

Node.js

- [BISO](#)
- [App](#)
- [Node.js](#)
- [LAMP](#)
- [LAMP](#) [Linux](#) [Apache](#) [MySQL](#) [PHP](#)
- [LEMP](#) [Linux](#) [Nginx](#) [MySQL](#) [PHP](#)
- [Node.js](#) [macOS](#)
- [Node.js](#) [CentOS 8](#)

BISO

<https://www.bilibili.com/video/BV1rQ4y1M7r5?p=1>

1	BIOS	G BIOS A BIOS D BIOS		
2				

<https://www.bilibili.com/video/BV1jf4y1S7XJ?p=1>

1	HP	DOS		
2	PLASHEFI			

fptw.exe -d image.bin

App

Web 2.0

<https://unhosted.org/apps/>

Node.js

Node.js

Node.js	JavaScript	JavaScript	Node.js
	Node.js		
macOS	Node.js		
Node.js	JavaScript	JavaScript	Node.js...
CentOS 8	Node.js		
Cefdnf CentOSNodejsStrnodejs rnvmmode node			
Ubuntu 18.04	Node.js		
Node.js	JavaScript	Ubuntu 18.04	Node.js
Debian 10	Node.js		
Node.js	JavaScript	JavaScript	Node.js Web ...
Ubuntu 16.04	Node.js		
Node.js	Javascript	Web	Ubuntu 16.04 Node.js
Debian 9	Node.js		
Node.js	JavaScript		
	Debian 9	Node.js	
Debian 8	Node.js		
Node.js	JavaScript	JavaScript	...
CentOS 7	Node.js		
Node.js	Javascript	CentOS 7	Node.js
Ubuntu 20.04	Node.js		
	Ubuntu 20.04	Node.js	apt Ubuntu nodejs apt PPA

LAMP

VPS CentOS LAMP

LAMP Linux Apache MySQL/MariaDB PHP

Linux PHP MySQL/MariaDB Apache

1.

VPS CentOS 7 VPS DNS

2. **Apache**

Apache CentOS

```
yum -y install httpd
```

Apache
Apache

```
systemctl start httpd.service
```

Apache

```
systemctl stop httpd.service
```

Apache

```
systemctl restart httpd.service
```

Apache

```
systemctl enable httpd.service
```

Apache

```
systemctl disable httpd.service
```

Apache

```
systemctl status httpd.service
```

http://domain.com

http://ip VPS IP

Apache Test Page

http://localhost

http://127.0.0.1

Apache Test Page

```
vi /etc/httpd/conf/httpd.conf
```

80 -> 8080

```
#Listen 12.34.56.78:80
```

```
Listen 8080
```

Apache

```
systemctl restart httpd.service
```

apache

/etc/httpd/conf/httpd.conf :

/etc/httpd/conf.d/*.conf : CentOS

httpd.conf

apache

/usr/lib/httpd/modules :apache

/var/www/html : CentOS " "

/var/www/error :

/var/www/icons : apache

/var/www/cgi-bin : CGI

/var/log/httpd :

/usr/sbin/apachectl : Apache

shell script ,

Apache

/usr/sbin/httpd : apache

/usr/bin/htpasswd : Apache

3. MySQL/MariaDB

MySQL/MariaDB

```
yum -y install mariadb mariadb-server
```

MySQL/MariaDB

MySQL/MariaDB

```
systemctl start mariadb
```

MySQL/MariaDB

```
systemctl stop mariadb
```

MySQL/MariaDB

```
systemctl restart mariadb
```

MySQL/MariaDB

```
systemctl enable mariadb
```

root

MySQL

```
mysqladmin -u root -p password 12345678
```

Enter password: root root 12345678

MySQL/MariaDB

```
systemctl restart mariadb
```

MySQL

/etc/my.cnf : Mysql mysql

/usr/lib/mysql : MySQL

4. PHP

PHP Apache PHP

```
yum -y install php
```

Apache :

```
systemctl restart httpd.service
```

Apache

/var/www/html

PHP

http://localhost/en

PHP

PHP

MySQL

PHP

MySQL

php-mysql

5. PHP PHP5 MySQL

```
yum search php
```

```
yum -y install php-mysql php-gd php-imap php-ldap php-odbc php-pear php-xml php-xmlrpc php-mbstr
```

```
yum -y check-update
```

```
yum -y update
```

php.ini

```
vi /etc/php.ini
```

Apache

```
systemctl restart httpd.service
```

http://localhost/env.php

PHP MySQL

6. Apache MySQL

```
systemctl enable httpd.service  
systemctl enable mariadb
```

“ Apache MySQL ”

OK LAMP

1. phpMyAdmin

```
yum -y install phpMyAdmin
```

2. putty SSH

```
chown root:root /var/www/html/ -R
```

```
chmod -R 777 /var/www/html/
```

zip

```
yum -y install unzip
```

WordPress

1. Wget

```
yum -y install wget
```

2.

```
cd /var/www/html
```

3. WordPress

```
wget http://wordpress.org/latest.tar.gz  
tar zxvf latest.tar.gz
```

4. WordPress

```
chmod -R 777 /var/www/html/wordpress  
chgrp -R ftp /var/www/html/wordpress  
chown -R www /var/www/html/wordpress
```

5. apache2 /etc/httpd/conf/httpd.conf <VirtualHost *:80> </VirtualHost>

```
<Directory "/var/www/html">  
Options FollowSymLinks IncludesNOEXEC Indexes  
DirectoryIndex index.html index.htm index.php  
AllowOverride all  
Order Deny,Allow  
Allow from all  
</Directory>
```

6. Apache

```
systemctl restart httpd.service
```

FTP

vsftpd pureftpd

WinSCP SFTP

LAMP Linux Apache MySQL

CentOS

7 Linux Apache MySQL PHP LAMP

“LAMP”

Web

Linux

Apache Web

CentOS 7 VPS

LAMP

CentOS

Linux

CentOS 7

1-4

— Apache

Apache Web

Web

C[yum]S

Apache

[yum]

CentOS

```
sudo yum install httpd
```

[sudo]

VPS Apache

```
sudo systemctl start httpd.service
```

IP

IP

```
http://your_server_IP_address/
```

CentOS 7 Apache

Image not found or type unknown

Web

Apache

```
sudo systemctl enable httpd.service
```

IP

IP

SSH

```
iproute2
```

```
ip addr show eth0 | grep inet | awk '{ print $2; }' | sed 's/\./.*$//'
```

IP

```
curl http://icanhazip.com
```

IP



MySQL MariaDB

Web

MySQL

MariaDB

MariaDB MySQL

```
yum
```

“ ”

```
sudo yum install mariadb-server mariadb
```

MariaDB

```
sudo systemctl start mariadb
```

MySQL

```
sudo mysql_secure_installation
```

|Y|

MySQL

MySQL

Enter current password for root (enter for none):

OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB root user without the proper authorization.

New password: password

Re-enter new password: password

Password updated successfully!

Reloading privilege tables..

... Success!

“ENTER”

MySQL

MariaDB

```
sudo systemctl enable mariadb.service
```

— — PHP

PHP

MySQL

Web

|yum|

php-mysql

```
sudo yum install php php-mysql
```

PHP

Apache Web

PHP

```
sudo systemctl restart httpd.service
```

PHP

PHP

PHP

```
yum search php-
```

```
php-bcmath.x86_64 : A module for PHP applications for using the bcmath library
php-cli.x86_64 : Command-line interface for PHP
php-common.x86_64 : Common files for PHP
php-dba.x86_64 : A database abstraction layer module for PHP applications
php-devel.x86_64 : Files needed for building PHP extensions
php-embedded.x86_64 : PHP library for embedding in applications
php-enchant.x86_64 : Enchant spelling extension for PHP applications
php-fpm.x86_64 : PHP FastCGI Process Manager
php-gd.x86_64 : A module for PHP applications for using the gd graphics library
. . .
```

```
yum info package_name
```

```
Description
```

```
php-fpm
```

```
yum info php-fpm
```

```
. . .
Summary      : PHP FastCGI Process Manager
URL          : http://www.php.net/
License      : PHP and Zend and BSD
Description  : PHP-FPM (FastCGI Process Manager) is an alternative PHP FastCGI
              : implementation with some additional features useful for sites of
              : any size, especially busier sites.
```

```
yum install
```

```
php-fpm
```

```
sudo yum install php-fpm
```

```
yum install
```

```
sudo yum install package1 package2 ...
```

LAMP

PHP

— Web PHP

PHP

PHP

`info.php` Apache

“ ”

CentOS `/var/www/html/`

```
sudo vi /var/www/html/info.php
```

PHP

```
<?php phpinfo(); ?>
```

HTTP HTTPS

```
sudo firewall-cmd --permanent --zone=public --add-service=http
sudo firewall-cmd --permanent --zone=public --add-service=https
sudo firewall-cmd --reload
```

Web

PHP

IP

```
http://your_server_IP_address/info.php
```

CentOS 7 PHP

Image not found or type unknown

PHP

PHP

```
sudo rm /var/www/html/info.php
```

LAMP

LEMP Linux Nginx MySQL

CentOS

7 Linux Nginx MySQL PHP LEMP

LEMP Web **PHP** Linux ENginx Web LAM
MariaDB CentOS 7 PHP 7.4 LEMP MariaDB MySQL

CentOS 7 1-4

1 — Nginx

Nginx Nginx EPEL CentOS 7

CentOS 7 EPEL

1. sudo yum install epel-release

??

Since we are using a `sudo` command, these operations get executed with root privileges. It will ask you for your regular user's password to verify that you have permission to run commands with root privileges. You'll also be prompted to confirm installation, so press `Y` to proceed.

`yum` EPEL Nginx

1. sudo yum install nginx

??

Nginx

1. sudo systemctl start nginx

??

IP

IP

Open in a web browser:

```
http://server_domain_name_or_IP/
```

CentOS 7 Nginx

CentOS 7 Nginx

Image not found or type unknown

Web

Nginx

1. sudo systemctl enable nginx

??

IP

IP

SSH

```
iproute2
```

1. ip addr show eth0 | grep inet | awk '{ print \$2; }' | sed 's/\/.*\$//'

??

IP

```
1. curl http://icanhazip.com
```

```
??
```

IP

2 — MariaDB

Web

MySQL

MariaDB

MariaDB

MySQL

```
|yum|
```

```
1. sudo yum install mariadb-server mariadb
```

```
??
```

MariaDB

```
1. sudo systemctl start mariadb
```

```
??
```

MariaDB

```
1. sudo mysql_secure_installation
```

```
??
```

```
|root|MariaDB
```

```
Maria|root||Y|
```

MariaDB

mysql_secure_installation prompts:

```
Enter current password for root (enter for none):
```

```
OK, successfully used password, moving on...
```

Setting the root password ensures that nobody can log into the MariaDB root user without the proper authorisation.

```
Set root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!
```

“ENTER”

MySQL

MariaDB

1. sudo systemctl enable mariadb

??

3 — PHP

PHP

MySQL

Web

CentOS 7

PHP

Remi

CentOS

CentOS 7

CentOS 7 Remi

1. sudo yum install <http://rpms.remirepo.net/enterprise/remi-release-7.rpm>

??

PHP

Remi

PHP 7+

1. yum --disablerepo="*" --enablerepo="remi-safe" list php[7-9][0-9].x86_64

??

Output

```
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * remi-safe: mirrors.ukfast.co.uk

Available Packages
php70.x86_64                                2.0-
1.el7.remi                                  remi-safe
php71.x86_64                                2.0-
1.el7.remi                                  remi-safe
php72.x86_64                                2.0-
1.el7.remi                                  remi-safe
php73.x86_64                                2.0-
1.el7.remi                                  remi-safe
php74.x86_64                                1.0-
3.el7.remi                                  remi-safe
php80.x86_64                                1.0-
3.el7.remi                                  remi-safe
```

PHP 7.4

PHP

Remi

PHP 7.4

1. `sudo yum-config-manager --enable remi-php74`

??

yum

PHP

Nginx

PHP 7.4

MySQL

1. `sudo yum install php php-mysqlnd php-fpm`

??

PHP

1. `php --version`

??

Output

```
PHP 7.4.5 (cli) (built: Apr 14 2020 12:54:33) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
```

PHP nano vi CentOS

1. sudo yum install nano

??

nano /etc/php-fpm.d/www.conf

1. sudo nano /etc/php-fpm.d/www.conf

??

user group nano CTRL + W

/etc/php-fpm.d/www.conf

```
...
; Unix user/group of processes
; Note: The user is mandatory. If the group is not set, the default user's group
;      will be used.
; RPM: apache user chosen to provide access to the same directories as httpd
user = apache
; RPM: Keep a group allowed to write in log dir.
group = apache
...
```

user group apache tonginx

/etc/php-fpm.d/www.conf

```
...
; RPM: apache user chosen to provide access to the same directories as httpd
user = nginx
; RPM: Keep a group allowed to write in log dir.
```

```
group = nginx
```

```
...
```

```
|listen| |php-fpm|    |listen|
```

/etc/php-fpm.d/www.conf

```
listen = /var/run/php-fpm/php-fpm.sock;
```

Finally, we'll need to change the owner and group settings for the socket file we just defined within the `|listen|` directive. Locate the `|listen.owner|`, `|listen.group|` and `|listen.mode|` directives. These lines are commented out by default. Uncomment them by removing the preceding `|#|` sign at the beginning of the line. Then, change the owner and group to `|nginx|`:

/etc/php-fpm.d/www.conf

```
listen.owner = nginx
```

```
listen.group = nginx
```

```
listen.mode = 0660
```

```
|nano| |CTRL + X| |Y| |ENTER|
```

```
|php-fpm|
```

1. sudo systemctl start php-fpm

??

PHP |php-fpm| PHPNginx

4 — Nginx PHP

Nginx PHP

Nginx Apache

|/etc/nginx/nginx.conf| PHP

```
|/etc/nginx/conf.d|
```

1. sudo nano /etc/nginx/conf.d/default.conf

??

server_name IP

/etc/nginx/conf.d/default.conf

```
server {
    listen      80;
    server_name server_domain_or_IP;

    root    /usr/share/nginx/html;
    index index.php index.html index.htm;

    location / {
        try_files $uri $uri/ =404;
    }
    error_page 404 /404.html;
    error_page 500 502 503 504 /50x.html;

    location = /50x.html {
        root /usr/share/nginx/html;
    }

    location ~ \.php$ {
        try_files $uri =404;
        fastcgi_pass unix:/var/run/php-fpm/php-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}
```

Nginx

1. sudo systemctl restart nginx

??

PHP Nginx

5 — Web PHP

`\php-fpm` `.php` Nginx

Nginx sudo

`sammy`

1. `sudo chown -R sammy.sammy /usr/share/nginx/html/`

??

PHP Web

`/usr/share/nginx/html/info.php` PHP

1. `nano /usr/share/nginx/html/info.php`

??

PHP PHP

`/usr/share/nginx/html/info.php`

```
<?php

phpinfo();
```

??

`/info.php` PHP IP

`http://server_host_or_IP/info.php`

CentOS 7 PHP 7.4

Image not found or type unknown

PHP `rm` PHP CentOS

1. rm /usr/share/nginx/html/info.php

??

PHP

php-fpm

PHP

nginx

Web

MariaDB

PHP

Nginx

Node.js macOS

macOS Node.js

High Sierra macOS

1 — macOS

Node.js Node.js

To access the command line interface, you'll use the Terminal application provided by macOS. Like any other application, you can find it by going into Finder, navigating to the Applications folder, and then into the Utilities folder. From here, double-click the Terminal application to open it up. Alternatively, you can use Spotlight by holding down the `COMMAND` key and pressing `SPACE` to find Terminal by typing it out in the box that appears.

macOS

Image not found or type unknown

<https://linuxdigitalocean.com/community/tutorials/an-introduction-to-the-linux-terminal>

macOS

Node.js

2 — Xcode

Xcode (IDE) macOS Xcode Node.js Node.js Xcode

“ ”

```
1. xcode-select --install
```

??

Homebrew

Node.js

3 —

macOS

Homebrew Unix

macOS

Homebrew

Node.js

Homebrew “ ”

1. `/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"`

??

`|curl|` GitHub Homebrew Git Homebrew

`|curl|`

- `-f|--fail` HTML
- `-s|--silent|curl|-S|--show-error|curl|`
- `-L|--location|curl|`

`|curl|` macOS Ruby

Homebrew

You'll be prompted to enter your password during the process. However, when you type your password, your keystrokes will not display in the Terminal window. This is a security measure and is something you'll see often when prompted for passwords on the command line. Even though you don't see them, your keystrokes are being recorded by the system, so press the `|RETURN|` key once you've entered your password.

`|y|` “ ”

Homebrew

1. `brew doctor`

??

Output

Your system is ready to brew.

brew update Homebrew

Homebrew Node.js

4 — Node.js

Homebrew Node.js

Homebrew search Node.js

1. brew search nodejs

??

Output

=> Formulae
node.js
nodejs

nodejs node.js

nodejs

1. brew install nodejs

??

Homebrew Node.js

Output

```
==> Installing dependencies for node: icu4c
==> Installing node dependency: icu4c

==> Installing node
==> Downloading https://homebrew.bintray.com/bottles/node-11.0.0.sierra.bottle.tar.gz
##### 100.0%
==> Pouring node-11.0.0.sierra.bottle.tar.gz
...

==> Summary
/usr/local/Cellar/node/11.0.0: 3,936 files, 50.1MB
```

Node.js Homebrew

Node.js

Node.js

1. node -v

??

Node.js

Node.js

Output

v11.0.0

1. npm -v

??

Output

6.4.1

npm

Node.js

Homebrew

Node.js

-
1. brew update
 2. brew upgrade nodejs
-

??

Node.js

5 —

“ ”

Node.js

nano | hello.js |

-
1. nano hello.js
-

??

js

```
let message = "Hello, World!";  
console.log(message);
```

??

CTRL+X | y |

-
1. node hello.js
-

??

Output

```
Hello, World!
```

Node.js

N|npm|js

- [Vue.js](#) [Axios](#) [API](#)
- [Django](#) [Web](#) [Ubuntu 18.04](#)

Node.js CentOS 8

Node.js JavaScript JavaScript JavaScript Web

CentOS 8 Node.js

- `dnf` CentOS AppStream `nodejs`
- `nvm` `node`
- `node`

`dnf` `nvm` Node

`root` `sudo` CentOS 8

1— CentOS AppStream

Node.js CentOS AppStream `dnf` `nodejs`

1. `sudo dnf module list nodejs`

??

Output

Name	Stream	Summary
Profiles		
nodejs	10 [d]	common [d], development, minimal,
s2i	Javascript runtime	
nodejs	12	common, development, minimal,
s2i	Javascript runtime	

`10` `1[d]` 10 Node.js 12

1. sudo dnf module enable nodejs:12

??

CentOS AppStream

dnf | nodejs |

1. sudo dnf install nodejs

??

dnf | y | ENTER |

node |

1. node --version

??

Output

v12.13.1

Node | --version |

Node.jsNode.js

Installing the `nodejs` package should also install the `npm` *Node Package Manager* utility as a dependency. Verify that it was installed properly as well:

1. npm --version

??

Output

6.12.1

2—

Node.js

nvm Node

Node.js

Node

CentOS 8

[NVMHub](#)`| curl |`ME`bash| curl || bash`

1. `curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.35.3/install.sh`

??

`| bash |`” URL NVM

1. `curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.35.3/install.sh | bash`

??

`nvm| . bash_profile`

1. `source ~/.bash_profile`

??

NVM

`nvm list-remote`

. . .

`v12.13.0 (LTS: Erbium)``v12.13.1 (LTS: Erbium)``v12.14.0 (LTS: Erbium)``v12.14.1 (LTS: Erbium)``v12.15.0 (LTS: Erbium)`

```
v12.16.0 (LTS: Erbium)
v12.16.1 (Latest LTS: Erbium)
v13.0.0
v13.0.1
v13.1.0
v13.2.0
v13.3.0
v13.4.0
v13.5.0
v13.6.0
v13.7.0
v13.8.0
v13.9.0
v13.10.0
v13.10.1
v13.11.0
v13.12.0
```

v13.6.0

1. nvm install v13.6.0

??

```
nvm list
```

Output

```
-> v13.6.0
default -> v13.6.0
node -> stable (-> v13.6.0) (default)
stable -> 13.6 (-> v13.6.0) (default)
```

```
-> v13.6.0
```

```
C\system -> v12.13.1 | nvm use system
```

LTS

Output

```
lts/* -> lts/erbium (-> N/A)
lts/argon -> v4.9.1 (-> N/A)
lts/boron -> v6.17.1 (-> N/A)
lts/carbon -> v8.17.0 (-> N/A)
lts/dubnium -> v10.19.0 (-> N/A)
lts/erbium -> v12.16.1 (-> N/A)
```

|erbium|

1. nvm install lts/erbium

??

Output

```
Downloading and installing node v12.16.1...
...
Now using node v12.16.1 (npm v6.13.4)
```

|nvm use|

```
nvm use v13.6.0
```

```
Now using node v13.6.0 (npm v6.13.4)
```

```
node --version
```

Output

```
v13.6.0
```

|npm|

3——

Node.js

Node.js

SSH

1. cd ~

??

|curl|||tar xz|

1. curl <https://nodejs.org/dist/v12.16.1/node-v12.16.1.tar.gz> | tar xz

??

|curl||tar|tar

1. cd node-v*

??

|dnf|CentOS

1. sudo dnf install gcc-c++ make python2

??

|y||ENTER|

1. ./configure

2. make -j4

??

|-j4|

30

1. sudo make install

??

Node

1. node --version

??

v12.16.1

npm

Node

CentOS AppStream

Node Version Manager

Node.js

JavaScript

- [Javascript](#) Node.js JavaScript
- [N](#)