Create a list of differences: on-premise infra vs cloud.

**Based on Ownership**

Organization owns and operates all hardware and software.

Third-party cloud provider owns and manages the infrastructure.

**Based on Management**

Organization is responsible for all setup, maintenance, upgrades, and security.

Cloud provider is responsible for infrastructure management, maintenance, and often security (depending on the service model).

**Based on Cost Structure**

High upfront capital expenditure for hardware, software licenses, and physical space. Ongoing operational expenses for power, cooling, and IT staff.

Primarily operational expenditure with a pay-as-you-go or subscription model. No large upfront investments.

**Based on Scalability**

Limited and often slow to scale. Requires purchasing and installing new hardware, which can be time-consuming and expensive.

Highly elastic and scalable. Resources can be provisioned or de-provisioned in minutes, adjusting quickly to demand.

**Based on Control**

Full control over hardware, software, and data. High customization possibilities.

Less direct control over the underlying infrastructure, but still control over applications and data. Customization is limited by provider offerings.

**Based on Accessibility**

Primarily designed for on-site access, with remote access often requiring additional configuration and potentially less convenience.

Accessible from anywhere with an internet connection, promoting remote work and global collaboration.

**Based on Security**

Organization is fully responsible for implementing and maintaining all security measures. Can be highly tailored but requires significant in-house expertise.

Cloud providers typically offer robust security measures and dedicated security teams. Shared responsibility model, where the provider secures the "cloud" and the user secures what's "in the cloud."

**Based on Maintenance**

Requires dedicated IT staff for continuous monitoring, patching, hardware repairs, and troubleshooting.

Managed by the cloud provider, reducing the burden on internal IT teams, who can focus on strategic initiatives.

**Based on Disaster Recovery**

Requires significant investment and planning for redundant systems and off-site backups.

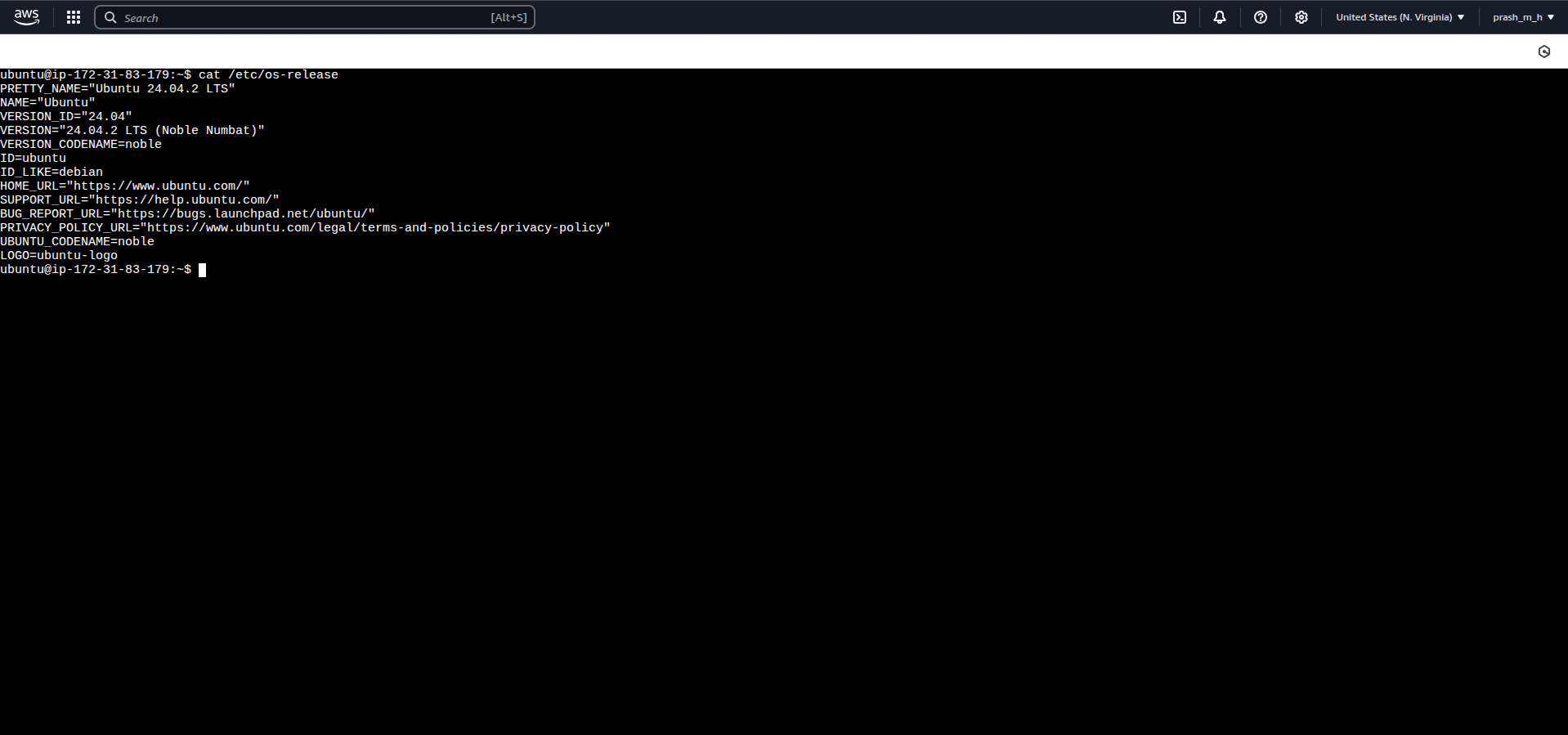
Often built-in disaster recovery and backup solutions provided by the cloud vendor, leveraging distributed data centers.

**Based on Performance**

Can offer low-latency connections and superior performance for specific workloads due to proximity and dedicated resources.

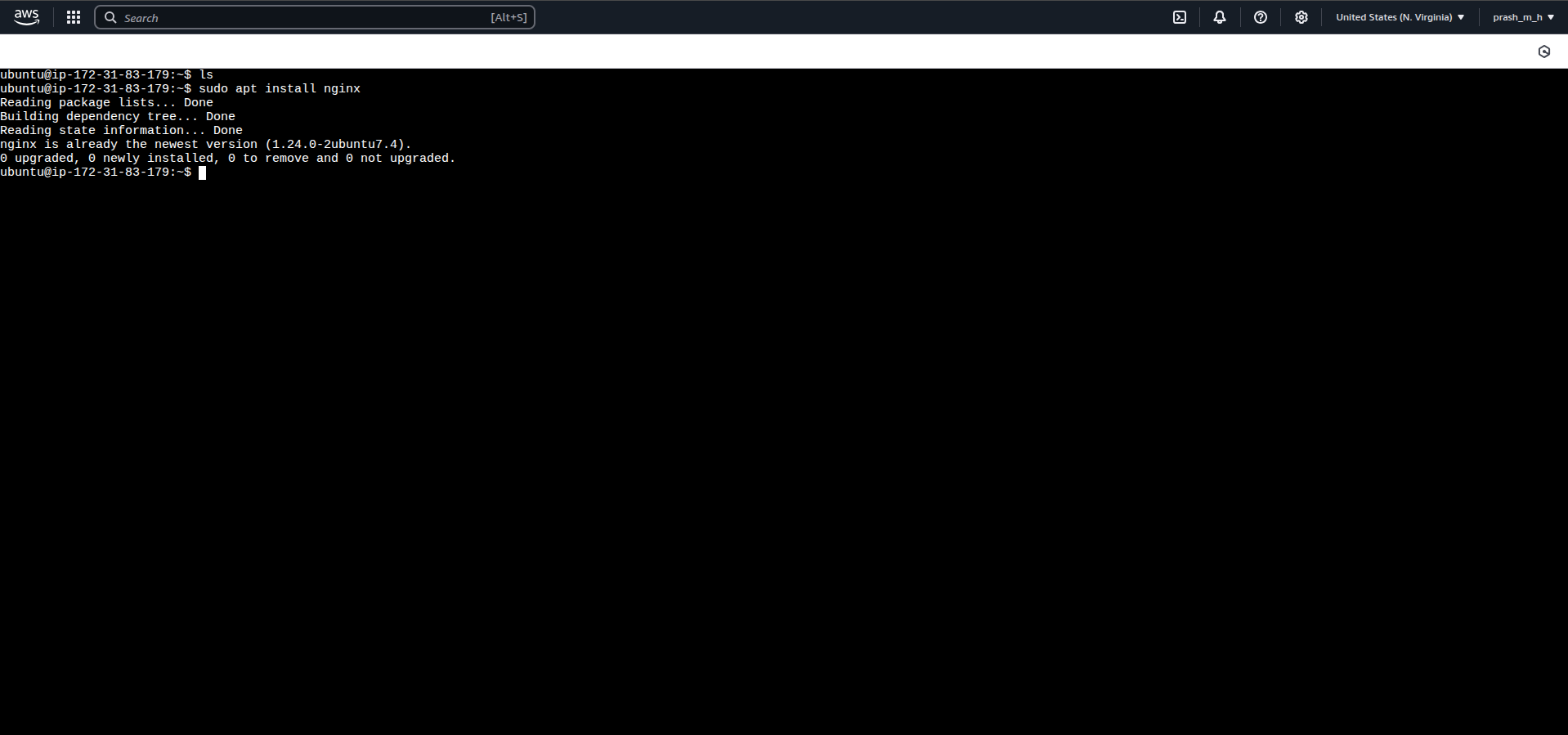
Performance depends on internet connectivity and network architecture. Can be excellent, but latency can be a factor for highly sensitive applications.

EC2 instance on AWS

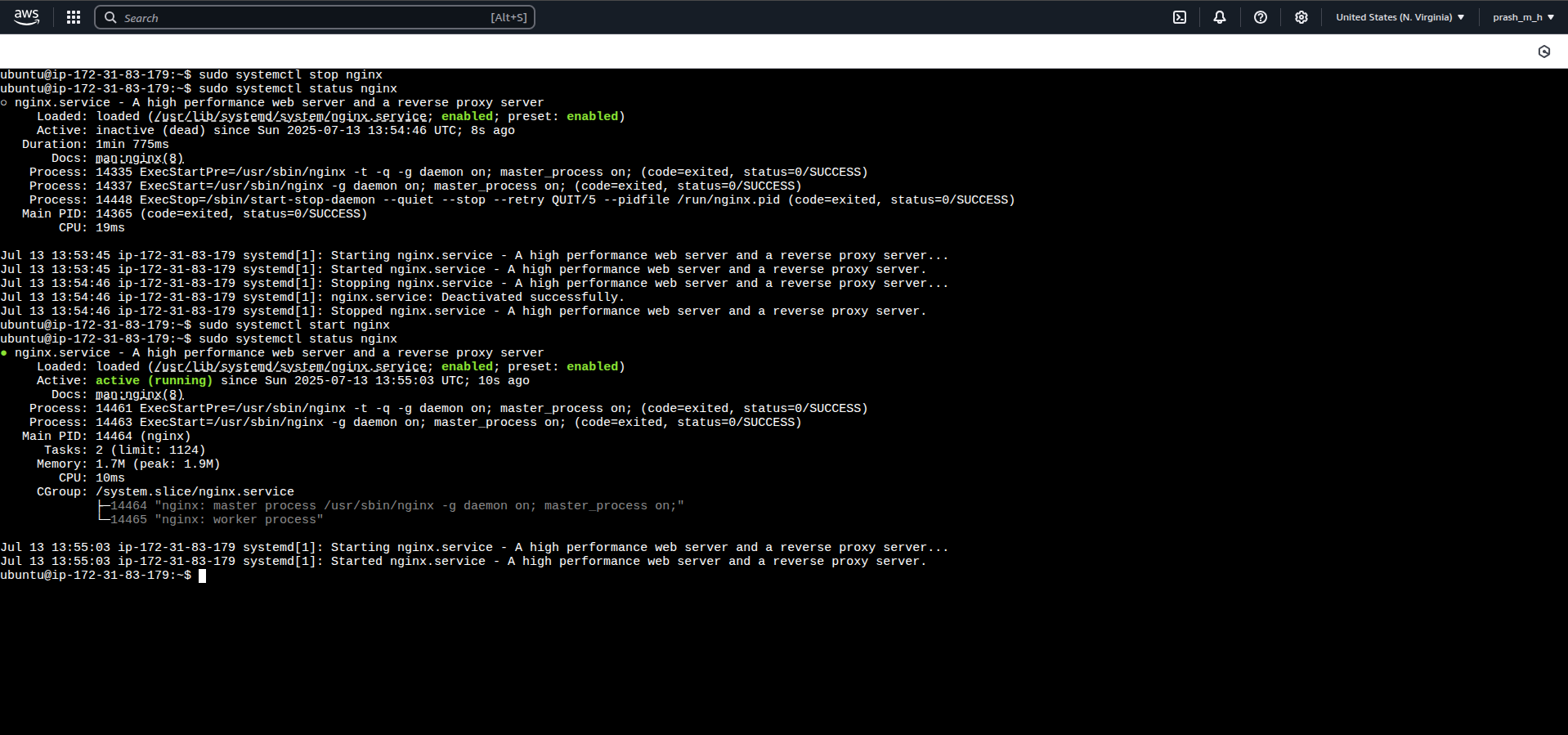


Screenshots or a markdown document explaining service states using nginx

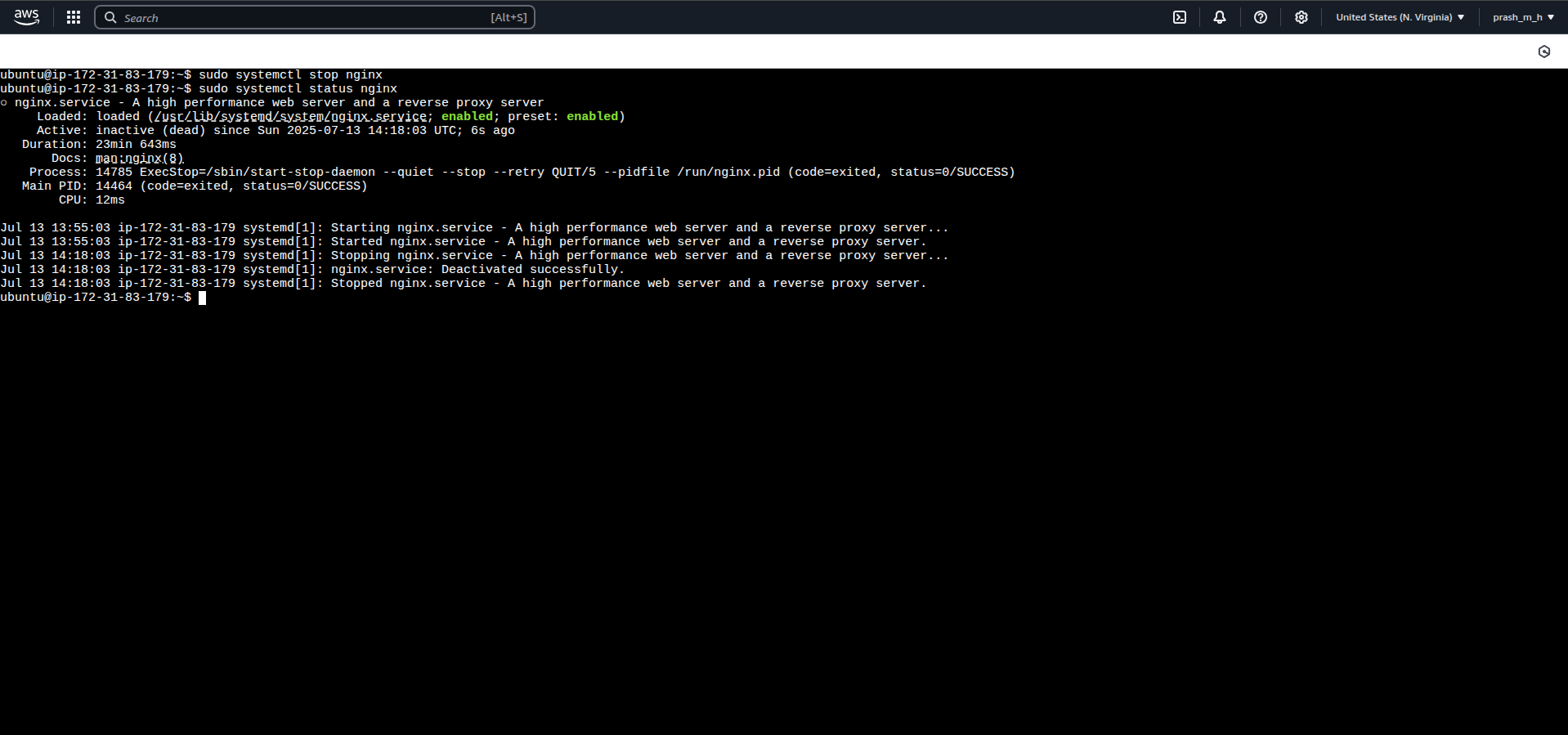
> sudo apt install nginx (to install nginx)



-> sudo systemctl status nginx (to check the status of nginx)

status is active and running

-> sudo systemctl stop (to stop nginx and status will be inactive)

 status is inactive (dead)

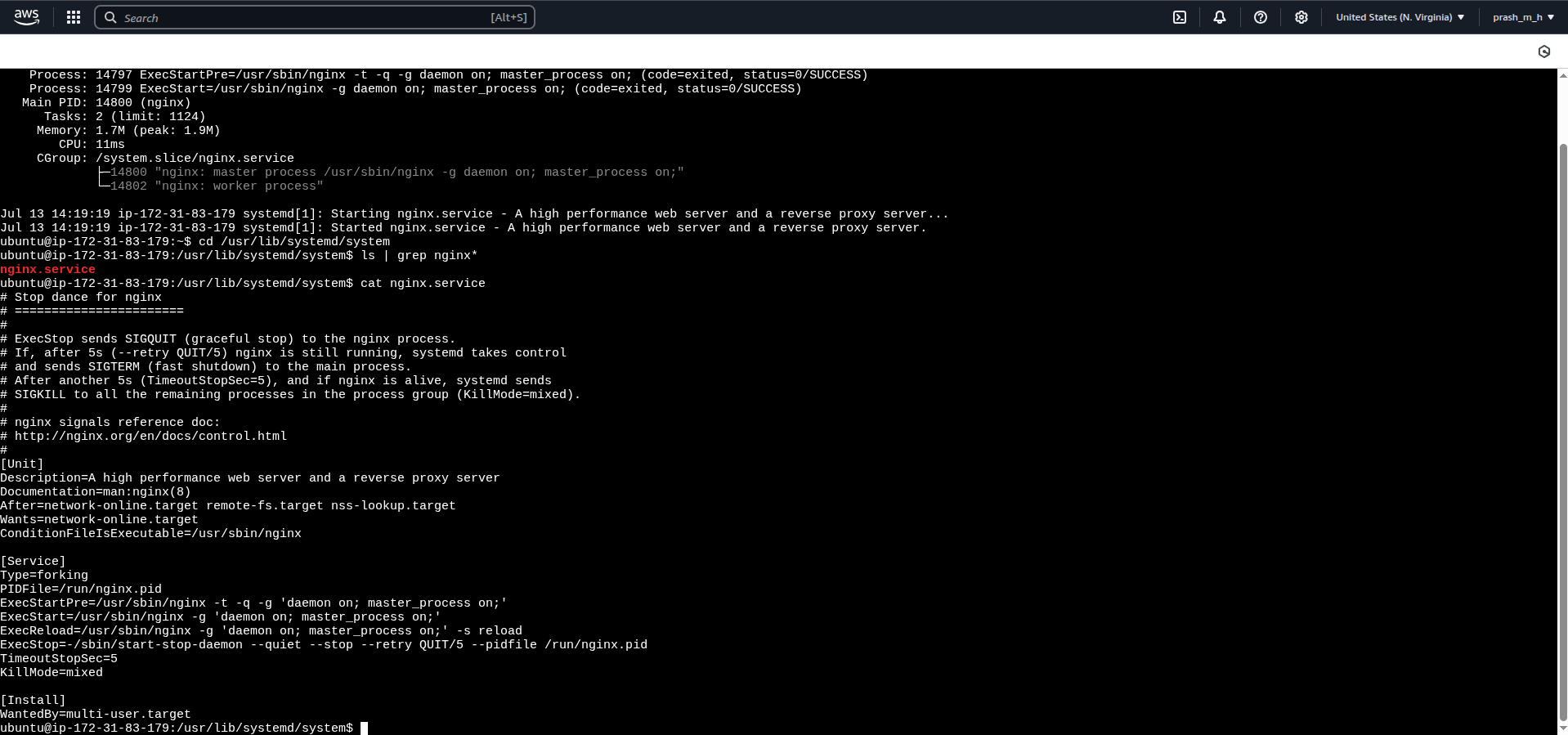
#to enable nginx as a system service

-> sudo systemctl enable nginx

# a service with name nginx.service will be created in *usr/lib/systemd/system/nginx.service*

*# n*ow nginx.service will start as soon as the system boots up

*nginx.service file*



**commands to create new user (prashant) and adding the user to sudo group**

**->** sudo adduser <username>

-> sudo adduser -aG sudo <username> (to add the user to the sudo group)

**command to check if the user is added to sudo group**

-> groups <username>

-> sudo gpasswd -d <username> <groupname>

**Create a file hello.txt, set different permissions (read/write/execute) for user/group/others, and document the changes.**

-> touch hello.txt

-> chmod 777 hello.txt (to set r/w/x permissions to user/group/others)

**to change the ownership of a file**

-> sudo chown <username> hello.txt

**to change the group ownership of a file**

-> sudo chown :<groupname> hello.txt