png` 。您将在同行作业中收到上传截图的提示。

7. Open the `deployment.yml` file in the `v1/guestbook` directory & view the code for the deployment of the application:
在 `v1/guestbook` 目录中打开 `deployment.yml` 文件，查看应用程序部署的代码：

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: guestbook
  labels:
    app: guestbook
spec:
  replicas: 1
  selector:
    matchLabels:
      app: guestbook
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: guestbook
    spec:
      containers:
      - image: us.icr.io/<your sn labs namespace>/guestbook:v1
        imagePullPolicy: Always
        name: guestbook
        ports:
        - containerPort: 3000
          name: http
        resources:
          limits:
            cpu: 50m
          requests:
            cpu: 20m
```

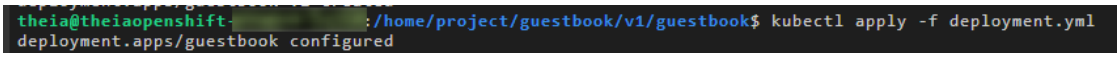
Note: Replace `<your sn labs namespace>` with your SN labs namespace. To check your SN labs namespace, please run the command `ibmcloud cr namespaces`
注意：将 `<your sn labs namespace>` 替换为您的 SN 实验室命名空间。要检查您的 SN 实验室命名空间，请运行命令 `ibmcloud cr namespaces`

- It should look as below:
  它应该如下所示：



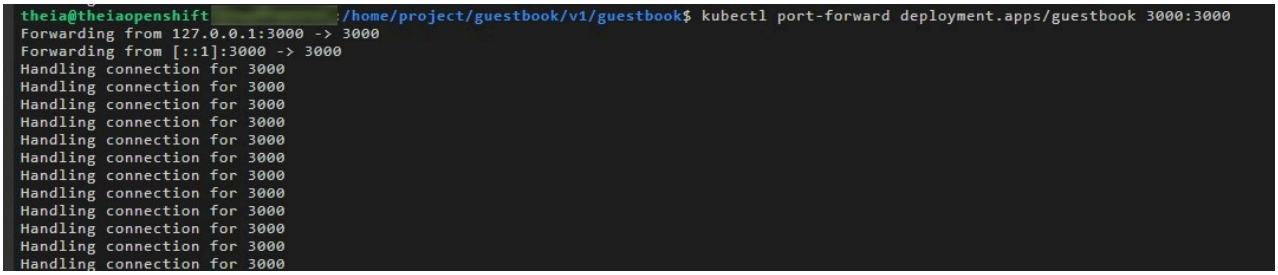
8. Apply the deployment using:
   使用以下方式部署：

   ```
   kubectl apply -f deployment.yml
   ```

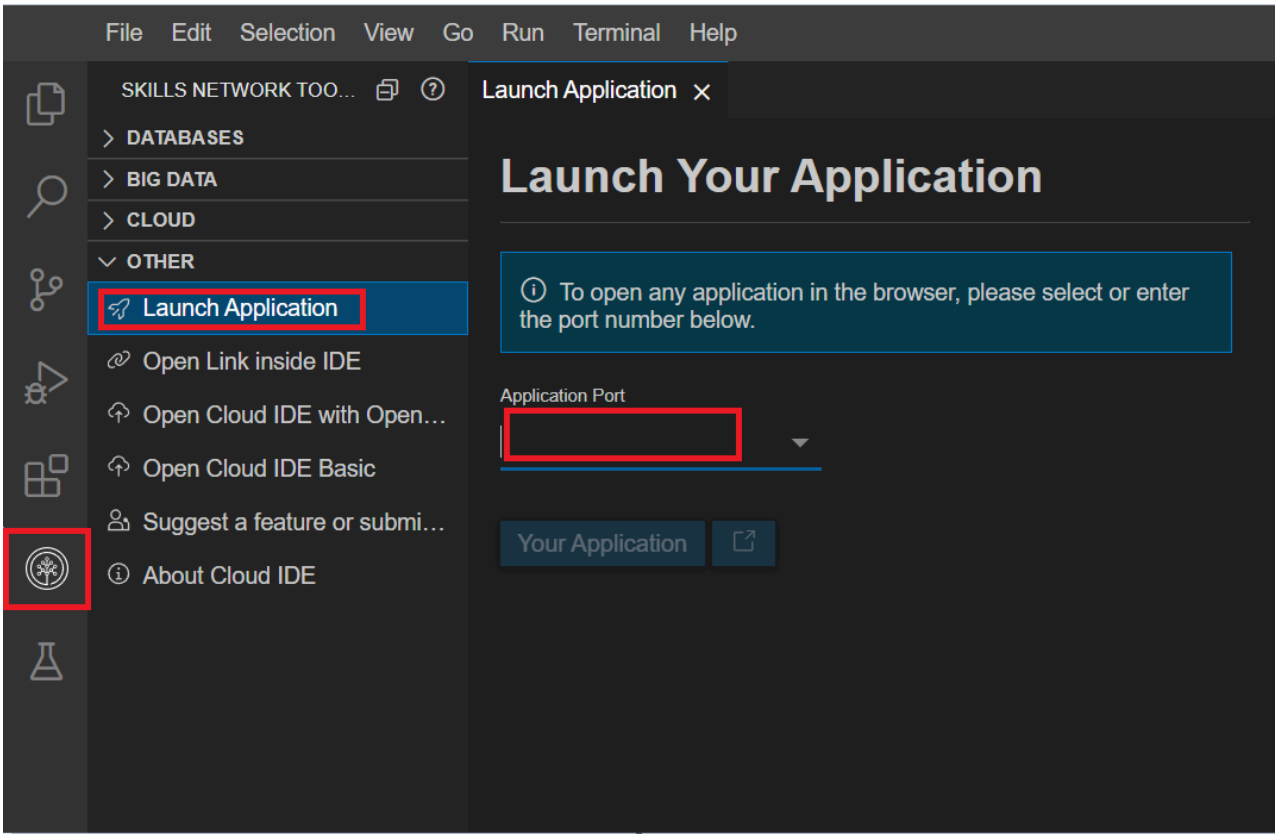


9. Open a New Terminal and enter the below command to view your application:
   打开一个新的终端，输入以下命令查看您的应用程序：

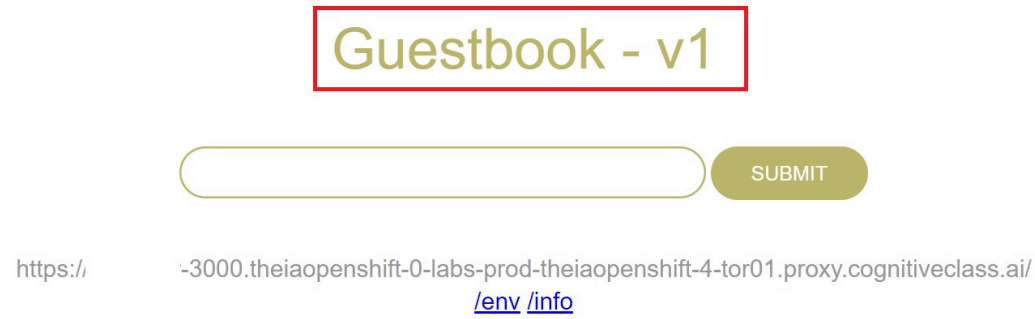
   ```
   kubectl port-forward deployment.apps/guestbook 3000:3000
   ```



10. Launch your application on port 3000. Click on the Skills Network button on the right, it will open the **"Skills Network Toolbox"**. Then click the **Other** then **Launch Application**. From there you should be able to enter the port and launch.
    在端口 3000 上启动您的应用程序。点击右侧的"Skills Network"按钮，它将打开"Skills Network Toolbox"。然后点击"其他"，再点击"启动应用程序"。从那里您应该能够输入端口并启动。



11. Now you should be able to see your running application. Please copy the app URL which will be given as below:
现在你应该能够看到你的运行应用程序了。请复制下面提供的应用程序 URL：



Take a screenshot of your deployed application and save it as a .jpg or .png with the filename `app.png`. You will be prompted to upload the screenshot in the Peer Assignment.
对你的部署应用程序进行截图，并将其保存为.jpg 或.png 格式，文件名使用 `app.png` 。在同行作业中，系统会提示你上传截图。

12. Try out the guestbook by putting in a few entries. You should see them appear above the input box after you hit **Submit**.
尝试通过输入一些条目来使用留言簿。当你点击提交后，你应该会看到它们出现在输入框上方。

Autoscale the Guestbook application using Horizontal Pod Autoscaler
使用水平 Pod 自动缩放器自动缩放 Guestbook 应用程序

1. Autoscale the Guestbook deployment using `kubectl autoscale deployment`
使用 `kubectl autoscale deployment` 自动扩展留言簿部署

▼ Hint! 提示!

```
kubectl autoscale deployment guestbook --cpu-percent=5 --min=1 --max=10
```

```
theia@theiaopenshift-[redacted]:/home/project$ kubectl autoscale deployment guestbook --cpu-percent=5 --min=1 --max=10
horizontalpodautoscaler.autoscaling/guestbook autoscaled
```

2. You can check the current status of the newly-made HorizontalPodAutoscaler, by running:
   您可以通过运行以下命令检查新创建的水平 Pod 自动缩放器的当前状态：

   ```
   kubectl get hpa guestbook
   ```

The current replicas is 0 as there is no load on the server.
由于服务器上没有负载，当前副本数为 0。

   Take a screenshot of your Horizontal Pod Autoscaler and save it as a .jpg or .png with the filename `hpa.png`. You will be prompted to upload the screenshot in the Peer Assignment.
   对你的水平 Pod 自动缩放器进行截图，并将其保存为 .jpg 或 .png 格式，文件名命名为 `hpa.png` 。你将在同行作业中收到提示上传截图。

3. Open another new terminal and enter the below command to generate load on the app to observe the autoscaling (Please ensure your port-forward command is running. In case you have stopped your application, please run the port-forward command to re-run the application at port 3000.)
   打开另一个新的终端，输入以下命令以生成应用程序负载以观察自动缩放（请确保你的端口转发命令正在运行。如果你已经停止了你的应用程序，请运行端口转发命令以在端口 3000 上重新运行应用程序。）

   ```
   kubectl run -i --tty load-generator --rm --image=busybox:1.36.0 --restart=Never -- /bin/sh -c "while sleep 0.01; do wget -q -O- <your app URL>; done"
   ```

- Please replace your app URL in the `<your app URL>` part of the above command.
  请在上面的命令中的 `<your app URL>` 部分替换你的应用程序 URL。

  Note: Use the same copied URL which you obtained in step 11 of the previous task.
  注意：使用你在上一个任务的第 11 步中获得的相同 URL。

The command will be as below:
命令如下：

```
theia@theiaopenshift-[redacted]:/home/project$ kubectl run -i --tty load-generator-7 --rm --image=busybox:1.28 --restart=Never -- /bin/sh -c "while sleep 0.01; do wget -q -O- https://[redacted]300
0.theiaopenshift-2-labs-prod-theiaopenshift-4-tor01.proxy.cognitiveclass.ai/; done"
```

  Note: In case you get a `Load generator already exists` error, please suffix a number after `load-generator` eg. `load-generator-1`, `load-generator-2`.
  注意：如果出现 `Load generator already exists` 错误，请在 `load-generator` 后添加数字，例如 `load-generator-1` ， `load-generator-2` 。

- You will keep getting an output similar as below which will indicate the increasing load on the app:
  您将不断得到类似以下的输出，这将表明应用程序负载正在增加：

```
    </div>

    <div id="guestbook-entries">
      <link href="https://afeld.github.io/emoji-css/emoji.css" rel="stylesheet">
      <p>Waiting for database connection... <i class='em em-boat'></i></p>

    </div>

    <div>
      <form id="guestbook-form">
        <input autocomplete="off" id="guestbook-entry-content" type="text">
        <a href="#" id="guestbook-submit">Submit</a>
      </form>
    </div>

    <div>
      <p><h2 id="guestbook-host-address"></h2></p>
      <p><a href="env">/env</a>
      <a href="info">/info</a></p>
    </div>
    <script src="jquery.min.js"></script>
    <script src="script.js"></script>
  </body>
</html>
```

  Note: Continue further commands in the 1st terminal
  注意：在第一个终端中继续执行后续命令。

4. Run the below command to observe the replicas increase in accordance with the autoscaling:
   运行以下命令以观察副本根据自动扩展增加：

   ```
   kubectl get hpa guestbook --watch
   ```

```
theia@theiaopenshift-                :/home/project$ kubectl get hpa guestbook --watch
NAME        REFERENCE                TARGETS         MINPODS   MAXPODS   REPLICAS   AGE
guestbook   Deployment/guestbook     <unknown>/5%    1         10        0          51s
```

5. Run the above command again after 5–10 minutes and you will see an increase in the number of replicas which shows that your application has been autoscaled.
   5–10 分钟后再次运行上述命令，您将看到副本数量增加，这表明您的应用程序已自动扩展。

```
^Ctheia@theiaopenshift-                :/home/project$ kubectl get hpa guestbook --watch
NAME        REFERENCE                TARGETS         MINPODS   MAXPODS   REPLICAS   AGE
guestbook   Deployment/guestbook     <unknown>/5%    1         10        0          2m54s
guestbook   Deployment/guestbook     <unknown>/5%    1         10        1          3m31s
```

   Take a screenshot of your Autoscaler details and save it as a .jpg or .png with the filename `hpa2.png`. You will be prompted to upload the screenshot in the Peer Assignment.
   对您的自动扩展器详细信息进行截图，并将其保存为.jpg 或.png 格式，文件名以 `hpa2.png` 命名。您将在同行评估中收到上传截图的提示。

6. Run the below command to observe the details of the horizontal pod autoscaler:
   运行以下命令以观察水平 Pod 自动扩展器的详细信息：

   ```
   kubectl get hpa guestbook
   ```

```
^Ctheia@theiaopenshift-                :/home/project$ kubectl get hpa guestbook
NAME        REFERENCE                TARGETS         MINPODS   MAXPODS   REPLICAS   AGE
guestbook   Deployment/guestbook     <unknown>/5%    1         10        1          8m2s
```

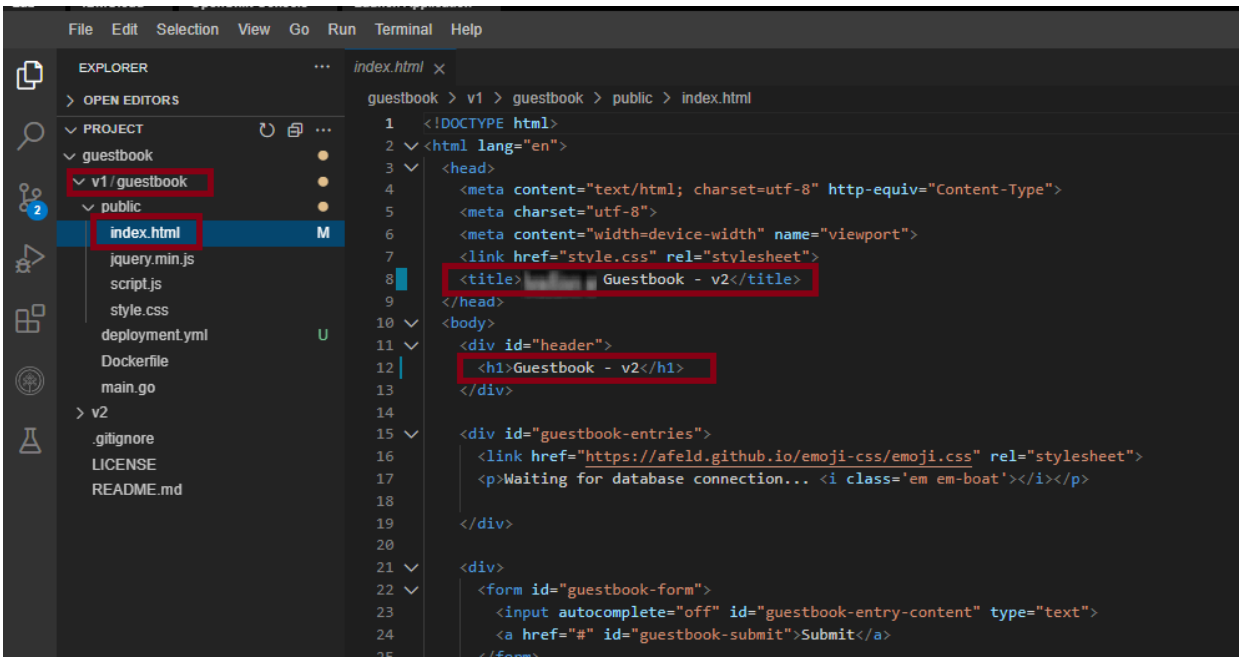
- Please close the other terminals where load generator and port-forward commands are running.
  请关闭运行负载生成器和端口转发命令的其他终端。

Perform Rolling Updates and Rollbacks on the Guestbook application
对留言簿应用程序执行滚动更新和回滚

Note: Please run all the commands in the 1st terminal unless mentioned to use a new terminal.
注意：除非特别说明使用新终端，否则请将所有命令在第一个终端中运行。

1. Please update the title and header in `index.html` to any other suitable title and header like **<Your name> Guestbook – v2** & **Guestbook – v2**.
   请将 `index.html` 中的标题和页眉更新为其他合适的标题和页眉，例如<您的名字>留言簿 – v2 & 留言簿 – v2。

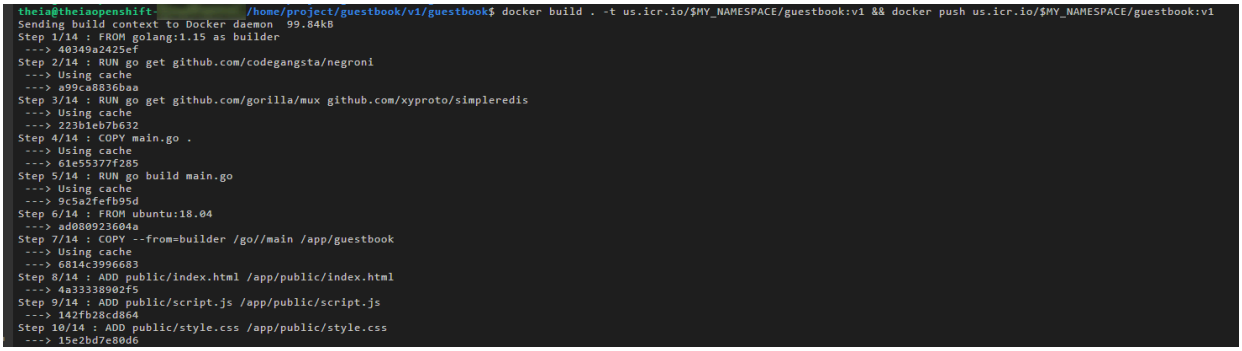
▼ Hint! 提示!



2. Run the below command to build and push your updated app image:
   运行以下命令来构建并推送您更新的应用镜像：

▼ Hint! 提示!

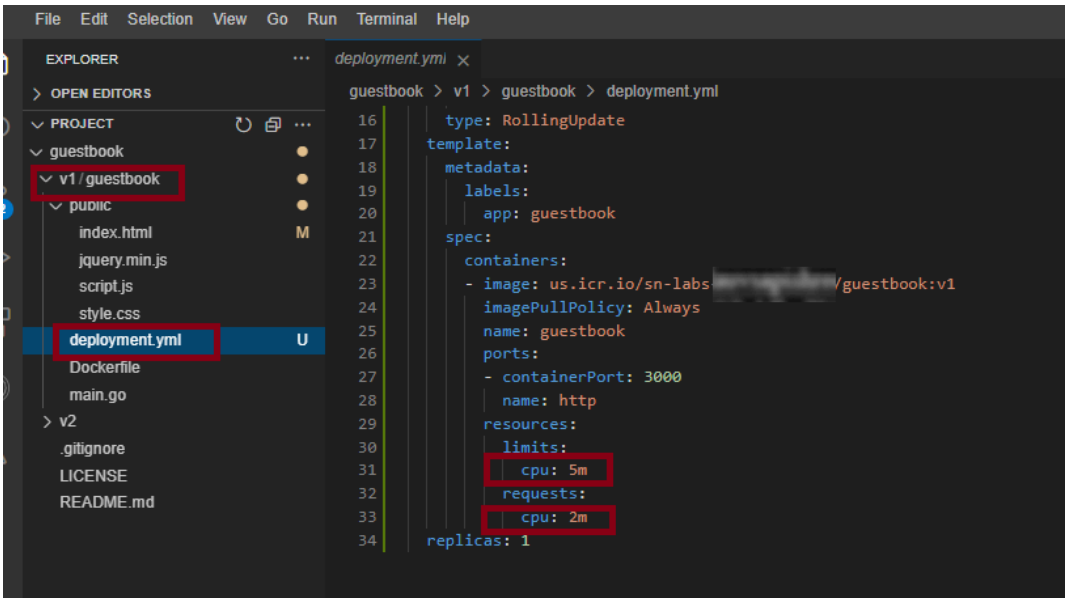
```
docker build . -t us.icr.io/$MY_NAMESPACE/guestbook:v1 && docker push us.icr.io/$MY_NAMESPACE/guestbook:v1
```



Take a screenshot of your updated image and save it as a .jpg or .png with the filename `upguestbook.png`. You will be prompted to upload the screenshot in the Peer Assignment.
将更新后的图像进行截图，并保存为 .jpg 或 .png 格式，文件名命名为 `upguestbook.png` 。您将在同行作业中收到上传截图的提示。

3. Update the values of the CPU in the `deployment.yml` to **cpu: 5m** and **cpu: 2m** as below:
将 `deployment.yml` 中的 CPU 值更新为 cpu: 5m 和 cpu: 2m，如下所示：

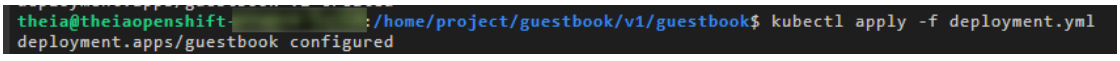
▼ Hint! 提示!



4. Apply the changes to the `deployment.yml` file.
将更改应用到 `deployment.yml` 文件。

▼ Hint! 提示!

```
kubectl apply -f deployment.yml
```



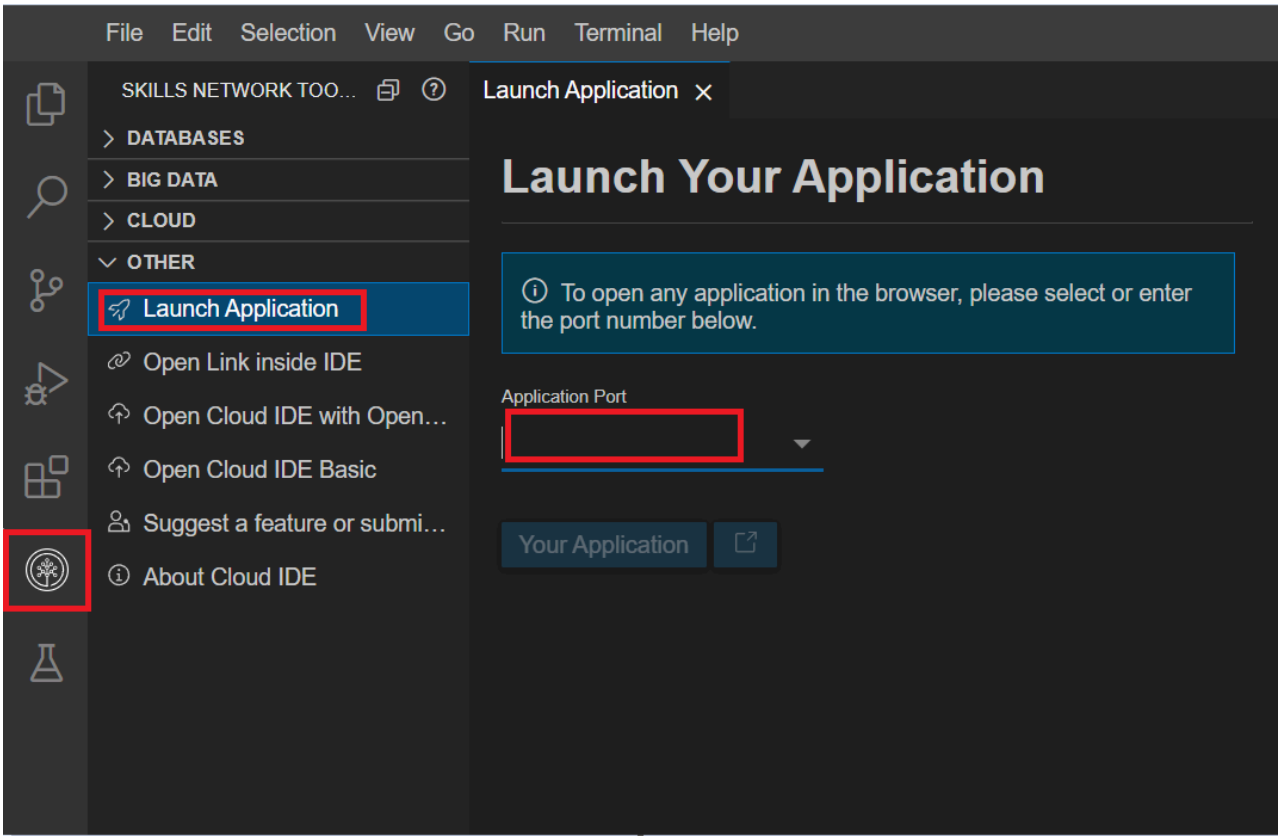
📷 Take a screenshot of the details of the output of Step 4 and save it as a .jpg or .png with the filename `deployment.png`. You will be prompted to upload the screenshot in the Peer Assignment.
📷 对步骤 4 的输出详情进行截图，并保存为.jpg 或.png 格式的文件，文件名使用 `deployment.png` 。在同行作业中，系统将提示您上传该截图。

5. Run the port-forward command again to start the app:
再次运行端口转发命令以启动应用程序：

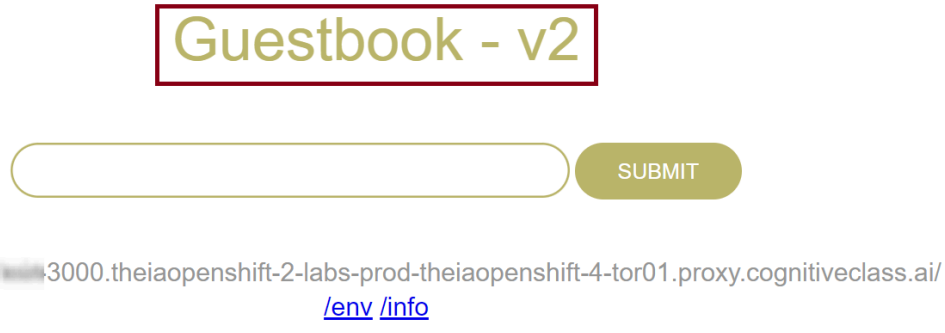
```
kubectl port-forward deployment.apps/guestbook 3000:3000
```

```
theia@theiaopenshift:/home/project/guestbook/v1/guestbook$ kubectl port-forward deployment.apps/guestbook 3000:3000
Forwarding from 127.0.0.1:3000 -> 3000
Forwarding from [::1]:3000 -> 3000
Handling connection for 3000
Handling connection for 3000
Handling connection for 3000
Handling connection for 3000
Handling connection for 3000
Handling connection for 3000
Handling connection for 3000
Handling connection for 3000
Handling connection for 3000
Handling connection for 3000
Handling connection for 3000
```

6. Launch your application on port 3000. Click on the Skills Network button on the right, it will open the **"Skills Network Toolbox"**. Then click the **Other** then **Launch Application**. From there you should be able to enter the port and launch.
在端口 3000 上启动您的应用程序。点击右侧的"Skills Network"按钮，它将打开"Skills Network Toolbox"。然后点击"其他"，再点击"启动应用程序"。从那里您应该能够输入端口并启动。



7. You will notice the updated app content as below:
您将注意到更新的应用内容如下：



Take a screenshot of your updated application and save it as a .jpg or .png with the filename `up-app.png`. You will be prompted to upload the screenshot in the Peer Assignment.
对更新后的应用程序进行截图，并将其保存为 .jpg 或 .png 格式，文件名命名为 `up-app.png` 。您将在同行作业中收到上传截图的提示。

Note: Please stop the application before running the next steps.
注意：在执行下一步操作之前，请停止应用程序。

8. Run the below command to see the history of deployment rollouts:
运行以下命令以查看部署分发的历史记录：

```
kubectl rollout history deployment/guestbook
```

```
theia@theiaopenshift-                /home/project/guestbook/v1/guestbook$ kubectl rollout history deployment/guestbook
deployment.apps/guestbook
REVISION  CHANGE-CAUSE
1         <none>
2         <none>
```

9. Run the below command to see the details of Revision of the deployment rollout:
运行以下命令以查看部署回滚的详细信息：

```
kubectl rollout history deployments guestbook --revision=2
```

```
theia@theiaopenshift-                /home/project/guestbook/v1/guestbook$ kubectl rollout history deployments guestbook --revision=2
deployment.apps/guestbook with revision #2
Pod Template:
  Labels:       app=guestbook
        pod-template-hash=6bdb7c74c6
  Containers:
   guestbook:
    Image:      us.icr.io/sn-labs-                /guestbook:v1
    Port:       3000/TCP
    Host Port:  0/TCP
    Limits:
      cpu:      5m
    Requests:
      cpu:      2m
    Environment:        <none>
    Mounts:     <none>
  Volumes:      <none>
```

Take a screenshot of the details of the correct Revision and save it as a .jpg or .png with the filename `rev.png`. You will be prompted to upload the screenshot in the Peer Assignment.
对正确的修订版本详细信息进行截图，并将其保存为 .jpg 或 .png 格式，文件名以 `rev.png` 结尾。您将在同行作业中收到上传截图的提示。

10. Run the below command to get the replica sets and observe the deployment which is being used now:
运行以下命令以获取副本集并观察当前正在使用的部署：

```
kubectl get rs
```

```
theia@theiaopenshift-                /home/project/guestbook/v1/guestbook$ kubectl get rs
NAME                   DESIRED   CURRENT   READY   AGE
guestbook-5997448ccb   0         0         0       17m
guestbook-6bdb7c74c6   1         1         1       105s
```

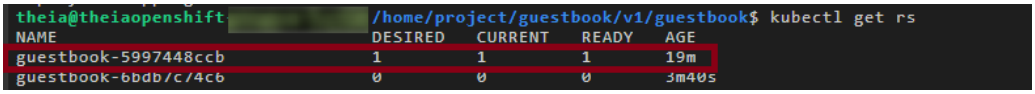
11. Run the below command to undo the deploymnent and set it to Revision 1:
运行以下命令以撤销部署并将其设置为修订版本 1：

```
kubectl rollout undo deployment/guestbook --to-revision=1
```

```
theia@theiaopenshift                :/home/project/guestbook/v1/guestbook$ kubectl rollout undo deployment/guestbook --to-revision=1
deployment.apps/guestbook rolled back
```

12. Run the below command to get the replica sets after the Rollout has been undone. The deployment being used would have changed as below:
运行以下命令以获取回滚操作后的副本集。使用的部署将如下更改：

```
kubectl get rs
```



Take a screenshot of the output of Step 9 and save it as a .jpg or .png with the filename `rs.png`. You will be prompted to upload the screenshot in the Peer Assignment.
将步骤 9 的输出进行截图，并保存为.jpg 或.png 格式的文件，文件名命名为 `rs.png` 。您将在同行评审任务中收到上传截图的提示。

Congratulations! You have completed the final project for this course. Do not log out of the lab environment (you can close the browser though) or delete any of the artifacts created during the lab, as these will be needed for the next lab,`Optional: Deploy Guestbook App from the OpenShift Internal Registry`.
恭喜！您已完成本课程的最终项目。不要注销实验室环境（但可以关闭浏览器）或删除在实验室期间创建的任何工件，因为这些将在下一个实验室中需要， `Optional: Deploy Guestbook App from the OpenShift Internal Registry` 。

© IBM Corporation. All rights reserved.