Internship Task Report: Deploying WordPress Using Monolithic and Microservices Architectures

Introduction

This report details the steps taken to deploy WordPress using two different architectures: monolithic and microservices. The task involves configuring EC2 instances on AWS, deploying WordPress and MySQL, and setting up a welcome page as the homepage.

Architectures Overview

Monolithic Architecture

In a monolithic architecture, all components of the application run on a single server. For this task, we deployed both WordPress and MySQL on a single EC2 instance.

Microservices Architecture

In a microservices architecture, different components of the application are isolated into separate services. For this task, we deployed WordPress on one EC2 instance and MySQL on another.

EC2 Instances Configuration

Instance Details

- Instance Type: t3-micro
- AMI: Ubuntu 20.04 LTS (ami-0a91cd140a1fc148a)

Security Groups

- Allow HTTP (port 80) for WordPress.
- Allow MySQL (port 3306) only from the WordPress instance for enhanced security.

- Allow SSH (port 22) for administration purposes.

Deployment Steps

Monolithic Architecture

1. **Launch EC2 Instance**
    - Create a t3-micro EC2 instance with the specified AMI (Ubuntu 20.04 LTS).
    - Assign a security group allowing HTTP, MySQL, and SSH access.

2. **Install and Configure Apache**
   ->sudo apt update
   ->sudo apt install apache2
   ->sudo systemctl start apache2
   ->sudo systemctl enable apache2

3. **Install and Configure MySQL**
   ->sudo apt install mysql-server
   ->sudo mysql_secure_installation
   ->sudo mysql
   ->CREATE DATABASE wordpress;
   ->CREATE USER 'wp_user'@'localhost' IDENTIFIED BY 'password';
   ->GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'localhost';

->FLUSH PRIVILEGES;

4. Install PHP and WordPress

->sudo apt install php libapache2-mod-php php-mysql
->cd /tmp
->wget https://wordpress.org/latest.tar.gz
->tar -xzvf latest.tar.gz
->sudo mv wordpress /var/www/html/
->sudo chown -R www-data:www-data /var/www/html/wordpress
->sudo chmod -R 755 /var/www/html/wordpress
->sudo systemctl restart apache2

5. Configure WordPress

- Navigate to http://your_instance_ip/wordpress in a web browser.
- Complete the WordPress setup wizard, connecting it to the MySQL database created earlier.
- Create a welcome page and set it as the homepage.

Microservices Architecture

1. Launch EC2 Instances

- Create two t3-micro EC2 instances with the specified AMI (Ubuntu 20.04 LTS).
- One instance for WordPress (WP-Instance) and another for MySQL (DB-Instance).

- Assign appropriate security groups.

2. Configure MySQL on DB-Instance

->sudo apt update

->sudo apt install mysql-server

->sudo mysql_secure_installation

-> sudo mysql

->CREATE DATABASE wordpress;

->CREATE USER 'wp_user'@'%' IDENTIFIED BY 'password';

->GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'%';

-> FLUSH PRIVILEGES;

->sudo ufw allow 3306

3. Install and Configure Apache and WordPress on WP-Instance

->sudo apt update

->sudo apt install apache2 php libapache2-mod-php php-mysql

->cd /tmp

>wget https://wordpress.org/latest.tar.gz

->tar -xzvf latest.tar.gz

->sudo mv wordpress /var/www/html/

->sudo chown -R www-data:www-data /var/www/html/wordpress

->sudo chmod -R 755 /var/www/html/wordpress

->sudo systemctl start apache2

->sudo systemctl enable apache2

4. Configure WordPress to Use Remote MySQL Database

- -Update the WordPress configuration to point to the DB-Instance.

-> // In /var/www/html/wordpress/wp-config.php

->define('DB_NAME', 'wordpress');

->define('DB_USER', 'wp_user');

->define('DB_PASSWORD', 'password');

->define('DB_HOST', 'DB-Instance_IP');

5. Complete WordPress Setup

- Navigate to `http://WP-Instance_IP/wordpress` in a web browser.
- Complete the WordPress setup wizard, connecting it to the remote MySQL database.
- Create a welcome page and set it as the homepage.

Conclusion

The deployment of WordPress using both monolithic and microservices architectures demonstrates different approaches to application architecture. The monolithic setup consolidates all services onto a single instance, simplifying management but potentially limiting scalability. The microservices setup separates the database and application server, enhancing scalability and fault isolation at the cost of increased complexity. Both setups were configured to ensure security and functionality, culminating in the creation of a welcome page on WordPress.

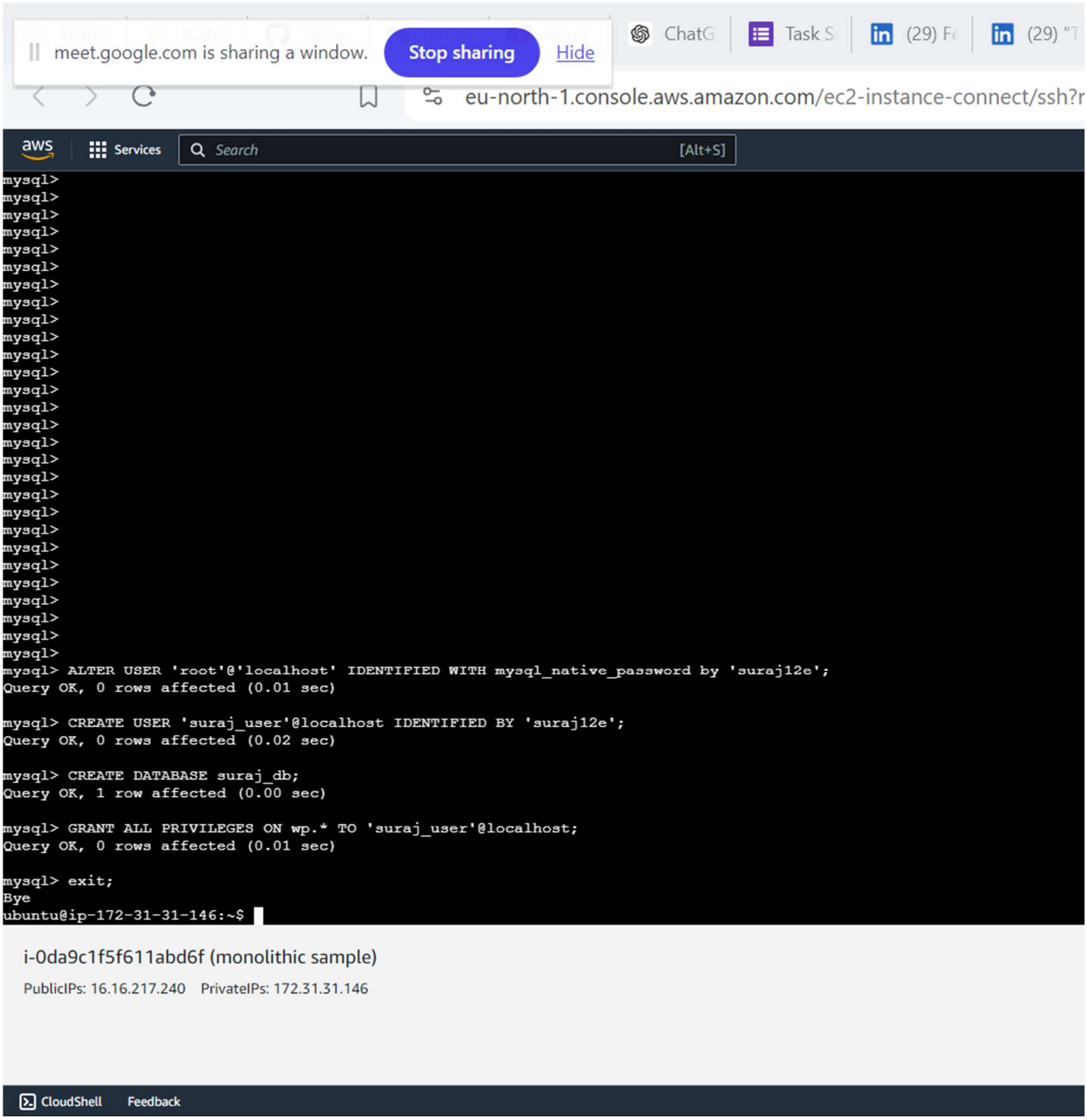
This task provided valuable insights into the strengths and trade-offs of different architectural styles, preparing for more advanced deployment scenarios in real-world environments.

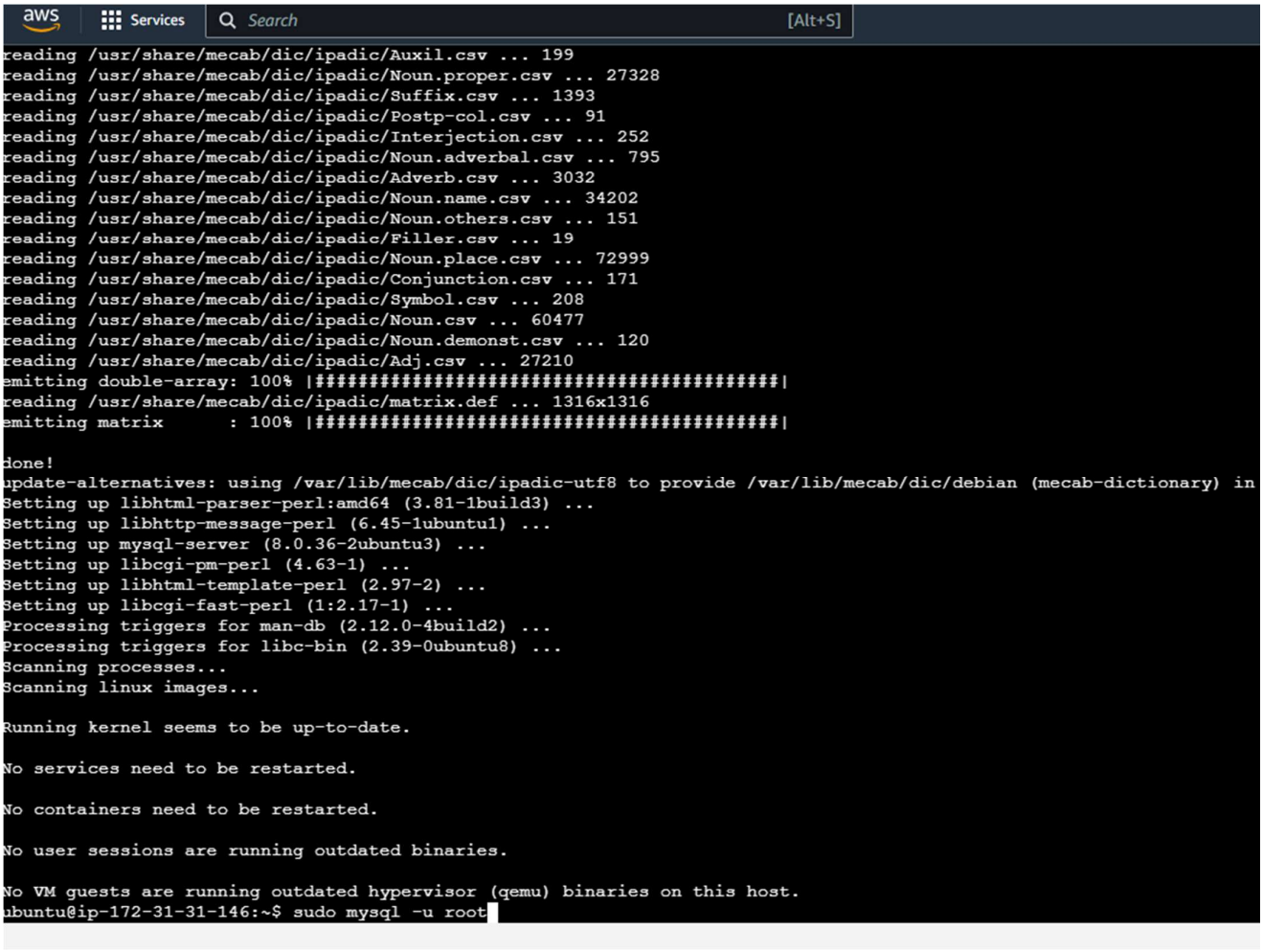
Reference

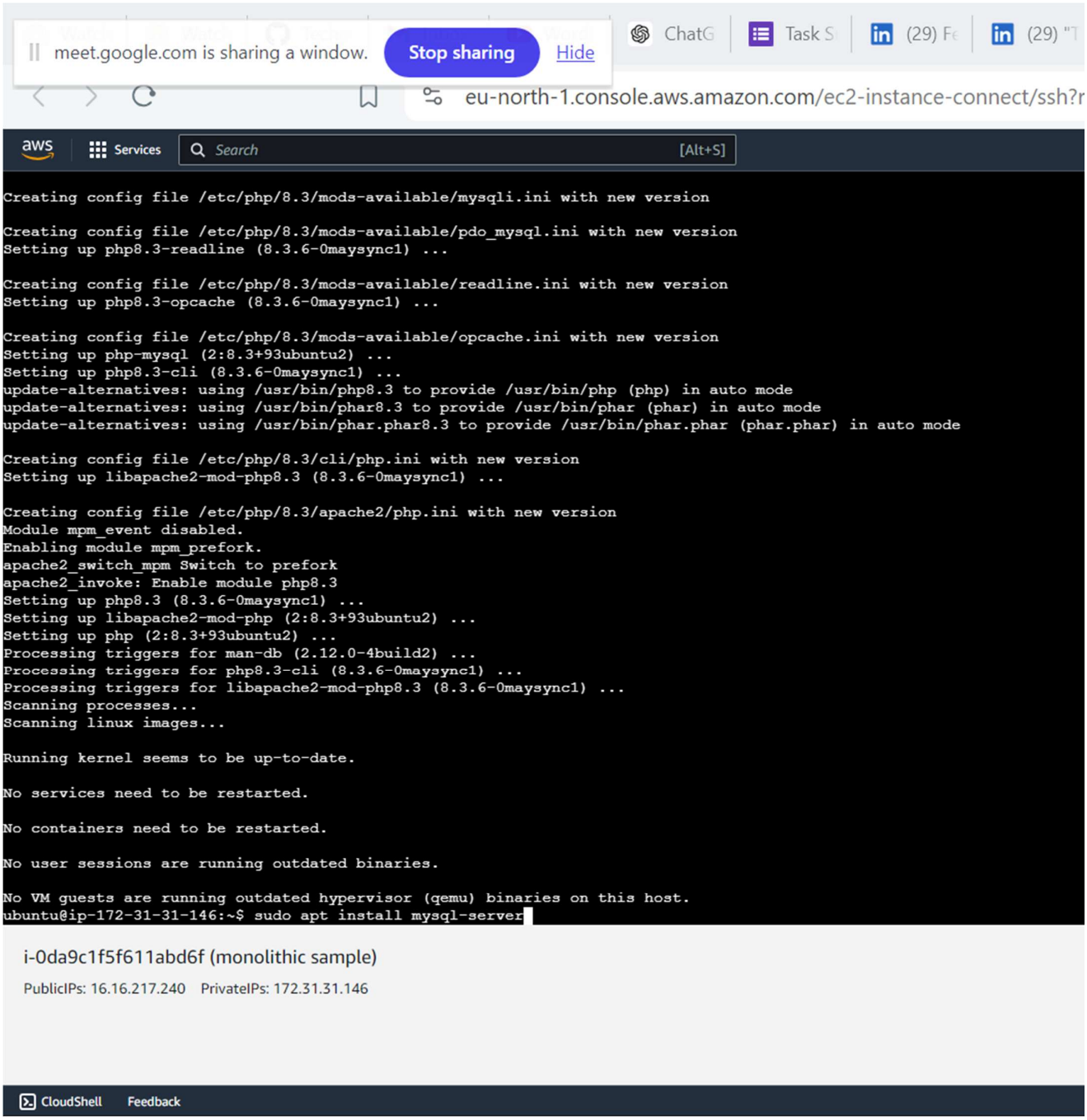
- https://aws.amazon.com/compare/the-difference-between-monolithic-and-microservices-architecture/
- https://docs.aws.amazon.com/codedeploy/latest/userguide/tutorials-wordpress.html

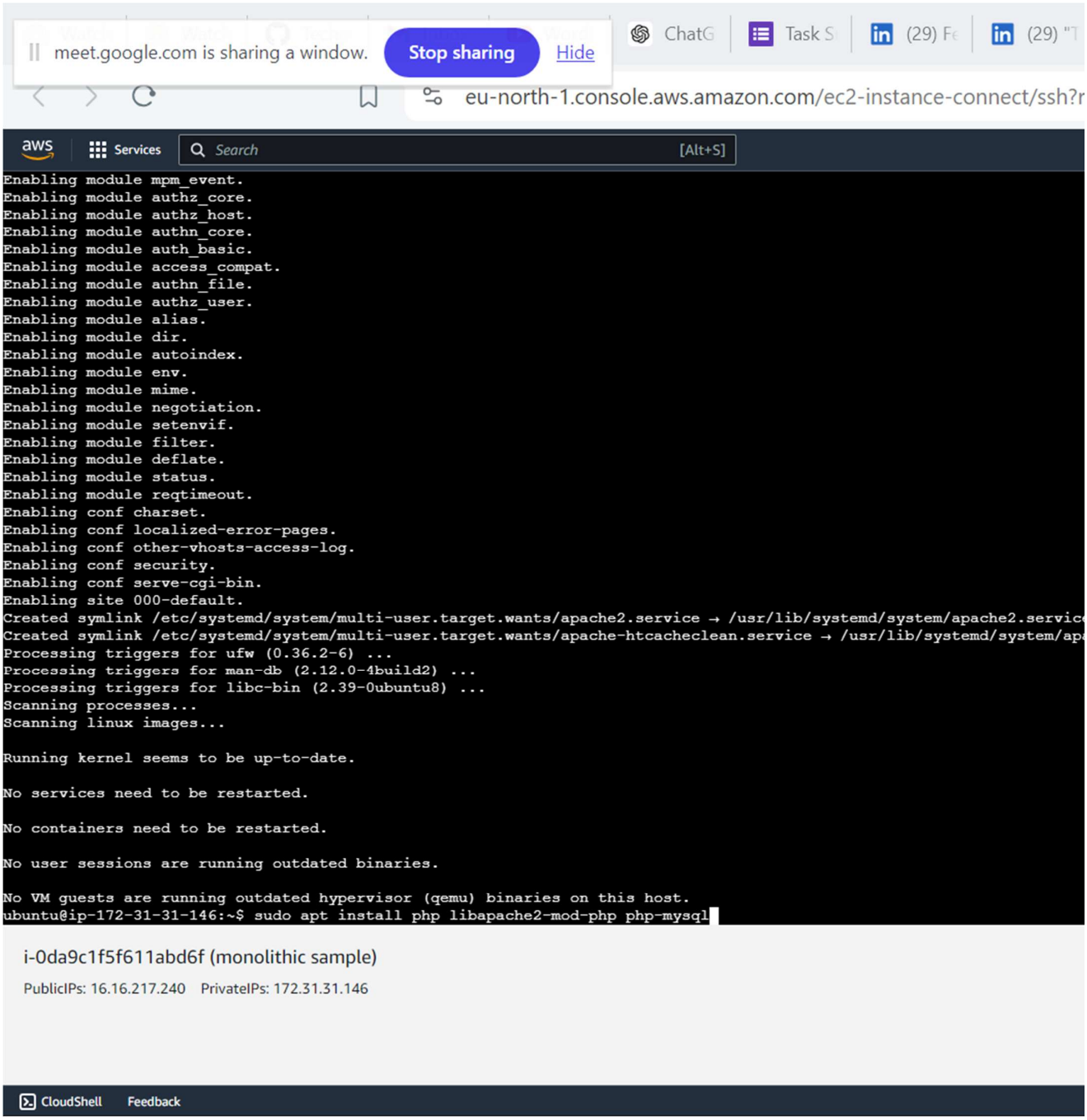
- https://thorntech.com/microservices-vs-monoliths-whats-right-architecture-software/

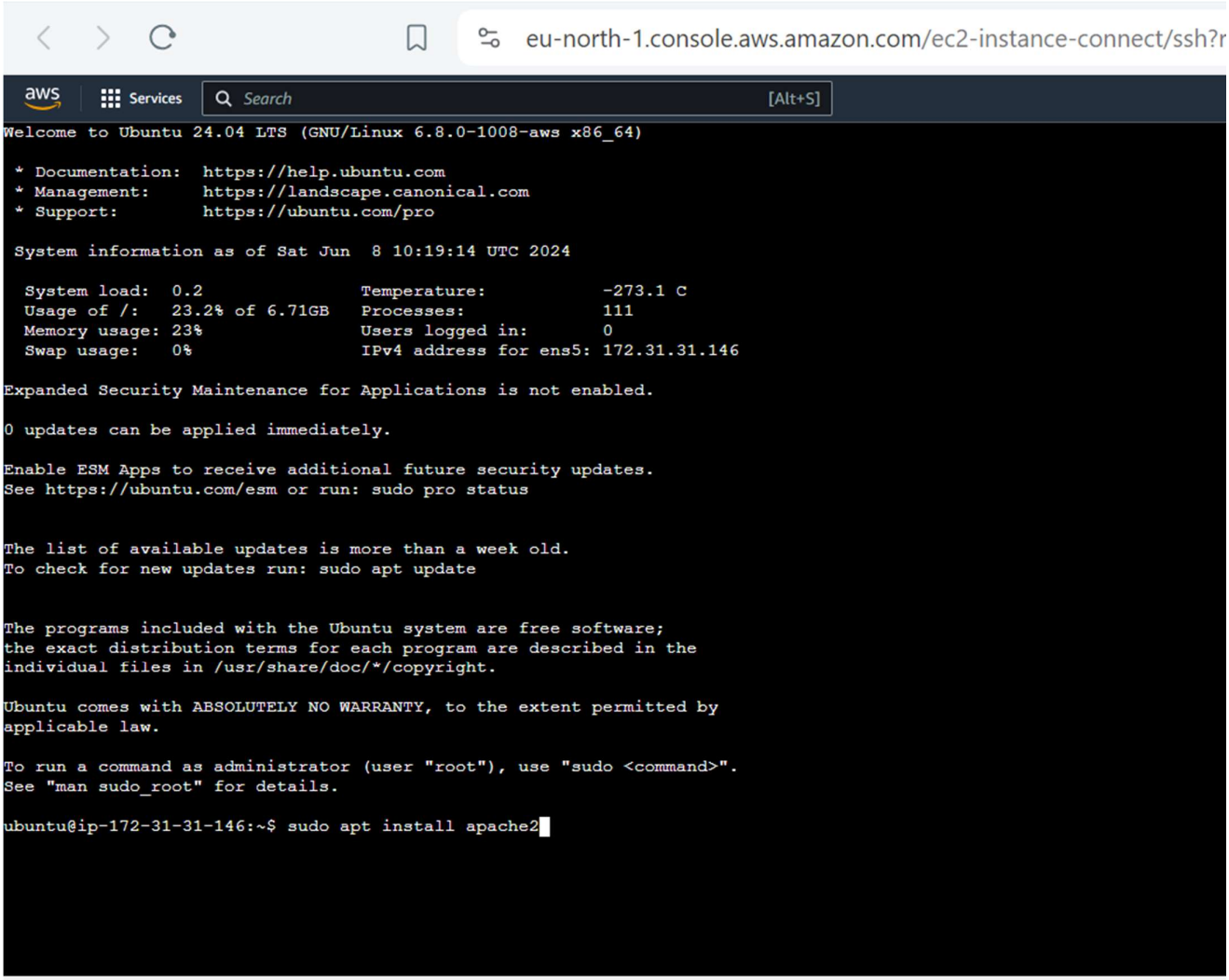
SCREENSHOT

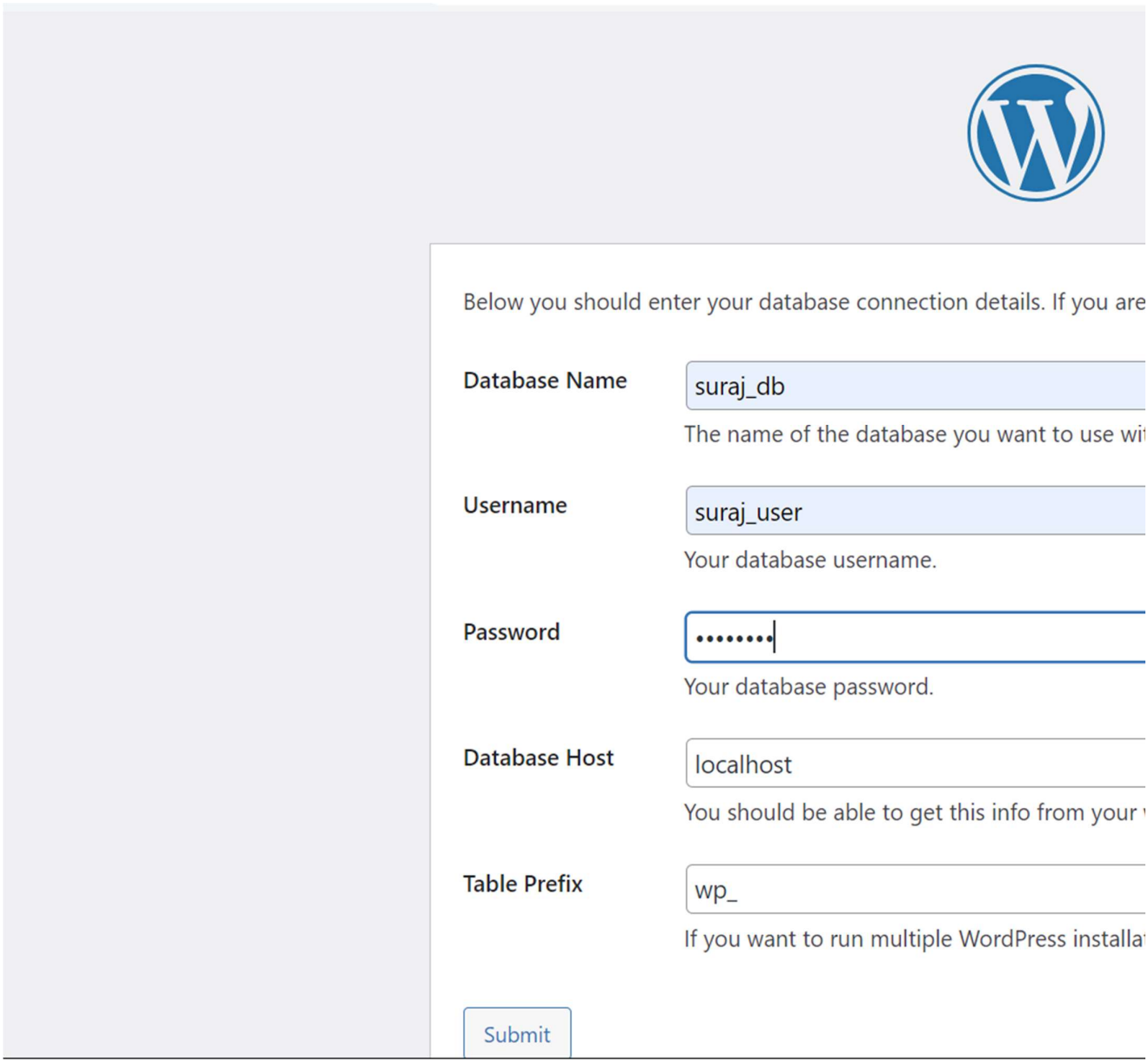

meet.google.com is sharing a window.
Stop sharing
Hide
ChatG
Task S
(29) Fe
(29) "T
eu-north-1.console.aws.amazon.com/ec2-instance-connect/ssh?r
aws
Services
Search
[Alt+S]
mysql>
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password by 'suraj12e';
Query OK, 0 rows affected (0.01 sec)
mysql> CREATE USER 'suraj_user'@localhost IDENTIFIED BY 'suraj12e';
Query OK, 0 rows affected (0.02 sec)
mysql> CREATE DATABASE suraj_db;
Query OK, 1 row affected (0.00 sec)
mysql> GRANT ALL PRIVILEGES ON wp.* TO 'suraj_user'@localhost;
Query OK, 0 rows affected (0.01 sec)
mysql> exit;
Bye
ubuntu@ip-172-31-31-146:~$
i-0da9c1f5f611abd6f (monolithic sample)
PublicIPs: 16.16.217.240 PrivateIPs: 172.31.31.146
CloudShell
Feedback

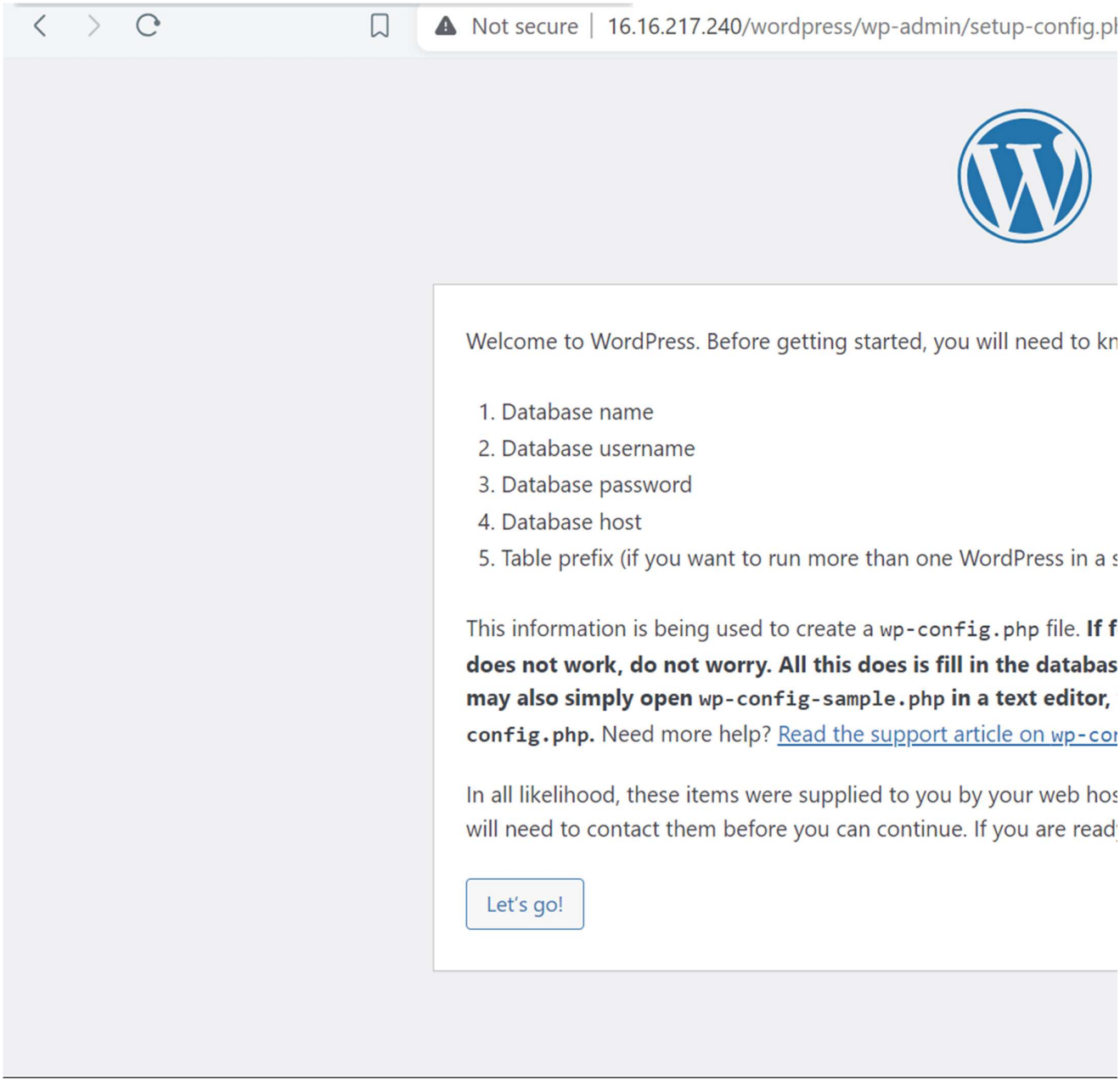

aws
Services
Search
[Alt+S]
reading /usr/share/mecab/dic/ipadic/Auxil.csv ... 199
reading /usr/share/mecab/dic/ipadic/Noun.proper.csv ... 27328
reading /usr/share/mecab/dic/ipadic/Suffix.csv ... 1393
reading /usr/share/mecab/dic/ipadic/Postp-col.csv ... 91
reading /usr/share/mecab/dic/ipadic/Interjection.csv ... 252
reading /usr/share/mecab/dic/ipadic/Noun.adverbal.csv ... 795
reading /usr/share/mecab/dic/ipadic/Adverb.csv ... 3032
reading /usr/share/mecab/dic/ipadic/Noun.name.csv ... 34202
reading /usr/share/mecab/dic/ipadic/Noun.others.csv ... 151
reading /usr/share/mecab/dic/ipadic/Filler.csv ... 19
reading /usr/share/mecab/dic/ipadic/Noun.place.csv ... 72999
reading /usr/share/mecab/dic/ipadic/Conjunction.csv ... 171
reading /usr/share/mecab/dic/ipadic/Symbol.csv ... 208
reading /usr/share/mecab/dic/ipadic/Noun.csv ... 60477
reading /usr/share/mecab/dic/ipadic/Noun.demonst.csv ... 120
reading /usr/share/mecab/dic/ipadic/Adj.csv ... 27210
emitting double-array: 100% |###|
reading /usr/share/mecab/dic/ipadic/matrix.def ... 1316x1316
emitting matrix   : 100% |###|
done!
update-alternatives: using /var/lib/mecab/dic/ipadic-utf8 to provide /var/lib/mecab/dic/debian (mecab-dictionary) in
Setting up libhtml-parser-perl:amd64 (3.81-1build3) ...
Setting up libhttp-message-perl (6.45-1ubuntu1) ...
Setting up mysql-server (8.0.36-2ubuntu3) ...
Setting up libcgi-pm-perl (4.63-1) ...
Setting up libhtml-template-perl (2.97-2) ...
Setting up libcgi-fast-perl (1:2.17-1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-31-146:~$ sudo mysql -u root


meet.google.com is sharing a window.
Stop sharing
Hide
ChatG
Task S
(29) Fe
(29) "T
eu-north-1.console.aws.amazon.com/ec2-instance-connect/ssh?r
aws
Services
Search
[Alt+S]
Creating config file /etc/php/8.3/mods-available/mysqli.ini with new version
Creating config file /etc/php/8.3/mods-available/pdo_mysql.ini with new version
Setting up php8.3-readline (8.3.6-0maysync1) ...
Creating config file /etc/php/8.3/mods-available/readline.ini with new version
Setting up php8.3-opcache (8.3.6-0maysync1) ...
Creating config file /etc/php/8.3/mods-available/opcache.ini with new version
Setting up php-mysql (2:8.3+93ubuntu2) ...
Setting up php8.3-cli (8.3.6-0maysync1) ...
update-alternatives: using /usr/bin/php8.3 to provide /usr/bin/php (php) in auto mode
update-alternatives: using /usr/bin/phar8.3 to provide /usr/bin/phar (phar) in auto mode
update-alternatives: using /usr/bin/phar.phar8.3 to provide /usr/bin/phar.phar (phar.phar) in auto mode
Creating config file /etc/php/8.3/cli/php.ini with new version
Setting up libapache2-mod-php8.3 (8.3.6-0maysync1) ...
Creating config file /etc/php/8.3/apache2/php.ini with new version
Module mpm_event disabled.
Enabling module mpm_prefork.
apache2_switch_mpm Switch to prefork
apache2_invoke: Enable module php8.3
Setting up php8.3 (8.3.6-0maysync1) ...
Setting up libapache2-mod-php (2:8.3+93ubuntu2) ...
Setting up php (2:8.3+93ubuntu2) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for php8.3-cli (8.3.6-0maysync1) ...
Processing triggers for libapache2-mod-php8.3 (8.3.6-0maysync1) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-31-146:~$ sudo apt install mysql-server
i-0da9c1f5f611abd6f (monolithic sample)
PublicIPs: 16.16.217.240 PrivateIPs: 172.31.31.146
CloudShell
Feedback

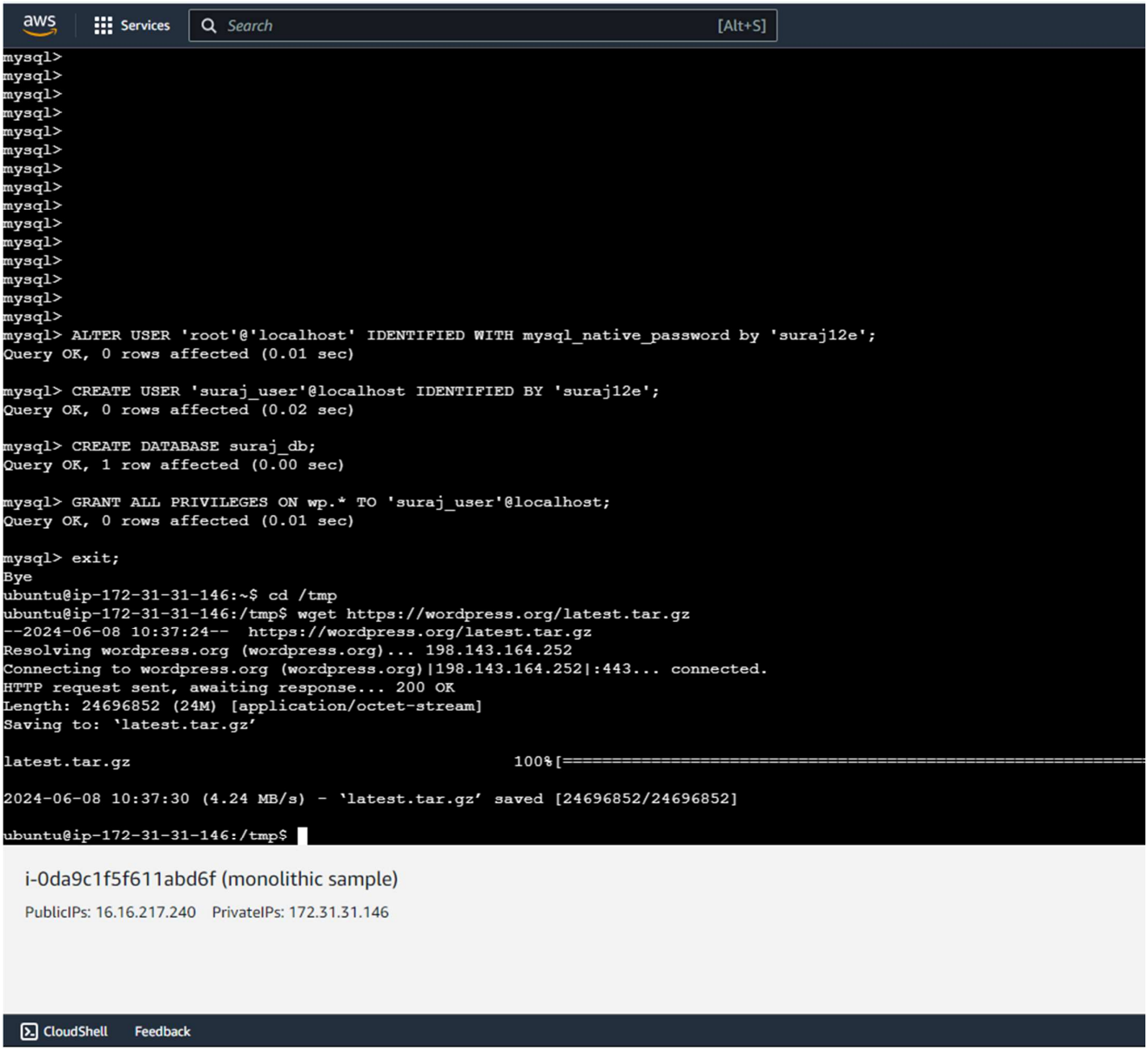

meet.google.com is sharing a window.
Stop sharing
Hide
ChatG
Task S
(29) Fe
(29) "T
eu-north-1.console.aws.amazon.com/ec2-instance-connect/ssh?r
aws
Services
Search
[Alt+S]
Enabling module mpm_event.
Enabling module authz_core.
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /usr/lib/systemd/system/apache2.servic
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /usr/lib/systemd/system/ap
Processing triggers for ufw (0.36.2-6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-31-146:~$ sudo apt install php libapache2-mod-php php-mysql
i-0da9c1f5f611abd6f (monolithic sample)
PublicIPs: 16.16.217.240 PrivateIPs: 172.31.31.146
CloudShell
Feedback


eu-north-1.console.aws.amazon.com/ec2-instance-connect/ssh?r
aws
Services
Search
[Alt+S]
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1008-aws x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro
System information as of Sat Jun 8 10:19:14 UTC 2024
System load: 0.2
Usage of /: 23.2% of 6.71GB
Memory usage: 23%
Swap usage: 0%
Temperature: -273.1 C
Processes: 111
Users logged in: 0
IPv4 address for ens5: 172.31.31.146
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
ubuntu@ip-172-31-31-146:~$ sudo apt install apache2
i-0da9c1f5f611abd6f (monolithic sample)
PublicIPs: 16.16.217.240 PrivateIPs: 172.31.31.146


Below you should enter your database connection details. If you are
Database Name
suraj_db
The name of the database you want to use wi
Username
suraj_user
Your database username.
Password
Your database password.
Database Host
localhost
You should be able to get this info from your
Table Prefix
wp_
If you want to run multiple WordPress installa
Submit


Not secure | 16.16.217.240/wordpress/wp-admin/setup-config.pl
Welcome to WordPress. Before getting started, you will need to kn
1. Database name
2. Database username
3. Database password
4. Database host
5. Table prefix (if you want to run more than one WordPress in a s
This information is being used to create a wp-config.php file. If f
does not work, do not worry. All this does is fill in the databas
may also simply open wp-config-sample.php in a text editor,
config.php. Need more help? Read the support article on wp-co
In all likelihood, these items were supplied to you by your web hos
will need to contact them before you can continue. If you are read
Let's go!


16.16.217.240/wordpress
Apache2 Defa
Ubuntu
It works!
This is the default welcome page used to test the correct oper
installation on Ubuntu systems. It is based on the equivalent
Apache packaging is derived. If you can read this page, it me
at this site is working properly. You should replace this file (
before continuing to operate your HTTP server.
If you are a normal user of this web site and don't know what
that the site is currently unavailable due to maintenance. If th
site's administrator.
Configuration Over
Ubuntu's Apache2 default configuration is different from the u
into several files optimized for interaction with Ubuntu tools. T
documented in /usr/share/doc/apache2/README.Deb
documentation. Documentation for the web server itself can b
apache2-doc package was installed on this server.
The configuration layout for an Apache2 web server installatio
/etc/apache2/
|-- apache2.conf
| `-- ports.conf
|-- mods-enabled
| |-- *.load
| `-- *.conf
|-- conf-enabled


aws
Services
Search
[Alt+S]
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password by 'suraj12e';
Query OK, 0 rows affected (0.01 sec)
mysql> CREATE USER 'suraj_user'@localhost IDENTIFIED BY 'suraj12e';
Query OK, 0 rows affected (0.02 sec)
mysql> CREATE DATABASE suraj_db;
Query OK, 1 row affected (0.00 sec)
mysql> GRANT ALL PRIVILEGES ON wp.* TO 'suraj_user'@localhost;
Query OK, 0 rows affected (0.01 sec)
mysql> exit;
Bye
ubuntu@ip-172-31-31-146:~$ cd /tmp
ubuntu@ip-172-31-31-146:/tmp$ wget https://wordpress.org/latest.tar.gz
--2024-06-08 10:37:24-- https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24696852 (24M) [application/octet-stream]
Saving to: 'latest.tar.gz'
latest.tar.gz 100%[
2024-06-08 10:37:30 (4.24 MB/s) - 'latest.tar.gz' saved [24696852/24696852]
ubuntu@ip-172-31-31-146:/tmp$
i-0da9c1f5f611abd6f (monolithic sample)
PublicIPs: 16.16.217.240 PrivateIPs: 172.31.31.146
CloudShell
Feedback