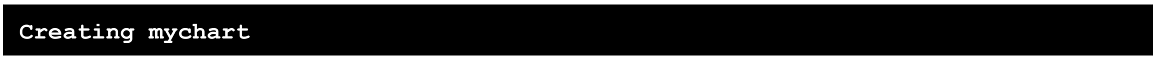
1.1 Helm 包管理工具

1.1.1 自定义创建 Chart

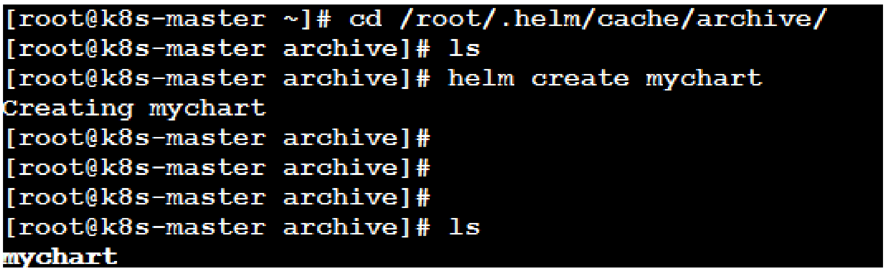
步骤 1　生成一个空的 Charts。

```
[root@k8s-master ~]# mkdir -p /root/.helm/cache/archive
[root@k8s-master ~]# cd /root/.helm/cache/archive
[root@k8s-master archive]# helm create mychart
```



```
Creating mychart
```

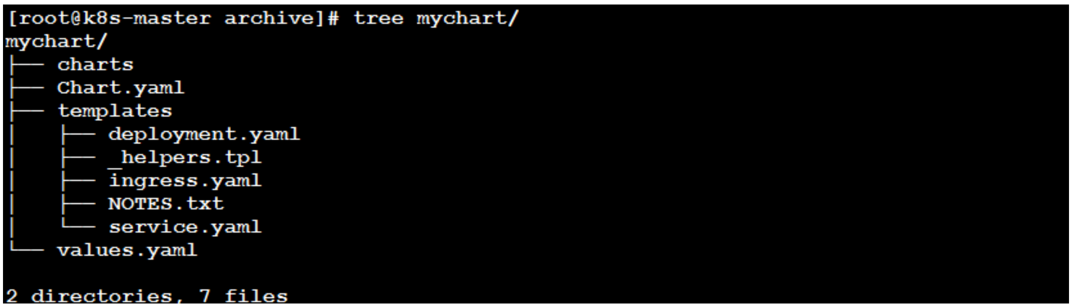
注：该命令能够在当前目录中创建一个名为 mychart 的图表，建议进入 chart 包下载默认路径下创建，默认路径为/root/.helm/cache/archive



```
[root@k8s-master ~]# cd /root/.helm/cache/archive/
[root@k8s-master archive]# ls
[root@k8s-master archive]# helm create mychart
Creating mychart
[root@k8s-master archive]#
[root@k8s-master archive]#
[root@k8s-master archive]#
[root@k8s-master archive]# ls
mychart
```

查看一下该 Charts 文件的目录结构，有以下几个核心文件：

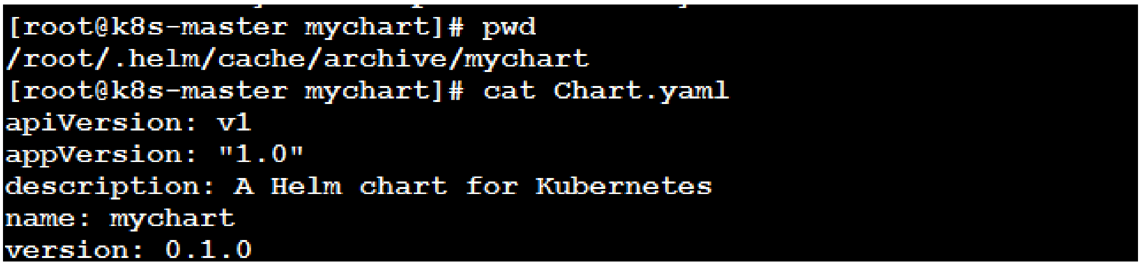
```
[root@k8s-master archive]# tree mychart/
```



```
[root@k8s-master archive]# tree mychart/
mychart/
├── charts
├── Chart.yaml
├── templates
│   ├── deployment.yaml
│   ├── _helpers.tpl
│   ├── ingress.yaml
│   ├── NOTES.txt
│   └── service.yaml
└── values.yaml

2 directories, 7 files
```

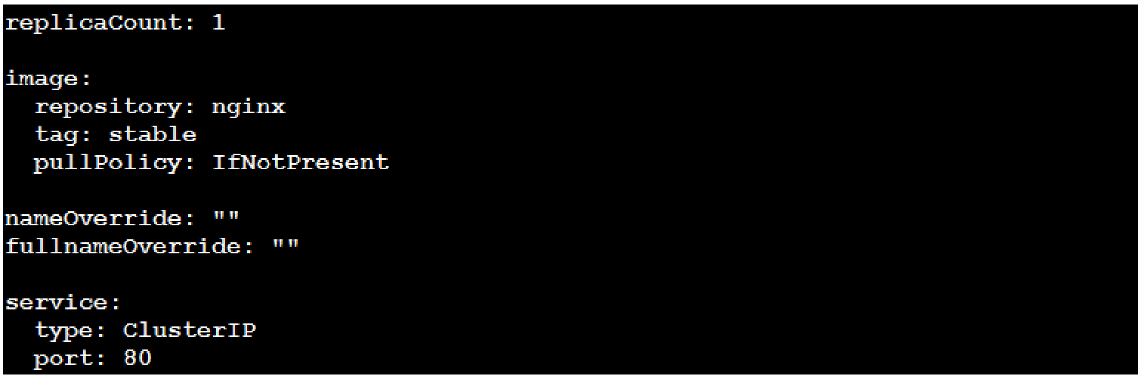
进入以下目录查看 Charts 元数据文件：

```
[root@k8s-master ~]# cd /root/.helm/cache/archive/mychart
[root@k8s-master mychart]# cat Chart.yaml
```



```
[root@k8s-master mychart]# pwd
/root/.helm/cache/archive/mychart
[root@k8s-master mychart]# cat Chart.yaml
apiVersion: v1
appVersion: "1.0"
description: A Helm chart for Kubernetes
name: mychart
version: 0.1.0
```

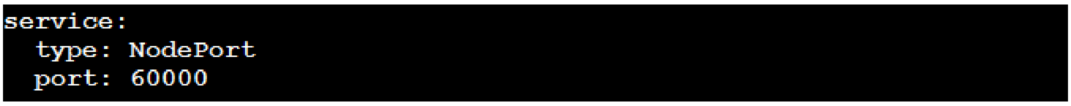
以下为 values.yaml 文件内容中的一段，可以看到定义了一个可以直接安装容器化 **Nginx** 应用的 Charts：



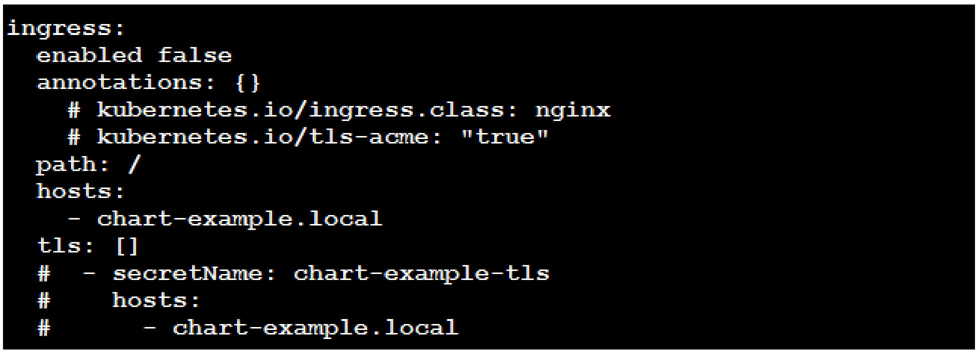
```
replicaCount: 1

image:
  repository: nginx
  tag: stable
  pullPolicy: IfNotPresent

nameOverride: ""
fullnameOverride: ""

service:
  type: ClusterIP
  port: 80
```

步骤 2 修改 Charts 以部署自定义服务。

```
[root@k8s-master mychart]# vim values.yaml
```



```
service:
  type: NodePort
  port: 60000
```

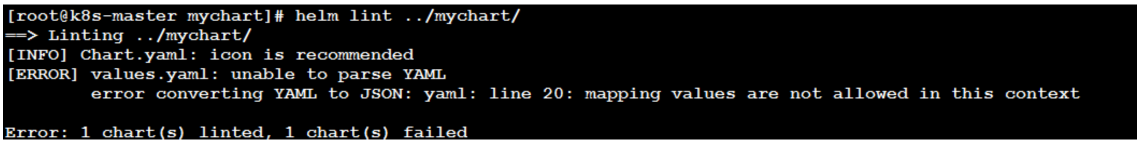


```
ingress:
  enabled false
  annotations: {}
    # kubernetes.io/ingress.class: nginx
    # kubernetes.io/tls-acme: "true"
  path: /
  hosts:
    - chart-example.local
  tls: []
  #  - secretName: chart-example-tls
  #    hosts:
  #      - chart-example.local
```

注：此处故意漏掉了 enabled 后的 “:”

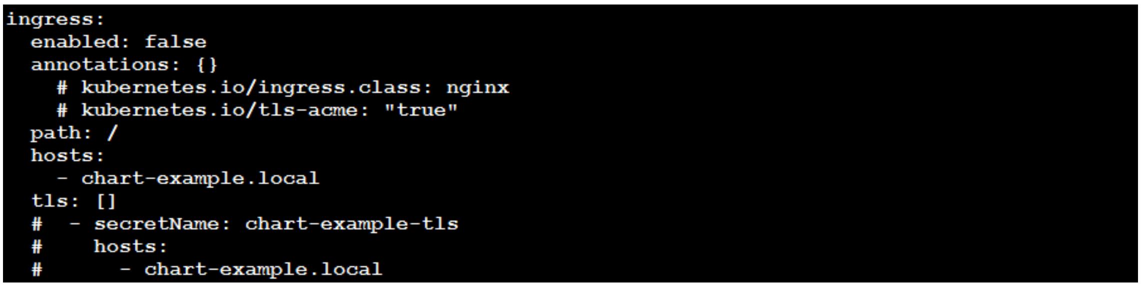
步骤 3 检查 chart 是否有语法错误，注意在创建的 chart 所在的目录运行。

```
[root@k8s-master ~]# cd /root/.helm/cache/archive
[root@k8s-master archive]# helm lint mychart
```



```
[root@k8s-master mychart]# helm lint ../mychart/
==> Linting ../mychart/
[INFO] Chart.yaml: icon is recommended
[ERROR] values.yaml: unable to parse YAML
        error converting YAML to JSON: yaml: line 20: mapping values are not allowed in this context

Error: 1 chart(s) linted, 1 chart(s) failed
```

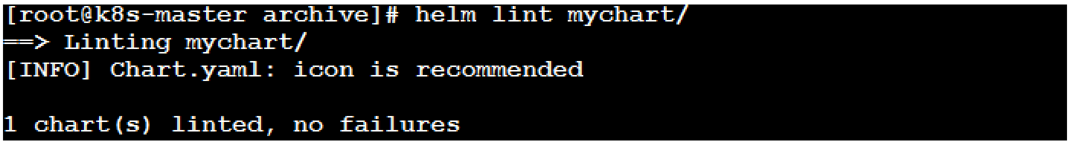
可以看到 helm lint 检查出了错误所在，修改之后



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

运行 helm lint 再次检查正常

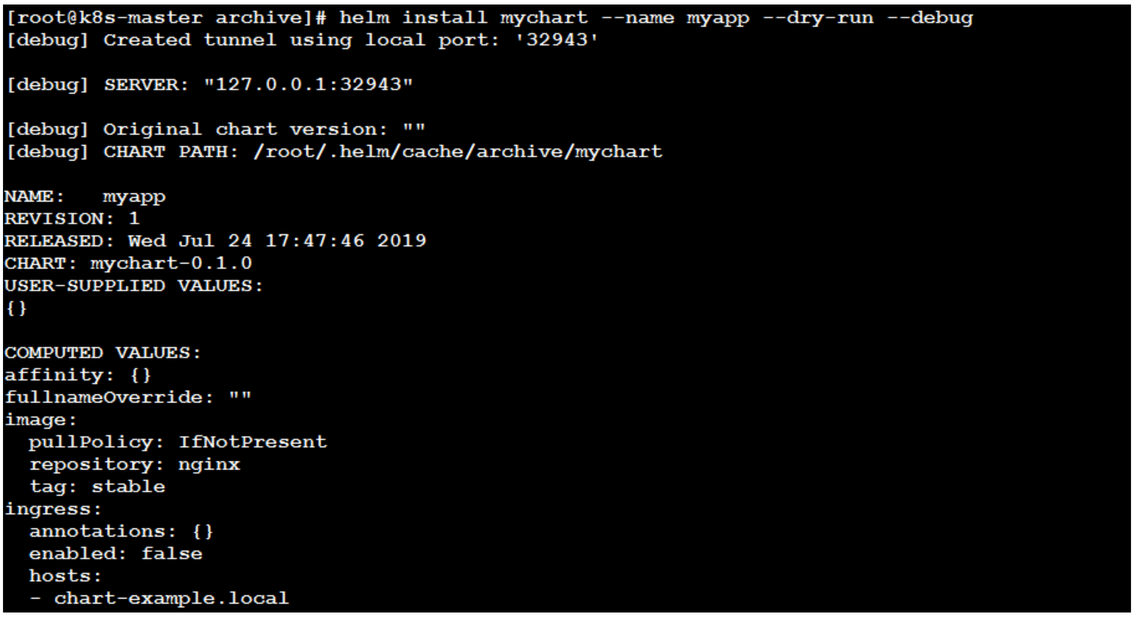
```
[root@k8s-master archive]# helm lint mychart
```



```
[root@k8s-master archive]# helm lint mychart/
==> Linting mychart/
[INFO] Chart.yaml: icon is recommended

1 chart(s) linted, no failures
```

步骤 4 使用--dry-run 和--debug 参数调试 chart 包。

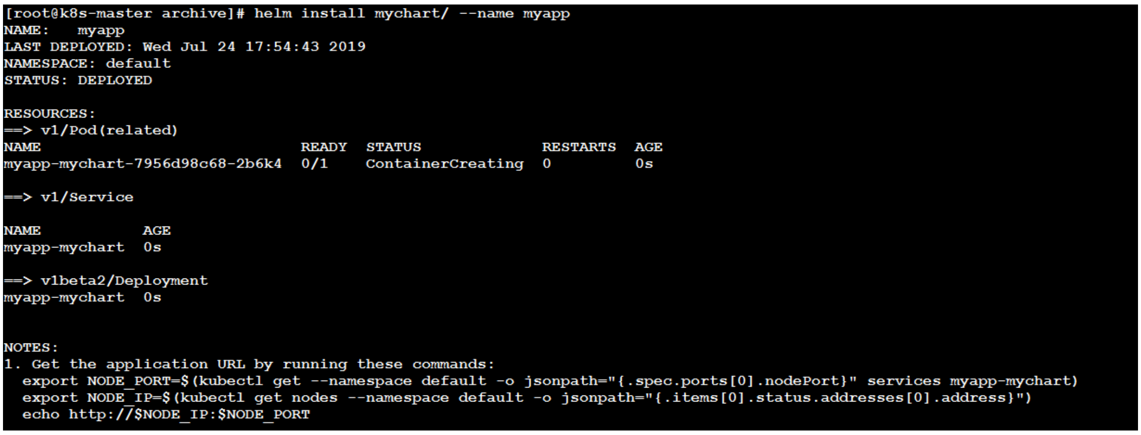
```
[root@k8s-master ~]# cd /root/.helm/cache/archive
[root@k8s-master archive]# helm install mychart --name myapp --dry-run --debug
```



```
[root@k8s-master archive]# helm install mychart --name myapp --dry-run --debug
[debug] Created tunnel using local port: '32943'

[debug] SERVER: "127.0.0.1:32943"

[debug] Original chart version: ""
[debug] CHART PATH: /root/.helm/cache/archive/mychart

NAME:   myapp
REVISION: 1
RELEASED: Wed Jul 24 17:47:46 2019
CHART: mychart-0.1.0
USER-SUPPLIED VALUES:
{}

COMPUTED VALUES:
affinity: {}
fullnameOverride: ""
image:
  pullPolicy: IfNotPresent
  repository: nginx
  tag: stable
ingress:
  annotations: {}
  enabled: false
  hosts:
  - chart-example.local
```

步骤 5 确认上述输出无误后，移除--dry-run 参数再运行一次 helm install 命令。

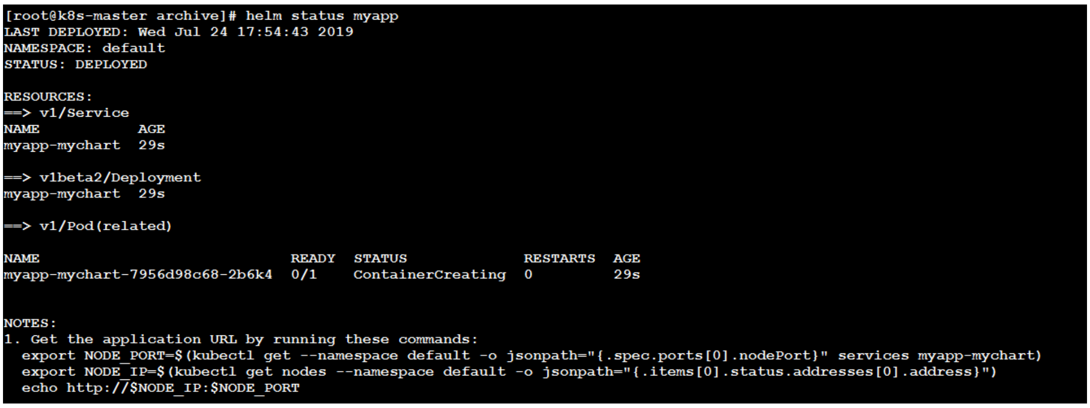
```
[root@k8s-master archive]# helm install mychart/ --name myapp
```



```
[root@k8s-master archive]# helm install mychart/ --name myapp
NAME:   myapp
LAST DEPLOYED: Wed Jul 24 17:54:43 2019
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/Pod(related)
NAME                            READY  STATUS             RESTARTS  AGE
myapp-mychart-7956d98c68-2b6k4  0/1    ContainerCreating  0         0s

==> v1/Service

NAME           AGE
myapp-mychart  0s

==> v1beta2/Deployment
myapp-mychart  0s


NOTES:
1. Get the application URL by running these commands:
  export NODE_PORT=$(kubectl get --namespace default -o jsonpath="{.spec.ports[0].nodePort}" services myapp-mychart)
  export NODE_IP=$(kubectl get nodes --namespace default -o jsonpath="{.items[0].status.addresses[0].address}")
  echo http://$NODE_IP:$NODE_PORT
```

步骤 6　　查看 helm 状态。

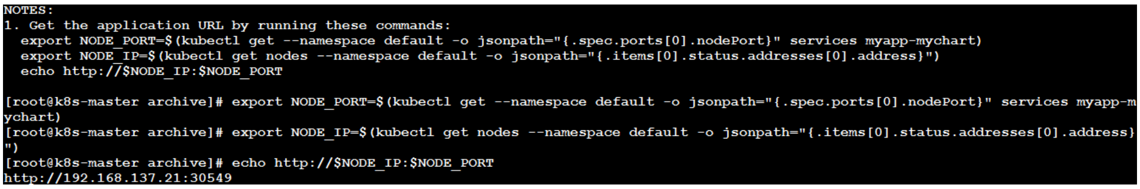
```
[root@k8s-master archive]# helm status myapp
```



```
[root@k8s-master archive]# helm status myapp
LAST DEPLOYED: Wed Jul 24 17:54:43 2019
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/Service
NAME           AGE
myapp-mychart  29s

==> v1beta2/Deployment
myapp-mychart  29s

==> v1/Pod(related)

NAME                            READY  STATUS             RESTARTS  AGE
myapp-mychart-7956d98c68-2b6k4  0/1    ContainerCreating  0         29s


NOTES:
1. Get the application URL by running these commands:
  export NODE_PORT=$(kubectl get --namespace default -o jsonpath="{.spec.ports[0].nodePort}" services myapp-mychart)
  export NODE_IP=$(kubectl get nodes --namespace default -o jsonpath="{.items[0].status.addresses[0].address}")
  echo http://$NODE_IP:$NODE_PORT
```

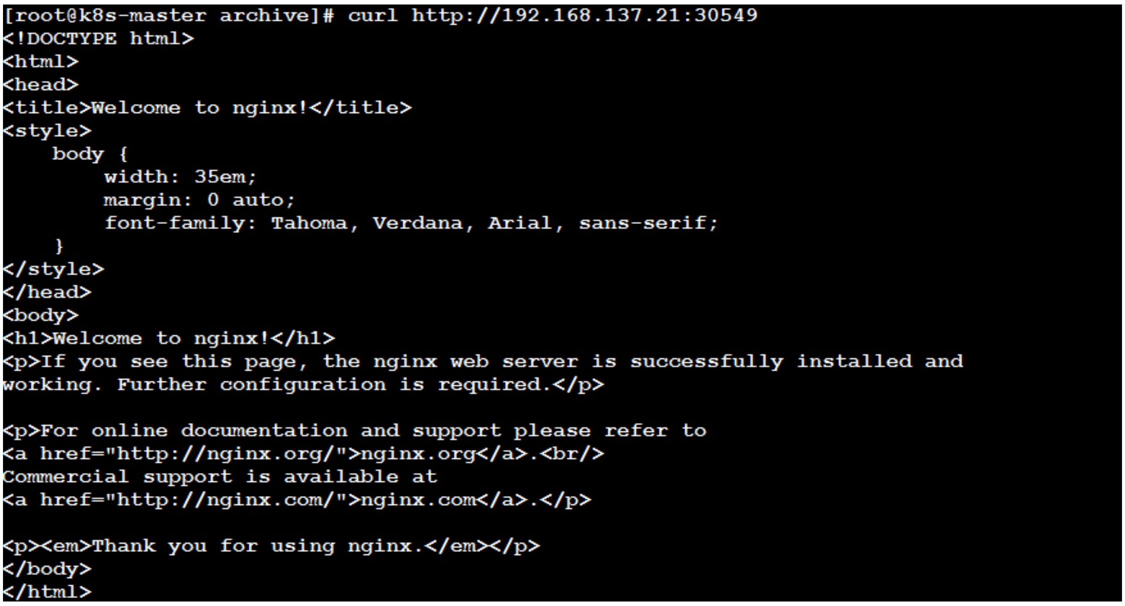
步骤 7　　通过 status 中输出的 NOTES 提示运行相关命令验证服务。

```
export NODE_PORT=$([root@k8s-master ~]# kubectl get --namespace default -o
jsonpath="{.spec.ports[0].nodePort}" services myapp-mychart)

export NODE_IP=$([root@k8s-master ~]# kubectl get nodes --namespace default -o
jsonpath="{.items[0].status.addresses[0].address}")


echo http://$NODE_IP:$NODE_PORT
```



```
NOTES:
1. Get the application URL by running these commands:
  export NODE_PORT=$(kubectl get --namespace default -o jsonpath="{.spec.ports[0].nodePort}" services myapp-mychart)
  export NODE_IP=$(kubectl get nodes --namespace default -o jsonpath="{.items[0].status.addresses[0].address}")
  echo http://$NODE_IP:$NODE_PORT

[root@k8s-master archive]# export NODE_PORT=$(kubectl get --namespace default -o jsonpath="{.spec.ports[0].nodePort}" services myapp-m
ychart)
[root@k8s-master archive]# export NODE_IP=$(kubectl get nodes --namespace default -o jsonpath="{.items[0].status.addresses[0].address}
")
[root@k8s-master archive]# echo http://$NODE_IP:$NODE_PORT
http://192.168.137.21:30549
```

```
curl 192.168.137.11:30549（此处 IP 地址以实际输出为准）
```

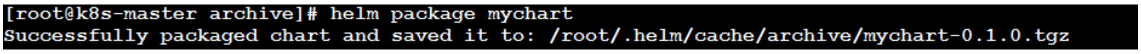


```
[root@k8s-master archive]# curl http://192.168.137.21:30549
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

1.1.2 将自定义的应用发布到本地 Repository

步骤 1　打包自己定义的应用。

```
[root@k8s-master ~]# cd /root/.helm/cache/archive
[root@k8s-master archive]# helm package mychart
```

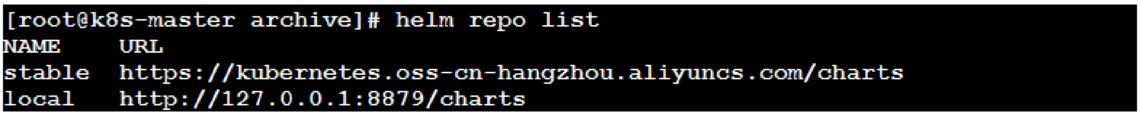


```
[root@k8s-master archive]# helm package mychart
Successfully packaged chart and saved it to: /root/.helm/cache/archive/mychart-0.1.0.tgz
```

注：请在 mychart 文件所在位置执行 package 命令

步骤 2　新版的 Helm 在安装的过程中，提供了两个仓库：stable 和 local。

```
[root@k8s-master ~]# helm repo list
```



```
[root@k8s-master archive]# helm repo list
NAME    URL
stable  https://kubernetes.oss-cn-hangzhou.aliyuncs.com/charts
local   http://127.0.0.1:8879/charts
```

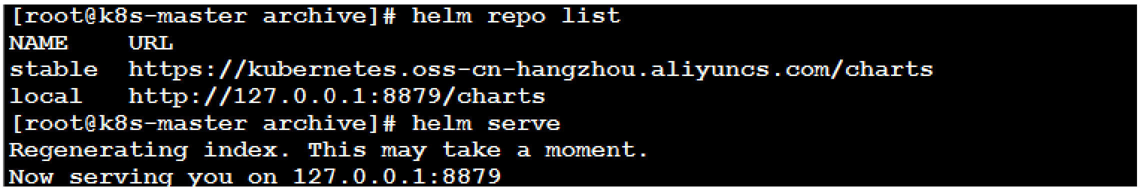
Stable 是搭建手册中阿里提供的默认仓库（因 google 仓库无法连接，采用国内阿里的仓库）

local 即本地仓库

步骤 3　开启配置的默认 Repository Server

```
[root@k8s-master archive]# helm serve
```

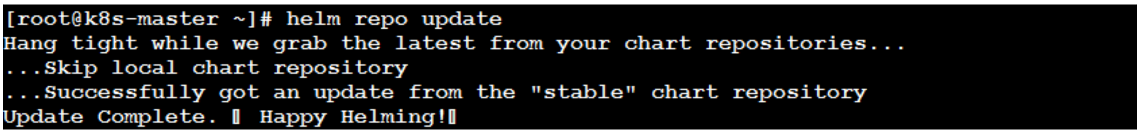
(该命令执行后光标会悬停在另一行闪烁，请另开一个窗口执行下面的其它命令）



```
[root@k8s-master archive]# helm repo list
NAME    URL
stable  https://kubernetes.oss-cn-hangzhou.aliyuncs.com/charts
local   http://127.0.0.1:8879/charts
[root@k8s-master archive]# helm serve
Regenerating index. This may take a moment.
Now serving you on 127.0.0.1:8879
```

步骤 4　更新仓库

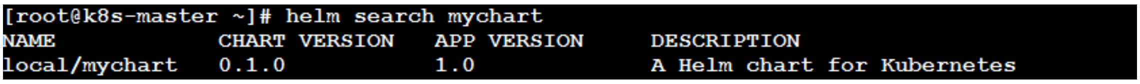
```
[root@k8s-master ~]# helm repo update
```



```
[root@k8s-master ~]# helm repo update
Hang tight while we grab the latest from your chart repositories...
...Skip local chart repository
...Successfully got an update from the "stable" chart repository
Update Complete. ▯ Happy Helming!▯
```

步骤 5　查看我们自定义的 chart

```
[root@k8s-master ~]# helm search mychart
```

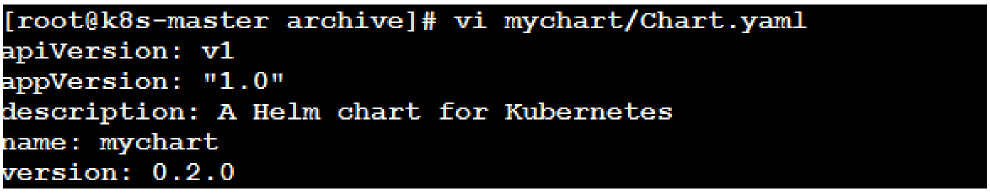


```
[root@k8s-master ~]# helm search mychart
NAME            CHART VERSION   APP VERSION     DESCRIPTION
local/mychart   0.1.0           1.0             A Helm chart for Kubernetes
```

1.1.3 helm 升级和回退一个应用

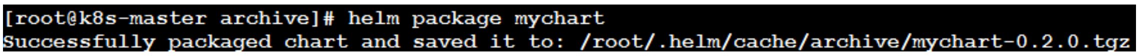
步骤 1　打包自己定义的应用。修改 Chart.yaml 文件，将版本号从 0.1.0 修改为 0.2.0，然后使用 helm package 命令打包并发布到本地仓库。

```
[root@k8s-master ~]# cd /root/.helm/cache/archive
[root@k8s-master archive]# vi mychart/Chart.yaml
```



```
[root@k8s-master archive]# vi mychart/Chart.yaml
apiVersion: v1
appVersion: "1.0"
description: A Helm chart for Kubernetes
name: mychart
version: 0.2.0
```

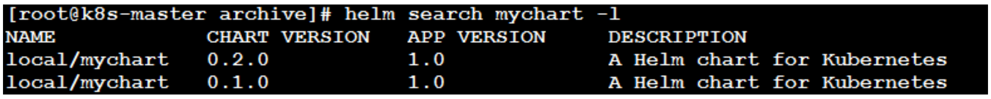
```
[root@k8s-master archive]# helm package mychart
```



```
[root@k8s-master archive]# helm package mychart
Successfully packaged chart and saved it to: /root/.helm/cache/archive/mychart-0.2.0.tgz
```

步骤 2　查询本地仓库中 mychart 的信息，注意加上-l 参数显示所有版本号。

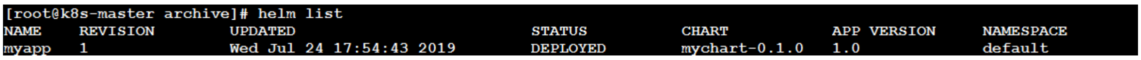
```
[root@k8s-master ~]# helm search mychart  -l
```



```
[root@k8s-master archive]# helm search mychart -l
NAME            CHART VERSION   APP VERSION     DESCRIPTION
local/mychart   0.2.0           1.0             A Helm chart for Kubernetes
local/mychart   0.1.0           1.0             A Helm chart for Kubernetes
```

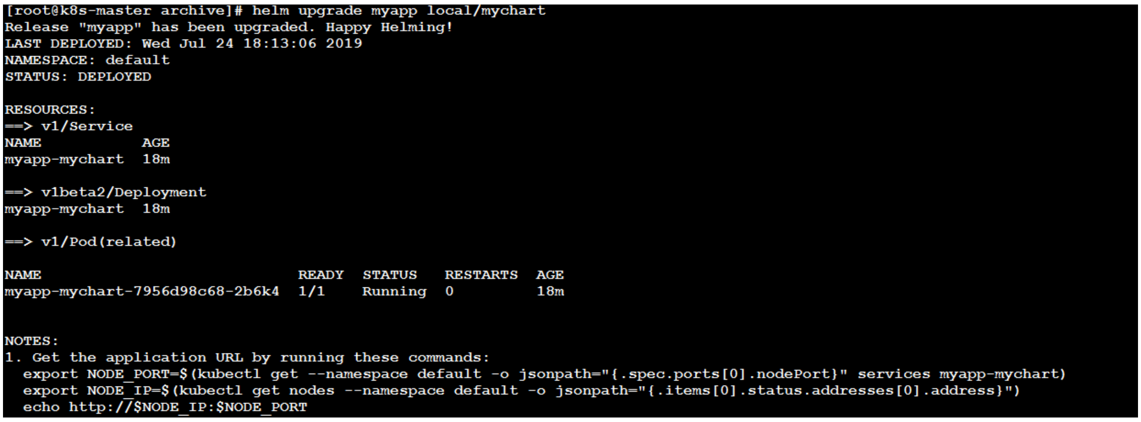
步骤 3　使用 helm upgrade 已经部署的 release，可以通过--version 来指定版本号，如果未有指定则自动升级到最新版本。

```
[root@k8s-master archive]# helm list    #查询使用 mychart 部署的 release 名称
```



```
[root@k8s-master archive]# helm list
NAME    REVISION        UPDATED                         STATUS          CHART           APP VERSION     NAMESPACE
myapp   1               Wed Jul 24 17:54:43 2019        DEPLOYED        mychart-0.1.0   1.0             default
```

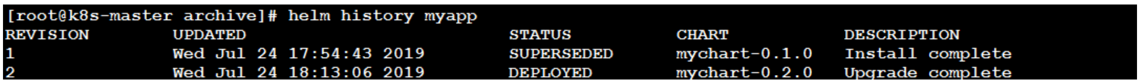
```
[root@k8s-master archive]# helm upgrade myapp local/mychart
```



```
[root@k8s-master archive]# helm upgrade myapp local/mychart
Release "myapp" has been upgraded. Happy Helming!
LAST DEPLOYED: Wed Jul 24 18:13:06 2019
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/Service
NAME           AGE
myapp-mychart  18m

==> v1beta2/Deployment
myapp-mychart  18m

==> v1/Pod(related)

NAME                            READY  STATUS   RESTARTS  AGE
myapp-mychart-7956d98c68-2b6k4  1/1    Running  0         18m


NOTES:
1. Get the application URL by running these commands:
  export NODE_PORT=$(kubectl get --namespace default -o jsonpath="{.spec.ports[0].nodePort}" services myapp-mychart)
  export NODE_IP=$(kubectl get nodes --namespace default -o jsonpath="{.items[0].status.addresses[0].address}")
  echo http://$NODE_IP:$NODE_PORT
```

步骤 4 使用 helm history 查看 release 的历史

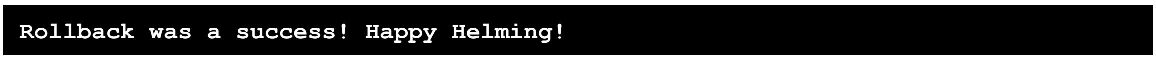
```
[root@k8s-master archive]# helm history myapp
```



```
[root@k8s-master archive]# helm history myapp
REVISION        UPDATED                         STATUS          CHART           DESCRIPTION
1               Wed Jul 24 17:54:43 2019        SUPERSEDED      mychart-0.1.0   Install complete
2               Wed Jul 24 18:13:06 2019        DEPLOYED        mychart-0.2.0   Upgrade complete
```

步骤 5 回退一个应用

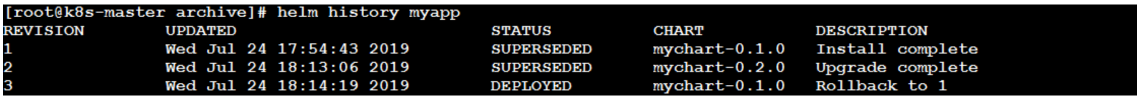
```
[root@k8s-master archive]# helm rollback myapp 1
```



```
Rollback was a success! Happy Helming!
```

步骤 6 回退后查看 release 历史

```
[root@k8s-master archive]# helm history myapp
```



```
[root@k8s-master archive]# helm history myapp
REVISION        UPDATED                         STATUS          CHART           DESCRIPTION
1               Wed Jul 24 17:54:43 2019        SUPERSEDED      mychart-0.1.0   Install complete
2               Wed Jul 24 18:13:06 2019        SUPERSEDED      mychart-0.2.0   Upgrade complete
3               Wed Jul 24 18:14:19 2019        DEPLOYED        mychart-0.1.0   Rollback to 1
```

1.1.4 搭建 helm 私有仓库

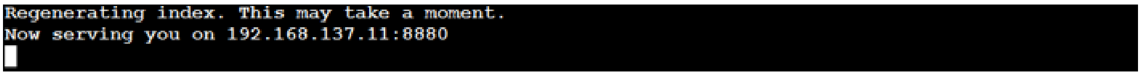
步骤 1 创建一个 repo 目录，用来存放 chart

```
[root@k8s-master ~]# mkdir -p /dcos/appstore/local-repo
```

步骤 2 启动本地 repo 仓库的服务

#IP 地址以实际情况为准（此处执行后光标在另一行悬停闪烁，请另外打开一个终端执行下列命令）

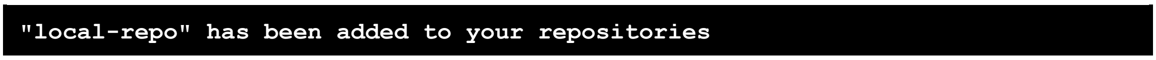
```
[root@k8s-master ~]# helm serve --address 192.168.137.11:8880 --repo-path
/dcos/appstore/local-repo
```



```
Regenerating index. This may take a moment.
Now serving you on 192.168.137.11:8880
█
```

步骤 3 将私有仓库添加到本地

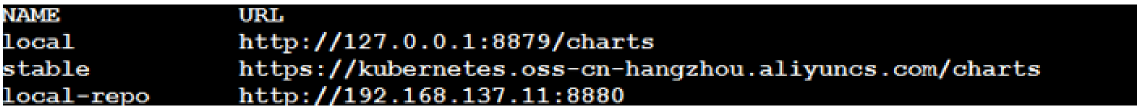
```
[root@k8s-master ~]# helm repo add local-repo http://192.168.137.11:8880
```



```
"local-repo" has been added to your repositories
```

步骤 4　查看是否添加成功

```
[root@k8s-master ~]# helm repo list
```



```
NAME            URL
local           http://127.0.0.1:8879/charts
stable          https://kubernetes.oss-cn-hangzhou.aliyuncs.com/charts
local-repo      http://192.168.137.11:8880
```

步骤 5　拷贝一份 chart 包到对应路径下

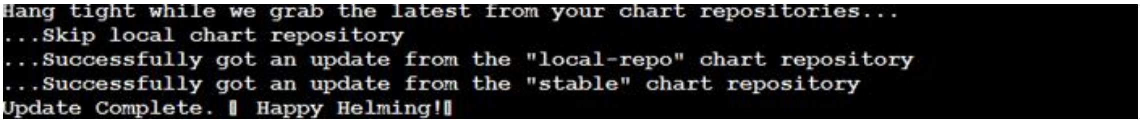
```
[root@k8s-master ~]# cp ~/.helm/cache/archive/mychart-0.1.0.tgz
/dcos/appstore/local-repo
```

步骤 6　生成仓库 index 文件

```
[root@k8s-master ~]# cd /dcos/appstore/local-repo
[root@k8s-master local-repo]# helm repo index . --
url=http://192.168.137.11:8880
```

步骤 7　更新仓库

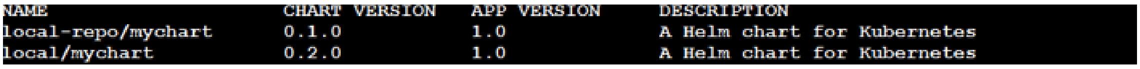
```
[root@k8s-master ~]# helm repo update
```



```
Hang tight while we grab the latest from your chart repositories...
...Skip local chart repository
...Successfully got an update from the "local-repo" chart repository
...Successfully got an update from the "stable" chart repository
Update Complete. ▮ Happy Helming!▮
```

步骤 8　查看 chart

```
[root@k8s-master ~]# helm search mychart
```



```
NAME                    CHART VERSION   APP VERSION     DESCRIPTION
local-repo/mychart      0.1.0           1.0             A Helm chart for Kubernetes
local/mychart           0.2.0           1.0             A Helm chart for Kubernetes
```