### Centos7发布说明

**环境说明：**

服务器系统：CentOS 7.2.1511

相关工具：Xshel、Xftp

服务器软件软件：.netcore、nginx、supervisor

**准备好发布的程序**

**安装.NET Core SDK for CentOS7**

打开网址：<https://www.microsoft.com/net/core#linuxcentos>

复制如下命令，单步执行：

sudo yum install libunwind libicu

curl -sSL -o dotnet.tar.gz <https://go.microsoft.com/fwlink/?LinkID=835019>

sudo mkdir -p /opt/dotnet && sudo tar zxf dotnet.tar.gz -C /opt/dotnet

sudo ln -s /opt/dotnet/dotnet /usr/local/bin

输入 dotnet –info 来查看是否安装成功

**配置Nginx**

下载安装Nginx，单步执行如下命令：

curl -o nginx.rpm

<http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7.ngx.noarch.rpm>

rpm -ivh nginx.rpm

yum install nginx

systemctl start nginx 来启动nginx

systemctl enable nginx 来设置nginx的开机启动（linux宕机、重启会自动运行nginx不需要连上去输入命令）。

**配置防火墙**

命令：firewall-cmd --zone=public --add-port=80/tcp --permanent（开放80端口）

命令：systemctl restart firewalld（重启防火墙以使配置即时生效）

测试nginx是否可以访问。

**配置nginx对ASP.NET Core应用的转发**

修改 /etc/nginx/conf.d/default.conf 文件，将文件内容替换为：

server {

listen 80;

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;

}

}

上传至CentOS进行覆盖。

执行：nginx –s reload 使其即时生效

运行ASP.NET Core应用程序，偿试再次访问。如果访问不了是由于SELinux保护机制所导致，我们需要将nginx添加至SELinux的白名单，我们通过一些命令解决这个问题。

yum install policycoreutils-python

sudo cat /var/log/audit/audit.log | grep nginx | grep denied | audit2allow -M mynginx

sudo semodule -i mynginx.pp

再次访问应该没有问题。

**配置守护服务（Supervisor）**

安装Supervisor

yum install python-setuptools

easy\_install supervisor

配置Supervisor

mkdir /etc/supervisor

echo\_supervisord\_conf > /etc/supervisor/supervisord.conf

修改supervisord.conf文件，将文件尾部的配置为：

[include]

files = conf.d/\*.ini

或

[include]

files = conf.d/\*.conf

配置对ASP.NET Core应用的守护，创建如下INI文件：

[program:AspNetCoreWebApi]

command=dotnet AspNetCoreWebApi.dll

directory=/root/wwwroot/

autostart=true

autorestart=true

stderr\_logfile=/var/log/AspNetCoreWebApi.err.log

stdout\_logfile=/var/log/AspNetCoreWebApi.out.log

environment=ASPNETCORE\_ENVIRONMENT=Production

user=root

startsecs=1

stopsignal=INT

将文件拷贝至：“/etc/supervisor/conf.d/

运行supervisord，查看是否生效，命令如下：

supervisord -c /etc/supervisor/supervisord.conf

ps -ef | grep AspNetCoreWebApi

如果服务已启动，修改配置文件可用“supervisorctl reload”命令来使其生效。

**配置Supervisor开机启动**

新建一个“supervisord.service”文件

# dservice for systemd (CentOS 7.0+)

# by ET-CS (https://github.com/ET-CS)

[Unit]

Description=Supervisor daemon

[Service]

Type=forking

ExecStart=/usr/bin/supervisord -c /etc/supervisor/supervisord.conf

ExecStop=/usr/bin/supervisorctl shutdown

ExecReload=/usr/bin/supervisorctl reload

KillMode=process

Restart=on-failure

RestartSec=42s

[Install]

WantedBy=multi-user.target

将文件拷贝至：“/usr/lib/systemd/system/supervisord.service”

执行命令：systemctl enable supervisord

执行命令：systemctl is-enabled supervisord #来验证是否为开机启动

使用 supervisorctl

Supervisorctl 是 supervisord 的一个命令行客户端工具，启动时需要指定与 supervisord 使用同一份配置文件，否则与 supervisord 一样按照顺序查找配置文件。

supervisorctl -c /etc/supervisord.conf

上面这个命令会进入 supervisorctl 的 shell 界面，然后可以执行不同的命令了：

> status # 查看程序状态

> stop usercenter # 关闭 usercenter 程序

> start usercenter # 启动 usercenter 程序

> restart usercenter # 重启 usercenter 程序

> reread ＃ 读取有更新（增加）的配置文件，不会启动新添加的程序

> update ＃ 重启配置文件修改过的程序

> status # 查看程序状态