

Deployment of Web Client Manual

Overview

This manual is help you to deploy the web client on your server and how to run the netdisk service system on your computer.

Deploy

This part is to configure NGINX Plus utilizes `keepalived` to provide high availability (HA) in a standard active-passive fashion.

Configuring keepalived for an Additional Passive Node

1. Install the **nginx-plus** and **nginx-ha-keepalived** packages on the new node.
2. Copy `/etc/keepalived/keepalived.conf` from the secondary node to the same location on the new node.
3. Edit **keepalived.conf** on the new node:
 - Lower the priority on any `vrrp_instance` .
 - Change `unicast_src_ip` .
 - Add the IP address of the secondary node to the `unicast_peer` section.
4. Edit **keepalived.conf** on the other nodes, adding the IP address of the new passive node to the `unicast_peer` section.
5. Restart `keepalived` on all nodes.
6. Test by stopping NGINX Plus on the first two nodes.

Configuring keepalived for Active–Active HA

1. Edit **keepalived.conf** on the secondary node:
 - Copy the entire `vrrp_instance` block `VI_1` section and paste it below the existing block
 - In the copied `vrrp_instance` section:
 - Rename the new `vrrp_instance` to `VI_2`
 - Change the `virtual_router_id` to 61

- Change the `virtual_ipaddress` to an available IP address on the same subnet (in this example 192.168.10.101)
 - Change the `priority` value to 100
2. Edit **keepalived.conf** on the primary node:
- Repeat the edits performed on the secondary node.
 - Set the `priority` within the new `vrrp_instance` to 99 or a value lower than on the secondary node.
3. Restart `keepalived` on all nodes.

Configuring NGINX Plus for Active–Active HA

Configuring NGINX Plus for Different Applications on Each Node

Example:

```
server {  
    listen 192.168.10.100:80;  
  
    location / {  
        root /application1;  
    }  
}  
  
server {  
    listen 192.168.10.101:80;  
  
    location / {  
        root /application2;  
    }  
}
```

Configuring NGINX Plus for All Applications on All Nodes

Example:

```
server {  
    listen *:80;  
  
    location /app1 {  
        root /application1;  
    }  
  
    location /app2 {  
        root /application2;  
    }  
}
```

Combining and Expanding Methods

Configuring All-Active HA on Three or More Nodes

Example:

```
vrrp_script chk_nginx_service {  
    script    "/usr/lib/keepalived/nginx-ha-check"  
    interval 3  
    weight   50  
}  
  
vrrp_instance VI_1 {  
    interface      eth0  
    state         BACKUP  
    priority      101  
    virtual_router_id 51  
    advert_int     1  
    accept  
    unicast_src_ip 192.168.10.10  
  
    unicast_peer {  
        192.168.10.11  
        192.168.10.12  
        192.168.10.13  
    }  
  
    virtual_ipaddress {  
        192.168.10.100  
    }
```

```
track_script {
    chk_nginx_service
}

notify "/usr/lib/keepalived/nginx-ha-notify"
}

vrrp_instance VI_2 {
    interface      eth0
    state         BACKUP
    priority      100
    virtual_router_id 61
    advert_int     1
    accept
    unicast_src_ip   192.168.10.10

    unicast_peer {
        192.168.10.11
        192.168.10.12
        192.168.10.13
    }

    virtual_ipaddress {
        192.168.10.101
    }

    track_script {
        chk_nginx_service
    }

    notify "/usr/lib/keepalived/nginx-ha-notify"
}

vrrp_instance VI_3 {
    interface      eth0
    state         BACKUP
    priority      99
    virtual_router_id 71
    advert_int     1
    accept
    unicast_src_ip   192.168.10.10

    unicast_peer {
        192.168.10.11
        192.168.10.12
    }
}
```

```

        192.168.10.13
    }

    virtual_ipaddress {
        192.168.10.102
    }

    track_script {
        chk_nginx_service
    }

    notify "/usr/lib/keepalived/nginx-ha-notify"
}

```

Configuring Active–Active–Passive HA

Example:

```

vrrp_script chk_nginx_service {
    script    "/usr/lib/keepalived/nginx-ha-check"
    interval 3
    weight   50
}

vrrp_instance VI_1 {
    interface      eth0
    state         BACKUP
    priority      99
    virtual_router_id 51
    advert_int    1
    accept
    unicast_src_ip 192.168.10.12

    unicast_peer {
        192.168.10.10
        192.168.10.11
    }

    virtual_ipaddress {
        192.168.10.100
    }

    track_script {

```

```

        chk_nginx_service
    }

    notify "/usr/lib/keepalived/nginx-ha-notify"
}

vrrp_instance VI_2 {
    interface          eth0
    state              BACKUP
    priority           99
    virtual_router_id  61
    advert_int         1
    accept
    unicast_src_ip    192.168.10.12

    unicast_peer {
        192.168.10.10
        192.168.10.11
    }

    virtual_ipaddress {
        192.168.10.101
    }

    track_script {
        chk_nginx_service
    }

    notify "/usr/lib/keepalived/nginx-ha-notify"
}

```

Configure for Nodes

1. Mount NFS to local /mnt on each node.

```
root # mount -t nfs 192.168.2.1:/srv/nfs/work /home/work
```

2. Install MySQL and Node.js on each node.

```
apt-get install mysql-server
```

```
curl -sL https://deb.nodesource.com/setup_7.x | sudo -E bash -
```

```
apt-get update
```

```
apt-get install -y nodejs
```

3. Change app_storepath.js to /mnt/allFiles on each node.

4. Change the location of MySQL database to `/mnt/mysql` on each node.

```
1.Check where is the database
```

```
    mysql -u root -prootadmin  
    # Open Database  
    show variables like '%dir%';  
    # Show sql store path  
    (Look at datadir line)  
    quit;
```

```
2.Stop MySQL Service
```

```
    service mysqld stop
```

```
3.Create new folder to store database.
```

```
    mkdir /mnt/mysql
```

```
4.Copy or move the old data to new location
```

```
    cp -R /usr/local/mysql/data/* /mnt/mysql/
```

```
    or
```

```
    mv /usr/local/mysql/data/* /mnt/mysql
```

```
5.Change MySQL database permission and configuration
```

```
    chown mysql:mysql -R /mnt/mysql/
```

```
    vim /etc/my.cnf
```

```
    datadir=/mnt/mysql
```

```
    vim /etc/init.d/mysqld
```

```
    datadir=/mnt/mysql
```

```
6.Start MySQL Service
```

```
    service mysqld start
```

5. Create database on only **ONE** node. It's available on all three nodes.

```
#create database :
CREATE DATABASE `supfile` CHARACTER SET 'utf8' COLLATE
'utf8_general_ci';

use supfile;

#create user_table:
CREATE TABLE user_table(ID int(255) NOT NULL
AUTO_INCREMENT,username varchar(255) NOT NULL,password varchar(255)
NOT NULL, used_memory varchar(255) NOT NULL, max_memory
varchar(255) NOT NULL, mailbox varchar(255) NOT NULL,active
varchar(255) NOT NULL,code varchar(255) NOT NULL,PRIMARY KEY (ID));

#create allFiles table:
CREATE TABLE allFiles(ID int(255) NOT NULL AUTO_INCREMENT,fileName
varchar(255) NOT NULL,hashName varchar(255) NOT NULL,lastTime
varchar(255) NOT NULL,typeis varchar(255),size varchar(255) NOT
NULL,downloadTime varchar(255) NOT NULL,fileOwner varchar(255) NOT
NULL,PRIMARY KEY (ID));
```

6. npm install
7. node n.js

Run

This part will help you to run the service system on your own computer.

Installation

Node.js

Download [here](#).

在你的平台上下载 Node.js 源码或预编译安装包，然后即可马上进行开发。

LTS 推荐大多数用户使用	当前版本 最新功能
 Windows Installer node-v8.11.2-x86.msi	 macOS Installer node-v8.11.2.pkg
Windows Installer (.msi)	32-bit 64-bit
Windows Binary (.zip)	32-bit 64-bit
macOS Installer (.pkg)	64-bit
macOS Binary (.tar.gz)	64-bit
Linux Binaries (x86/x64)	32-bit 64-bit
Linux Binaries (ARM)	ARMv6 ARMv7 ARMv8
Source Code	node-v8.11.2.tar.gz

其他平台

SunOS Binaries	32-bit 64-bit
Docker Image	Official Node.js Docker Image
Linux on Power Systems	64-bit
Linux on System z	64-bit

MySQL

Depending your OS, choose [wamp](#) or [mamp](#) or you can just install [MySQL](#).

wampserver.com

FRANÇAIS RUССКИЙ

WampServer
Apache, PHP, MySQL on Windows

START DOWNLOAD FORUM

WAMP SERVER,
a Windows web development environment.

WampServer is a Windows web development environment. It allows you to create web applications with Apache2, PHP and a MySQL database. Alongside, PhpMyAdmin allows you to manage easily your databases.

CONTRIBUTION
ALTER WAY

START USING WAMP SERVER



START WITH WAMP SERVER

WampServer installs automatically all you need to start developing web applications and is very intuitive to use. You will be able to tune your server without even touching the setting files.

INSTALLING FUNCTIONALITIES

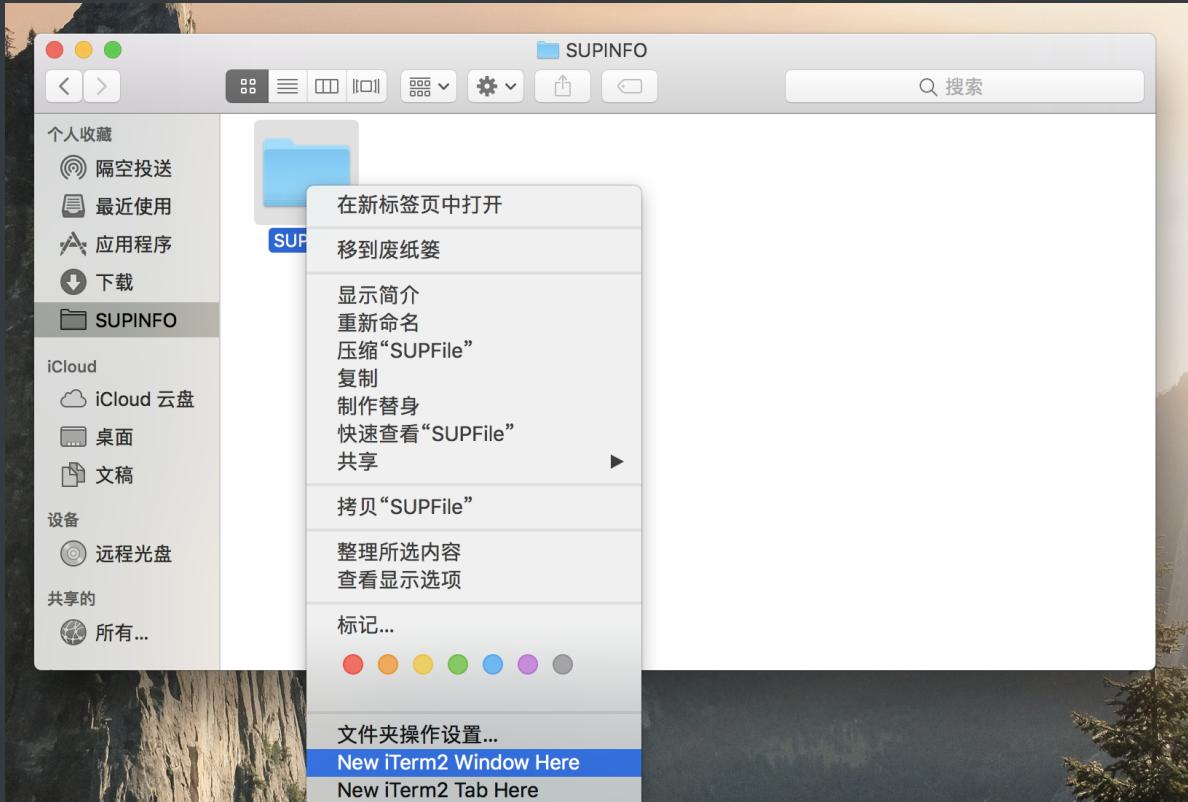
The screenshot shows the official website for MAMP and MAMP PRO. At the top, there's a navigation bar with links for 'MAMP', 'MAMP PRO', 'DOWNLOADS', 'MORE', a language selector (UK), and a 'STORE' button. The main header features a large white elephant logo on the left, followed by the text 'MAMP & MAMP PRO' and 'Now available!'. A large number '4' is prominently displayed on the right. Below the header, there's a section for 'MAMP: My Apache - MySQL - PHP' available 'For macOS and Windows'. It shows two versions: 'MAMP' with a 'FREE' badge and 'MAMP PRO'. The footer contains copyright information ('© Copyright 2004 - 2018 appslute GmbH') and links for 'Legal Notice' and 'Privacy Policy'.

The screenshot shows the official MySQL website. The header includes the MySQL logo, a search bar, and links for 'Contact MySQL', 'Login', and 'Register'. Below the header, there's a navigation bar with links for 'MYSQL.COM', 'DOWNLOADS', 'DOCUMENTATION', 'DEVELOPER ZONE', and social media icons for Facebook, Twitter, LinkedIn, Google+, and YouTube. The main content area features a large image of a cloud with the MySQL logo and the text 'New! Oracle MySQL Cloud Service'. It also mentions 'MySQL Enterprise Edition powered by Oracle Cloud' and a 'LEARN MORE' button. Below this, there are sections for 'MySQL Enterprise Edition' (with a stack of disks icon) and 'Oracle MySQL Cloud Service' (with a cloud icon). Both sections provide brief descriptions and links to learn more.

Configuration

Node.js

1. Go to **SUPFile** folder and open terminal

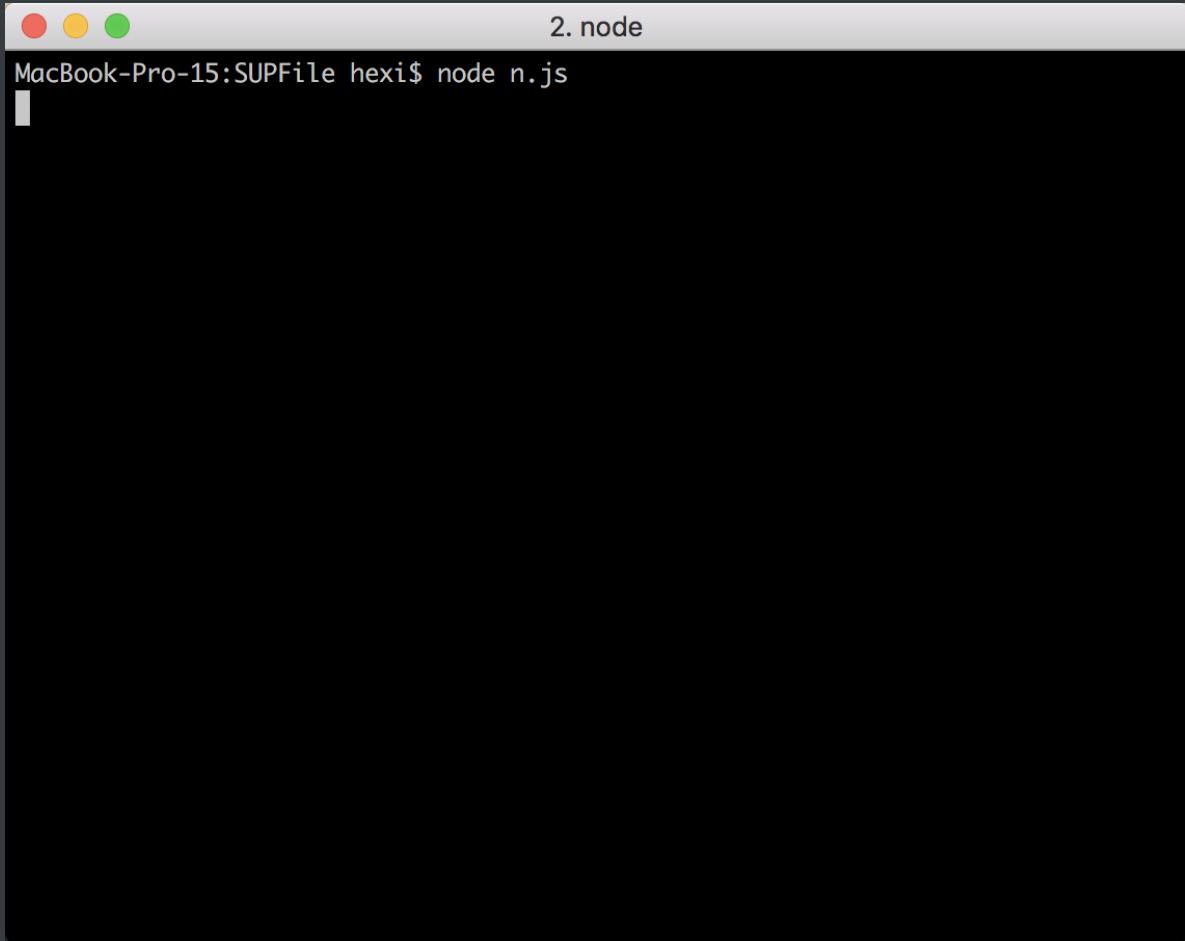


2. Run npm install

```
2. bash
Last login: Fri May 25 16:54:31 on ttys000
MacBook-Pro-15:SUPFile hexi$ npm install
npm WARN supfile@1.0.0 No repository field.

up to date in 0.453s
MacBook-Pro-15:SUPFile hexi$
```

3. Run node n.js



2. node

MacBook-Pro-15:SUPFile hexi\$ node n.js

MySQL

1. Create database:

```
CREATE DATABASE `supfile`  
CHARACTER SET 'utf8'  
COLLATE 'utf8_general_ci';
```

MacBook-Pro-15:bin hexi\$ sudo ./mysql -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.11 MySQL Community Server - GPL

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE `supfile`
->
-> CHARACTER SET 'utf8'
->
-> COLLATE 'utf8_general_ci';
Query OK, 1 row affected, 1 warning (0.02 sec)

mysql> |

2. USE supfile;

```
mysql> USE supfile;  
Database changed
```

3. create user_table:

```
CREATE TABLE user_table(ID int(255) NOT NULL AUTO_INCREMENT,username varchar(255)  
NOT NULL,password varchar(255) NOT NULL, PRIMARY KEY (ID));
```

```
mysql> CREATE TABLE user_table(ID int(255) NOT NULL AUTO_INCREMENT,username varc  
har(255) NOT NULL,password varchar(255) NOT NULL, PRIMARY KEY (ID));  
Query OK, 0 rows affected (0.04 sec)
```

4. create allFiles table: CREATE TABLE allFiles(ID int(255) NOT NULL
AUTO_INCREMENT,fileName varchar(255) NOT NULL,hashName varchar(255) NOT
NULL,lastTime varchar(255) NOT NULL,typeis varchar(255),size varchar(255) NOT
NULL,downloadTime varchar(255) NOT NULL,fileOwner varchar(255) NOT NULL,PRIMARY
KEY (ID));

```
mysql> CREATE TABLE allFiles(ID int(255) NOT NULL AUTO_INCREMENT,fileName varcho  
r(255) NOT NULL,hashName varchar(255) NOT NULL,lastTime varchar(255) NOT NULL,ty  
peis varchar(255),size varchar(255) NOT NULL,downloadTime varchar(255) NOT NULL,  
fileOwner varchar(255) NOT NULL,PRIMARY KEY (ID));  
Query OK, 0 rows affected (0.10 sec)
```

JavaScript

Edit these lines in the `/js/config_sql.js` according to your own information, such as **user** and **password**.

```
1 const mysql = require('mysql');
2 var databasePool = mysql.createPool({
3   'host':'localhost',
4   'user':'root',
5   'password':'MySQL185788688'
6   'database':'supfile'
7
8   // 'host':'172.17.0.2',
9   // 'user':'root',
10  // 'password':'111',
11  // 'database':'supfile'
12 });
13
14 module.exports = databasePool;
```

One more thing ...

Open browser go to localhost:9001

SUPFile

Store everything you want
Share with your friends
Safe and convenient

Done!