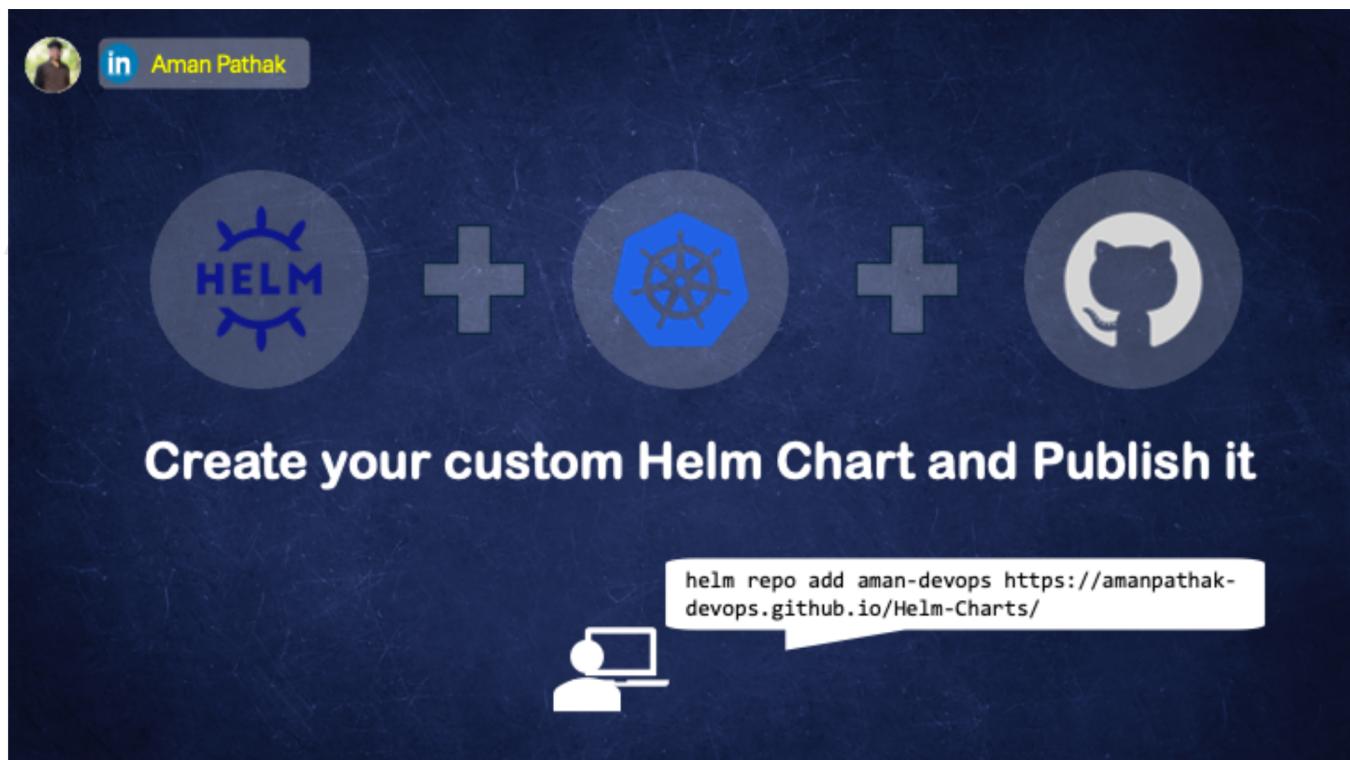


# Package and Publish Your Custom Multiple Helm Charts using GitHub Repo



# Introduction

In this guide, we will explore how to package and publish multiple Helm charts in a single GitHub repository. We'll deploy two distinct applications—a Go-based portfolio website and the classic Tetris game—on a Kubernetes cluster using Helm. By the end, you will have learned how to create, package, and publish Helm charts on GitHub and deploy them in a Kubernetes environment.

## Objective

The objective of this tutorial is to help you understand:

1. How to create multiple Helm charts.
2. How to package and publish those charts on a GitHub repository.
3. How to install and run those applications on Kubernetes using Helm.

## Pre-requisites

Before starting, ensure you have the following tools installed:

- **kubectl** for interacting with Kubernetes clusters.
- **Helm** for managing Kubernetes applications.
- **Minikube** for local Kubernetes cluster setup (optional).
- A public **GitHub repository** to publish your Helm charts.

Repository for this Hands-On Project-

<https://github.com/AmanPathak-DevOps/Helm-Charts/tree/master>

## Hands-On Guide



Create a new GitHub repository

**Note:** It is required to keep your repository public to publish the helm charts

The screenshot shows a GitHub repository page for 'Helm-Charts'. The repository is public and has one branch ('master') and one commit ('Initial commit'). The commit was made by 'AmanPathak-DevOps' 20 hours ago. The repository contains files: LICENSE, README.md, and README. The README file is described as containing multiple helm charts for deploying applications with minimum commands. The repository has 0 stars, 1 watching, and 0 forks. There are no releases published.

I am cloning the same repo on my EC2 machine. You can do the same on your local machine with the above-required tool installation.

```
ubuntu@ip-172-31-31-201:~$ git clone https://github.com/AmanPathak-DevOps/Helm-Charts.git
Cloning into 'Helm-Charts'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (4/4), done.
ubuntu@ip-172-31-31-201:~$ 
ubuntu@ip-172-31-31-201:~$ cd Helm-Charts/
ubuntu@ip-172-31-31-201:~/Helm-Charts$ 
ubuntu@ip-172-31-31-201:~/Helm-Charts$ ls
LICENSE  README.md
ubuntu@ip-172-31-31-201:~/Helm-Charts$ 
```

We are going to create two different charts. So we can learn how to package multiple charts in one repo and publish them on Git Hub as well. The first chart has a portfolio website in Go Lang and the second chart is a game known as Tetris. You don't need to follow the Kubernetes manifest file if you don't want to.

To create a first helm chart, run the below command

```
helm create go-portfolio-app
```

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts]$ helm create go-portfolio-app
Creating go-portfolio-app
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ 
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ tree
.
├── LICENSE
├── README.md
└── go-portfolio-app
    ├── Chart.yaml
    ├── charts
    ├── templates
    │   ├── NOTES.txt
    │   ├── _helpers.tpl
    │   ├── deployment.yaml
    │   ├── hpa.yaml
    │   ├── ingress.yaml
    │   ├── service.yaml
    │   └── serviceaccount.yaml
    └── tests
        └── test-connection.yaml
values.yaml

5 directories, 12 files
ubuntu@ip-172-31-31-201:~/Helm-Charts$ 
```

We already have our own Manifest file content and values file. So, empty the templates folder and remove the values.yaml file

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ cd go-portfolio-app/
[ubuntu@ip-172-31-31-201:~/Helm-Charts/go-portfolio-app$ rm -rf templates/* values.yaml
[ubuntu@ip-172-31-31-201:~/Helm-Charts/go-portfolio-app$ 
[ubuntu@ip-172-31-31-201:~/Helm-Charts/go-portfolio-app$ tree
.
├── Chart.yaml
└── charts

3 directories, 1 file
ubuntu@ip-172-31-31-201:~/Helm-Charts/go-portfolio-app$ 
```

Now, create a deployment.yaml file and service.yaml file then copy the content from below.

### namespace.yaml

```
apiVersion: v1
kind: Namespace
metadata:
  name: go-app
```

### deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: go-portfolio
```

```
namespace: go-app
labels:
  role: go-portfolio
  env: dev
spec:
  replicas: {{ .Values.replicaCount }} # Use replica
count from values.yaml
  selector:
    matchLabels:
      role: go-portfolio
  template:
    metadata:
      labels:
        role: go-portfolio
    spec:
      containers:
        - name: go-portfolio
          image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}" # Pull image and tag from
values.yaml
          imagePullPolicy: {{ .Values.image.pullPolicy }}
        ports:
          - containerPort: 8080
        resources:
          requests:
            memory: "256Mi"
            cpu: "500m"
          limits:
            memory: "512Mi"
            cpu: "1"
```

## service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: go-portfolio-svc
  namespace: go-app
spec:
  ports:
    - port: 8080
      targetPort: 8080
    type: ClusterIP
  selector:
    role: go-portfolio
```

## values.yaml

```
replicaCount: 1

image:
  repository: avian19/go-port
  pullPolicy: IfNotPresent
  # Overrides the image tag whose default is the chart
  appVersion.
  tag: "10201891038"
```

Hope you have added the above YAML code in your respective files.

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts/go-portfolio-app$ vim templates/namespace.yaml
[ubuntu@ip-172-31-31-201:~/Helm-Charts/go-portfolio-app$ vim templates/deployment.yaml
[ubuntu@ip-172-31-31-201:~/Helm-Charts/go-portfolio-app$ vim templates/service.yaml
[ubuntu@ip-172-31-31-201:~/Helm-Charts/go-portfolio-app$ vim values.yaml
[ubuntu@ip-172-31-31-201:~/Helm-Charts/go-portfolio-app$ tree
.
├── Chart.yaml
└── charts
    └── templates
        ├── deployment.yaml
        ├── namespace.yaml
        └── service.yaml
└── values.yaml

3 directories, 5 files
ubuntu@ip-172-31-31-201:~/Helm-Charts/go-portfolio-app$ ]
```

Before going to package your helm chart, you can validate whether the helm chart is correct or not by deploying it.

```
helm install <chart-name> <chart-dir>
```

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts/go-portfolio-app$ helm install go-portfolio-app .
NAME: go-portfolio-app
LAST DEPLOYED: Sun Oct 13 09:12:23 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
[ubuntu@ip-172-31-31-201:~/Helm-Charts/go-portfolio-app$ kubectl get all -n go-app
NAME           READY   STATUS    RESTARTS   AGE
pod/go-portfolio-7f9bd85775-hdvqn   1/1     Running   0          7s

NAME              TYPE      CLUSTER-IP       EXTERNAL-IP      PORT(S)      AGE
service/go-portfolio-svc   ClusterIP   10.101.215.117   <none>        8080/TCP   7s

NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/go-portfolio   1/1     1           1          7s

NAME           DESIRED   CURRENT   READY   AGE
replicaset.apps/go-portfolio-7f9bd85775   1         1         1         7s
[ubuntu@ip-172-31-31-201:~/Helm-Charts/go-portfolio-app$ 
[ubuntu@ip-172-31-31-201:~/Helm-Charts/go-portfolio-app$ kl go-portfolio-7f9bd85775-hdvqn -n go-app
kl: command not found
[ubuntu@ip-172-31-31-201:~/Helm-Charts/go-portfolio-app$ kubectl logs go-portfolio-7f9bd85775-hdvqn -n go-app
2024/10/13 09:12:25 Server started on :8080
ubuntu@ip-172-31-31-201:~/Helm-Charts/go-portfolio-app$ ]
```

The Go App is running fine. So, don't forget to uninstall it

```
helm uninstall <chart-name>
```

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts/go-portfolio-app$ helm uninstall go-portfolio-app
release "go-portfolio-app" uninstalled
ubuntu@ip-172-31-31-201:~/Helm-Charts/go-portfolio-app$ ]
```

Now, we are ready to package our helm chart.

Go to the GitHub root directory and check the status

I am pushing my changes to the master branch(to keep track) which is not required in the process of packaging and publishing the helm chart.

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts/go-portfolio-app$ cd ..
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    go-portfolio-app/

nothing added to commit but untracked files present (use "git add" to track)
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ git add go-portfolio-app/
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ git commit -m "Added Go Portfolio YAMLS"
[master 3ae296] Added Go Portfolio YAMLS
Committer: Ubuntu <ubuntu@ip-172-31-31-201.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

  git config --global --edit

After doing this, you may fix the identity used for this commit with:

  git commit --amend --reset-author

6 files changed, 102 insertions(+)
create mode 100644 go-portfolio-app/.helmignore
create mode 100644 go-portfolio-app/chart.yaml
create mode 100644 go-portfolio-app/templates/deployment.yaml
create mode 100644 go-portfolio-app/templates/namespace.yaml
create mode 100644 go-portfolio-app/templates/service.yaml
create mode 100644 go-portfolio-app/values.yaml
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ git push origin master
Username for 'https://github.com': AmanPathak-DevOps
Password for 'https://AmanPathak-DevOps@github.com':
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 2 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 2.00 KiB | 1023.00 KiB/s, done.
Total 10 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/AmanPathak-DevOps/Helm-Charts.git
 f247930..3ae296 master -> master
ubuntu@ip-172-31-31-201:~/Helm-Charts$ ]
```

To Package your Helm Chart and publish it, the gh-pages branch is required in our GitHub Repo. So, create a branch called gh-pages

```
git checkout -b gh-pages
```

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ git branch
* master
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ ]
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ git checkout -b gh-pages
Switched to a new branch 'gh-pages'
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ git branch
* gh-pages
  master
ubuntu@ip-172-31-31-201:~/Helm-Charts$ ]
```

Once, the branch gets created. We need to create a package of our go-portfolio-app.

To generate the package, you need to run the below command which will generate a file having tgz extension.

**Note:** The branch will be master as it contains the Chart.yaml file

```
helm package <chart-name(dir)>
```

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ ls
LICENSE README.md go-portfolio-app
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ helm package go-portfolio-app/
Successfully packaged chart and saved it to: /home/ubuntu/Helm-Charts/go-portfolio-app-0.1.0.tgz
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ ls
LICENSE README.md go-portfolio-app go-portfolio-app-0.1.0.tgz
ubuntu@ip-172-31-31-201:~/Helm-Charts$ ]
```

So, we packaged our helm chart. Now, we are ready to publish it.  
To publish, we need to create an index.yaml file that keeps track of all the packaged helm charts whenever you install them.  
So, create an index.yaml file at the same place

```
touch index.yaml
```

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ ls
LICENSE README.md go-portfolio-app
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ helm package go-portfolio-app/
Successfully packaged chart and saved it to: /home/ubuntu/Helm-Charts/go-portfolio-app-0.1.0.tgz
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ ls
LICENSE README.md go-portfolio-app go-portfolio-app-0.1.0.tgz
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ touch index.yaml
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ cat index.yaml
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ ]
```

As you can see, the index.yaml file did not have any content. But now, we need to generate the index file based on the charts present in our repository which is go-portfolio-app.

To generate the index.yaml file, run the below command

```
helm repo index <dir-where-tgz-file-located>
```

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ ls
LICENSE README.md go-portfolio-app
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ helm package go-portfolio-app/
Successfully packaged chart and saved it to: /home/ubuntu/Helm-Charts/go-portfolio-app-0.1.0.tgz
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ ls
LICENSE README.md go-portfolio-app go-portfolio-app-0.1.0.tgz
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ touch index.yaml
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ cat index.yaml
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ helm repo index .^C
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ helm repo index .
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ cat index.yaml
apiVersion: v1
entries:
  go-portfolio-app:
    - apiVersion: v2
      appVersion: 1.16.0
      created: "2024-10-13T09:26:51.636270883Z"
      description: A Helm chart for Kubernetes
      digest: 3b288e49945e044dcc2e6d9319cc2be3b204b5ce0eaae64c7ae127a14c799a03
      name: go-portfolio-app
      type: application
      urls:
        - go-portfolio-app-0.1.0.tgz
      version: 0.1.0
generated: "2024-10-13T09:26:51.635828164Z"
ubuntu@ip-172-31-31-201:~/Helm-Charts$ ]
```

Now, Index.yaml is not empty as it holds the chart name go-portfolio-app  
Finally, we need to push our gh-pages branch changes to GitHub.

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ git status
On branch gh-pages
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    go-portfolio-app-0.1.0.tgz
    index.yaml

nothing added to commit but untracked files present (use "git add" to track)
ubuntu@ip-172-31-31-201:~/Helm-Charts$ git add go-portfolio-app-0.1.0.tgz index.yaml
ubuntu@ip-172-31-31-201:~/Helm-Charts$ git status
On branch gh-pages
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   go-portfolio-app-0.1.0.tgz
    new file:   index.yaml

ubuntu@ip-172-31-31-201:~/Helm-Charts$ git commit -m "Added go-portfolio-app chart"
[gh-pages 5bc779d] Added go-portfolio-app chart
  Committer: Ubuntu <ubuntu@ip-172-31-31-201.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
  git config --global --edit

After doing this, you may fix the identity used for this commit with:
  git commit --amend --reset-author

2 files changed, 14 insertions(+)
create mode 100644 go-portfolio-app-0.1.0.tgz
create mode 100644 index.yaml
ubuntu@ip-172-31-31-201:~/Helm-Charts$ git push -f origin gh-pages
Username for 'https://github.com': AmanPathak-DevOps
Password for 'https://AmanPathak-DevOps@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 1.68 KiB | 1.68 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'gh-pages' on GitHub by visiting:
remote:   https://github.com/AmanPathak-DevOps/Helm-Charts/pull/new/gh-pages
remote:
To https://github.com/AmanPathak-DevOps/Helm-Charts.git
 * [new branch]  gh-pages -> gh-pages
ubuntu@ip-172-31-31-201:~/Helm-Charts$ ]
```

As soon as you push your changes to the GitHub repo.

Go to the browser and check the GitHub repository, you will see there will a GitHub Actions workflow getting triggered in Orange color.

The screenshot shows the GitHub repository page for 'Helm-Charts'. At the top, there are navigation links: Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. The 'Actions' tab is highlighted with a red underline. Below the header, the repository name 'Helm-Charts' is shown as 'Public'. The main content area displays a list of files and their recent commits:

File / Commit	Description	Time Ago
Ubuntu Added go-portfolio-app chart	Added go-portfolio-app chart	5bc779d - 1 minute ago
go-portfolio-app	Added Go Portfolio YAMLS	14 minutes ago
LICENSE	Initial commit	yesterday
README.md	Initial commit	yesterday
go-portfolio-app-0.1.0.tgz	Added go-portfolio-app chart	1 minute ago
index.yaml	Added go-portfolio-app chart	1 minute ago

Below the file list, there are links for 'README' and 'License'. A large 'Helm-Charts' heading is present, followed by a descriptive text: 'This repository will contain the multiple helm charts. You can deploy applications with minimum commands'. To the right of the file list, there is an 'About' section with a detailed description of the repository's purpose and links to 'Readme', 'MIT license', 'Activity', 'Stars', 'Watching', and 'Forks'. There are also sections for 'Releases' (No releases published) and 'Packages' (No packages published).

Once you click on it you will see the deployment gets completed. But where is it deployed?

The screenshot shows the same GitHub repository page for 'Helm-Charts'. The 'Actions' tab is still highlighted. A modal window is open over the file list, indicating that all checks have passed:

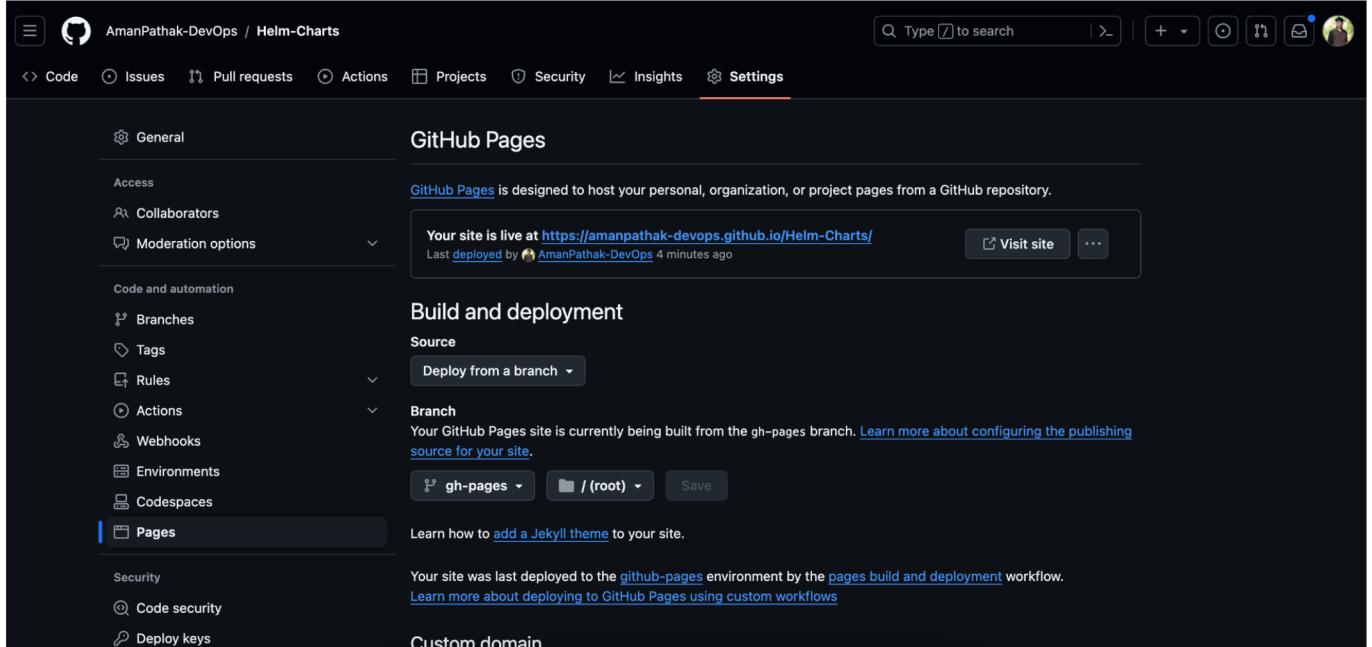
**All checks have passed**  
3 successful checks

- ✓ pages build and deployment / build (dynamic) Successful in 21s Details
- ✓ pages build and deployment / report-build-status (dynamic) Successful in 2s Details
- ✓ pages build and deployment / deploy (dynamic) Successful in 7s Details

Below the modal, the file list shows the same items as before, with the most recent commit being 'Added go-portfolio-app chart' at '3 minutes ago'. The 'About' section and other repository details are visible on the right side of the screen.

To check the deployed place,  
Go to the Settings of your GitHub repository and navigate to Pages where we are hosting our helm charts

As you can see in the below snippet, Your site is live at <https://amanpathak-devops.github.io/Helm-Charts/>. This will be our URL to deploy Helm charts on any Kubernetes Cluster.



The screenshot shows the GitHub Pages settings for a repository named 'Helm-Charts'. The 'General' tab is selected. A prominent message states 'Your site is live at <https://amanpathak-devops.github.io/Helm-Charts/>'. Below this, under 'Build and deployment', it shows the source is 'Deploy from a branch' and the branch is 'gh-pages'. There are tabs for 'Pages', 'Security', 'Code security', and 'Deploy keys', with 'Pages' currently active.

Now, add the helm chart repo using the below command

```
helm repo add aman-devops  
https://amanpathak-devops.github.io/Helm-Charts/
```

```
[ubuntu@ip-172-31-31-201:~$ helm repo list  
Error: no repositories to show  
[ubuntu@ip-172-31-31-201:~$  
[ubuntu@ip-172-31-31-201:~$ helm repo add aman-devops https://amanpathak-devops.github.io/Helm-Charts/  
"aman-devops" has been added to your repositories  
[ubuntu@ip-172-31-31-201:~$  
[ubuntu@ip-172-31-31-201:~$ helm repo list  
NAME          URL  
aman-devops   https://amanpathak-devops.github.io/Helm-Charts/  
ubuntu@ip-172-31-31-201:~$
```

Now, we are ready to install our helm chart through the helm install command

```
helm install go-app aman-devops/go-portfolio-app
```

```
[ubuntu@ip-172-31-31-201:~$ helm install go-app aman-devops/go-portfolio-app
NAME: go-app
LAST DEPLOYED: Sun Oct 13 09:59:36 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
ubuntu@ip-172-31-31-201:~$ ]
```

You can run the below command to list the created Kubernetes resources through the helm chart

```
kubectl get all -n go-app
```

```
[ubuntu@ip-172-31-31-201:~$ kubectl get all -n go-app
NAME                           READY   STATUS    RESTARTS   AGE
pod/go-portfolio-7f9bd85775-md5zd   1/1     Running   0          118s

NAME            TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
service/go-portfolio-svc   ClusterIP   10.104.112.40   <none>        8080/TCP   118s

NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/go-portfolio   1/1     1           1          118s

NAME           DESIRED   CURRENT   READY   AGE
replicaset.apps/go-portfolio-7f9bd85775   1         1         1       118s
ubuntu@ip-172-31-31-201:~$ ]
```

Run the below command to access your application

```
kubectl port-forward svc/tetris-service -n tetris
3000:3000
```

To access your application on the browser, you can hit the showing link in your browser

The screenshot shows a web browser window with the URL "localhost:8080" in the address bar. The page title is "Aman Pathak - DevOps Engineer". Below the title is a navigation bar with four buttons: "Home" (highlighted in red), "Projects", "Achievements", and "Contact". The main content area has a yellow header and an orange background. The "Home" section contains a heading "Home", a brief introduction about the author's expertise in DevOps, and a mission statement about streamlining deployment processes. At the bottom of the page is a dark footer bar with the copyright notice "© 2024 Aman Pathak. All rights reserved."

If you are using EC2 Server and want to access applications in your browser then, the kubectl port-forward command won't be enough. To access your browser, run one more command on your local machine it is almost the same command as ssh does.

```
ssh -L 8080:localhost:8080 <ec2-user>@<public-ip> -i  
<Pem-file>
```

```

amanpathak@MacBookAir Downloads % ssh -L 8080:localhost:8080 ubuntu@184.73.41.161 -i Aman-Pathak.pem
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Sun Oct 13 10:07:13 UTC 2024

System load: 0.11 Temperature: -273.1 C
Usage of /: 28.8% of 28.02GB Processes: 169
Memory usage: 27% Users logged in: 1
Swap usage: 0% IPv4 address for ens5: 172.31.31.201

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

5 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sun Oct 13 09:49:24 2024 from 152.58.65.8
ubuntu@ip-172-31-31-201:~$ 

```

We have successfully packaged and published our first Helm chart. If you feel confident with this process, you can go ahead and exit this blog or document. I hope you found something new to learn. However, if you're interested in learning how to create multiple charts within a single GitHub repository, I invite you to continue with me in this guide.

Let's go ahead and create another chart, which should be quick if you already grasp the process!

So, I switched back to my master branch in my GitHub repository  
`git switch master`

```

[ubuntu@ip-172-31-31-201:~/Helm-Charts$ ls
LICENSE README.md go-portfolio-app go-portfolio-app-0.1.0.tgz index.yaml
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ 
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ git switch master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ ls
LICENSE README.md go-portfolio-app
ubuntu@ip-172-31-31-201:~/Helm-Charts$ 

```

Create a new helm chart named tetris-game

```
helm create tetris-game
```

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ helm create tetris-game
Creating tetris-game
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ ls
LICENSE README.md go-portfolio-app tetris-game
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ tree
.
├── LICENSE
├── README.md
└── go-portfolio-app
    ├── Chart.yaml
    ├── charts
    └── templates
        ├── deployment.yaml
        ├── namespace.yaml
        ├── service.yaml
        └── values.yaml
└── tetris-game
    ├── Chart.yaml
    ├── charts
    └── templates
        ├── NOTES.txt
        ├── _helpers.tpl
        ├── deployment.yaml
        ├── hpa.yaml
        ├── ingress.yaml
        ├── service.yaml
        ├── serviceaccount.yaml
        └── tests
            └── test-connection.yaml
    └── values.yaml

8 directories, 17 files
ubuntu@ip-172-31-31-201:~/Helm-Charts$ ]
```

Empty the templates folder and remove the values.yaml file

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ rm -rf tetris-game/templates/ tetris-game/values.yaml
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ tree
.
├── LICENSE
├── README.md
└── go-portfolio-app
    ├── Chart.yaml
    ├── charts
    └── templates
        ├── deployment.yaml
        ├── namespace.yaml
        ├── service.yaml
        └── values.yaml
└── tetris-game
    ├── Chart.yaml
    └── charts

6 directories, 8 files
ubuntu@ip-172-31-31-201:~/Helm-Charts$ ]
```

Now, add your YAML files given below

### namespace.yaml

```
apiVersion: v1
kind: Namespace
metadata:
  name: tetris
```

### deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: tetris
  namespace: tetris
spec:
  replicas: 3
  selector:
    matchLabels:
      app: tetris
  template:
    metadata:
      labels:
        app: tetris
    spec:
      containers:
        - name: tetris
          image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"
      ports:
        - containerPort: 3000
```

## service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: tetris-service
  namespace: tetris
spec:
  selector:
    app: tetris
```

```
ports:  
- protocol: TCP  
  port: 3000  
  targetPort: 3000  
type: ClusterIP
```

## values.yaml

```
replicaCount: 1  
  
image:  
  repository: avian19/tetrisv2  
  pullPolicy: IfNotPresent  
  # Overrides the image tag whose default is the chart  
appVersion.  
  tag: "1"
```

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts/tetris-game]$ mkdir templates  
[ubuntu@ip-172-31-31-201:~/Helm-Charts/tetris-game$  
[ubuntu@ip-172-31-31-201:~/Helm-Charts/tetris-game$ vim templates/namespace.yaml  
[ubuntu@ip-172-31-31-201:~/Helm-Charts/tetris-game$ vim templates/deployment.yaml  
[ubuntu@ip-172-31-31-201:~/Helm-Charts/tetris-game$ vim templates/service.yaml  
[ubuntu@ip-172-31-31-201:~/Helm-Charts/tetris-game$ vim values.yaml  
[ubuntu@ip-172-31-31-201:~/Helm-Charts/tetris-game$ ]
```

Before going to package your helm chart, you can validate whether the helm chart is correct or not by deploying it.

```
helm install <chart-name> <chart-dir>
```

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts/tetris-game$ helm install tetris .
NAME: tetris
LAST DEPLOYED: Sun Oct 13 10:31:02 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
[ubuntu@ip-172-31-31-201:~/Helm-Charts/tetris-game$ kubectl get all -n tetris
NAME READY STATUS RESTARTS AGE
pod/tetris-55cdd6c6b9-6rt66 1/1 Running 0 8s
pod/tetris-55cdd6c6b9-fvflx 1/1 Running 0 8s
pod/tetris-55cdd6c6b9-lbxl5 1/1 Running 0 8s

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
service/tetris-service ClusterIP 10.102.205.204 <none> 3000/TCP 8s

NAME READY UP-TO-DATE AVAILABLE AGE
deployment.apps/tetris 3/3 3 3 8s

NAME DESIRED CURRENT READY AGE
replicaset.apps/tetris-55cdd6c6b9 3 3 3 8s
```

The Tetris App is running fine. So, don't forget to uninstall it

```
helm uninstall <chart-name>
```

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts/tetris-game$ helm uninstall tetris
release "tetris" uninstalled
ubuntu@ip-172-31-31-201:~/Helm-Charts/tetris-game$ ]
```

Now, we are ready to package our helm chart.

Go to the GitHub root directory and check the status

I am pushing my changes to the master branch(to keep track) which is not required in the process of packaging and publishing the helm chart.

```

ubuntu@ip-172-31-201:/Helm-Charts/tetris-game$ cd ..
ubuntu@ip-172-31-201:/Helm-Charts$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    tetris-game/

nothing added to commit but untracked files present (use "git add" to track)
ubuntu@ip-172-31-201:/Helm-Charts$ git add tetris-game/
ubuntu@ip-172-31-201:/Helm-Charts$ git commit -m "Added Tetris-game YAMLS"
[master fcbef20] Added Tetris-game YAMLS
  Committer: Ubuntu <ubuntu@ip-172-31-201.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

  git config --global --edit

After doing this, you may fix the identity used for this commit with:

  git commit --amend --reset-author

6 files changed, 91 insertions(+)
create mode 100644 tetris-game/.helmignore
create mode 100644 tetris-game/Chart.yaml
create mode 100644 tetris-game/templates/deployment.yaml
create mode 100644 tetris-game/templates/namespace.yaml
create mode 100644 tetris-game/templates/service.yaml
create mode 100644 tetris-game/values.yaml
ubuntu@ip-172-31-201:/Helm-Charts$ git push origin master
Username for 'https://github.com': AmanPathak-DevOps
Password for 'https://AmanPathak-DevOps@github.com':
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 2 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 1.82 KiB | 930.00 KiB/s, done.
Total 10 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/AmanPathak-DevOps/Helm-Charts.git
  3ae296..fcbef20 master -> master
ubuntu@ip-172-31-201:/Helm-Charts$ 

```

Package the Tetris application by running the below command on the master branch

```
helm package <chart-name(dir)>
```

```

[ubuntu@ip-172-31-201:/Helm-Charts$ git switch master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
[ubuntu@ip-172-31-201:/Helm-Charts$ helm package tetris-game/
Successfully packaged chart and saved it to: /home/ubuntu/Helm-Charts/tetris-game-0.1.0.tgz
[ubuntu@ip-172-31-201:/Helm-Charts$ tree
.
├── LICENSE
├── README.md
└── go-portfolio-app
    ├── Chart.yaml
    ├── charts
    ├── templates
    │   ├── deployment.yaml
    │   ├── namespace.yaml
    │   └── service.yaml
    └── values.yaml
└── tetris-game
    ├── Chart.yaml
    ├── charts
    ├── templates
    │   ├── deployment.yaml
    │   ├── namespace.yaml
    │   └── service.yaml
    └── values.yaml
tetris-game-0.1.0.tgz

7 directories, 13 files
ubuntu@ip-172-31-201:/Helm-Charts$ 

```

Now, switch to gh-pages to package the helm chart and publish it  
git switch gh-pages

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ git switch gh-pages
Switched to branch 'gh-pages'
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ 
ubuntu@ip-172-31-31-201:~/Helm-Charts$ ]
```

As you can see in the below snippet, the index.yaml file has a go-app chart configuration. Now, we need to update the index file to have the Tetris app configuration.

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ git switch gh-pages
Switched to branch 'gh-pages'
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ git status
On branch gh-pages
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    tetris-game-0.1.0.tgz

nothing added to commit but untracked files present (use "git add" to track)
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ cat index.yaml
apiVersion: v1
entries:
  go-portfolio-app:
  - apiVersion: v2
    appVersion: 1.16.0
    created: "2024-10-13T09:26:51.636270883Z"
    description: A Helm chart for Kubernetes
    digest: 3b288e49945e044dcc2e6d9319cc2be3b204b5ce0eaae64c7ae127a14c799a03
    name: go-portfolio-app
    type: application
    urls:
    - go-portfolio-app-0.1.0.tgz
    version: 0.1.0
generated: "2024-10-13T09:26:51.635828164Z"
ubuntu@ip-172-31-31-201:~/Helm-Charts$ ]
```

Run the below command to update the index file for the Tetris app

```
helm repo index <dir-where-tgz-file-located>
```

```

ubuntu@ip-172-31-31-201:~/Helm-Charts$ helm repo index .
ubuntu@ip-172-31-31-201:~/Helm-Charts$ cat index.yaml
apiVersion: v1
entries:
  - go-portfolio-app:
      apiVersion: v2
      appVersion: 1.16.0
      created: "2024-10-13T10:46:04.789790135Z"
      description: A Helm chart for Kubernetes
      digest: 3b288e4945e04d0cc2e6d9319cc2be3b204b5ce0aae64c7ae127a14c799a03
      name: go-portfolio-app
      type: application
      urls:
        - go-portfolio-app-0.1.0.tgz
        version: 0.1.0
  - tetris-game:
      apiVersion: v2
      appVersion: 1.16.0
      created: "2024-10-13T10:46:04.790085871Z"
      description: A Helm chart for Kubernetes
      digest: 16617ed7a55301985b5ca4124405786c0d40cf66b688445e86ccbe28c719a16b
      name: tetris-game
      type: application
      urls:
        - tetris-game-0.1.0.tgz
        version: 0.1.0
generated: "2024-10-13T10:46:04.789303275Z"
ubuntu@ip-172-31-31-201:~/Helm-Charts$ 

```

Now, Index.yaml contains the configuration for chart tetris.  
So we can push our gh-pages branch changes to GitHub.

```

ubuntu@ip-172-31-31-201:~/Helm-Charts$ git status
On branch gh-pages
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   index.yaml

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    tetris-game-0.1.0.tgz

no changes added to commit (use "git add" and/or "git commit -a")
ubuntu@ip-172-31-31-201:~/Helm-Charts$ git add index.yaml tetris-game-0.1.0.tgz
ubuntu@ip-172-31-31-201:~/Helm-Charts$ git commit -m "Added tetris app to helm charts"
[gh-pages 1f9ed7a] Added tetris app to helm charts
Committer: Ubuntu <ubuntu@ip-172-31-31-201.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

  git config --global --edit

After doing this, you may fix the identity used for this commit with:

  git commit --amend --reset-author

2 files changed, 13 insertions(+), 2 deletions(-)
create mode 100644 tetris-game-0.1.0.tgz
ubuntu@ip-172-31-31-201:~/Helm-Charts$ git push origin gh-pages
Username for 'https://github.com': AmanPathak-DevOps
Password for 'https://AmanPathak-DevOps@github.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 1.41 KiB | 1.41 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/AmanPathak-DevOps/Helm-Charts.git
  5bc779d..1f9ed7a  gh-pages -> gh-pages
ubuntu@ip-172-31-31-201:~/Helm-Charts$ 

```

As soon as you push your changes to the GitHub repo.  
Go to the browser and check the GitHub repository, you will see there will a GitHub Actions workflow getting triggered in Orange color.

The screenshot shows the GitHub repository page for 'Helm-Charts'. The 'Code' tab is selected. At the top, there are navigation links: Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. Below the header, the repository name 'Helm-Charts' is shown with a public status. A search bar and a 'Pin' button are also present. The main content area displays a list of files and commits. On the right side, there are sections for 'About', 'Releases', and 'Packages'.

You can check the process of deployments by click on Orange icon

The screenshot shows the GitHub repository page for 'Helm-Charts'. The 'Actions' tab is selected. At the top, there are navigation links: Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. Below the header, the repository name 'Helm-Charts' is shown with a public status. A search bar and a 'Pin' button are also present. The main content area displays a list of files and deployment status. A modal window titled 'Some checks haven't completed yet' is open, showing three pending checks: 'pages build and deployment / report-build-status (dynamic)', 'pages build and deployment / deploy (dynamic)', and 'pages build and deployment / build (dynamic)'. The 'Details' for each check are visible on the right. The 'About' section is partially visible on the right.

You can also go to Actions to view the GitHub workflow

Now, when you are trying to install the Tetris app you might get errors as the old chart doesn't have Tetris application configuration including index.yaml file.

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ helm install tetris aman-devops/tetris-game
Error: INSTALLATION FAILED: chart "tetris-game" matching not found in aman-devops index. (try 'helm repo update'): no chart name found
ubuntu@ip-172-31-31-201:~/Helm-Charts$ ]
```

Before installing the Tetris app, we need to update our helm chart

**helm repo update**

Now, we are ready to install our helm chart through the helm install command

**helm install tetris aman-devops/tetris-game**

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ helm install tetris aman-devops/tetris-game
Error: INSTALLATION FAILED: chart "tetris-game" matching not found in aman-devops index. (try 'helm repo update'): no chart name found
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "aman-devops" chart repository
Update Complete. *Happy Helming!*
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ 
[ubuntu@ip-172-31-31-201:~/Helm-Charts$ helm install tetris aman-devops/tetris-game
NAME: tetris
LAST DEPLOYED: Sun Oct 13 10:54:00 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
ubuntu@ip-172-31-31-201:~/Helm-Charts$ ]
```

You can run the below command to list the created Kubernetes resources through the helm chart

```
kubectl get all -n tetris
```

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts]$ kubectl get all -n tetris
NAME           READY   STATUS    RESTARTS   AGE
pod/tetris-55cdd6c6b9-b44d1  1/1     Running   0          21s
pod/tetris-55cdd6c6b9-dxqtr  1/1     Running   0          21s
pod/tetris-55cdd6c6b9-frpt7  1/1     Running   0          21s

NAME            TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
service/tetris-service  ClusterIP  10.99.49.223  <none>        3000/TCP   21s

NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/tetris  3/3      3           3           21s

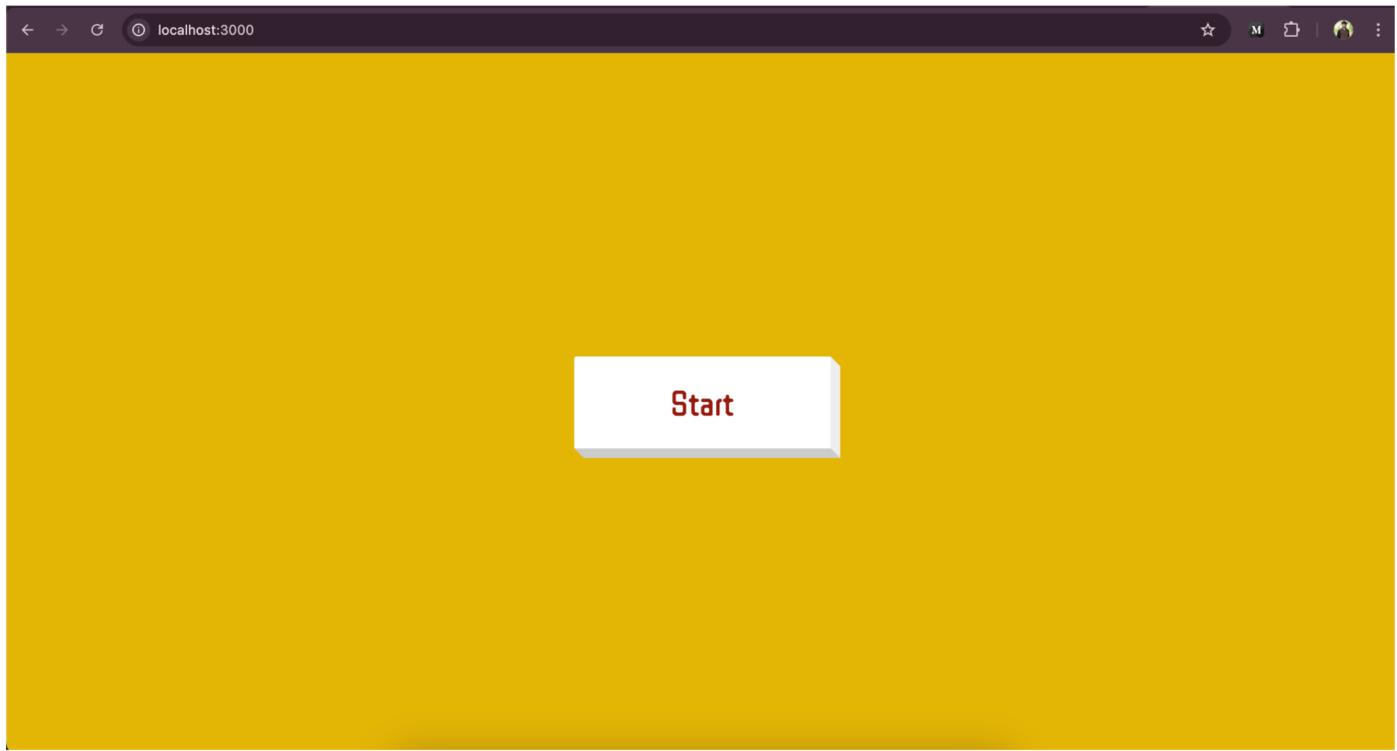
NAME           DESIRED  CURRENT   READY   AGE
replicaset.apps/tetris-55cdd6c6b9  3       3         3        21s
ubuntu@ip-172-31-31-201:~/Helm-Charts$
```

Run the below command to access your application

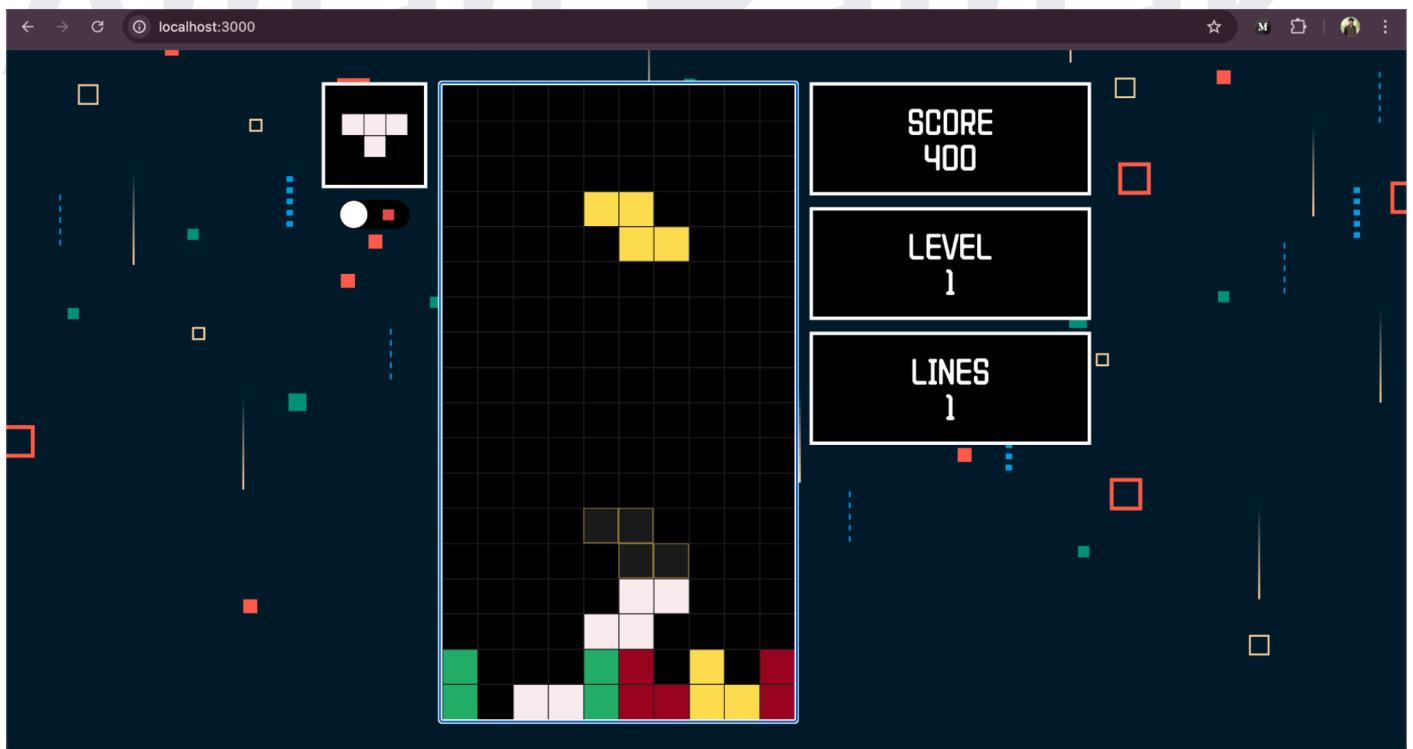
```
kubectl port-forward svc/tetris-service -n tetris
3000:3000
```

```
[ubuntu@ip-172-31-31-201:~/Helm-Charts]$ kubectl port-forward svc/tetris-service -n tetris 3000:3000
Forwarding from 127.0.0.1:3000 -> 3000
Forwarding from [::1]:3000 -> 3000
```

To access your application on the browser, you can hit the showing link in your browser



Enjoy the Game!



If you are using EC2 Server and want to access applications in your browser then, the kubectl port-forward command won't be enough. To access your browser, run one more command on your local machine it is almost the same command as ssh does.

```
ssh -L 3000:localhost:3000 <ec2-user>@<public-ip> -i  
<Pem-file>
```

```
amanpathak@MacBookAir Downloads % ssh -L 3000:localhost:3000 ubuntu@184.73.41.161 -i Aman-Pathak.pem  
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)  
  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/pro  
  
System information as of Sun Oct 13 10:23:47 UTC 2024  
  
System load: 0.22 Temperature: -273.1 C  
Usage of /: 28.8% of 28.02GB Processes: 173  
Memory usage: 28% Users logged in: 1  
Swap usage: 0% IPv4 address for ens5: 172.31.31.201  
  
* Ubuntu Pro delivers the most comprehensive open source security and  
compliance features.  
  
https://ubuntu.com/aws/pro  
  
Expanded Security Maintenance for Applications is not enabled.  
  
5 updates can be applied immediately.  
To see these additional updates run: apt list --upgradable  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
Last login: Sun Oct 13 10:23:48 2024 from 152.58.65.8  
ubuntu@ip-172-31-31-201:~$
```

That's it for today. I hope you learned something new today.

## Conclusion

By the end of this guide, you successfully packaged, published, and deployed two different applications using Helm charts in a Kubernetes environment. You've also learned how to manage multiple charts within a single GitHub repository. With this foundation, you can now apply the same process to other applications or scale your deployment strategies further!



THANK YOU

**FOR YOUR ATTENTION**

**Follow for more**

Join Discord Community: <https://lnkd.in/dsEdxpst>

Follow on GitHub: <https://github.com/AmanPathak-DevOps/>

Follow on LinkedIn- <https://www.linkedin.com/in/aman-devops/>