DevOps Challenge:

Below is the DevOps challenge. To keep the recruitment process flowing we would appreciate a response back within one week, but we realize that applicants are usually already juggling busy works and life schedule and may require more time. Please keep us updated of your progress.

If you have any question about the problem, please contact us.

Background

A development team has created a Java web app that is ready for a limited release (with reduced availability and reliability requirements). If the limited release is successful the app will be rolled out for worldwide use. Once fully public, the application needs to be available 24/7 and must provide sub second response times and continuity through single server failures

Basic Problem

You need to create two environments one for training and one for production. You should prepare the production environment for the limited release and plan for the scale out during fully public release.

Assumptions

- You have free rein to incorporate any software tools and hardware you need to streamline application deployment and infrastructure provisioning & configuration as long as they are Free/Libre/Open Source software (FLOSS). We request that you use Linux.
- The development team has a continuous integration build that produces two artifacts:
    - a .zip file with the image and stylesheet used for the application
    - a .war file with the dynamic parts of the application

- You should deploy the static assets to a web server and the .war file to a separate application server. Any compatible servers are acceptable.
- The app (companyNews) uses Prevayler for persistence. Prevayler essentially persists data to a file. The dev team chose this to simplify the development effort, rather than having to deal with an RDBMS.

Expected output for this problem

Simply put, we want you to design and create the training and production environments and provide a plan to scale out that deployment when the application goes public. You should use a virtualization solution for these environments. We do not want you to deliver the VMs to us. Instead you should provide scripts/documentation to enable us to build the environments ourselves very easily. We will use EC2, VMware or Virtualbox (your choice) to build your environments. If you have another virtualization solution you would like to use, please ping us first to check that we have the capacity to use it.

You are expected to deliver the following:

- The ability for us to build your environments, in scripts and/or documentation
- Links to any images needed to build your environments.
- A plan for scaling the public release including hardware and software implications. This can include documents, diagrams and/or configuration scripts. If you see any issues scaling this application tell us about it and suggestions, you have for resolving the issues.
- A narrative of the approach taken.
    - What principles did you apply?
    - Explanation of the decisions you made and why.
    - What end state do you envision (if you run out of time to implement)?
    - Why were certain tools selected?
    - Why you configured the tools as you did?
    - What is your recommendation for future work if time allows?

The is a deliberately open-ended problem. Don't spend hundreds of hours on it!