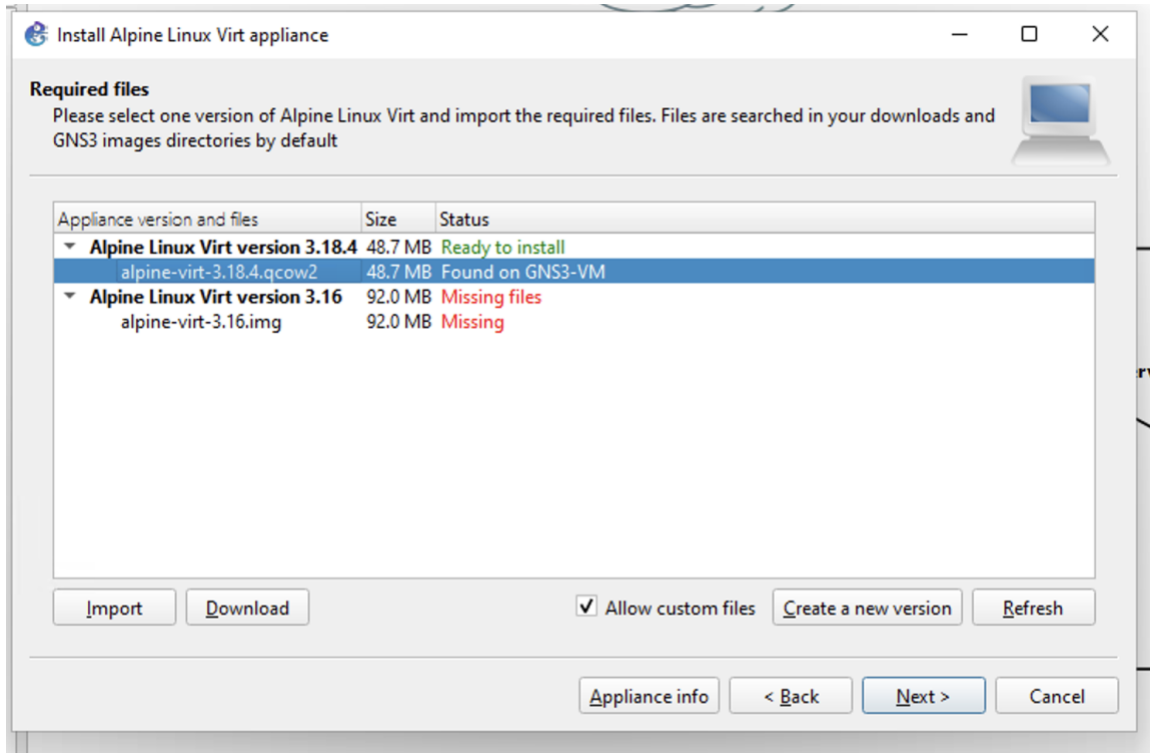


# Lab Guide: Alpine Linux Deployment, Configuring Services and FortiGate Integration

## 1. Adding Alpine Linux to GNS3



- In the GNS3 left-hand menu, click the "+" (**New Template**) icon.
- Select **"Install an appliance from the GNS3 server (recommended)"**.
- Search for "Alpine".
- Select **"Alpine Linux virt"** and click **Install**. Follow the Wizard.
- Choose the latest version click on download button to have GNS3 fetch it and download automatically
- Import the .qcow2 file when completed.
- Click **Next** and **Finish**. The template will now appear in your "End Devices" list.

## 2. Deploying Services on Alpine Linux

### Step 1: Configure Static IP

- **Access Interfaces File:** `vi /etc/network/interfaces`
- **Configure Static IP:** (In the vi editor Press `i` to insert, paste the content below, then press `Esc`, type `:wq`, and press `Enter`)

```
auto eth0
iface eth0 inet static
    address 10.0.4.50
    netmask 255.255.255.0
    gateway 10.0.4.1
```

- **Apply Changes:** rc-service networking restart
- **Update System:** apk update

## Step 2: OpenSSH Server Configuration

- **Install Package:** apk add openssh
- **Enable & Start:**
  - rc-update add sshd default
  - rc-service sshd start
- **Root Access (Optional):**
  - Edit /etc/ssh/sshd\_config and set PermitRootLogin yes.
  - Restart service: rc-service sshd restart.

## Step 3: Nginx Web Server Configuration

- **Install Package:** apk add nginx
- **Create Configuration File:** vi /etc/nginx/nginx.conf
- *(In the vi editor Press i to insert, paste the content below, then press Esc, type :wq, and press Enter)*

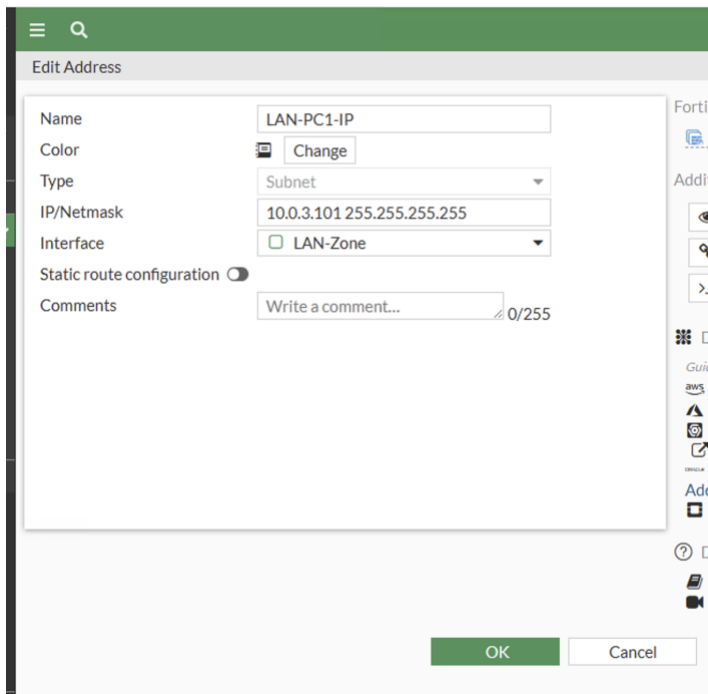
```
worker_processes 1;
events { worker_connections 1024; }
http {
    server {
        listen 80;
        location / {
            root /var/www/localhost/htdocs;
            index index.html;
        }
    }
}
EOF
```

- **Deploy Content & Permissions:**
  - echo "<h1>Lab Success!</h1>" >  
/var/www/localhost/htdocs/index.html
  - chown -R nginx:nginx /var/www/localhost/htdocs
  - chmod -R 755 /var/www/localhost/htdocs
- **Enable & Start:**
  - rc-update add nginx default
  - rc-service nginx start

### 3. Firewall Integration (FortiGate UI)

To ensure your LAN PCs can reach the Alpine servers, you must define the "Objects" (IP addresses) and then create a "Policy" to allow the traffic.

#### Step 1: Create Address Objects



The screenshot shows the 'Edit Address' configuration window in the FortiGate UI. The window has a green header bar with a menu icon and a search icon. The title 'Edit Address' is displayed. The configuration fields are as follows:

- Name:** LAN-PC1-IP
- Color:** A color selection icon followed by a 'Change' button.
- Type:** Subnet (selected from a dropdown menu)
- IP/Netmask:** 10.0.3.101 255.255.255.255
- Interface:** LAN-Zone (selected from a dropdown menu)
- Static route configuration:** A toggle switch that is currently turned off.
- Comments:** Write a comment... (with a character count of 0/255)

At the bottom of the window are 'OK' and 'Cancel' buttons. On the right side, there is a vertical toolbar with various icons for actions like 'Add', 'Edit', 'Delete', and 'Help'.

- Go to **Policy & Objects > Addresses**.
- Click **Create New > Address**. (repeat for all 4 servers).
  - **Name:** Server01-IP-Alpine
  - **Type:** Subnet/IP Range.
  - **IP/Netmask:** 10.0.4.50/255.255.255.255 (Ensure this matches the IP set on your end devices).
  - **Interface:** Select the interface the Alpine server is connected to (or any).
  - Click **OK**.

## Step 2: Create Firewall Policy

| Name                                  | Source     | Destination        | Schedule | Service       | Action   | NAT        | Security Profile |
|---------------------------------------|------------|--------------------|----------|---------------|----------|------------|------------------|
| [-] [-] LAN-Zone -> [-] Server-Zone 3 |            |                    |          |               |          |            |                  |
| LAN-To-Server01-Allow_WEB             | LAN-PC1-IP | Server01-IP-Alpine | always   | HTTP<br>HTTPS | ✓ ACCEPT | ✓ Enabled  | SSL              |
| LAN-To-Server01-Allow_SSH             | LAN-PC2-IP | Server01-IP-Alpine | always   | SSH           | ✓ ACCEPT | ✓ Enabled  | SSL              |
| LAN-to-Server-PING-ALL                | all        | all                | always   | PING          | ✓ ACCEPT | ✗ Disabled | SSL              |
| [+] [-] LAN-Zone -> [-] WAN-Zone 1    |            |                    |          |               |          |            |                  |
| [+] [-] Server-Zone -> [-] LAN-Zone 1 |            |                    |          |               |          |            |                  |
| [+] [-] Server-Zone -> [-] WAN-Zone 1 |            |                    |          |               |          |            |                  |
| [+] Implicit 1                        |            |                    |          |               |          |            |                  |

- Go to **Policy & Objects > Firewall Policy**.
- Click **Create New**.
  - **Name:** Allow\_LAN\_to\_Alpine\_Services.
  - **Incoming Interface:** Select the interface facing your **LAN PCs**.
  - **Outgoing Interface:** Select the interface facing your **Alpine Servers**.
  - **Source:** Select the **all** object or the specific address object of your LAN PCs.
  - **Destination:** Select your **Alpine\_Server\_1** (and any other server objects).
  - **Schedule:** **always**.
  - **Service:** Select **SSH** and **HTTP** (or create a custom group containing TCP 22 and 80).
  - **Action:** **ACCEPT**.
  - **NAT:** Usually **OFF** if you are routing between subnets in a lab, or **ON** if you are doing simple port forwarding.
- **Logging:** In the Firewall Policy settings, ensure Log Allowed Traffic is set to All Sessions.

## 4. Lab Verification Checklist

- **SSH Test:** Run `ssh root@192.168.1.50` from your LAN-PCs.
- **Web Test:** Navigate to `http://192.168.1.50` in your LAN-PCs.

Record which specific PCs were able to successfully reach the services to verify your firewall policy implementation.