

```

File Edit View Search Terminal Tabs Help
root@DevImranOps:~/kube          imran@DevOps: ~/_/kops
Port:      3000/TCP
Requests:
  cpu:      100m
State:     Running
  Started:  Sun, 02 Apr 2017 09:25:06 +0000
  Ready:    True
  Restart Count: 0
Volume Mounts:
  /etc/creds from cred-vd3ume (ro)
  /var/run/secrets/kubernetes.io/serviceaccount from default-token-h7kw0 (ro)
Environment Variables: <none>
Conditions:
  Type Status
  Initialized True
  Ready True
  PodScheduled True
Volumes:
  cred-volume:
    Type: Secret (a volume populated by a Secret)
    SecretName: db-secrets
  default-token-h7kw0:
    Type: Secret (a volume populated by a Secret)
    SecretName: default-token-h7kw0
  QoS Class: Burstable
  Tolerations: <none>
Events:
FirstSeen   LastSeen   Count  From           SubObjectPath   Type   Reason  M
message
-----  -----
25s        25s        1      {default-scheduler }               Normal   Scheduled
Successfully assigned helloworld-deployment-1674349767-33nc to ip-172-20-63-195.us-west-1.compute.internal
25s        25s        1      {kublet ip-172-20-63-195.us-west-1.compute.internal} spec.containers{k8s-demo}  Normal   Pulling p
23s        23s        1      {kubelet ip-172-20-63-195.us-west-1.compute.internal} spec.containers{k8s-demo}  Normal   Pulled S
successfully pulled image "visualpath/k8s-demo"
23s        23s        1      {kublet ip-172-20-63-195.us-west-1.compute.internal} spec.containers{k8s-demo}  Normal   Created C
reated container with docker id a0a442912537; Security:{seccomp=unconfined}
23s        23s        1      {kubelet ip-172-20-63-195.us-west-1.compute.internal} spec.containers{k8s-demo}  Normal   Started S
tarted container with docker id a0a442912537
root@DevImranOps:~/kube#

```

- ◆ Login to any of the pod and check mount point /etc/creds to verify our db-secrets.

```

root@DevImranOps:~/kube# kubectl get pod
kubectNAME          READY   STATUS    RESTARTS   AGE
helloworld-deployment-1674349767-33nc  1/1     Running   0          3m
helloworld-deployment-1674349767-czmp8  1/1     Running   0          3m
helloworld-deployment-1674349767-sgx62  1/1     Running   0          3m
nodehelloworld.example.com  1/1     Running   0          3h
root@DevImranOps:~/kube# kubectl exec -i -t helloworld-deployment-1674349767-33nc -- /bin/bash
root@helloworld-deployment-1674349767-33nc:/app# ls /etc/creds/
.. 4984 02 04 09 25 04.703846137/ ..data/                               password
root@helloworld-deployment-1674349767-33nc:/app# cat /etc/creds/username
rootroot@helloworld-deployment-1674349767-33nc:/app# cat /etc/creds/password
passwordroot@helloworld-deployment-1674349767-33nc:/app#

```

```

passwordroot@helloworld-deployment-1674349767-33nc:/app# df -h
Filesystem      Size  Used Avail Use% Mounted on
overlay       19G   3.5G  15G  20% /
tmpfs         498M     0  498M  0% /dev
tmpfs         498M     0  498M  0% /sys/fs/cgroup
tmpfs         498M   8.0K  498M  1% /etc/creds
/dev/xvda1     19G   3.5G  15G  20% /etc/hosts
shm            64M     0   64M  0% /dev/shm
tmpfs         498M   12K  498M  1% /run/secrets/kubernetes.io/serviceaccount

```

## Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in)

## 19. WordPress Deployment.

- ◆ Creating secrets for WordPress deployment.

```
root@DevImranOps:~/kube# cat wordpress/wordpress-secrets.yml
apiVersion: v1
kind: Secret
metadata:
  name: wordpress-secrets
type: Opaque
data:
  db-password: cGFzc3dvcmQ=
root@DevImranOps:~/kube# kubectl create -f wordpress/wordpress-secrets.yml
secret "wordpress-secrets" created
root@DevImranOps:~/kube#
```

- ◆ Creating work press deployment.

```
File Edit View Search Terminal Tabs Help
root@DevImranOps:~/kube
root@DevImranOps:~/kube# cat wordpress/wordpress-single-deployment-no-volumes.yml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: wordpress-deployment
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: wordpress
    spec:
      containers:
        - name: wordpress
          image: wordpress:4-php7.0
          ports:
            - name: http-port
              containerPort: 80
          env:
            - name: WORDPRESS_DB_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: wordpress-secrets
                  key: db-password
            - name: WORDPRESS_DB_HOST
              value: 127.0.0.1
        - name: mysql
          image: mysql:5.7
          ports:
            - name: mysql-port
              containerPort: 3306
          env:
            - name: MYSQL_ROOT_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: wordpress-secrets
                  key: db-password
root@DevImranOps:~/kube# kubectl create -f wordpress/wordpress-single-deployment-no-volumes.yml
deployment "wordpress-deployment" created
root@DevImranOps:~/kube#
root@DevImranOps:~/kube#
```

- ◆ Creating a service for WordPress.

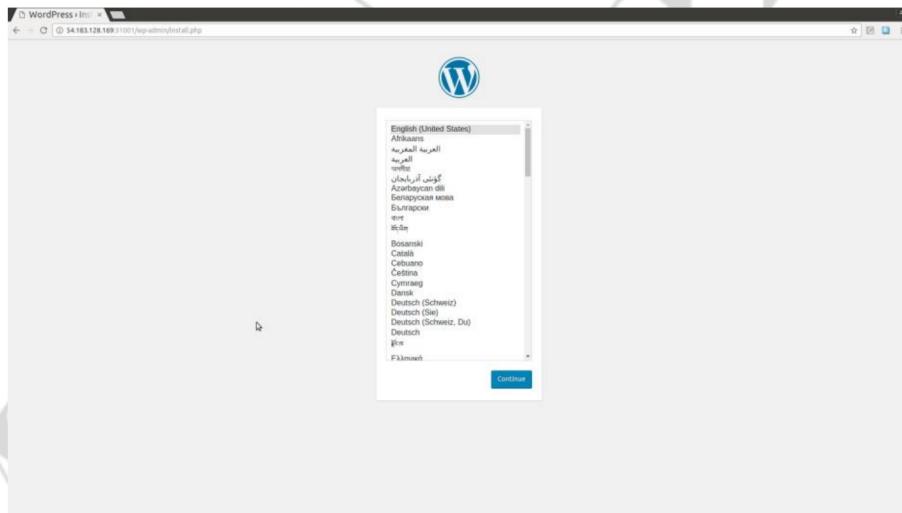
```

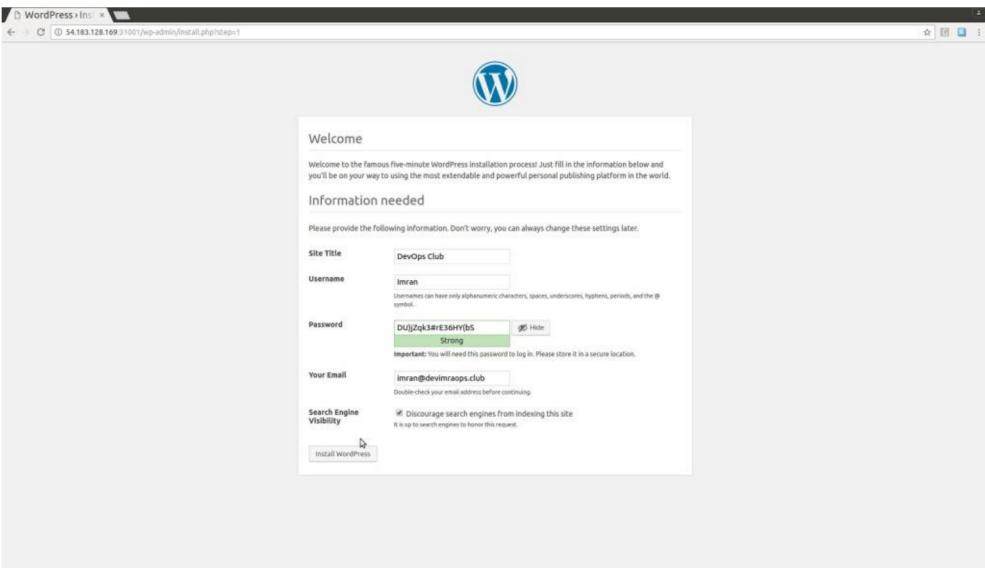
root@DevImranOps:~/kube# cat wordpress/wordpress-service.yml
apiVersion: v1
kind: Service
metadata:
  name: wordpress-service
spec:
  ports:
    - port: 31001
      nodePort: 31001
      targetPort: http-port
      protocol: TCP
  selector:
    app: wordpress
  type: NodePort
root@DevImranOps:~/kube# kubectl create -f wordpress/wordpress-service.yml
service "wordpress-service" created
root@DevImranOps:~/kube# kubectl get svc
NAME           CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE
kubernetes     100.64.0.1   <none>        443/TCP       5h
wordpress-service 100.64.1.146 <nodes>       31001:31001/TCP 1m
root@DevImranOps:~/kube#

```

◆ Accessing your WordPress deployment from outside world.

- Access your master node public IP on port 31001 as shown above in service port.
- Master node Security group should allow access on port 31001 from MyIP/Anywhere.





Pods or containers are stateless, they does not save data if killed or dead.

We will delete our WordPress pod as a demo to show that behaviour.

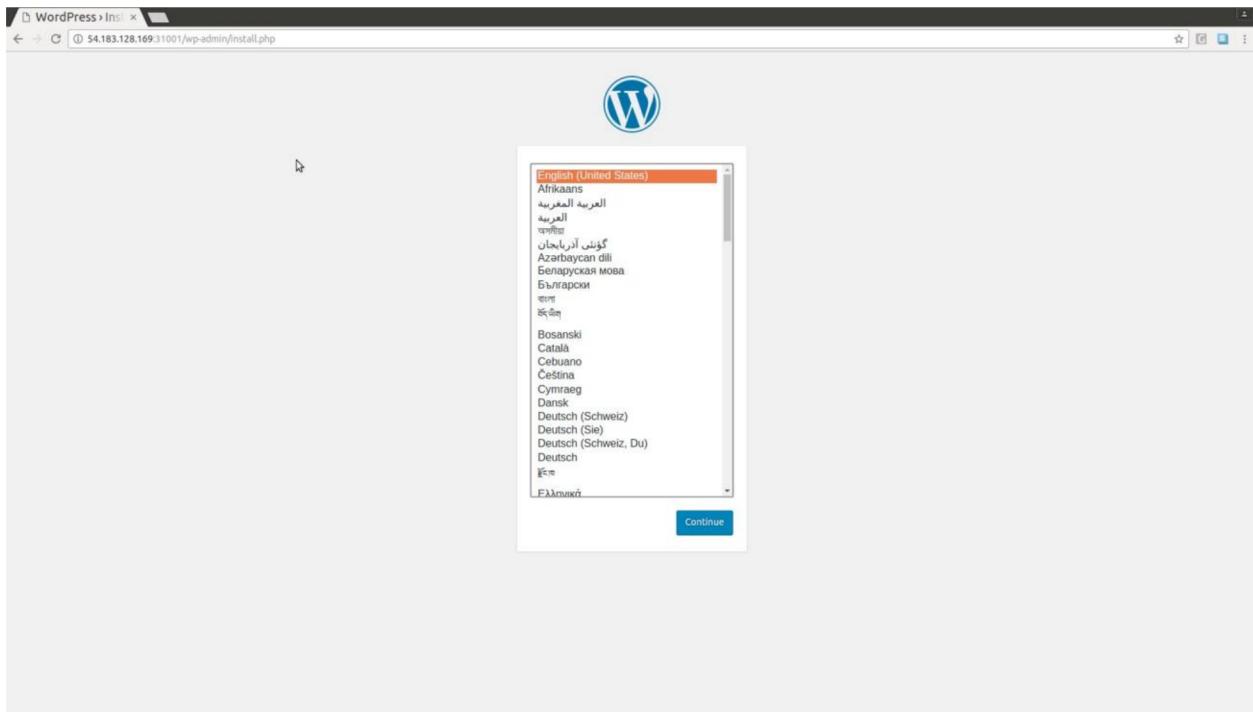
```
root@DevImranOps:~/kube# kubectl get pod
NAME                      READY   STATUS    RESTARTS   AGE
wordpress-deployment-1969810892-z76tv   2/2     Running   0          16m
root@DevImranOps:~/kube# kubectl delete pod delete wordpress-deployment-1969810892-z76tv
pod "wordpress-deployment-1969810892-z76tv" deleted
Error from server (NotFound): pods "delete" not found
root@DevImranOps:~/kube# kubectl get pod
NAME                      READY   STATUS    RESTARTS   AGE
wordpress-deployment-1969810892-1hkxz   2/2     Running   0          7s
root@DevImranOps:~/kube#
```

- ✓ Pod get recreated by deployment replication set if we delete it manually.
- ✓ All the data from previous pod gets deleted as we delete the pod.

## Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in)

- ✓ Check the WordPress app from browser, we would have lost the previous installation.



LAMRAVY  
VISUALPATH

# XVI. Nagios

**Nagios** is an [open source and free computer-software application](#) can be used to [monitor](#) systems, [networks](#) and infrastructure. Nagios offers monitoring and alerting services for servers, switches, applications and services. It alerts users when things go wrong and alerts them a second time after the problem has been resolved.

## 1. Network Monitoring

When it comes to open source network monitoring tools, the World's largest organizations turn to Nagios. Nagios monitors the network for problems caused by overloaded data links or network connections, as well as monitoring routers, switches and more. Easily able to monitor availability, uptime and response time of every node on the network, Nagios can deliver the results in a variety of visual representations and reports.

## 2. Server Monitoring

Nagios is known for being the best server monitoring software on the market. Server monitoring is made easy in Nagios because of the flexibility to monitor your servers with both agent-based and agentless monitoring. With over 5000 different addons available to monitor your servers, the community at the Nagios Exchange have left no stone unturned.

## 3. Application Monitoring

Implementing effective application monitoring with Nagios allows your organization to quickly detect application, service, or process problems, and take action to eliminate downtime for your application users. Nagios provides tools for monitoring of applications and application state – including Windows applications, Linux applications, UNIX applications, and Web applications.

## 4. Prerequisites

To follow this tutorial, you must have superuser privileges on the Ubuntu 14.04 server that will run Nagios. Ideally, you will be using a non-root user with superuser privileges.

## 5. LAMP Stack Script.

Nagios runs on LAMP stack which can be installed by below mentioned script.

```
$ cat lampstack.sh
```

```
apt-get update
```

```
apt-get install apache2 mysql-server php5-mysql vim php5 libapache2-mod-php5 php5-mcrypt  
sudo mysql_install_db  
sudo mysql_secure_installation  
echo -e "<IfModule mod_dir.c>\n  DirectoryIndex index.php index.html index.cgi index.pl  
index.xhtml index.htm\n</IfModule>" > /etc/apache2/mods-enabled/dir.conf  
  
service apache2 restart  
apt-get install php5-cli  
LAMP stack is also required.
```

## 6. Nagios Setup

### Create the Nagios user and group

Create a nagios user and a group.

```
sudo adduser nagios  
sudo addgroup nagcmd  
sudo usermod -a -G nagcmd nagios
```

### Install the dependencies

First, update your apt-get package lists:

```
sudo apt-get update
```

Let's install all the dependencies we will need to install Nagios with this command

```
sudo apt-get install build-essential libgd2-xpm-dev openssl libssl-dev xinetd apache2-utils unzip
```

apache2-utils, which will be used to set up the Nagios web interface.

As we are building the nagios core from the source, we should need all these libraries to be installed.

### Install Nagios core

For installing we have placed all the commands together and made as a script

```
Vi naginstall.sh
```

### Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

```
# move to the home directory
cd ~

#Download the Nagios tar file from this URL
curl -L -O https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.1.1.tar.gz

#Extract the Nagios archive with this command:
tar xvf nagios-*.tar.gz

#Then change to the extracted directory:
cd nagios-*

#Before building Nagios, we must configure it.
./configure --with-nagios-group=nagios --with-command-group=nagcmd --with-mail=/usr/sbin/sendmail

#Now compile Nagios with this command:
make all

#Now we can run these make commands below to build and install Nagios, init scripts, and sample configuration files:
sudo make install
sudo make install-commandmode
sudo make install-init
sudo make install-config
sudo /usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-available

#To issue external commands via the web interface to Nagios, we must add the web server user, www-data, to the nagcmd group:
sudo usermod -G nagcmd www-data

Execute the above script to install nagios core

#./ naginstall.sh
```

## Install the Nagios Plugins

```
Vi npinstall.sh
#move to user's home directory
cd ~
```

## Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in)

```
#download the tar file for the nagios pulgins using the command  
curl -L -O http://nagios-plugins.org/download/nagios-plugins-2.1.1.tar.gz  
  
#Extract Nagios Plugins archive with this command:  
tar xvf nagios-plugins-*.tar.gz  
  
#Then change to the extracted directory:  
cd nagios-plugins-*  
  
#Before building Nagios Plugins, we must configure it. Use this command:  
.configure --with-nagios-user=nagios --with-nagios-group=nagios --with-openssl
```

#Now compile Nagios Plugins with this command:

Make

#Then install it with this command:

sudo make install

Execute the above script to complete the installation with the nagios plugins

./ npinstall.sh

## Install NRPE

Find the source code for the latest stable release of NRPE at the [NRPE downloads page](#).

Download the latest version to your Nagios server.

At the time of this writing, the latest release is 2.15.:

Vi nrpeinstall.sh

```
#move to the home directory for the user  
cd ~
```

```
#download the tar file to our home directory with curl  
curl -L -O http://downloads.sourceforge.net/project/nagios/nrpe-2.x/nrpe-2.15/nrpe-2.15.tar.gz
```

#Extract the NRPE archive with this command:

tar xvf nrpe-\*.tar.gz

## Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in)

```
#Then change to the extracted directory:  
cd nrpe-*  
  
#Configure NRPE with these commands:  
.configure --enable-command-args --with-nagios-user=nagios --with-nagios-group=nagios --  
with-ssl=/usr/bin/openssl --with-ssl-lib=/usr/lib/x86_64-linux-gnu  
  
#Now build and install NRPE and its xinetd startup script with these commands:  
make all  
sudo make install  
sudo make install-xinetd  
sudo make install-daemon-config
```

Now open the xinetd startup script in an editor:  
sudo vi /etc/xinetd.d/nrpe

In the file modify the **only\_from** line by adding the private IP address of the your Nagios server to the end (substitute in the actual IP address of your server):

only\_from = 127.0.0.1 192.168.1.20

```
: default: on  
: description: NRPE (Nagios Remote Plugin Executor)  
service nrpe  
  
    flags          = REUSE  
    socket_type   = stream  
    port          = 5666  
    wait          = no  
    user          = nagios  
    group         = nagios  
    server        = /usr/local/nagios/bin/nrpe  
    server_args   = -c /usr/local/nagios/etc/nrpe.cfg --i  
    log_on_failure += USERID  
    disable       = no  
    only_from     = 127.0.0.1 192.168.1.20
```

Save and exit. Only the Nagios server will be allowed to communicate with NRPE.