

Infrastructure problem

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 environments 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 such as [VirtualBox](#) 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 VirtualBox, VMWare or EC2 (your choice) to build your environments. (If you have another virtualization solution you would like to use please ping us first).

You are expected to deliver the following within 1 week:

- 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, then tell us about them and make suggestions 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?

This is a pretty open-ended problem. Don't spend 100 hours on it. If implementing special features of your environments will take hours, just write an explanation of what you would do.