# 安装.netcore运行时

安装 .NET

sudo dnf update -y

安装 ASP.NET Core 运行时

sudo dnf install aspnetcore-runtime-8.0

# 验证dotnet core runtime是否安装成功

dotnet --info

# 解压安装包

解压或者拷贝部署文件到指定的文件夹。

创建并进入部署包文件夹，例如我们打算在 /var/www 中运行：

创建并进入文件夹：

mkdir -m 666 /var/www/

cd /var/www

解压安装包至 /var/www 文件夹中并删除安装包：

tar -xzf xblms-linux-x64.tar.gz

rm xblms-linux-x64.tar.gz -f

# **安装**Nginx

# 安装nginx

sudo dnf install nginx

# 启动nginx（需要先确保80端口未被其他程序占用）

systemctl start nginx

# 设为开机启动

systemctl enable nginx

# **设置**反向**代理**

安装完毕 Nginx后，请修改 /etc/nginx/nginx.conf 配置文件，在文本编辑器中打开它，并将 server 内容替换为以下内容：

# XBLMS Nginx Config ...

server {

listen 80;

server\_name \_;

location / {

proxy\_pass http://localhost:5000;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection keep-alive;

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_connect\_timeout 600;

proxy\_send\_timeout 600;

proxy\_read\_timeout 600;

send\_timeout 600;

}

}

使用上述配置文件，Nginx 接受端口 80 上的流量并将请求转接到 http://localhost:5000。

如果希望指定域名，可以将 server\_name \_; 替换为 server\_name example.com;

除此之外，由于 Nginx 默认允许上传文件的大小是 1M，通常还需要设置 Nginx 允许上传文件的大小：

client\_max\_body\_size 500m;

500m表示最大上传500M，可以根据实际需要进行设置。

以下是完整的 Nginx 配置文件示例，具体配置可以根据实际情况更改：

user nginx;

worker\_processes auto;

error\_log /var/log/nginx/error.log notice;

pid /var/run/nginx.pid;

events {

worker\_connections 1024;

}

http {

include /etc/nginx/mime.types;

default\_type application/octet-stream;

log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

'$status $body\_bytes\_sent "$http\_referer" '

'"$http\_user\_agent" "$http\_x\_forwarded\_for"';

access\_log /var/log/nginx/access.log main;

sendfile on;

keepalive\_timeout 65;

client\_max\_body\_size 2048m;

# XBLMS Nginx Config ...

server {

listen 80;

server\_name \_;

location / {

proxy\_pass http://localhost:5000;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection keep-alive;

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_connect\_timeout 600;

proxy\_send\_timeout 600;

proxy\_read\_timeout 600;

send\_timeout 600;

}

}

}

完成配置 Nginx 后，运行 sudo nginx -t 来验证配置文件的语法。

sudo nginx -t

如果配置文件测试成功，可以通过运行 sudo nginx -s reload 强制 Nginx 重新载入。

sudo nginx -s reload

# **设置进程管理器**

创建服务定义文件 xblms.service：

sudo touch /etc/systemd/system/xblms.service

将以下内容保存至 /etc/systemd/system/xblms.service：

[Unit]

Description=XBLMS

[Service]

WorkingDirectory=/var/www

ExecStart=/usr/bin/dotnet /var/www/XBLMS.Web.dll

Restart=always

# Restart service after 10 seconds if the xlms service crashes:

RestartSec=10

KillSignal=SIGINT

SyslogIdentifier=xblms

User=root

Environment=ASPNETCORE\_ENVIRONMENT=Production

Environment=DOTNET\_PRINT\_TELEMETRY\_MESSAGE=false

[Install]

WantedBy=multi-user.target

保存文件并启用服务。

sudo systemctl enable xblms.service

运行服务，并确认它正在运行。

sudo systemctl start xblms.service

sudo systemctl status xblms.service

● xblms.service - XBLMS

Loaded: loaded (/etc/systemd/system/xblms.service; enabled; vendor preset: disabled)

Active: active (running) since Sun 2020-08-30 04:40:33 CST; 9s ago

Main PID: 17983 (dotnet)

CGroup: /system.slice/xblms.service

└─17983 /usr/bin/dotnet /var/www/XBLMS.Web.dll

通过浏览器从地址 http://<IP地址或域名> 进行访问

# **创建空数据库**

省略...

# 访问并安装系统

浏览器访问 <http://localhost/admin/install>

省略...