部署文档

Docusign测试服务器配置

测试环境配置：

名称	规格
CPU	2核
内存	4G

硬盘	SSD 30G
静态IP	163.228.234.38

正式环境推荐配置：

名称	规格
CPU	2核
内存	4G
硬盘	SSD 500G
静态IP	是

虚拟机创建

1）主页点击虚拟机

2）点击创建

- 选择资源组，这里选择创建好的资源组，测试环境选择了docusign-test
- 配置虚拟机名称
- 选择镜像，这里选择了ubuntu20.04
- 选择虚拟机配置，测试环境选择了2H 4G
- 选择验证方式，测试环境选择了用户名密码

然后点击下一步配置磁盘

4）选择磁盘类型是标准ssd

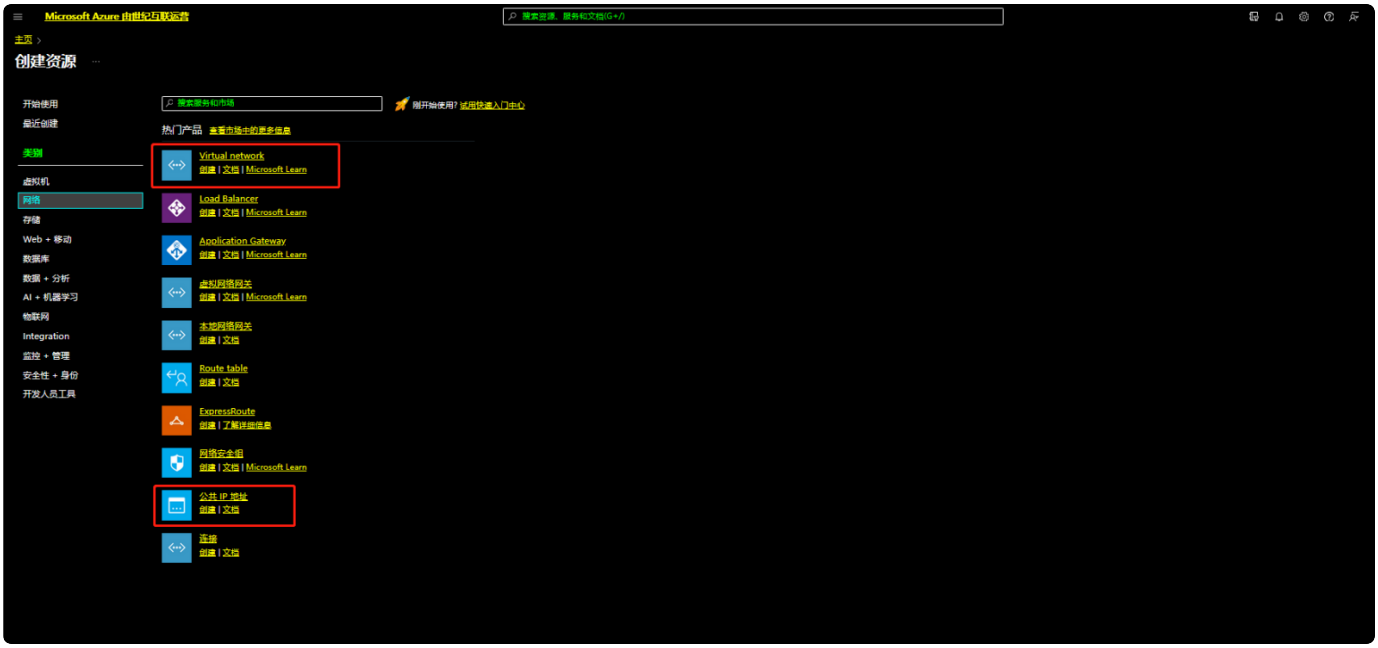
创建一个自定义的30G新标注ssd磁盘

点击下一步网络

5）在配置网络之前我们需要购买一个静态ip

回到主页点击创建资源

创建公共IP和子网



选择资源组

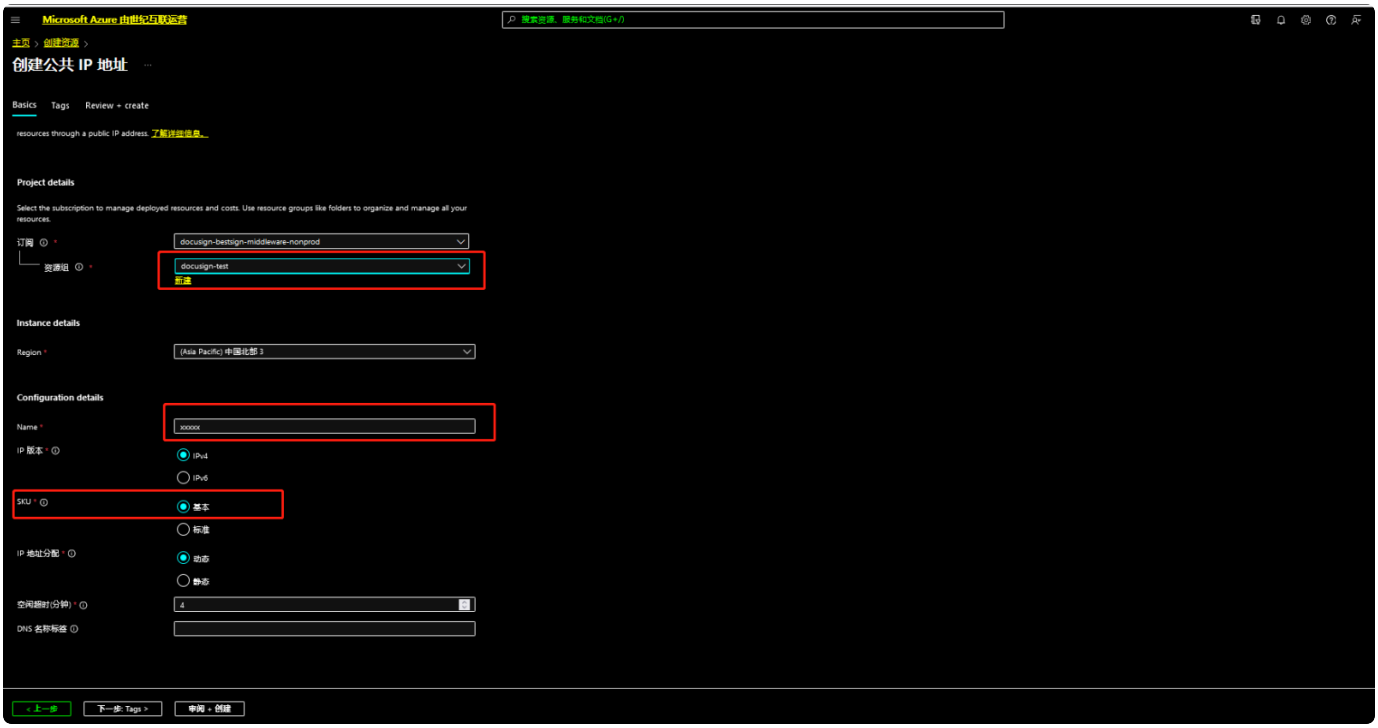
选择创建名字

选择SKU基本

下一步Tags

没有配置继续下一步

然后创建

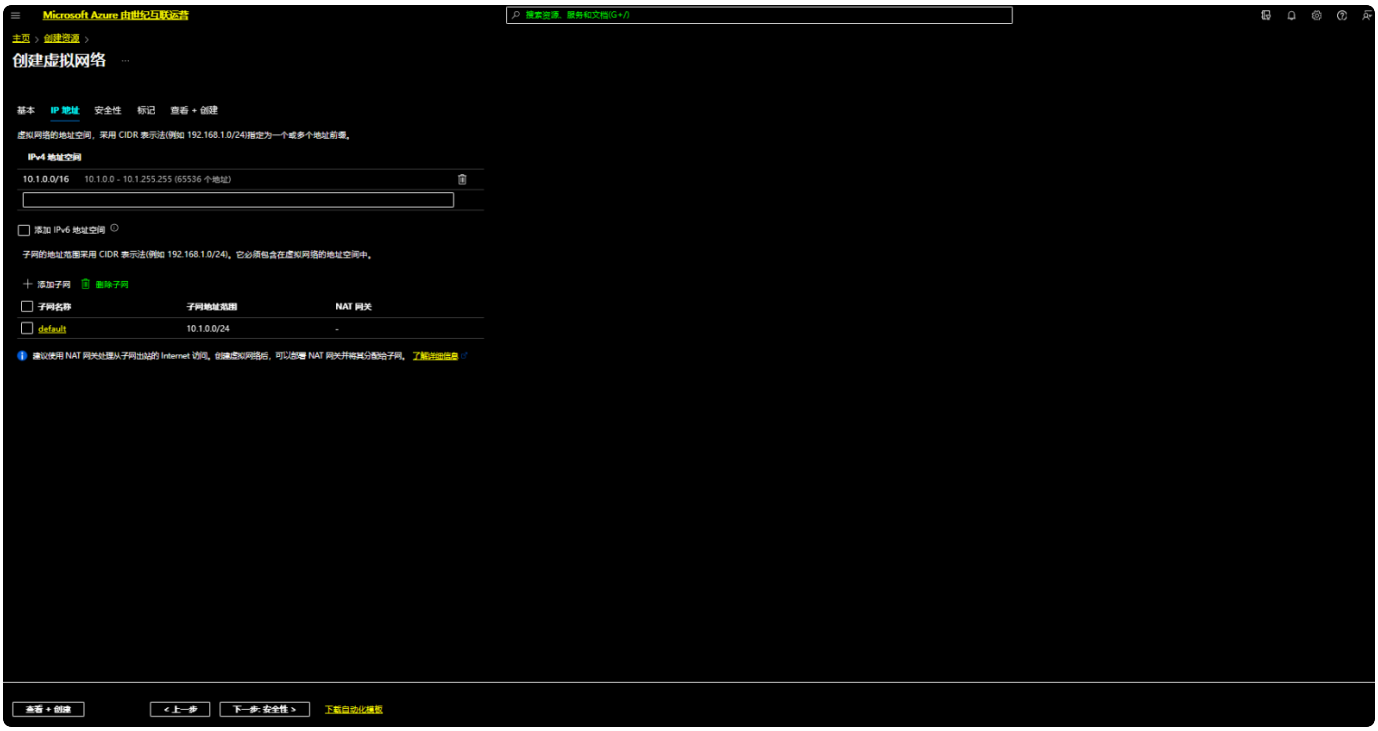


创建子网

选择资源组

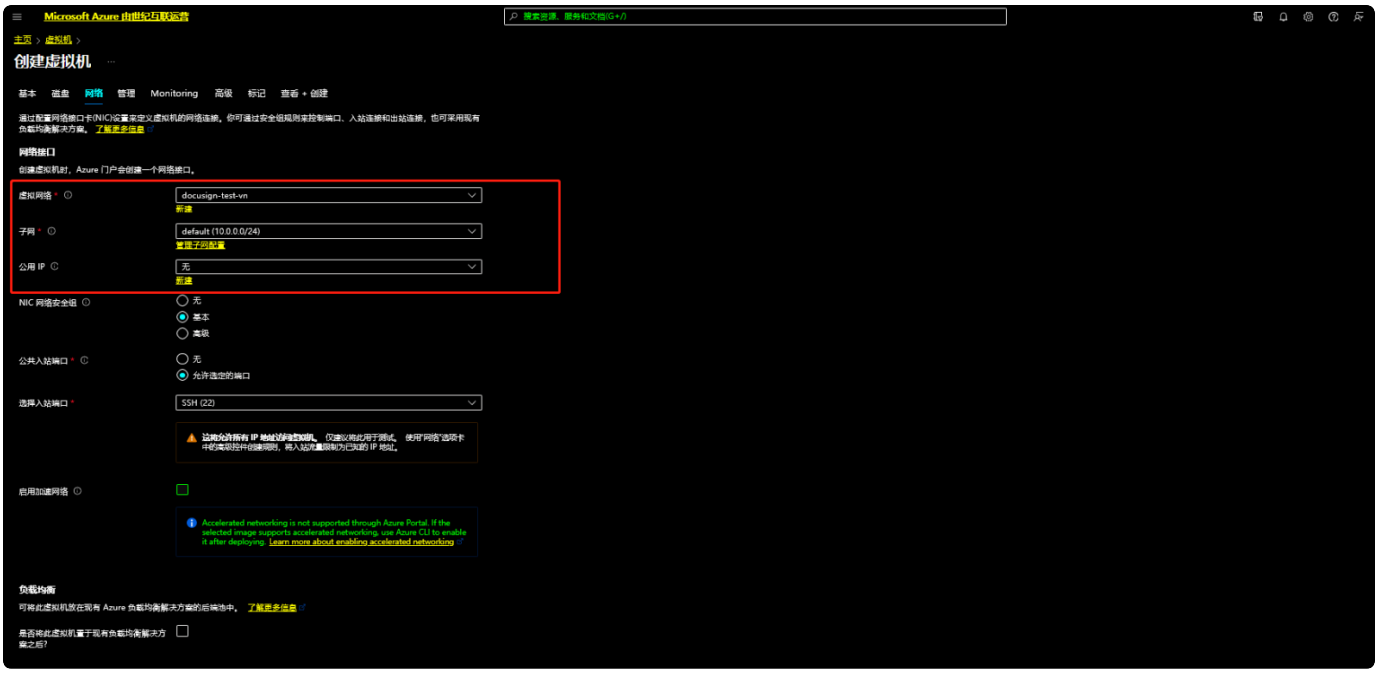
选择名称

下一步



然后下一步直到创建

6）选择刚刚创建好的子网和公网ip然后下一步直到创建完成



Docusign服务器所需服务安装

通过创建服务器的用户名和密码连接到服务器

Docker 安装

更新系统

在开始安装之前，先更新 Ubuntu 系统。可以通过以下命令更新 Bash 复制代码

```
sudo apt-get update sudo apt-get upgrade
```

安装 Docker

执行以下命令，添加 Docker 的 GPG 密钥： Bash 复制代码

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

添加完 GPG 密钥后，添加 Docker 软件包源： Bash 复制代码

```
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

更新源信息： Bash 复制代码

```
sudo apt-get update
```

最后，安装 Docker： Bash 复制代码

```
sudo apt-get install docker-ce
```

启动 Docker

安装完成后，启动 Docker： Bash 复制代码

```
sudo systemctl start docker
```

为了防止每次开机都需要手动启动 Docker，还可以设置 Docker 服务开机自启 Bash 复制代码

```
sudo systemctl enable docker
```

现在 Docker 已经安装完成，可以通过运行以下命令检查 Docker 是否安装成... Bash 复制代码

```
docker --version
```

这将显示安装的 Docker 版本信息。

Nginx 安装📎9806095_www.iscn.com.cn_nginx.zip

创建nginx.conf文件

创建nginx.conf文件 Bash 复制代码

```
vim nginx.conf
```

nginx.conf JSON 复制代码

```
events {
    worker_connections  1024;
}

http {
    include       mime.types;
    default_type  application/octet-stream;
    sendfile        on;
    keepalive_timeout  65;

    server {
        listen       443 ssl;
        server_name  www.iscn.com.cn;

        ssl_certificate      /etc/ssl/certs/cert.pem;
        ssl_certificate_key  /etc/ssl/private/key.pem;
        ssl_protocols        TLSv1.2 TLSv1.3;
        ssl_ciphers          HIGH:!aNULL:!MD5;

        location / {
            # 这里将443端口上的流量代理到本机的80端口
            proxy_pass http://127.0.0.1:80;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Proto https;
        }
        location /mmjj.jpg {
            # 这里将https://163.228.234.38:443/mmjj.jpg代理到本机端口443上
            proxy_pass http://222.128.117.139:8868/mmjj.jpg;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Proto https;
        }

        error_page 502 /502.html;
        location = /502.html {
            root /var/www/errors/;
            internal;
        }
    }
}
```

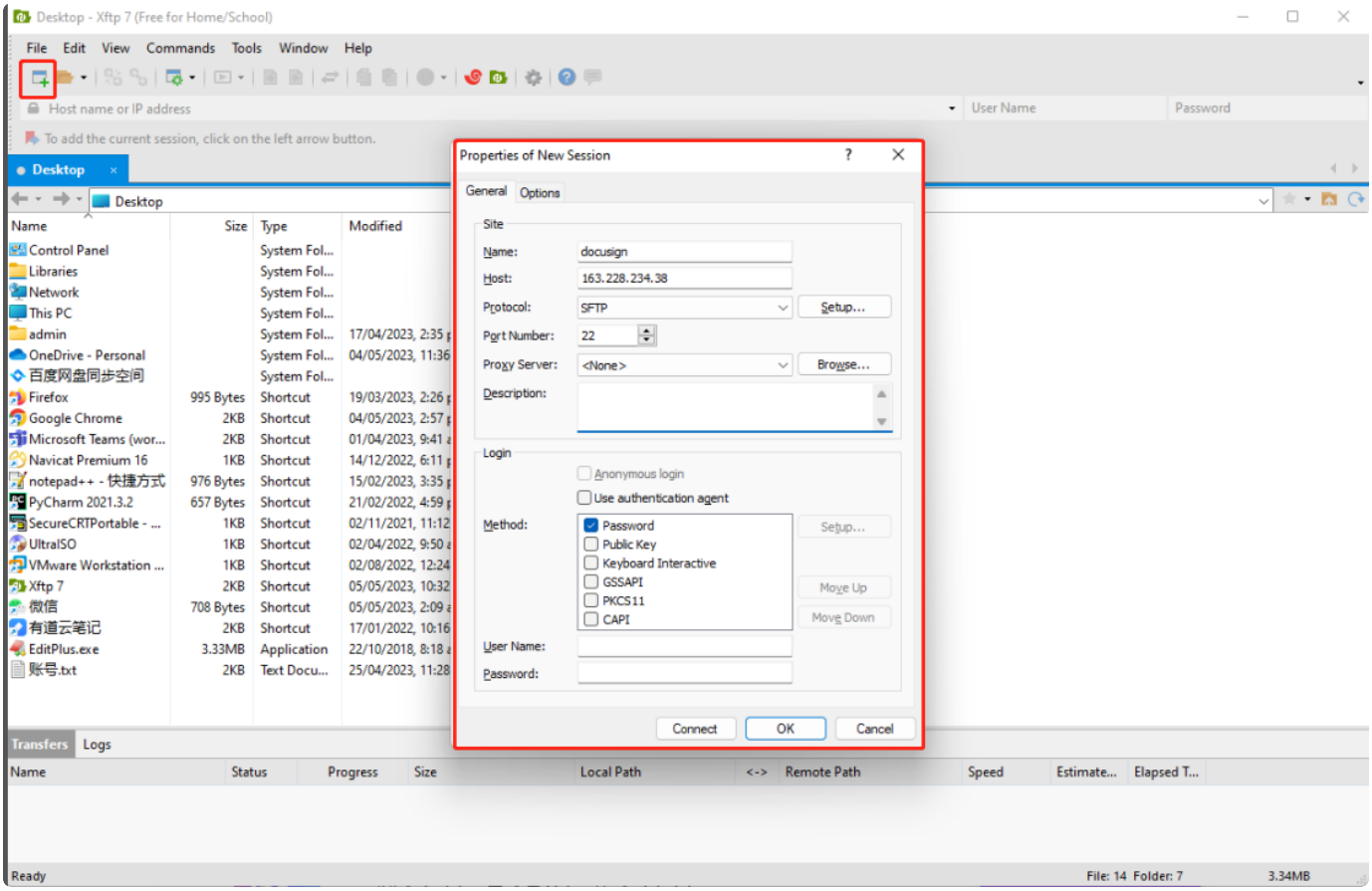
证书准备

将证书下载到本地服务器中在Dockerfile中打包后再本地服务器删除

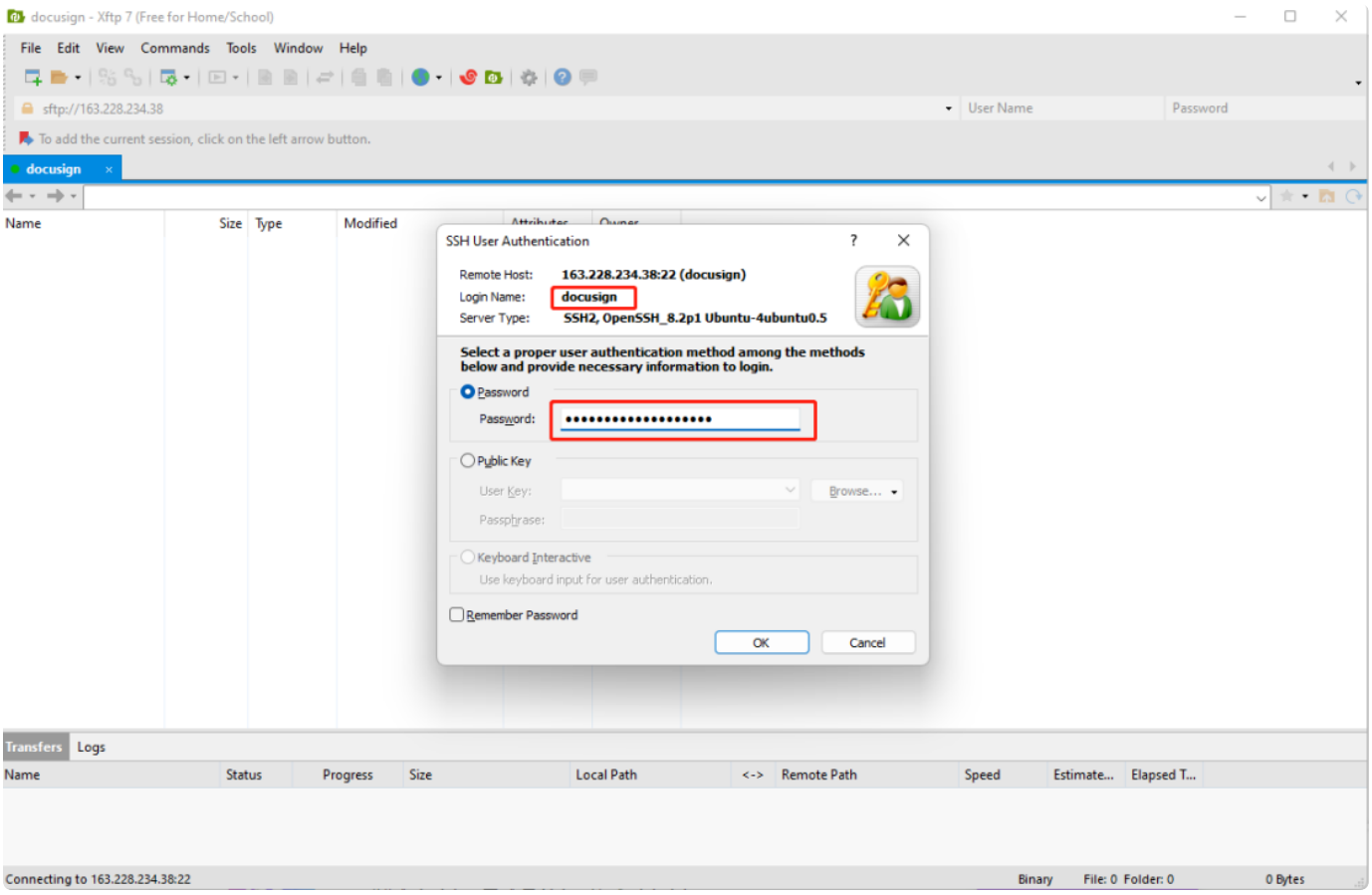
https://www.xshell.com/zh/xshell/ （xshell 服务器ssh工具）

https://www.xshell.com/zh/xftp/ （xftp ftp工具）

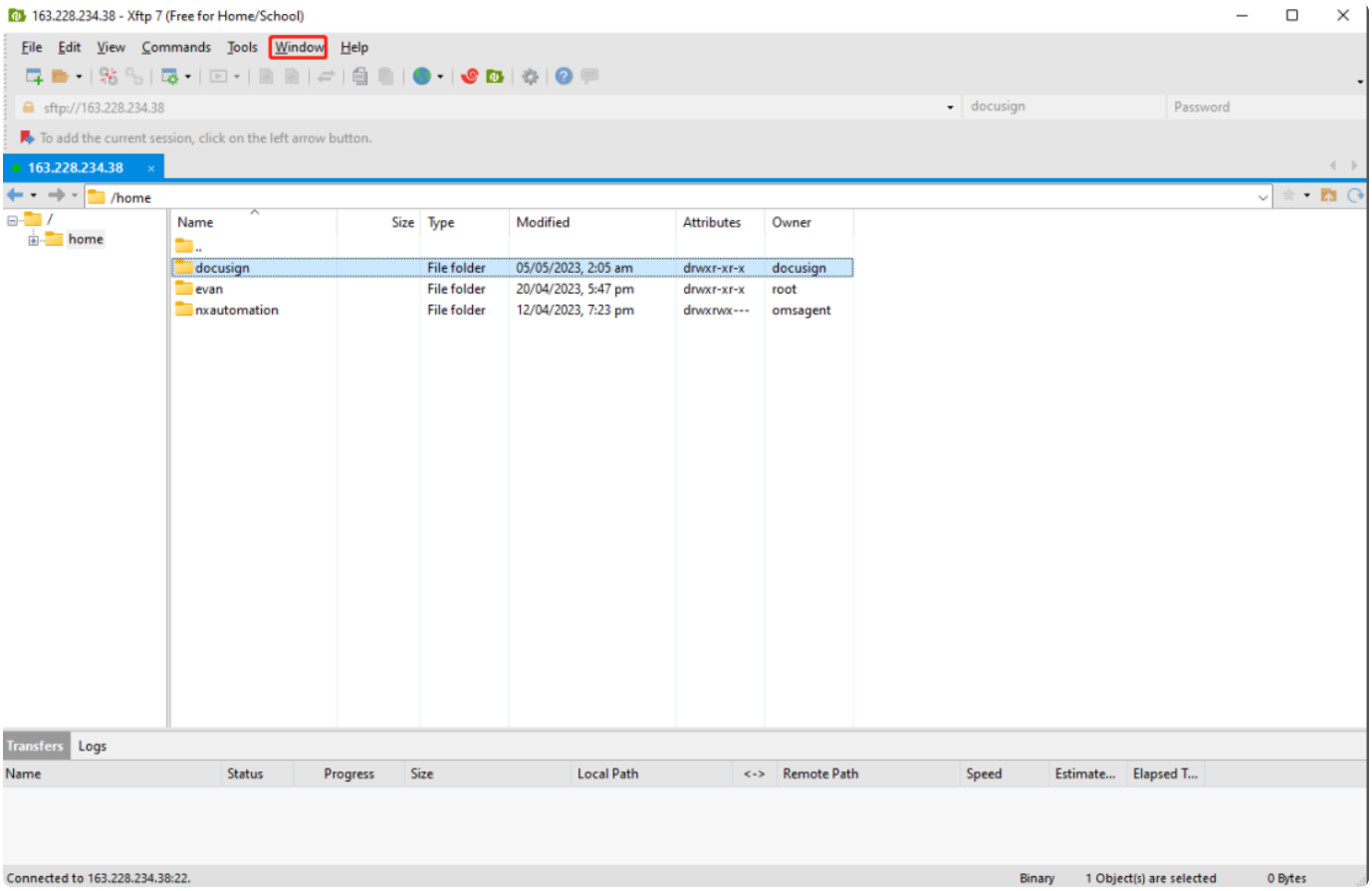
打开下载好的xftp上传本地的证书文件



输入用户名和密码（确保自己的ip在ssh白名单中）



点击window创建一个windows和xftp的链接用来上传文件



将证书文件上传到服务器然后改名为cert.pem 和key.pem

Bash 复制代码

```bash
mv 9806095_www.iscn.com.cn.pem cert.pem
mv 9806095_www.iscn.com.cn.key key.pem
```

打包镜像

创建Dockerfile

Dockerfile

Dockerfile 复制代码

```dockerfile
FROM nginx:latest
# Install required packages
RUN apt-get update && \
    apt-get install -y curl jq && \
    rm -rf /var/lib/apt/lists/*

# Copy the script and make it executable
COPY get_keyvault_cert.sh /get_keyvault_cert.sh
RUN chmod +x /get_keyvault_cert.sh

# Copy the Nginx configuration file
COPY nginx.conf /etc/nginx/nginx.conf
COPY cert.pem /etc/ssl/certs/cert.pem
COPY key.pem /etc/ssl/private/key.pem
COPY 5021.html /var/www/errors/502.html
COPY mmjj.jpg /var/www/errors/mmjj.jpg

# Change ownership and permissions of error page and image
RUN chown -R nginx:nginx /var/www/errors \
    && chmod 644 /var/www/errors/*.html \
    && chmod 644 /var/www/errors/*.jpg
CMD  nginx -g "daemon off;"
```

运行Dockerfile

运行Dockerfile

Bash 复制代码

```bash
docker build -t my-nginx .
```

运行成功后删除本地证书 Bash 复制代码

```
rm -rf cert.pem key.pem
```

中间件安装📎app_publish.tar

发布程序

1. 通过 terminal 进入系统源代码目录下 app 目录。
2. 发布程序

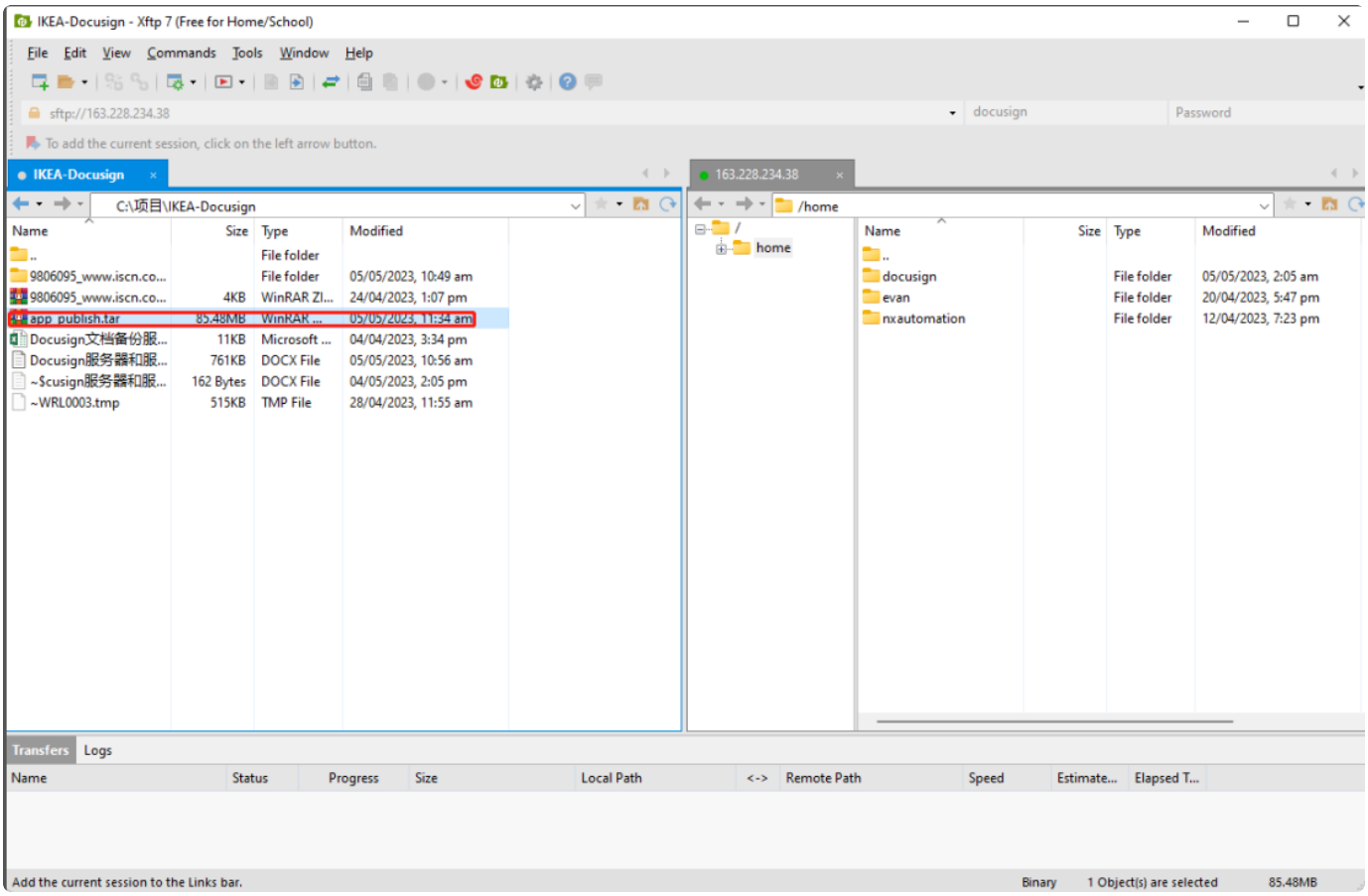
发布程序 Bash 复制代码

```
dotnet publish -c Release
```

3. 将文件打包上传至服务器

部署文件上传

使用xftp工具上传publish.tar 到服务器解压



构建镜像

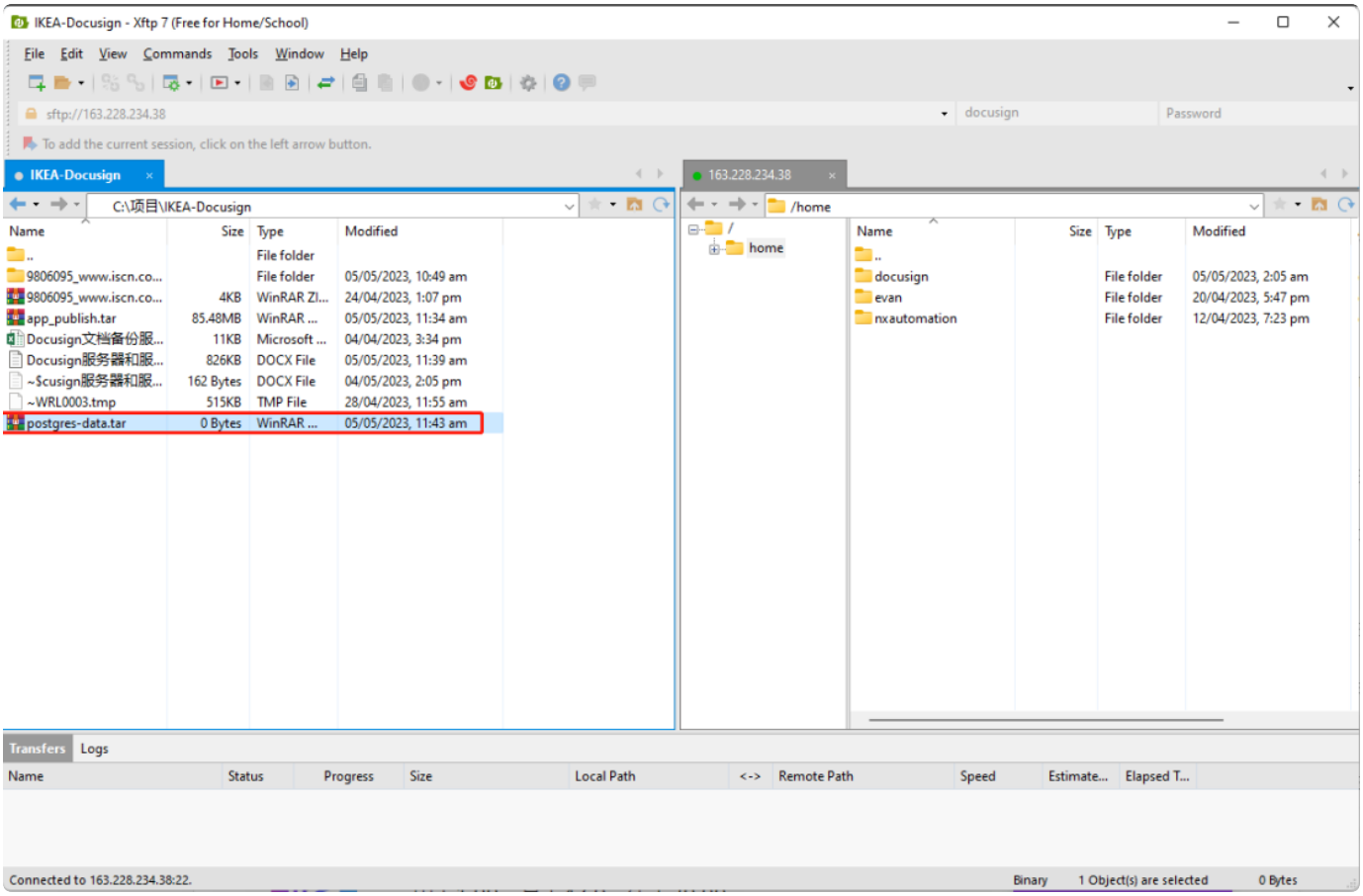
构建镜像 Bash 复制代码

```
tar xf app_publish.tar
cd app_publish
docker build -t app .
```

Postgres 安装📎postgres-data.tar

上传静态文件postgres-data到服务器

使用xftp工具将postgres-data.tar上传到服务解压



解压postgres-data.tar Bash 复制代码

```
tar xf postgres-data.tar
```

Docker Compose 配置

创建docker-compose Bash 复制代码

```
vim docker-compose.yml
```

docker-compose.yml Plain Text 复制代码

```
version: '3.7'
services:
  db:
    image: postgres
    restart: always
    environment:
      POSTGRES_PASSWORD: MyP@ssw0rd123!
      POSTGRES_PASSWORD_REQUIREMENTS: 'length:8,uppercase:1,lowercase:1,number:1,special:1'
    volumes:
      - ./postgres-data:/var/lib/postgresql/data
    ports:
      - "5432:5432"

  nginx:
    image: my-nginx
    restart: always
    volumes:
      - ./nginx.conf:/etc/nginx/nginx.conf
#    ports:
#      - "443:443"
    network_mode: host

  app:
    image: app
    environment:
      TZ: Asia/Shanghai
    restart: always
    ports:
      - "80:80"
#    volumes:
#      - ./appsettings.json:/app/appsettings.json
```

启动程序

启动docker-compose Bash 复制代码

```
docker-compose up -d
docker-compose ps
```

```
root@docusign-test-vm:/home/docusign/cert# docker-compose ps
    Name                   Command               State                    Ports
-------------------------------------------------------------------------------------------
cert_app_1     dotnet App.dll                   Up      0.0.0.0:80->80/tcp,:::80->80/tcp
cert_db_1      docker-entrypoint.sh postgres    Up      0.0.0.0:5432->5432/tcp,:::5432->5432/tcp
cert_nginx_1   /docker-entrypoint.sh /bin ...   Up
root@docusign-test-vm:/home/docusign/cert#
```