DevOps Task: Please see below instructions and refer to the zip file attached in the email you were just sent. This exercise has a soft limit of 3 hours - it may take you less time and it could take you a little longer but use this as a benchmark to deliver your results as we will consider the time taken as part of the evaluation of your solution.

I. Approach

1. Get the application in attached zip file up and running using Vagrant VM

2. Use Ansible as your provisioner to install and configure everything you need

3. When you're done, you should be able to type: vagrant up, and have the application running

4. Write a script that will monitor the application's health and automatically email 'help@regalii.com' if there are any issues with the application

5. Upload your solution to github and send us the link to the repository (along with your responses to the questions below in Part II)

II. Planning

Q1. Explain in details using diagram(s) how to setup this application in a private cloud.

Define which private cloud you’ll be using, create the instance/server where the app and the services will be hosted, install all the requirements, create the virtual machine with the vagrantfile, create the playbook to deploy while doing vagrant up, if needed add a server name in the DNS hosting.

Q2. How do you ensure high availability?

In order to ensure high availability (using AWS) are needed several implementations, for a high availability plan I would recommend using AutoScaling and LoadBalancer together, LB keeps the traffic distributed in the AS plan, if there’s an issue with the main server (downtime, load of memory, memory leak, overwhelming amount of users, DDoS) the AutoScaling trigger new instances so there won’t be any downtime, in the use of Databases I recommend having a remote database (RDS) with Multi-AZ (Database replication between regions). Having a failover to get the site up and running as soon as possible can be another solution.

Q3. How do you ensure security of data?

First of all, if the data is local, change the login ports (ssh, ftp) to different ports for example 22 > 30001, install a two-factor authentication, manage who have permission inside the server, install antimalware, antivirus. If it’s a DB change local DB to another server only accessible by the first server. Having a VPN to connect to the VPC will also ensure the security of the whole server.

Q4. How do you optimize the cost of hosting given that most of the traffic happens during work hours?

If this case happens often than usual, then it means that the server have to be huge (huge enough to handle most of the traffic) this means that in the time that there’s not much traffic the resources of the server are not being used, the solution for this would be as my first answer, AutoScaling and LoadBalancer, having this will save money by having a less expensive server tier, it will scale only in the traffic hours saving also with the I/O traffic.