

DEVOPS

ASSIGNMENT 1

Installation of Nginx.

Task Description:

This task involves creating a Jenkins Freestyle job that automates the installation of Nginx on an Ubuntu server using the command `sudo apt-get install -y nginx`. This automation ensures consistent server setup and can be easily repeated, making it ideal for automated environments.

Steps:

1. **Update Package List:**

- Run the following command to update the package list:
- `sudo apt update`

2. **Install Nginx:**

- Install Nginx using the following command:
- `sudo apt-get install -y nginx`

3. **Verify Installation:**

- Check if Nginx is running with the command:
- `systemctl status nginx`

4. **Create Jenkins Freestyle Job:**

- In Jenkins, click New Item and create a Freestyle project (e.g., `Install_Nginx`).

5. **Configure Job:**

- Add a build step of type Execute shell.
- In the shell command box, enter:
- `sudo apt update`
- `sudo apt-get install -y nginx`

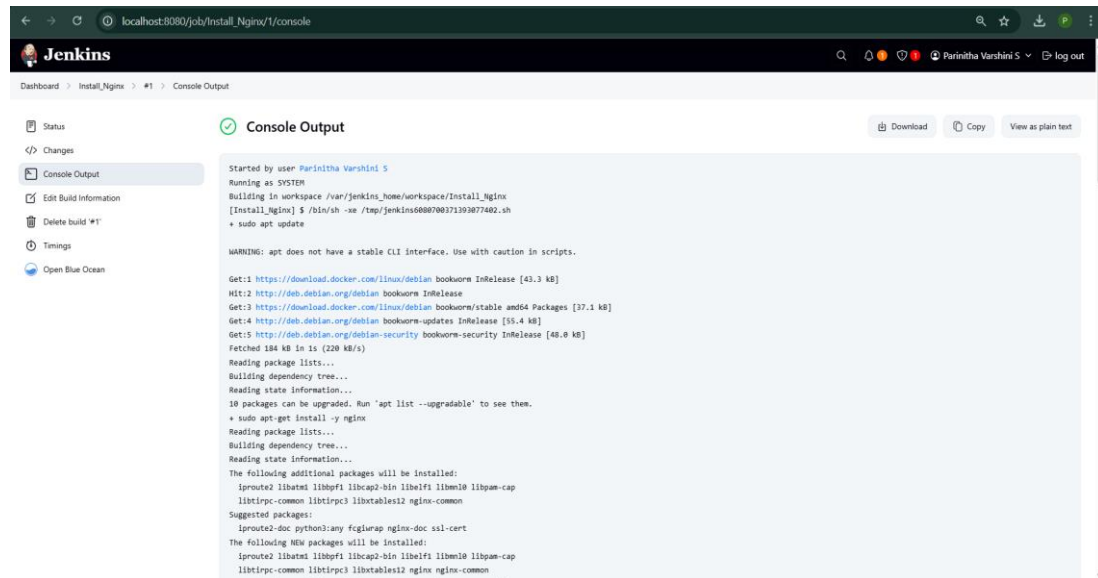
6. **Save and Run Job:**

- Save the job and click Build Now to run it.

7. Check Console Output:

- After the job completes, view the **Console Output** to confirm the successful installation of Nginx.

OUTPUT:

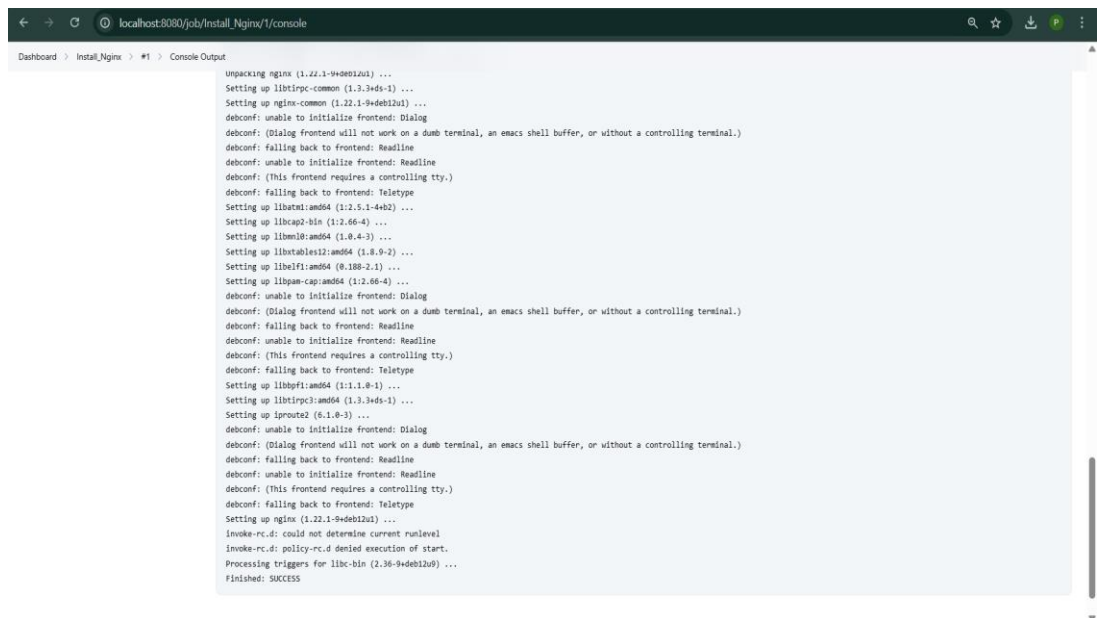


The screenshot shows the Jenkins web interface at localhost:8080/job/Install_Nginx/1/console. The left sidebar contains links for Status, Changes, Console Output (selected), Edit Build Information, Delete build #1, Timings, and Open Blue Ocean. The main area displays the console output for the 'Install_Nginx' job, started by user 'Parinitha Varshini S'. The output shows the execution of 'sudo apt update' and the installation of Nginx. It lists several additional packages that will be installed along with Nginx, including libatni, libbpf, libcap2-bin, libelf1, libm10, libpam-cap, libtirpc-common, libtirpc, libtstables12, and nginx-common. It also suggests packages like python3-any, fcgiwrap, nginx-doc, and ssl-cert.

```
Started by user Parinitha Varshini S
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/Install_Nginx
[Install_Nginx] $ /bin/sh -xe /tmp/jenkins6890700371393877482.sh
+ sudo apt update

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Get:1 https://download.docker.com/linux/debian bookworm InRelease [49.3 kB]
Hit:2 http://deb.debian.org/debian bookworm InRelease
Get:3 https://download.docker.com/linux/debian bookworm/stable amd64 Packages [37.1 kB]
Get:4 http://deb.debian.org/debian bookworm-updates InRelease [55.4 kB]
Get:5 http://deb.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Fetched 184 kB in 1s (220 kB/s)
Reading package lists...
Building dependency tree...
Reading state information...
10 packages can be upgraded. Run 'apt list --upgradable' to see them.
+ sudo apt-get install -y nginx
Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  iproute2 libatni libbpf libcap2-bin libelf1 libm10 libpam-cap
  libtirpc-common libtirpc libtstables12 nginx-common
Suggested packages:
  iproute2-doc python3:any fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  iproute2 libatni libbpf libcap2-bin libelf1 libm10 libpam-cap
  libtirpc-common libtirpc libtstables12 nginx nginx-common
.
```



This screenshot shows the continuation of the console output from the previous image. It details the unpacking of Nginx and its dependencies, the setting up of various components like libtirpc-common, nginx-common, and libbpf, and the successful completion of the installation process. The output includes messages from debconf about frontend initialization and the final 'Finished: SUCCESS' status.

```
Unpacking nginx (1.22.1-9+deb12u1) ...
Setting up libtirpc-common (1.3.3+ds-1) ...
Setting up nginx-common (1.22.1-9+deb12u1) ...
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or without a controlling terminal.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
Setting up libatni:amd64 (1:2.5.1-4+b2) ...
Setting up libcap2-bin (1:2.66-4) ...
Setting up libm10:amd64 (1:8.4-3) ...
Setting up libtstables12:amd64 (1.8.9-2) ...
Setting up libelf1:amd64 (8.180-2.1) ...
Setting up libpam-cap:amd64 (1:2.66-4) ...
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or without a controlling terminal.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
Setting up libbpf1:amd64 (1:1.1.0-1) ...
Setting up libtirpc3:amd64 (1.3.3+ds-1) ...
Setting up iproute2 (6.1.0-3) ...
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or without a controlling terminal.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
Setting up nginx (1.22.1-9+deb12u1) ...
invoke-rc.d: could not determine current runlevel
invoke-rc.d: policy-rc.d denied execution of start.
Processing triggers for libc-bin (2.36-9+deb12u9) ...
Finished: SUCCESS
```