Complete Guide to Installing NGINX, Configuring a Self-Signed SSL Certificate, and Testing on Ubuntu

This guide compiles all the steps discussed for installing NGINX on Ubuntu, configuring it with a self-signed SSL certificate, verifying setups, troubleshooting common issues, and using essential tools like `nano` and `netstat`. It's tailored for Ubuntu 20.04 LTS or later (tested on a setup as of September 17, 2025), assuming a server like `project-1-12.eduhk.hk` with IP `192.168.56.182`.

Prerequisites

- Ubuntu server (20.04 LTS or later).
- User with `sudo` privileges.
- Terminal access.
- Internet connection.
- Basic command-line knowledge.

Using Nano Text Editor

`nano` is a simple terminal-based editor used for editing NGINX configs and files. Key shortcuts (especially for the ones mentioned):

- **Open a file**: `sudo nano /path/to/file` (e.g., `sudo nano /etc/nginx/sites-available/default`).
- **Remove all lines (clear content)**: Press `Ctrl + K` repeatedly to cut (delete) lines one by one, or hold it to clear faster.
- **Paste content**: Right-click in the terminal to paste (or use `Ctrl + U` to uncut/paste previously cut text).
- **Save and exit**: Press `Ctrl + X`, then `Y` (yes) to confirm saving changes.
- **Other useful shortcuts**:
  - `Ctrl + O`: Save without exiting.
  - `Ctrl + W`: Search.
  - `Ctrl + G`: View all shortcuts.

Always test configs after editing (e.g., `sudo nginx -t`).

Part 1: Installing NGINX

Step 1: Update Package Index

```
sudo apt update
```

Step 2: Install NGINX

```
sudo apt install nginx
```

Step 3: Verify and Start NGINX

```
sudo systemctl status nginx
sudo systemctl start nginx  # If not running
sudo systemctl enable nginx  # Start on boot
```

Step 4: Test Basic Installation

Find your server IP:

```
ip addr show | grep inet  # Shows all IPs; look for e.g., 192.168.56.182 under ens33
# Or shorter: hostname -I
# Public IP (if needed): curl ifconfig.me
```

Visit `http://<your_ip>` (e.g., `http://192.168.56.182`) in a browser. You should see the NGINX welcome page.

Part 2: Installing OpenSSL (for Certificates)

```
sudo apt install openssl
```

Part 3: Generating a Self-Signed SSL Certificate

Step 1: Create SSL Directory

```
sudo mkdir /etc/nginx/ssl
```

Step 2: Generate Certificate

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/nginx/ssl/nginx.key -out /etc/nginx/ssl/nginx.crt
```

During prompts (Distinguished Name fields):

- **Country Name (2 letter code) [AU]**: `HK` (for Hong Kong; press Enter for defaults otherwise).
- **State or Province Name (full name) [Some-State]**: `Hong Kong` (or `.` to leave blank).
- **Locality Name (e.g., city) []**: `Hong Kong` (or `.`).
- **Organization Name (e.g., company) []**: `YourOrg` (or `.`).
- **Organizational Unit Name (e.g., section) []**: `IT` (or `.`).

- **Common Name (e.g., server FQDN) []**: `project-1-12.eduhk.hk` (use your domain/FQDN; verify with `ping project-1-12.eduhk.hk`).
- **Email Address []**: `admin@eduhk.hk` (or `.`).

This creates a 1-year valid, 2048-bit RSA self-signed cert. Secure files:

```
sudo chmod 600 /etc/nginx/ssl/nginx.key
sudo chmod 644 /etc/nginx/ssl/nginx.crt
```

Part 4: Configuring NGINX for HTTPS

Step 1: Edit Configuration

```
sudo nano /etc/nginx/sites-available/default
```

Use `Ctrl + K` to clear existing content if needed, paste the new config (right-click to paste), then `Ctrl + X` > `Y` to save/exit.

Replace with:

```
server {
    listen 80;
    listen [::]:80;
    server_name project-1-12.eduhk.hk;

    # Redirect HTTP to HTTPS
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    listen [::]:443 ssl;
    server_name project-1-12.eduhk.hk;

    ssl_certificate /etc/nginx/ssl/nginx.crt;
    ssl_certificate_key /etc/nginx/ssl/nginx.key;

    root /var/www/html;  # Default root; not /var/www/root
    index index.html index.htm index.nginx-debian.html;

    location / {
        try_files $uri $uri/ /index.html;
    }
}
```

- Replace `project-1-12.eduhk.hk` with your domain/IP if needed.

- Root is `/var/www/html` by default (verify: `cat /etc/nginx/sites-available/default | grep root`).

Step 2: Test and Reload

```
sudo nginx -t  # Check syntax
sudo systemctl reload nginx
```

Step 3: Firewall (if using UFW)

```
sudo ufw allow 'Nginx Full'  # Allows 80 and 443
sudo ufw status
```

Part 5: Creating Test Content

Default root: `/var/www/html` (not `/var/www/root`—that's non-standard).

```
ls -l /var/www/html  # Check contents
echo "<h1>Hello from project-1-12.eduhk.hk with HTTPS!</h1>" | sudo tee
/var/www/html/index.html
sudo chown www-data:www-data /var/www/html/index.html
sudo systemctl reload nginx
```

Part 6: Testing the Setup

Step 1: Check Listening Ports

Install `netstat` if missing:

```
sudo apt install net-tools
```

Then:

```
sudo netstat -tuln | grep ':80\|:443'
# Alternative (no install): sudo ss -tuln | grep ':80\|:443'
```

Expected:

```
tcp 0 0 0.0.0.0:80  0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:443 0.0.0.0:* LISTEN
```

```
tcp6 0 0 :::80 :::* LISTEN
tcp6 0 0 :::443 :::* LISTEN
```

Step 2: Test SSL with OpenSSL

Basic connection:

```
openssl s_client -connect 192.168.56.182:443  # Use your IP
```

Full HTTP test:

```
echo -e "GET / HTTP/1.1\r\nHost: project-1-12.eduhk.hk\r\nConnection: close\r\n\r\n" | openssl s_client -connect 192.168.56.182:443 -servername project-1-12.eduhk.hk
```

Expected: `HTTP/1.1 200 OK` with your `<h1>` content. Self-signed warning (`verify error:num=18`) is normal.

Step 3: Browser Test

- Visit `https://project-1-12.eduhk.hk` or `https://192.168.56.182`.
- Accept "not private" warning (self-signed).
- Should show your test page.

Part 7: Troubleshooting

Issue	Possible Cause	Solution
`netstat: command not found`	Not installed	`sudo apt install net-tools` or use `ss`.
No HTTP response in OpenSSL	Empty `/var/www/html` or bad config	Create `index.html`; check `sudo nginx -t`.
400 Bad Request	Incomplete request in basic `openssl s_client`	Use full GET command; normal for partial requests.
Port not listening	Config error or NGINX down	`sudo systemctl status nginx`; reload.
Browser mismatch warning	CN doesn't match access method	Use domain in browser; regenerate cert if using IP.
Certificate fields wrong (e.g., C=AU)	Default prompts	Regenerate with correct inputs (e.g., HK).

Logs

```
```

```
sudo tail -f /var/log/nginx/error.log  # Errors
sudo tail -f /var/log/nginx/access.log  # Access
sudo journalctl -u nginx  # System logs
```

Part 8: Security and Best Practices

- **Self-Signed Limits**: Warnings in browsers; use Let's Encrypt for production.
- **Updates**: `sudo apt update && sudo apt upgrade`.
- **Backup**: Copy `/etc/nginx/ssl/*` securely.
- **Trust Cert Locally** (testing): `sudo cp /etc/nginx/ssl/nginx.crt /usr/local/share/ca-certificates/nginx.crt && sudo update-ca-certificates`.

Conclusion

This covers installing NGINX, SSL setup, testing, and tools like `nano` and `netstat`. Your setup on `project-1-12.eduhk.hk` (IP: 192.168.56.182) should now serve HTTPS content. For advanced topics, see NGINX Docs.