**Document for Tutorial Jenkins Reverse proxy ,Domain assign & ssl Certification**

Jenkins installation

[Linux (jenkins.io)](https://www.jenkins.io/doc/book/installing/linux/#debianubuntu)

sudo apt update

sudo apt install fontconfig openjdk-17-jre

java -version

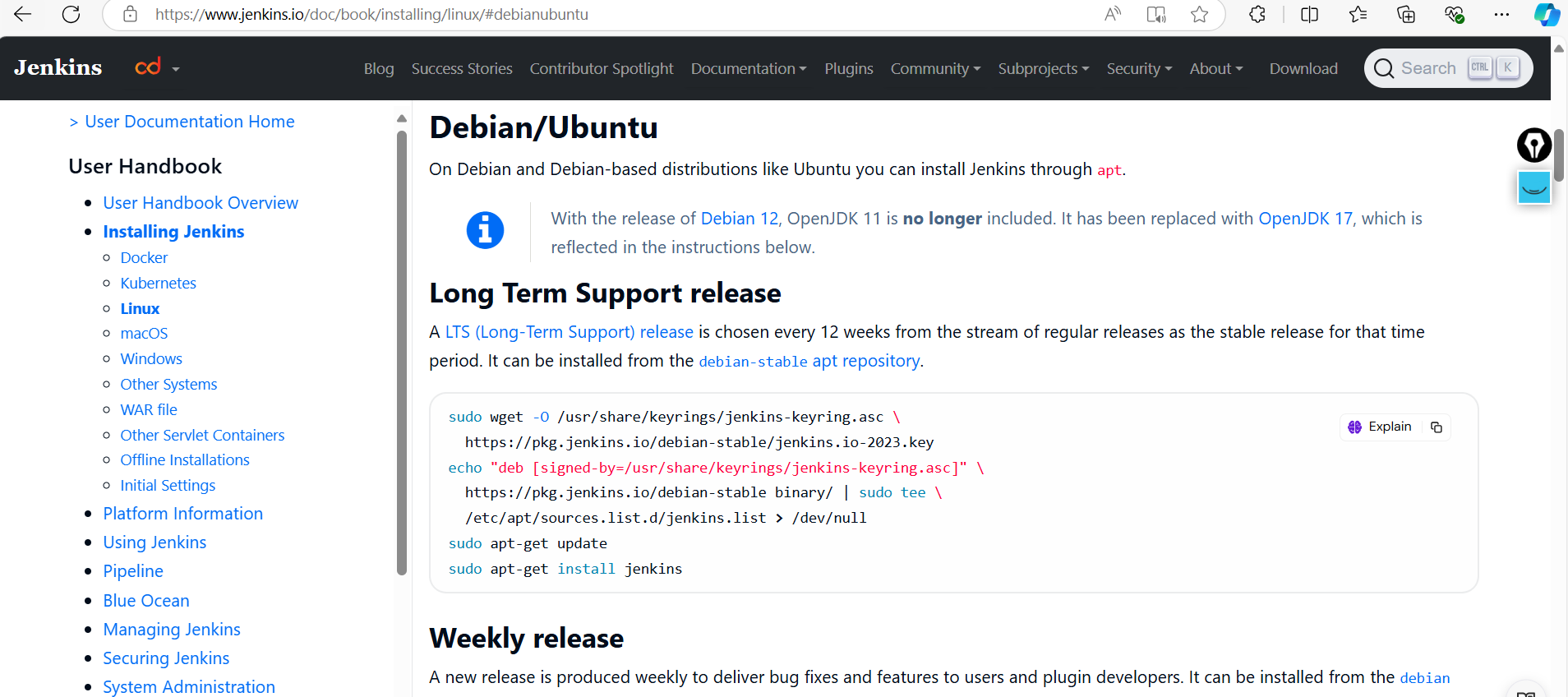
openjdk version "17.0.8" 2023-07-18

OpenJDK Runtime Environment (build 17.0.8+7-Debian-1deb12u1)

OpenJDK 64-Bit Server VM (build 17.0.8+7-Debian-1deb12u1, mixed mode, sharing)

[Linux (jenkins.io)](https://www.jenkins.io/doc/book/installing/linux/#debianubuntu)

**Long Term Support release**

Follow this link for updated on official website   
<https://www.jenkins.io/doc/book/installing/linux/#debianubuntu>   
  


sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \

https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key

echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \

https://pkg.jenkins.io/debian-stable binary/ | sudo tee \

/etc/apt/sources.list.d/jenkins.list > /dev/null

sudo apt-get update

sudo apt-get install jenkins

**Start Jenkins**

You can enable the Jenkins service to start at boot with the command:

sudo systemctl enable jenkins

You can start the Jenkins service with the command:

sudo systemctl start jenkins

You can check the status of the Jenkins service using the command:

sudo systemctl status jenkins

**Nginx Webserver is installation**

Install Nginx: Install Nginx using the following command:

sudo apt install nginx

Start Nginx: Once the installation is complete, start the Nginx service:

sudo service nginx start

This command starts the Nginx service, and it will now be running on your EC2 instance.

Enable Nginx to Start on Boot: To ensure Nginx starts automatically when your server restarts, enable it as a startup service:

sudo systemctl enable nginx

Verify Nginx Installation: Open your web browser and enter your EC2 instance's public IP address or domain name. You should see the default Nginx welcome page, indicating that Nginx is successfully installed and running.

Create a Jenkins Server Block:

Create a new Nginx server block configuration file for Jenkins. You can do this by creating a new file in the /etc/nginx/sites-available/ directory. Let's name it jenkins:

sudo nano /etc/nginx/sites-available/jenkins

Config File

server {

listen 80;

server\_name jenkins.example.com; # Replace with your domain

location / {

proxy\_pass http://localhost:8080; # Jenkins is running on the default port

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 $scheme;

}

location ~ /\. {

deny all;

}

}

Enable the Jenkins Server Block:

Create a symbolic link to the configuration file in the sites-enabled directory:

sudo ln -s /etc/nginx/sites-available/jenkins /etc/nginx/sites-enabled/

Test Nginx Configuration:

1. Before restarting Nginx, it's a good idea to test the configuration to make sure there are no syntax errors:

sudo nginx -t

If the test is successful, you should see: **nginx: configuration file /etc/nginx/nginx.conf test is successful**.

1. **Restart Nginx:** Restart Nginx to apply the changes:

sudo systemctl restart nginx

1. **Access Jenkins:** Open your web browser and navigate to **http://jenkins.example.com**. Replace **jenkins.example.com** with your actual domain. You should now be able to access Jenkins through Nginx.

SSL certificate applied to Domain .

sudo apt install python3-certbot-nginx

certbot --version

certbot --nginx -d xyz.com