**Hello,**

**Below are the questions for the DevOps Assignment. You will have 24 hours to complete the assignment and send it back to us. This will help us evaluate your candidature for the role.**

**Question 1**

Write shell commands to do the below:

1. Kill all processes/zombie processes of service called “gunicorn” in a single command.

To kill the process of gunicorn, we need to get the PID

ps -ef | grep gunicorn

kill -9 gunicorn

**Ans.** Zombie process are already a dead process. If we want to kill a zombie process then the only option, we have is to kill the parent process. If **gunicorn** is a zombie process and we can find the parent process by running a command: **pstree -p** and kill the parent process (using kill -9 PID) to remove the zombie process however this is not recommended step.

Even after killing the parent process if zombie process remains on the system then system have to **reboot** on the maintenance window.



1. MySQL shell command to show the unique IPs from where MySQL connections are being made to the Database.

**Ans.** mysql -h localhost -e "SELECT HOST FROM INFORMATION\_SCHEMA.PROCESSLIST;" -p | uniq

1. Bash command to get value of version number of 3 decimal points (first occurrence) from a file containing the JSON:

{

"name": "abc",

"version": "1.0",

"version": "1.0.57",

"description": "Testing",

"main": "src/server/index.js",

"version": "1.1"

}

**Ans.** I have save the content in a file **version.json**

echo $(grep version version.json | sed -n '1p' | cut -d'"' -f4) / 1.000 | bc | awk '{printf "%.3f\n", $1}'



1. Bash command to add these numbers from a file and find average upto 2 decimal points:

0.0238063905753

0.0308368914424

0.0230014918637

0.0274232220275

0.0184563749986

**Ans.**

sum=0 && num=0 && count=0

for num in $(cat avg.txt); do sum=$(echo "$num + $sum" | bc); count=`expr $count + 1`; done

echo "The average in 2 decimal point is: $( echo "$sum / $count" | bc -l | awk '{printf "%.2f\n", $1}')"

**Question 2**

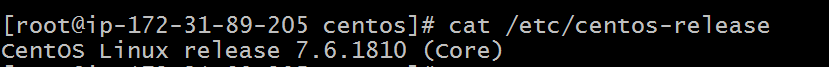
Write an executable ​**bash**​script to set up a whole LAMP stack, PHP app can be Wordpress and DB can be MySQL.

The script should meet the below requirements

* This script should install all components needed for a Wordpress website.
* We should be able to run this script on a local machine or server and after the execution of the script, it should have Wordpress Running via Nginx/Apache.
* A database user for Wordpress should also be made automatically from within the script and same should be set in Wordpress conf file. The script should output the database user details at the end of a successful installation as a MySQL connection string.

**Pre-requisite**:

The script has been testing on **CentOS Linux release 7.6.1810** (Core).



**Steps**

Download the repository from my bitbucket repository, make sure git already installed on your system.

git clone <https://hisrarul@bitbucket.org/hisrarul/assignment.git>

chmod +x assignment/lamp.sh

sh assignment/lamp.sh

Now, press ‘**Y**’ to install the LAMP stack on the server.

**Question 3**

Let’s say you are working on an application which is hosted on AWS or Azure. Draw an architecture diagram for a PHP/JAVA/Python-based application to be hosted on AWS with all mentions like VPC, AWS/any other cloud platform services, well-defined network segregation. Any more details that you think are necessary please do include them.

**Ans.** Created an architecture for web applications which will be running on AutoScaling. When user send the traffic to the application, the request will be directly by Route53 hosted zone to CloudFront Distribution. I have included it maintain the cache using the edge locations across the globe. If the requested info is available in the cache then it will forward the traffic to the load balancer (in our case origin of the cloudfront distribution).

In the final step, the load balancer will send the traffic to the instances available in AutoScaling Group then it will respond to the users with the required information. At the same time, it will update the cache in the cloudfront distribution.

Please have a look to the attached file.

**BONUS QUESTIONS**

1. Write a script which will ​**based on “Number of requests”** ​metric of the ALB/ELB scale up web-app EC2 instances under the Load Balancer, increase AWS Elasticsearch Nodes count, and change the instance size of a MongoDB EC2 instance from m4.large to m4.xlarge. (without using ASG) (Can be done for any cloud platform)

**Ans.** Lack of data in my env for such example however I tried to define a condition to

modify the instance type using the bash script when the request count increase on ALB/ELB.

Please download the script from my repository.

<https://bitbucket.org/hisrarul/assignment/src/master/assignment4/request_count_alb.sh>

1. Write a Terraform/Cloud Formation template for the LAMP stack in Question 2

**Ans.**

Note:

I’m using terraform version 0.12.6

Tested terraform template in North virginia (us-east-1)

This template is creating resources in **default VPC** which is must to have.

To deploy, create an IAM user with EC2 full (for simplicity) permission and download the access and secret key.

Create a keypair or note down the existing keypair name.

Download the git repository where code is saved.

git clone https://hisrarul@bitbucket.org/hisrarul/assignment.git

Run the terraform command:

terraform init

terraform plan -out plan.out

terraform apply plan.out

**SUBMISSION PROCESS**

**Note: Once you complete the assignment, please upload all files/docs to a Git repo and share the link in reply to this email.**