Deploying E commerce platform using Magento with Redis and varnish

**Magento:**

Magento is an open-source e Commerce platform written in PHP.

It provides a flexible shopping cart system, content management, and powerful SEO capabilities.

**Redis:**

Caching: Speeds up page load times by storing frequently accessed data.

Session storage: Manages user sessions more efficiently than file-based sessions.

**Varnish:**

Varnish is a reverse proxy and caching HTTP accelerator used to speed up Magento.

It stores pre-generated HTML pages and serves them quickly instead of generating them dynamically.

**Composer:**

Composer is a PHP dependency manager that helps install and update Magento and its extensions.

Magento 2 requires Composer to manage its modules and third-party libraries.

PHP8.1 setup:

sudo apt update && sudo apt install -y lsb-release ca-certificates apt-transport-https software-properties-common wget

sudo wget -O /etc/apt/trusted.gpg.d/php.gpg https://packages.sury.org/php/apt.gpg

echo "deb https://packages.sury.org/php/ $(lsb\_release -sc) main" | sudo tee /etc/apt/sources.list.d/php.list

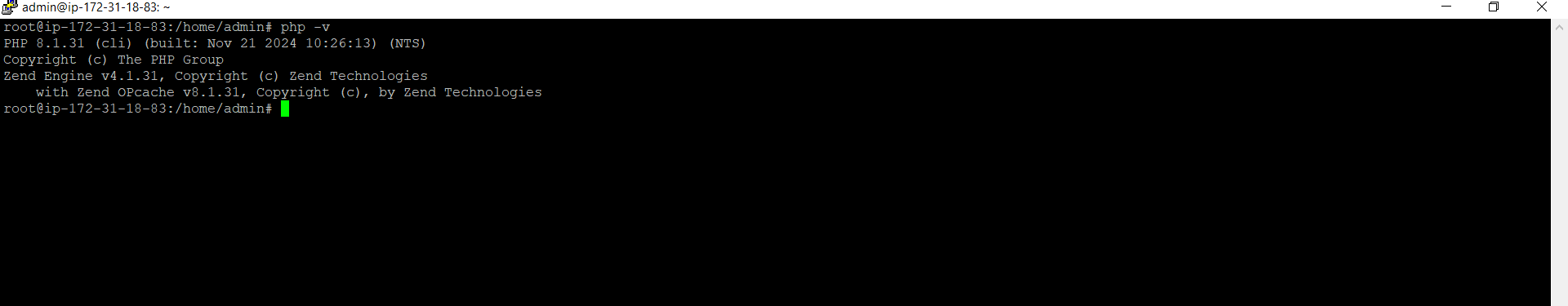
sudo apt update

sudo apt install -y php8.1 php8.1-fpm php8.1-cli php8.1-mysql php8.1-curl \

php8.1-gd php8.1-mbstring php8.1-xml php8.1-bcmath php8.1-soap \

php8.1-intl php8.1-zip php8.1-tokenizer php8.1-opcache php8.1-xsl \

php8.1-imagick php8.1-redis php8.1-common



Mysql 8 setup:

sudo apt remove --purge mysql\* mariadb\*

sudo apt autoremove

sudo apt autoclean

wget https://dev.mysql.com/get/mysql-apt-config\_0.8.29-1\_all.deb

sudo dpkg -i mysql-apt-config\_0.8.29-1\_all.deb

sudo apt update

sudo apt install -y mysql-server

sudo systemctl start mysql

sudo systemctl enable mysql

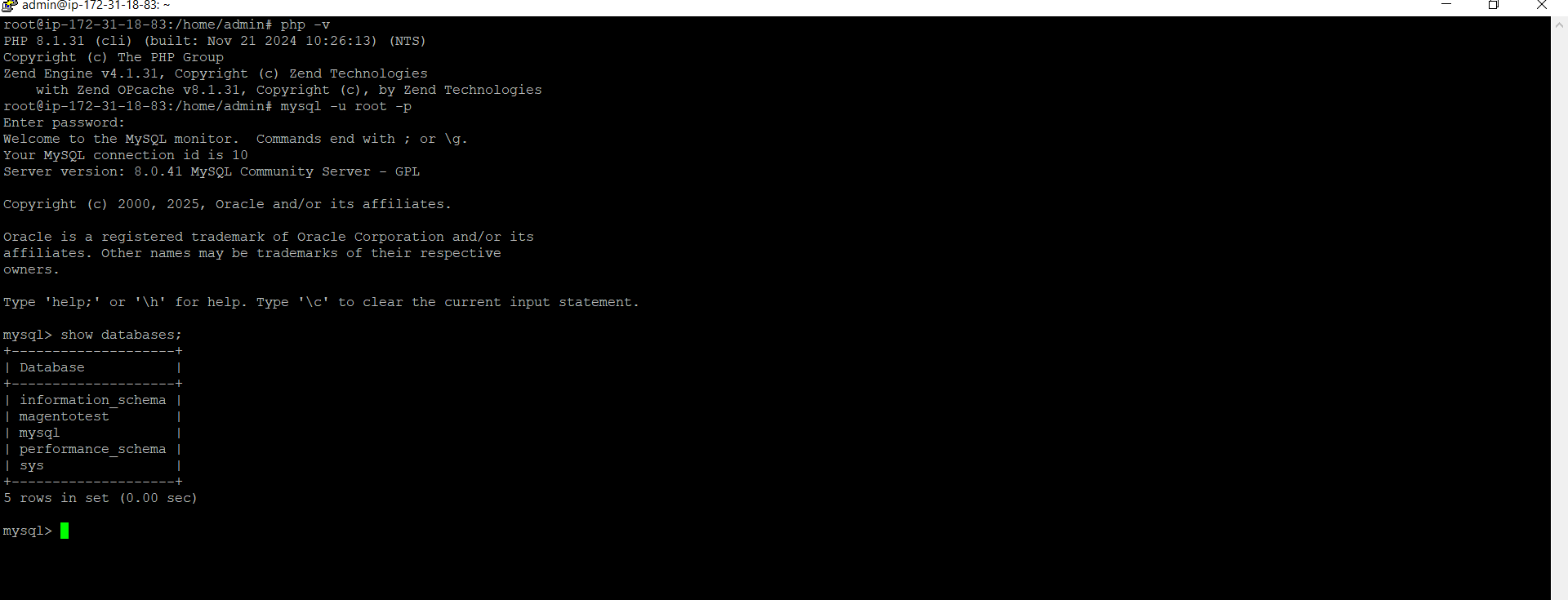
mysql -u root -p

CREATE DATABASE magentotest;

GRANT ALL PRIVILEGES ON magentotest.\* TO 'root'@'localhost';

FLUSH PRIVILEGES;

EXIT;



Nginx :

sudo apt install -y nginx

sudo systemctl start nginx

sudo systemctl enable nginx

Elastic search:

wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -

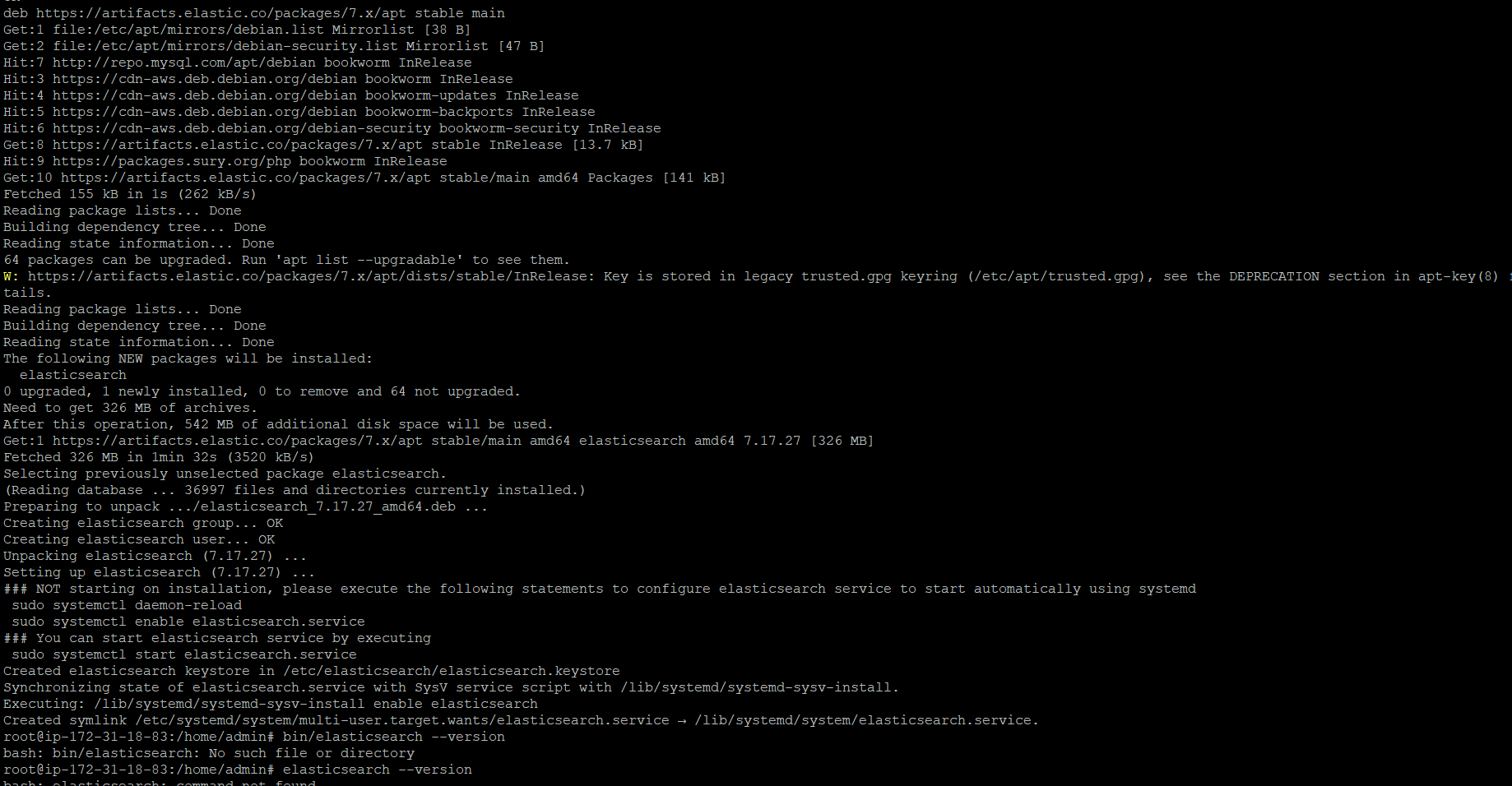
echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-7.x.list

sudo apt update

sudo apt install -y elasticsearch

sudo systemctl start elasticsearch

sudo systemctl enable elasticsearch



Composer:

sudo apt install -y composer

sudo mkdir -p /var/www/html/test.mgt.com ---change the dir name

useradd test-ssh

groupadd clp

usermod -aG clp test-ssh

passwd test-ssh

sudo chown -R test-ssh:clp /var/www/html/test.mgt.com

Create magento via composer:

cd /var/www/html/test.mgt.com

composer create-project --repository-url=https://repo.magento.com/ magento/project-community-edition .

sudo find var generated vendor pub/static pub/media app/etc -type f -exec chmod g+w {} +

sudo find var generated vendor pub/static pub/media app/etc -type d -exec chmod g+ws {} +

php bin/magento setup:install --base-url=https://47.128.241.69 \

--db-host=localhost --db-name=magentotest --db-user=root --db-password=mgt123 \

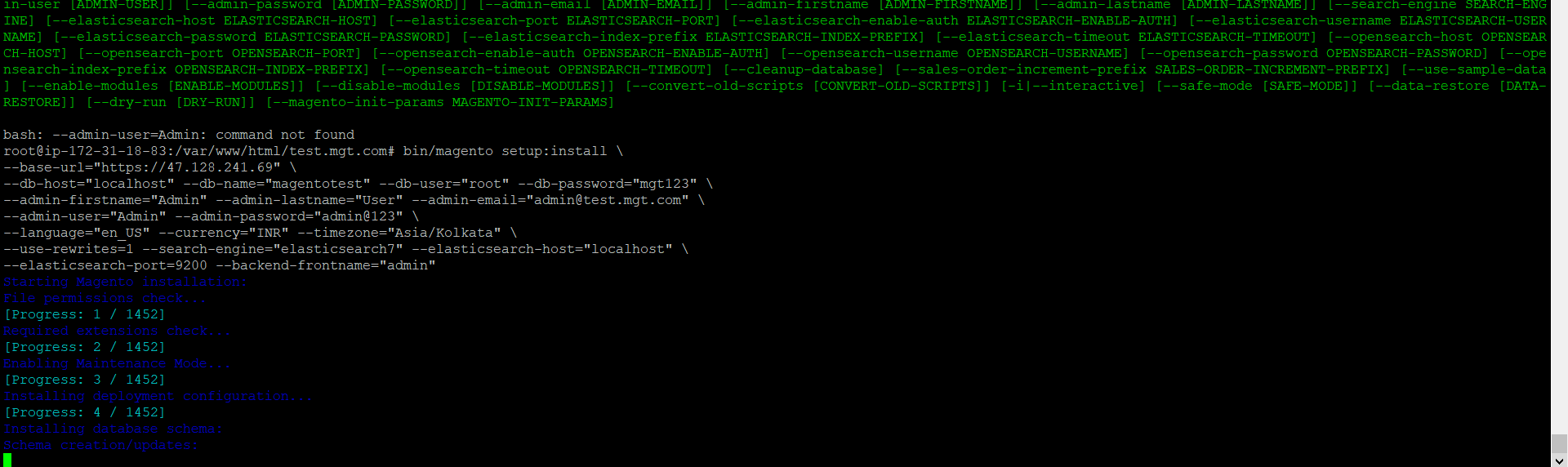
--admin-firstname=Admin --admin-lastname=User --admin-email=admin@test.mgt.com \

--admin-user=admin --admin-password=admin@123 \

--language=en\_US --currency=INR --timezone=Asia/Kolkata \

--use-rewrites=1 --search-engine=elasticsearch7 --elasticsearch-host=localhost \

--elasticsearch-port=9200 --backend-frontname=admin



Add sample data:

bin/magento sampledata:deploy

bin/magento setup:upgrade

Redis install:

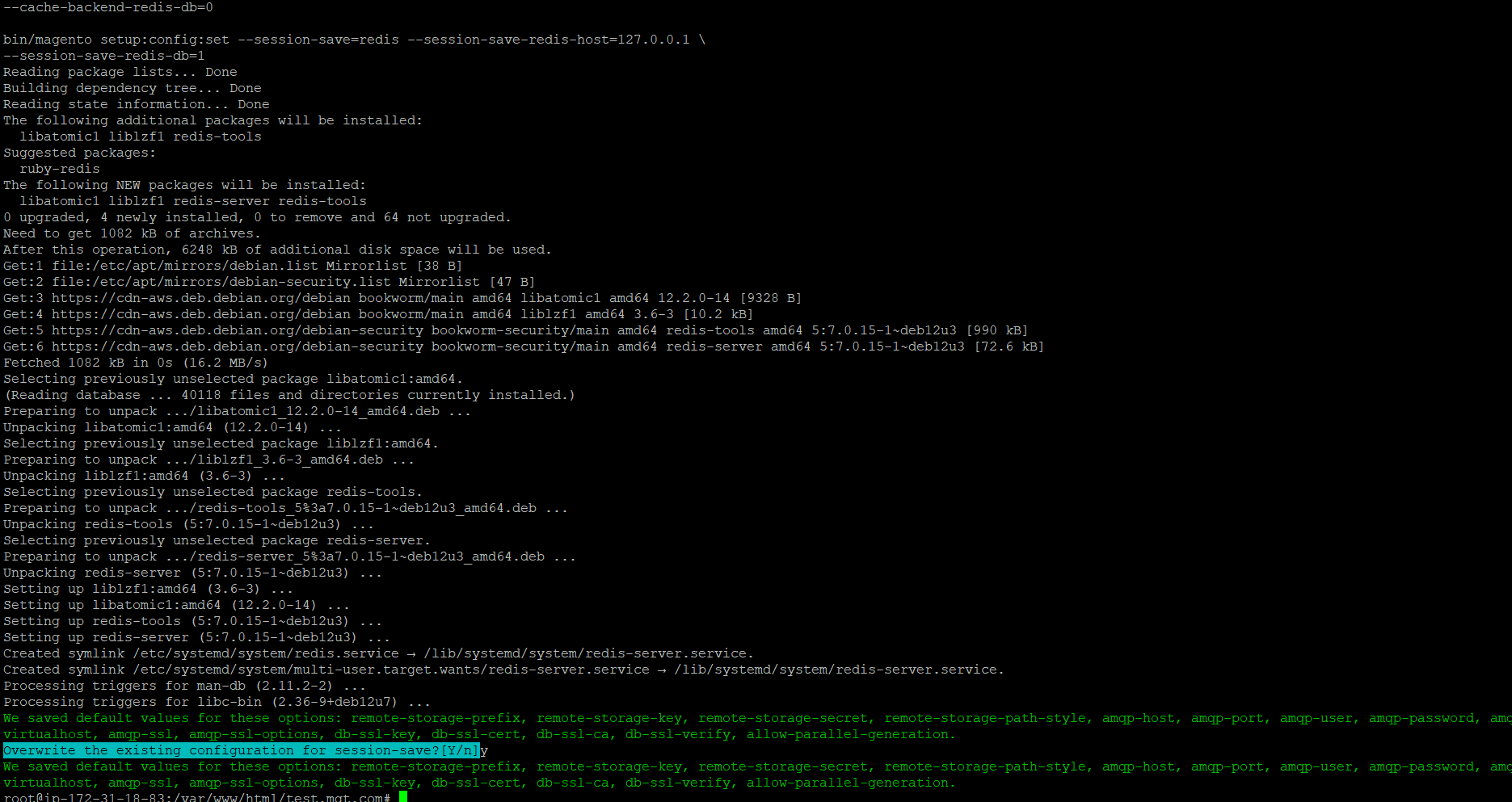
sudo apt install -y redis-server

bin/magento setup:config:set --cache-backend=redis --cache-backend-redis-server=127.0.0.1 \

--cache-backend-redis-db=0

bin/magento setup:config:set --session-save=redis --session-save-redis-host=127.0.0.1 \

--session-save-redis-db=1



Creating php fpm pool:

Creating php fpm pool:

sudo nano /etc/php/8.1/fpm/pool.d/test-ssh.conf

[test-ssh]

user = test-ssh

group = clp

listen = /run/php/php8.1-fpm-test-ssh.sock

listen.owner = test-ssh

listen.group = clp

listen.mode = 0660

pm = dynamic

pm.max\_children = 75

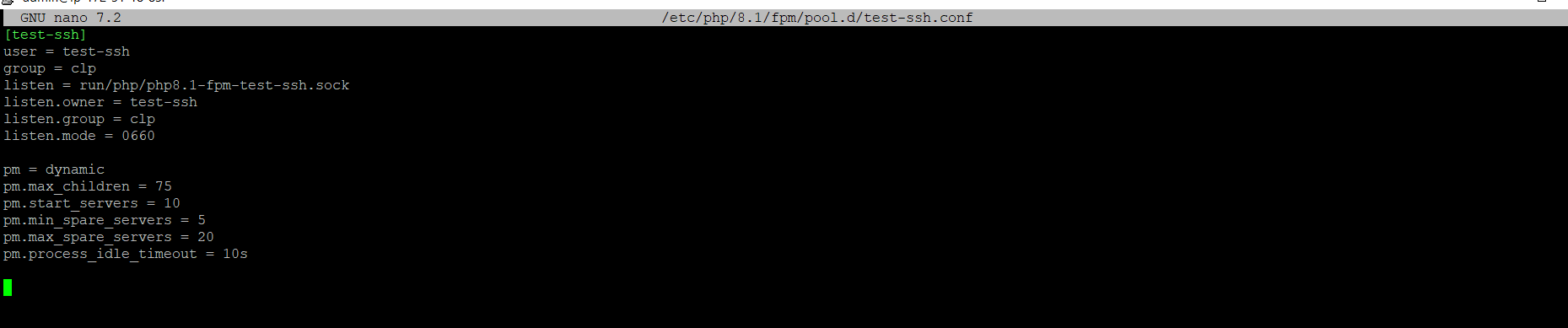
pm.start\_servers = 10

pm.min\_spare\_servers = 5

pm.max\_spare\_servers = 20

pm.process\_idle\_timeout = 10s

sudo chown -R test-ssh:clp /etc/php/8.1/fpm/pool.d/test-ssh.conf



changing nginx user:

sudo nano /etc/nginx/nginx.conf

user test-ssh;

worker\_processes auto;

events {

worker\_connections 1024;

}

http{

fastcgi\_buffers 16 16k;

fastcgi\_buffer\_size 32k;

sudo chown -R test-ssh:test-ssh /var/lib/nginx

sudo chown -R test-ssh:test-ssh /var/log/nginx

sudo chown -R test-ssh:test-ssh /var/cache/nginx

Creating nginx vhost:

sudo nano /etc/nginx/sites-available/test.mgt.com

upstream fastcgi\_backend {

server unix:/run/php/php8.1-fpm-test-ssh.sock;

}

server {

listen 80;

listen [::]:80;

server\_name 3.111.149.148; # Use your public IP

set $MAGE\_ROOT /var/www/html/test.mgt.com;

set $MAGE\_MODE developer; # Change to 'production' if needed

root $MAGE\_ROOT/pub; # Magento must serve from /pub

index index.php index.html index.htm;

access\_log /var/log/nginx/magento-access.log;

error\_log /var/log/nginx/magento-error.log error;

location / {

index index.php index.html;

try\_files $uri $uri/ /index.php?$args;

}

location /static/ {

location ~ ^/static/version {

expires max;

log\_not\_found off;

}

}

location /media/ {

try\_files $uri $uri/ /get.php?$args;

}

location ~ \.php$ {

fastcgi\_split\_path\_info ^(.+\.php)(/.+)$;

fastcgi\_pass unix:/run/php/php8.1-fpm-test-ssh.sock;

fastcgi\_index index.php;

include fastcgi.conf;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

fastcgi\_buffers 16 16k;

fastcgi\_buffer\_size 32k;

}

location ~\* \.(jpg|jpeg|gif|png|css|js|ico|xml|svg|ttf|woff|woff2|eot|otf)$ {

expires max;

log\_not\_found off;

}

}

server {

listen 443 ssl;

server\_name public-ip;

ssl\_certificate /etc/ssl/certs/selfsigned.crt;

ssl\_certificate\_key /etc/ssl/private/selfsigned.key;

set $MAGE\_ROOT /var/www/html/test.mgt.com;

set $MAGE\_MODE developer; # Change to 'production' if needed

root $MAGE\_ROOT/pub; # Magento must serve from /pub

index index.php index.html index.htm;

access\_log /var/log/nginx/magento-access.log;

error\_log /var/log/nginx/magento-error.log error;

location / {

index index.php index.html;

try\_files $uri $uri/ /index.php?$args;

}

location /static/ {

location ~ ^/static/version {

expires max;

log\_not\_found off;

}

}

location /media/ {

try\_files $uri $uri/ /get.php?$args;

}

location ~ \.php$ {

fastcgi\_split\_path\_info ^(.+\.php)(/.+)$;

fastcgi\_pass unix:/run/php/php8.1-fpm-test-ssh.sock;

fastcgi\_index index.php;

include fastcgi.conf;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

fastcgi\_buffers 16 16k;

fastcgi\_buffer\_size 32k;

}

location ~\* \.(jpg|jpeg|gif|png|css|js|ico|xml|svg|ttf|woff|woff2|eot|otf)$ {

expires max;

log\_not\_found off;

}

}



Creating certificate for https:

ls -l /etc/ssl/certs/selfsigned.crt /etc/ssl/private/selfsigned.key

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \

-keyout /etc/ssl/private/selfsigned.key \

-out /etc/ssl/certs/selfsigned.crt

sudo ln -s /etc/nginx/sites-available/test.mgt.com /etc/nginx/sites-enabled/

sudo systemctl restart nginx

Phpmyadmin:

sudo apt install -y phpmyadmin

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

server {

listen 8080;

server\_name 3.111.149.148; # Public ip

root /var/www/html/phpmyadmin;

index index.php index.html index.htm;

location /phpmyadmin {

alias /usr/share/phpmyadmin;

index index.php index.html;

}

location ~ \.php$ {

fastcgi\_split\_path\_info ^(.+\.php)(/.+)$;

fastcgi\_pass unix:/run/php/php8.1-fpm-test-ssh.sock; # Update this if using a different PHP version

fastcgi\_index index.php;

include fastcgi.conf;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

}

location ~\* \.(js|css|png|jpg|jpeg|gif|ico|svg|woff|woff2|ttf|eot|otf)$ {

expires max;

log\_not\_found off;

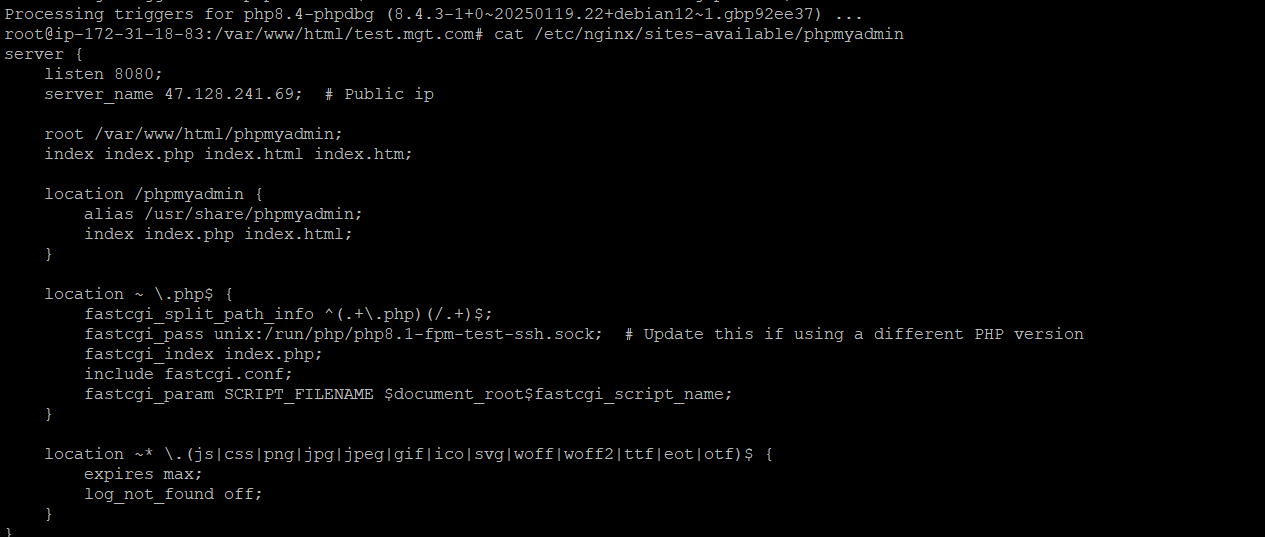
}

}

sudo ln -s /etc/nginx/sites-available/phpmyadmin /etc/nginx/sites-available/phpmyadmin

ln -s /usr/share/phpmyadmin /var/www/html/

chown -R test-ssh:clp /var/www/html/phpmyadmin



Set php-8.1:

sudo update-alternatives --set php /usr/bin/php8.1

Install varnish:

sudo apt install -y varnish

bin/magento setup:config:set --http-cache-hosts=127.0.0.1:6081

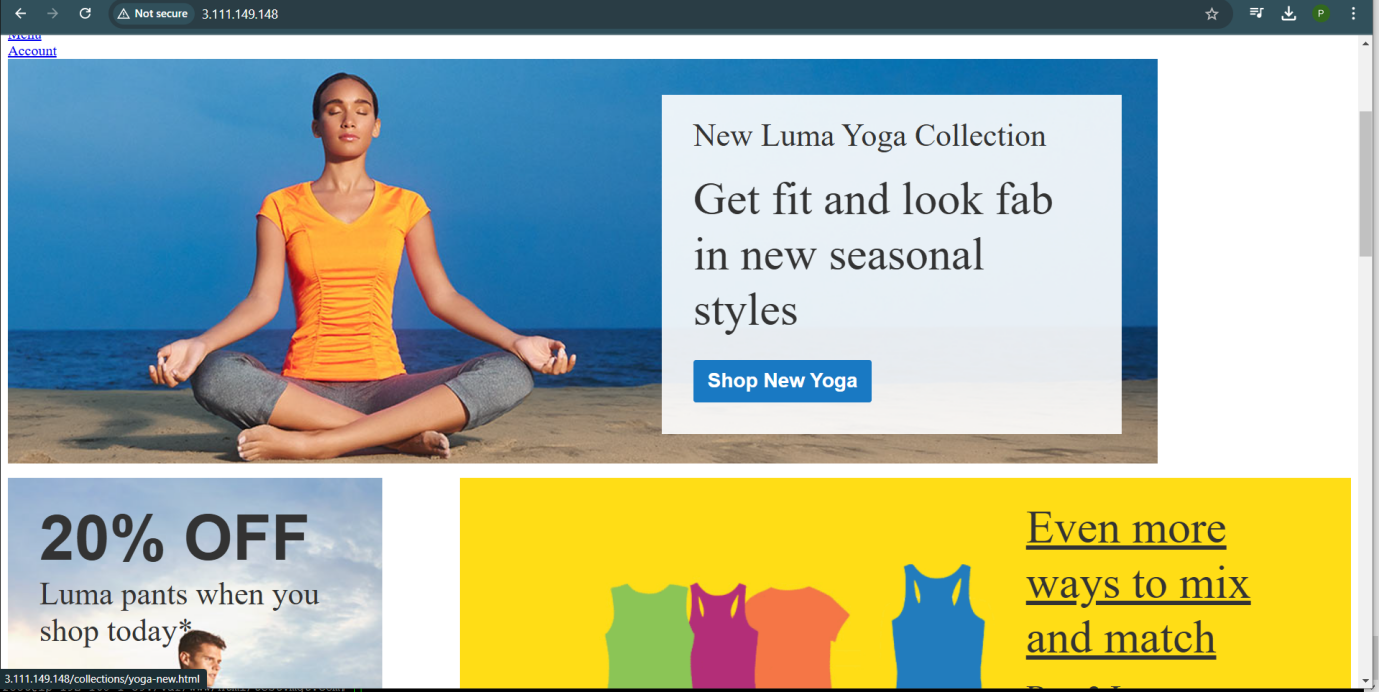
sudo bin/magento config:set --scope=default --scope-code=0 system/full\_page\_cache/caching\_application 2

sudo systemctl restart varnish

sudo systemctl restart nginx

Output:

Magneto ecommerce website:



Phpmyadmin:

