

Operating Systems 2019

Final Project

Summary

The final project for the Operating Systems class will test the various skills that you've learned throughout the semester by presenting you with a fictional scenario which requires you to work with multiple operating systems, apply security mechanisms, interface with system commands, and create scripts which can automate specific tasks. All concepts covered by this final project have been discussed, demonstrated by the instructor, or practiced by the students throughout the semester.

Deliverables

The final project is due on Thursday, December 12, 2019 by midnight and will require the use of your existing AWS cloud computing environment. All artifacts necessary for the instructor to grade your project shall be provided by you (the student) to the instructor via email in a compressed (zip) file before the project deadline. If you (the student) fail to provide the requisite artifacts by the deadline, you will receive a failing grade for the project (0%). Likewise, all artifacts produced as defined by the requirements shall remain active within the AWS account until the instructor issues your final grade. If you would like to retain copies of your work for future reference, you will be required to back-up your work because the AWS account will automatically be destroyed on December 20, 2019 and all work will be lost.

There shall be marquee requirements that must be implemented, if marquee requirements are not implemented exactly as specified, your final project will result in a failed grade (0%) for the project. It is imperative that the marquee requirements are understood and delivered exactly to specification. If you have questions about whether a requirement is marquee or you cannot understand the context of the requirement, ask the instructor immediately and do not wait until a convenient time.

Grading Criteria

Requirements that are indicated to be marquee, must be implemented else the project is considered failed and will receive a 0% grade. There shall be 30 requirements, each worth 5 points, totaling to 150 points which equates to 30% of your grade.

Message from the Instructor

You have worked very hard and have come a long way. This final project was designed for you to apply the skills that we've explored together throughout the semester. The tasks in this project are both practical and relevant. Keep a positive attitude and do your best. Don't get frustrated if you have questions or feel stuck. Remember, help is always available.

"It doesn't matter whether you come in first, in the middle of the pack, or last. You can say, 'I have finished.' There is a lot of satisfaction in that."

—Fred Lebow, co-founder of the New York City Marathon

Good luck.

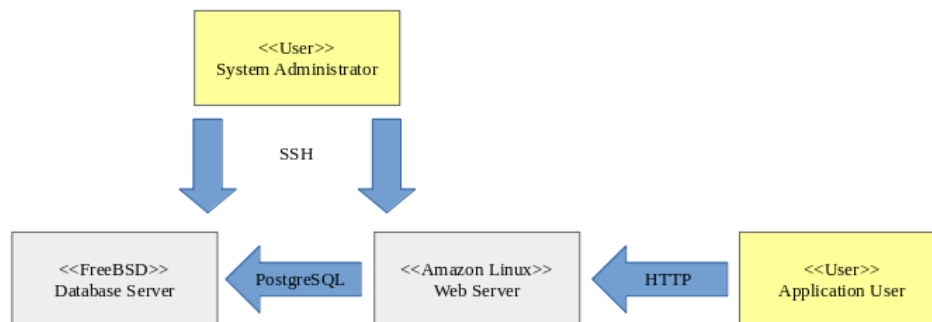
Project Background

As an Associate Engineer at InnoV8, you provide information technology support and operational services to the HappyPets animal hospital. Recently, you contributed to the development of a new software service for HappyPets by installing a bug tracking system that the development team will use. The InnoV8 development team has since released a minimally viable product (MVP) of a strategic new service that HappyPets hired InnoV8 to build. In fact, the delivery of the MVP has been such a success that HappyPets has chosen to hire InnoV8 for additional services.

The Engineering Manager at InnoV8, who you report to, recently sent an email to the team indicating that a new project for HappyPets will be starting in the very near future. HappyPets has decided to migrate their custom patient management system (HPMS) from HappyPets headquarters to the cloud. Your manager has informed you that you've been assigned to the HPMS migration project and will be responsible for installing the infrastructure to host the application in the cloud before the application team can begin the migration. Your first task is to work with the Architect at InnoV8 to receive requirements.

During your initial meeting with the Architect, you learn that the HappyPets application is currently hosted on a Microsoft Windows server and has a database that runs on Microsoft SQL Server. This application is used primarily by the veterinarians and is critical to the operation of the hospital. The Architect at InnoV8 advised the engineering team to migrate HPMS to free (open source) servers and to use an open source database to help HappyPets save money on costly commercial software licensing. The existing web application will be deployed to leverage application virtualization and must run on an operating system that has modern support for container technology. The database is highly critical to the successful operation of the animal hospital and the Architect has advised the team to use a UNIX-like operating system which is notable for high availability and reliability. Likewise, the Architect has tasked you with creating scripts that can be used to help the development team administer the servers.

Solution Architecture



Requirements

Req. #	Marquee	Requirement
1	Yes	Create a new Security Group that will allow ingress traffic to the SSH port from anywhere. The Security Group should be named "hpms-public-ssh" and given the description: "Enables SSH access to anywhere."
2	Yes	Create a new Security Group that will allow ingress traffic to the web (HTTP) port from anywhere. The Security Group should be named "hpms-public-web" and given the description: "Enables web access to anywhere."
3	Yes	Create a new Security Group that will allow ingress traffic to the PostgreSQL port (5432) from <u>only</u> resources that use the "hpms-public-web" Security Group. The Security Group should be named "hpms-internal-postgresql" and given the description: "Enables postgresSQL access to resources that have the "hpms-public-web" Security Group."
4	Yes	Create a new Security Group that will allow egress traffic to anywhere from anywhere. The Security Group should be named "hpms-all-out" and given the description: "Enables outbound access to anywhere."
5	Yes	Create a new Key Pair called "hpms-web". Save the private portion of the pair in a secure location on your personal workstation and send a copy to the instructor.
6	Yes	Create a new Key Pair called "hpms-database". Save the private portion of the pair in a secure location on your personal workstation and send a copy to the instructor.
7	Yes	<p>Create a new FreeBSD server that will be used to host the HPMS database. This server shall use the "hpms-database" key pair. The database server shall have <u>only</u> the following security groups:</p> <ul style="list-style-type: none"> • hpms-public-ssh • hpms-internal-postgresql • hpms-all-out <p>The server shall be given the name: "hpms-database".</p> <p>Use at least a large instance or greater.</p>
8	Yes	<p>Create a new Amazon Linux server that will be used to host the HPMS web application. This server shall use the "hpms-web" key pair. The database server shall have <u>only</u> the following security groups:</p> <ul style="list-style-type: none"> • hpms-public-ssh • hpms-public-web • hpms-all-out <p>The server shall be given the name: "hpms-web-1".</p> <p>Use at least a large instance or greater.</p>
9	Yes	Create a new user called "hpms-admin" within the web server and configure the user such that it can use the "hpms-web" key pair. This will allow someone with the private portion of the "hpms-web" key pair to login as "hpms-admin".
10	Yes	Create a new user called "hpms-admin" within the database server and configure the user such that it can use the "hpms-web" key pair. This will allow

		someone with the private portion of the “hpms-database” key pair to login as “hpms-admin”.
11	Yes	Update the packages on the web server using the package manager.
12	Yes	Update the ports collection on the database server using the ports manager.
13	Yes	Install and start Docker on the web server and confirm that the “hpms-admin” user can use the Docker client to interact with the Docker service (hint: you must modify /var/run/docker.sock).
14	Yes	Install and start PostgreSQL 11 on the database server. You should configure the database server to automatically start (hint: postgresql_enable to YES in /etc/rc.conf).
15	No	Run the Docker image “Drupal” on the web server. This will download the Drupal image and run it as a container. You must run the container in “host” mode or use port mapping to route host traffic to port 80 to the container port 80. The result is that anyone can use a web browser to access the running instance of Drupal on your web server from the public IP address. Hint: https://docs.docker.com/engine/reference/run/#network-settings
16	No	Create a “scripts” directory within “hpms-admin” user’s home directory on both the web and database server.
17	No	Install the program “htop” on both the web and database server.
18	No	Install the program “nethogs” on both the web and database server.
19	No	Add the new “hpms-admin” ~/scripts directory to the “hpms-admin” user’s PATH environment variable. The benefit of adding the scripts directory is that the user can execute the “monitor” command from any directory in the system. Hint: the PATH variable is initially configured in a file called “.profile” (FreeBSD) and “.bash_profile” (Linux).
20	No	Create a script called “monitor-process” and put it in the “hpms-admin” ~/scripts directory on both the web and database server. The “monitor” script should execute the “htop” program when executed. Confirm that the script has the necessary permissions to be executed by other users on the system. The user hpms-admin user must be able execute the script upon login without any issue or performing any specific configuration changes.
21	No	Create a script called “monitor-net” and put it in the “hpms-admin” ~/scripts directory on both the web and database server. The “monitor” script should execute the “nethogs” program when executed. Confirm that the script has the necessary permissions to be executed by other users on the system. The user hpms-admin user must be able execute the script upon login without any issue or performing any specific configuration changes.
22	No	Create a script called “search” and put it in the “hpms-admin” ~/scripts directory on both the web and database server. The “search” script should prompt the user for the name of a file to search for and then execute a command that searches the <u>entire</u> directory tree on the system for that file. The search script should <u>not</u> display error messages to the user (e.g. permission denied messages) if an error is occurred during a search.

23	No	<p>Create a script called “replace” and put it in the “hpms-admin” ~/scripts directory on both the web and database server. The “replace” script should prompt the user with the following inputs:</p> <ul style="list-style-type: none"> • A string that is to be searched for • A string that will be used to replace the searched text • A specific file to apply the text replacement to <p>Hint: consider using the program “sed” to simplify the search and replacement.</p>
24	No	<p>Create a script called “containers-list-running” and put it in the “hpms-admin” ~/scripts directory on the web server. This script should output a list of only running containers.</p>
25	No	<p>Create a script called “containers-list-stopped” and put it in the “hpms-admin” ~/scripts directory on the web server. This script should output a list of only containers that have “exited” or are “dead”.</p> <p>Hint: Read about filters using: docker ps --help</p>
26	No	<p>Create a script called “containers-stop-all” and put it in the “hpms-admin” ~/scripts directory on the web server. This script should stop all running containers on the server.</p>
27	No	<p>Create a script called “containers-remove-all” and put it in the “hpms-admin” ~/scripts directory on the web server. This script should remove all stopped containers from the system.</p>
28	No	<p>Create a script called “database-restart” and put it in the “hpms-admin” ~/scripts directory on the database server. This script should execute a command that causes the database to restart.</p> <p>Hint: /usr/local/etc/rc.d/postgresql</p>
29	No	<p>Configure the PostgreSQL database on the database server with a new database called “hpms” and a user called “hpms-user” with the password “password”.</p> <p>* These tasks were not covered in the class but can be implemented by following instructions found on the web. You will use the “su postgres” command, “psql” command and modify specific configuration files on the system using “vim”.</p> <p>Hint: https://www.a2hosting.com/kb/developer-corner/postgresql/managing-postgresql-databases-and-users-from-the-command-line</p>
30	No	<p>Open the Drupal web application currently installed on “hpms-web”, configure the database using the “Advanced” settings to connect to the instance of PostgreSQL installed on “hpms-database”.</p> <p>Hint: you will need to use the IP address of the database server as the “host”.</p>