



SHANE RAINVILLE

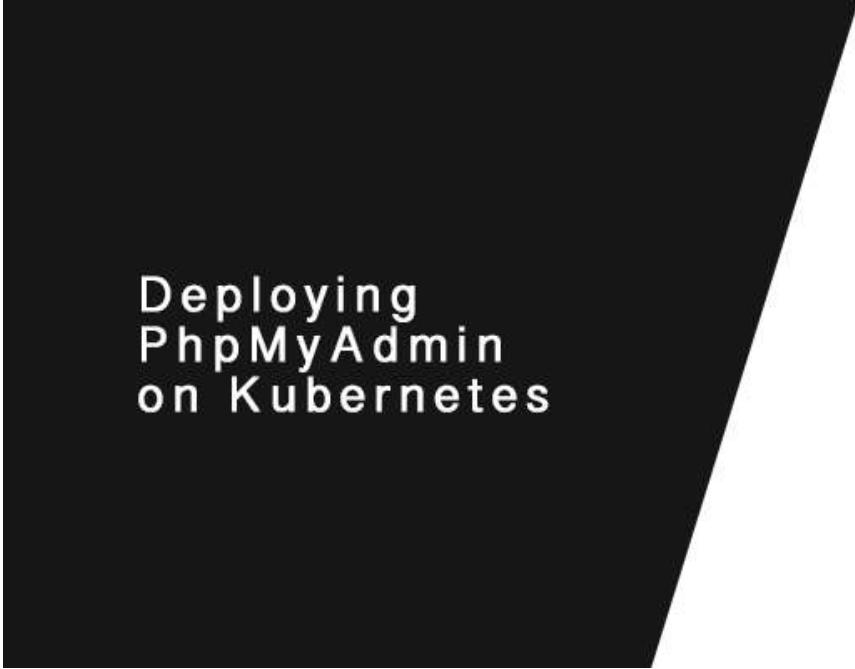
MAY 22, 2019

Deploy PhpMyAdmin on Kubernetes to Manage MySQL Pods

Contact & Enquire Now



Crest Plywood



Deploying
PhpMyAdmin
on Kubernetes



Overview

In this tutorial, you will learn how to deploy PhpMyAdmin to Kubernetes for managing your MySQL server pods.

It is very likely that your MySQL Server is only available from within the Kubernetes cluster, as services like this shouldn't be exposed to the public Internet. However, you will need a solution to manage your databases, and one that doesn't include running exec to gain access to the pod or pods directly.

PhpMyAdmin is a very popular web frontend for managing MySQL servers. It is a good candidate for being able to manage the MySQL servers hosted in Kubernetes, as it enables us to manage all of our databases easily and effectively.

Getting Started

Resources used for this tutorial are available from Github. You may freely download them to follow along or as templates for your own environment.

- [Kubernetes Templates](#)

Deploying PhpMyAdmin Pods

Secrets

Our PhpMyAdmin deployment will link to an already existing MySQL service, rather than allowing users to specify the address to the service. PhpMyAdmin will need to know the root password of the MySQL server it is connecting to.

A secret has already been created to store the root password for MySQL. We will target our PhpMyAdmin at this secret, that is shared with the MySQL pods. An example of the secret data is shown below for reference.

```
---  
apiVersion: v1  
kind: Secret  
metadata:  
  name: mysql-secrets  
  type: Opaque  
data:  
  root-password: c3VwZXItc2VjcmV0LXBhc3N3b3Jk
```

The root-password value is an arbitrary key name created to store a base64 encoded string of the MySQL server's root password. All secret values in a Secret resource file must be base64 encoded.

The PhpMyAdmin Kubernetes deployment will reference the root-password key to set the MYSQL_ROOT_PASSWORD environment variable, when the pod's container starts.

Create a Deployment Resource

Create a new file named deployment.yml.

```
touch deployment.yml
```

Add the following contents to the deployment.yml file.

```
---  
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: phpmyadmin-deployment  
  labels:  
    app: phpmyadmin  
spec:  
  replicas: 1  
  selector:  
    matchLabels:
```

```
app: phpmyadmin
template:
  metadata:
    labels:
      app: phpmyadmin
spec:
  containers:
    - name: phpmyadmin
      image: phpmyadmin/phpmyadmin
      ports:
        - containerPort: 80
      env:
        - name: PMA_HOST
          value: mysql-service
        - name: PMA_PORT
          value: "3306"
        - name: MYSQL_ROOT_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysql-secrets
              key: root-password
```

Create the deployment using the kubectl apply command

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

Create a Service Resource

Kubernetes pods are ephemeral and their IP address lives only as long as the pod does. This is problematic for long-term accessibility to PhpMyAdmin, which is why we create a service resource for our pods.

A service resource will be assigned a static IP address, and all requests to this IP address will be forwarded to the backend PhpMyAdmin pods. The service resources is coupled to the pods via labels.

Create a new file named service.yml

```
touch service.yml
```

Add the following contents to it. Since our PhpMyAdmin pods are assigned a label of `phpmyadmin`, we will use this as a selector in the service.

```
---
apiVersion: v1
kind: Service
metadata:
  name: phpmyadmin-service
spec:
  type: NodePort
  selector:
    app: phpmyadmin
  ports:
  - protocol: TCP
    port: 80
    targetPort: 80
```

To create the service resource we use the `kubectl apply` command.

```
kubectl apply -f service.yml
```

Ingress Controller

To simplify exposing PhpMyAdmin to the public Internet, an ingress resource will be created. The ingress resource will point to the PhpMyAdmin service resource.

Creating a Basic HTTP Ingress

Create a new file named ingress.yml

```
touch ingress.yml
```

Add the following contents to it.

```
---
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: phpmyadmin-http-ingress
spec:
  backend:
    serviceName: phpmyadmin-service
    servicePort: 80
```

Create the ingress resource using the kubectl apply command.

```
kubectl apply -f ingress.yml
```

Securing the Ingress with TLS

The above ingress example exposes PhpMyAdmin over HTTP, which means all sensitive information is accessible by any network our traffic is traversing through. This is obvious a bad practice for production environments.

To secure the Ingress endpoint you will need the following:

- Certificate Authority Certificate
- TLS Certificate
- TLS Key

All three files will need to be stored as Kubernetes secrets. Create a new secret using the following command, replacing the filenames with those that match your environment.

```
kubectl create secret generic ingress-tls-secret --from-
file=tls.crt=server.crt --from-file=tls.key=server.key --from-
file=ca.crt=ca.crt
```

Alternatively, you can create another secret resource file and store the base64 encoded string of each file. To base64 encode the files, use the following command against each file.

```
base64 -i <filename>
```

With the base64 encoded file values, the following is an example of a secrets file storing the required certificates and keys.

```
---
apiVersion: v1
kind: Secret
metadata:
  name: ingress-tls-secret
type: Opaque
data:
  ca.crt:
LS0tLS1CRUdJTiBQUk1WQVRFIEtFWS0tLS0tCk1JSUpRZ01CQURBTkJna3Foa2lHOXcwQkF
RRUZBQVNDQ1N3d2dnA29BZ0VBQW9JQ0FRREJKTUJvNVdkR1ZrYmIKWEY4QU1Eb01LdWgrbm
JGNTg0dE1RY1pGUXRJWU1hWnI4avloc2hUdFJXR1IxVWc5M0tnUW4wY2xSQWUwUEs5ZwppS
OFSeW5MaWdIOG5JZEFwY2tZcklr0Vkl25yYW8wYUtyL0M3U3hObFkxL2tnWS9NTFzkMVZW
eWtDMWNYS0J5CmE5bGt3M0hnQk1tOWt4eXdPZVdFcTJtN3ZnbVpvZmZRTmIvSUt0aTFwRTB
WWjhNdGZwb1Bva21EYz11MmkxWEUKVjNwaUtTRnkveW4wQnkzV1p4YzE5ais0Rm53M1ZURj
RnQUtPR3VTUUxOOhHMSWJBQUxFs3hGQmFhR11kd212Ygp0MFVxRWxBOEcvQkwxRVBLYUh3a
T11QXV1ZjAxNTBqSWVWSm5PTFhadWhBaHdxK2lpbEFTb1E4UVVxOWtzRk44Cn11Z2hjUnNV
bDBkN1N0VDE1Y0w3QVN2Z0djR2FPK3JIVUttRnU1OHEzMHVSYmZwL0QyRUUyZTk5eXvhT11
WT00KK3FDOTVTeDdqWHZ1a1JiRkd6c3N1NkN1eDMyT2JjU1VxZDJQTjg2S2hveXFUeHR2N0
xRU3V1bEMyeH1KN0tTZwpIc1ZQY3BEU3REZ1kzZ11NMEw1Wnk1TEJmd2tEMC91a0Q4ZE56K
1FPalVaY2cxS0gyY0NrK1VSM1NsZHRpNFdNCkdQMGFwOXhtalc3R255S1hFRFdjVFVHN01T
Swt2QzNtejBhRTkzQ0hSR1VlejdWbWJqeEtZbzU0N0JhbG9kcWgKOE9tdVpz1hhMTBDzz1
xMFRVWVpQbTNpQ3NOR2ZRWmFRc2FWNX1vazMrS1IxaU0yL21kQ3dQQ21TUUDac3p4bApIR2
9iZ0FFY3NRNk1iVXFEWwtXT09PcnY0bHZJWVFJREFRQUJBb01DQUZHqk90RUxQYkdvNXUzY
```

TvSD25QZFlFCmhXZ3BFN1BxNm5XYnJCZStRdnFDdVVBRVI2TWdpMFEExYXg4bmFmenhDTnM2
MjhNV0drb3M4Y1ZDOG9tUGJUdTQKRGZCdGpISnhEcnRWRVdEZf1GRy8vUnQxTjcwQ2FSZTB
iS25RN1dIdEpCd2MwcG90elhhVWdBREpFa3c3VGVyRApHbjh0Z11YQWFPSzdrenJlenc1T0
UwMkiOULNQUURRbX1ZQ204TzVaNXBtQ1dwTm1hT2wza3hIditHMUQvaUdSCkhCc21qR2xWe
DYyMTVOM0NYMWh1U3hvV3F2MWN5K2NyaUdRdjdnMULQVzNWdWpMc3Eyc1RiWDZpc0VWNHhU
TVkKZH1jSXhESytOT1JzV2Q1RERGK2E0bFAzSHBjQWozejV6N1R2aXo2RUw1WXM4VFpxTG1
uSFh2UG1JRnM4Wjdoawoxeks3NjBKK1VuemhVaERwSy9WVUdiOEc3cEkxMERwS1FISENUMm
1GSTZvdVZoUVRnd0NHUCtnTWVmSDg2VmU0CnVzZmMyZVJBZEE3ZVFrOVdrK0RpSWkvTz1YW
itSzzNraW1sRWs0eUFQUXZyUTg2b1I0UmxFampqZjhGQTNZTysKTzRbVQwWXpJb1Y0M290
bjhkNXdhRDhsb2hrcnZPaGNYZky0eGdCNWp1YW05NkhMc0N1V25Xbm8xeThPTFNXeApGWjA
2Y3Jkb3ptVzVSZkFpvCtIcndlDktSSmZqr1psd0dmaW1VTEszSVpQTjVjcG03ZjdMUDMwWX
EzQWFxaUJScjYxakp1QzV5c0pEWXFYWituRE9yN0RraU1OYWJpNWcvVm9RQ1BTcjIyTEcwe
UtGRGdpR0I0UmRCdzBlbFBaQkcKYS9aZm5VQmhYVC81TTlieWhiWWhBb01CQVFEchJ1RXFn
a1RZUmhYbncydmRuQkRrVW5r0WJYcGJuaFJQRzc4YQo2TGJqVzZXemhqVExZcmdHdFlnNUh
TSHJJavRrb0NyNFJ3M1N6W1NTMudJYTBoS1BvQnI5V1F1TUNvcjlyc1pYckNyQktQT1B1R2
ZWNX1NNHVMWmZSRTdFWHirTENWbUJLWmtJTFpwaTV2R3RvVzV0VG11VDFzck1KVzVhbHROV
DYKczhmdy9qYmh50W15dk5NWE1NaC9ZempnWmhFdFgrcCtXcmZ6RUZ4cS9tbzc0Qm110Edr
cDZTcDB1eU94QVBVVAowTm9CWjFZY0pZT2ViVHdvZzJNQ1F4N1ZqZxhUN2tQRk12ZzdnVEN
GY2ZTSC93dGw2Qz1SLzFJUmpwUjJIQS9ZCmRkN3dKTnZ6SFpYSFgvY2FDK1NSSUhqVUFTWD
doR0ZGNFhNbDBOOVB4cytNRkNzZkFvSUJBUURUbHNFeWZBY1cKVS96N3BndVJqdTFqQWZMN
zdpRm04VkJpZUpaajhWSvhCbjBmY3FyaG1ENDgvZ0d0M31jtZVIUm8ycExmcWQxbAorT1RJ
UytvYUtmavaUVTY2VrQnVMZWhnMDEyZGFbTkNMQmphZ0VkbXZhSDdTRXQzb2t2SnZCNWJWQ3J
jbT1JdExSCi8rd1FQc2oxWVFBMjNCWHBSSzBETXpJaW1xaHN4L3hFSjg1am5Qam4yOW0wOV
U5YWy0VXA2ZEhmeEt5NzJMwGoKdG1nST1XwmJZrjZXTnE2a1FCVXpQSDQzL11MMGpKQi81Y
ytmcyxUGZhci93eU1xT0trV09RQ0hnNjFMcEhtTApkbzFCbHdCM3RTMF1MVd2SmtTSXkz
RUQ5c0pUa1QwMkk5QU9ZRzcxU1N2SDVJTjYweEJwdm1LTVPyb2Qwd1Z4Ck1SQu9vdWN1L1J
4L0FvSUJBQXdqT1RiTUJWOW1aSRoTlowUkdoK0IyU0pEcVhOUXhxVkZQNnZhSW9XMG9PcX
UKTWRmt2pOahNYV0w2REDUMkpReHE5VkJvQUdQT29jVHVteG5oaDIXR1Y2Q0pHMGMrMVBGU
VBxVU92UkZ1RUVpeAp1SHBXMzc2dVI1Qm9nVzVGaUhXbm51QytadmZJb0c0bFJ6N29ZeXo0
ZFzmOXJDYjBSdUxjZmFOMzz3NnRaW1JjCnppaWFmaXhaVzVDAVNyWD1GeT1RS0J1Z0RHG5
sN1ZIQ2FXS2dqMno1d0p5UmU1Q09oeGJmR2xBRT1LMjExWW0KN0tzeDZVb0JhUWNMd1c3ck
xEYy96TzFWTkd1QzBrNWNHT05uYitRZkp6WBHT2dvaTdXR3g1dnFvMlc5eEY3ZwpUb05Ja
k1FaGxRR2t1aUhvN2zzWFSOEVFc01JRWxGR2RrbEVVdWtDZ2dFQVZ6NnZOUERkUDBDc2JO
cy9vK3BLClN1ld1ExRDZSzVZBQmRFa0N3ZjBNZTBBS3N1l25aQTE0bEUrc1E1UmpGNG50MGh
wbU8yRmdVQjY3dnpkR05UbWYKSm5VQm84b1BIREFzSEV0U1hoV2hTeXzzM3NJeG9JUTRQQT
BuaVNSWUdUQnR1Q3dPcGpWSXRweVBLdmVvaHoxVApDRXZob1FqWFNoMS9DN08zeFBzRk10a
3pDZkg4MUNYYjgvaSttck5BSmxNcVhkbEwzRHowYXd1WWhERWozb096CkhMODViYktheTBD
alR0VH1mWUhFUjJoc0h3N2ZsalljSDAvSuhtMjEwdHI0TStpqXVWTd1adm9qZnFzVDMwaE8
KZ2g1c3EwYmROWWFsZUJ2MGJ0N1JYw1NRd296UzA4R00xdVNMcmtaL1d3ck11ZfhwNGx4Tk
pqT2JJMW01ZDYrWQpZd0tDQVFFQXY5aVNGUkhFWmJkSWJaak12ekUvV3FaRDJWVVMxb05XV
zU5ZXJibzBqOFkzUWJNb1RPV1lEQzJrCnhLZG1JQ1pZNEZxeW9yd2h3aFd2aHNWajkzVzV0
VzRYTGx4SmI3cWI5QkhkcU1ORnc2SURCaFRMdWF0NFk4UWEKWGRUbWNCEk1EWEnrZmV3MjI

3RGk0bjFLVGpiMkt1K3FYUGHpeGNZUnN4cFdmbVFmS3pVK2EweFNNTy9vQWIwMAo0R1N0WG
9SY0xyRHczWnZ0cmZkK0ZWdytKZz1OeF11d2VhcxFWOVM5SnNHZFdoTHdZSk1vTUNGUGtHR
Wx2TTB5CjJvMFBzMk50aktvRWtvRzzuMzbqQ1A0cUVNMXJodURCa2p5amEwU1lqaklnOFJT
UGgrbzJobHU1b2VoL0ZBrn4Kd2dGS3NQb3g1cmptM09OS0NHSXVmV0NLRmUxMEDRPT0KLS0
tLS1FTkQgUFJJVkfURSBLRVktLS0tLQo=

tls.crt:

c3VwZXItc2VjcmV0LXBhc3N3b3JkLS0tLS1CRUdJTIBDRVJUSUZJQ0FURS0tLS0tCk1JSUZ
VakNDQXpvQ0NRQ0ZmV0owVURPYzFEQU5CZ2txaGtpRz13MEJBUXNGQURCck1Rc3dDUV1EV1
FRR0V3SkQKUVRFUU1BNEdBMVVFQ0F3SFQyNTBZWEpwYnpFUU1BNEdBMVVFQnd3SFZH0X1iM
jUwYnpFU01CQUdBMVVFQ2d3SgpVM1Z5ZG1WeWJHRmlNU1F3SwdZRFZRUUREQnRrWlcxdmNH
aHdiWGxoWkcxcGJpNXpaWEoyWlhKc11XSXVZMKV3CkhoY05NVgt3T1RJek1ESXdNe1U0V2h
jTk1qQxDove15TURJd016VTRXakJyTVFzd0NRWURUVFHRXdKRFFURVEKTUE0R0ExVUVQX
dIVDI1MF1YSnBiekVRTUE0R0ExVUVCd3dIVkc5eWIyNTBiekVTTUJBR0ExVUVdz3dKVTJWe
QpkbVZ5YkdGaU1TUXdJZ11EV1FRRERCdGtaVzf2Y0dod2JYbGhaRzFwYmk1elpYSjJaWEpz
WVdJdVkyRXdnZ0lpCk1BMEdu3fHu01iM0RRRUJBUVVBQTRJQ0R3QXdnZ01LQW9JQ0FRREJ
KTUJvNVdkR1ZrYmJYRjhBTURvSUT1aCsKbmJGNTg0dE1RY1pGUXRJWU1hWnI4aVloc2hUdF
JXR1IxVwc5M0tnUW4wY2xSQWUwUEs5Z21LQVJ5bkxpZ0g4bgpJZEFwY2tZck1rR0Vkl25yY
W8wYUtyL0M3U3hObFkxL2tnWS9NTFzkMVzWeWtDMWNYS0J5YTlsa3czSGdCTW05Cmt4eXdP
ZvdFcTJtN3ZnbVpvZmZRTmIvSUT0aTFwRTBWWjhNdGZwb1Bva21EYz11MmkxWEVWM3BpS1N
GeS95bjAKQnkzV1p4YzE5ais0Rm53M1ZURjRnQuTPR3VTUUxOOHhMSWJBQUxFs3hGQmFhR1
1kd212YnQwVXFFbEE4Ry9CTAoxRVBLYUh3aT11QXV1ZjAxNTBqSWVWSm5PTFhadWhBaHdxK
21pbEFTb1E4UVVxOWtzRk44eWVnaGNSc1VsMGQ3C1N0VDE1Y0w3QVN2Z0djR2FPK3JIVutt
RnU1OHEzMHVSYmZwL0QyRUUyZTk5eXvhT11WT00rcUM5Nvn4N2pYdnUKa1JiRkd6c3N1NkN
1eDMyT2JjU1VxZDJQTjg2S2hveXFUeHR2N0xRU3V1bEMyeH1KN0tTZ0hzV1BjcERTdERnWQ
ozZ11NMEw1Wnk1TEJmd2tEMC91a0Q4ZE56K1FPalVaY2cxS0gyY0NrK1VSM1NsZHRpNFdNR
1AwYXA5eG1qVzdHCm55S1hFRFdjVFVHN01TSwt2QzNtejBhRTkzQ0hSR1VlejdWbWJqeEtZ
bzU0N0JhbG9kcWg4T211WnNmWGExMEMKZz1xMFRVWVpQbTNpQ3NOR2ZRWmFRc2FWNx1vazM
rS1ixaU0yL21kQ3dQ21TUUDac3p4bEhHb2JnQUVjc1E2TQpiVXFewxtXT09PcnY0bhZJWV
FJREFRQUJNQTBHQ1NxR1NjYjNEUUVQC3dVQUE0SUNBUUFry0VYNE1MWkgwZ3cvCmd5NVFBd
G8rdW5qb2MrZ3FlazhKQytKVEwxS0ZuV1dIamk3Z1RzYzF6dkx0aNbd0x2VGFyS1ZDbDN1
VGN6SG8KR11qaFFib3ZMc3VjT31xY0xYcFJEg93QUN1M3hTWEhocDRtzGFZYmdmc09ps08
1cmI3U2J2bnZ3UGJ1U01IMgpQZUFkR3BvSWVhd1hKaW1SRxOrY0JPZFd5VTUrS0w2N3zjOD
JiQX1XV1RpVzdaSTBobHivTmkwTks1S1pmddRsCk1vQutabhKZDdochA5YzB2dERneE5RR
GEzQ0xLUm96UmpZcUt5dHzhZTFPTVhoL1NLY0NiakpDQVJYnm03YXQKZG9ieGlWSHVIOWhD
cUoyTmZoN2JFcNv6UE5Cz1zaV294dTE1SE8wdnRYODhzUE1YzxJzW1Rwell3TjdCUEpUaAp
LVi9GREh2ZWJzWFY1ZDVhSTR1RFY0RnpmUFM2b2hJRTZwc0VVOFQyTkkMZW5vVzdRdk1YVT
NwMm1HRG1JWGQzCndFS3ZXWkr6dXRzaGluWEN4RkZuNVNVNmhCUXJIR3UxWn15YjZwSEwvV
E1oTzVwbkF0ajA1W1N2a20yVUYYbjgKa3RKcDFLYWNST1pMOFFzOWhvYXhYcjZKYm91OTZT
N3djRDRFYkxwVkhvQjArbU91YWh0WUI5ZDUvMzhBa0tJTApPcz1la1FCc3J1VmZUWEEvUnN
GUHRBV3JpZ1NLVHNpUmp5V1BJSndlU52c2E2UHROenZnZTRhtU1PYkjKchG3CkxPLy93aX
hiL0dDSkg50XBBN2tSeHV2eUdDd081UzhjZwdzeEdxMHRBS01maHUyZE1zNmNZbHJinjFVb
1FoNHQKL2gwdTQ2TmNIY1RZeXVBbFF1MmNYeEdHvmcvU2pnPT0KLS0tLS1FTkQgQ0VSVE1G

SUNBVEUtLS0tLQo=

 tls.key:

 LS0tLS1CRUdJTIBDRVJUSUZJQ0FURS0tLS0tCk1JSUZVakNDQXpvQ0NRQ0ZmV0owVURPYzF
 EQU5CZ2txaGtpRz13MEJBUXNGQURCck1Rc3dDUV1EV1FRR0V3SkQKUVRFUU1BNEdBMVVFQ0
 F3SFQyNTBZWEpwYnpFUU1BNEdBMVVFQnd3SFZH0XliMjUwYnpFU01CQUDbMVVFQ2d3SgpVM
 1Z5ZG1WeWJHRmlNU1F3SwdZRFZRUUREQnRrWlcxdmNHaHdiWGxoWkcxcGJpNXpaWEoyWlhK
 c11XSXVZMKV3CkhoY05NVGt3T1RJek1ESXdNe1U0V2hjTk1qQXdOVE15TURJd016VTRXakJ
 yTVFd0NRWURWUVFHRXdKRFFURVEKTUE0R0ExVUVDQXdIVDI1MF1YSnBiekVRTUE0R0ExVU
 VCd3dIVkc5eWIyNTBiekVTTUJBR0ExVUVDZ3dKVTJWeQpkBVZ5YkdGaU1TUXdJZ11EV1FRR
 ERCdGtaVzf2Y0dod2JYbGhaRzFwYmk1elpYSjJaWEpzWVdJdVkyRXdnZ0lpCk1BMEddU3FH
 U01iM0RRRUJBUVVBQTRJQ0R3QXdnZ01LQW9JQ0FRREJKTUJvNVdkR1ZrYmJYRjhBTURvSUt
 1aCsKbmJGNTg0dE1RY1pGUXRJWU1hWnI4avloc2hUdFJXR1IxVWc5M0tnUW4wY2xSQWUwUE
 s5Z21LQVJ5bkxpZ0g4bgpJZEfwY2tZcklr0Vkl25yYW8wYUtyL0M3U3hObFkxL2tnWS9NT
 FZkMVZWeWtDMWNYs0J5YTlsa3czSGdCTW05Cmt4eXdpZVdFcTjtN3ZnbVpvZmZRTmIvsut0
 aTFwRTBWWjhNdGZwb1Bva21EYz11MmkxWEVWM3Bps1NGeS95bjAKQnkzV1p4YzE5ais0Rm5
 3M1ZURjRnQuTPR3VTUUxOOHhMSWJBQUxF3hGQmFhR11kd212YnQwVXFFbEE4Ry9CTAoxRV
 BLYUh3aT11QXV1ZjAxNTBqSWVWSm5PTFhadWhBaHdxK21pbEFTb1E4UVVxOWtzRk44eWVna
 GNSc1VsMGQ3C1N0VDE1Y0w3QVN2Z0djR2FPK3JIVttRnU1OHezMHVSYmZwL0QyRUUyZTk5
 eXvhT11WT00rcUM5NVN4N2pYdnUKa1JiRkd6c3N1NkN1eDMyT2JjU1VxZDJQTjg2S2hveXF
 UeHR2N0xRU3V1bEMyeH1KN0tTZ0hzV1BjcERTdERnWQozZ11NMEw1Wnk1TEJmd2tEMC91a0
 Q4ZE56K1FPalVaY2cxS0gyY0NrK1VSM1NsZHRpNFdNR1AwYXA5eG1qvzdHCm55S1hFRFdjV
 FVHN01TSWt2QzNtejBhRTkzQ0hSR1VlejdWbWJqeEtZbzUON0JhbG9kcWg4T211WnNmWGeX
 MEMKZz1xMFRVWVpQbTNpQ3NOR2ZRwmFRc2FWNx1vazMrS1IxaU0yL21kQ3dQQ21TUUDac3p
 4bEhHb2JnQUVjC1E2TQpiVXFewwtXT09PcnY0bhZJWVFJREFRQUJNQTBHQ1NxR1NJYjNEUU
 VCQ3dVQUE0SUNBUUFry0VYNE1MWkgwZ3cvCmd5NVFBdG8rdW5qb2MrZ3FlazhKQytKVEwxS
 0ZuVldIamk3Z1RzYzF6dkx0aXNbD0x2VGFyS1ZDbDN1VGN6SG8KR1lqaFFIb3ZMc3VjT31x
 Y0xYcFJEdG93QUN1M3hTWEhocDRTZGFZYmdmc09ps081cmI3U2J2bnZ3UGJ1U01IMgpQZUF
 kr3BvSWVHdlhKaW1SRXory0JPZFd5VTUrS0w2N3zjODJiQX1XV1RpVzdaSTBobHIvTmkwTk
 s1S1pmdDRsCk1vQutabnhKZDdocHA5YzB2dERNeE5RRGEzQ0xLUm96UmpZcUt5dHzhZTFPT
 VhoL1NLY0NiakpDQVJYNm03YXQKZG9ieGlWSHVIOWhDcUoyTmzoN2JFcNv6UE5CZ1zaV294
 dTE1SE8wdnRYODhzUE1YZXJZw1Rwell3TjdCUEpUaApLVi9GREh2ZWJzWFY1ZDVhSTR1RFY
 0RnpmUFM2b2hJRTZwc0VVOFQyTkxMZW5vVzdRdk1YVTNwMm1HRG1JWGQzCndFS3ZXwkR6dX
 RZaGluWEN4RkZuNVNVNmhcUXJIR3UxWn15YjZwSEwvVE1oTzVwbkF0aja1W1n2a20yVUYyb
 jgKa3RKcDFLYWNSt1pMOFFzOWhvYXhYcjZKYm91OTZTN3djRDRFYkxwVkhvQjArbU91YWh0
 WUI5ZDUvMzhBa0tJTApPczl1a1FCc3J1VmZUWEEvUnNGUHRBV3JpZ1NLVHNpUmp5V1BJSHd
 1cU52c2E2UHROenZnZTRhtU1PYkJKcHg3CkxPLy93aXhiL0dDSkg5OXBBN2tSeHV2eUdDd0
 81UzhjZwdzeEdxMHRBS01maHUyZE1zNmNZbHJiNjFVb1FoNHQKL2gwdTQ2TmNIY1RZeXVBb
 FF1MmNYeEdHvmcvU2pnPT0KLS0tLS1FTkQg0VSVE1GSUNBVEutLS0tLQo=

We enable TLS encryption for the ingress controller by adding the following annotations to the metadata section of the ingress.yml file.

We enable TLS encryption for the ingress controller by adding the following annotations to the metadata section of the ingress.yml file.

- nginx.ingress.kubernetes.io/auth-tls-verify-client
- nginx.ingress.kubernetes.io/auth-tls-secret
- nginx.ingress.kubernetes.io/auth-tls-verify-depth

Update the ingress.yml resource file to include annotations. The update file should look like the following example when done.

```
---
```

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: phpmyadmin-http-ingress
  annotations:
    # Enable client certificate authentication
    nginx.ingress.kubernetes.io/auth-tls-verify-client: "on"
    # Create the secret containing the trusted ca certificates
    nginx.ingress.kubernetes.io/auth-tls-secret: "default/ingress-tls-
secret"
    # Specify the verification depth in the client certificates chain
    nginx.ingress.kubernetes.io/auth-tls-verify-depth: "1"
spec:
  backend:
    serviceName: phpmyadmin-service
    servicePort: 80
```

No related posts found

 KUBERNETES

INGRESS CONTROLLER, MYSQL, PHPMYADMIN, TLS

No related posts found



Interactive DevOps Training

Ad Edureka

As Low As \$45.95 per Month - No Credit Card Required

Ad fmphost.com

Top rated Design Master Course

Ad Skill-Lync

FilmAddict's Photography Wedding.

Ad filmaddict

Airtel Broadband @499

Ad Airtel Xstream Fiber

Red-Green Handwoven Cotton Saree -HOLI | Chidiyaa

Ad Chidiyaa

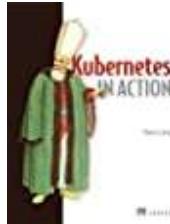
Top B.Tech Program in B'lore

Ad PES University

How-To Start Business - Host Billions

Ad whmcs

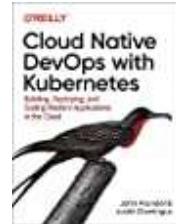
Shop Related Products



Kubernetes in Action

\$40.92 ~~\$50.00~~

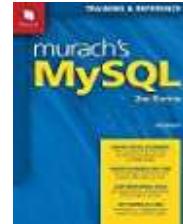
(245)



Cloud Native DevOps with Kubernetes: Building, ...

\$45.49 ~~\$70.00~~

(73)



Murach's MySQL, 2nd Edition

\$41.78 ~~\$54.50~~

(326)



Kubernetes: Up and Running: Dive into the Future of Infr...

\$6.57 ~~\$40.00~~

(124)

Ads by Amazon

ALSO ON SERVERLAB

[Learning Continuous Delivery, Part 1](#)

4 years ago • 1 comment

In this series of tutorials, we're going to show you a practical example of ...

[How to Proxy WSS WebSockets with ...](#)

3 years ago • 2 comments

Learn how to configure NGINX to support Secure WebSockets (WSS) to ...

[How to Install ImageMagick for ...](#)

3 years ago • 3 comments

Learn how to install the ImageMagick library and Imagick PHP extension ...

[Se
Ce](#)

3 y

Le
pr
Ce

[2 Comments](#)[Serverlab](#)[🔒 Disqus' Privacy Policy](#)[Login](#) ▾[Heart Favorite](#)[Twitter Tweet](#)[Facebook Share](#)[Sort by Best](#) ▾

Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS [?](#) Name**Adi Lakshmi Meda** • 2 years ago

Hi if wanna to modify upload_max_filesize and post_max_size parameters where should I define it

[^](#) | [▼](#) • Reply • Share >**Andres Fernando Leon** • 2 years ago

hi i followed your tutorial. What if i am using port 80. I tried using port 85 in a different namespace but it didnt work. what am i missing? i just replaced 80 for port 85. please help?

[^](#) | [▼](#) • Reply • Share >