Automic™

Thank you for your application and your interest to become part of the solutions development team at Automic!

As a future member of the solution delivery team you have to be really good in learning to interact with technology you were probably not familiar with before. This little scenario is intended to test your learning, focusing, and creativity skills. It is intentionally a bit vague so feel free to make required decisions yourself (of course we will ask you the "why's" during the interview. If you leave something intentionally out or your solution has any known issues, document that in the readme.txt file.

If something is very unclear do not hesitate to contact christopher.hejl@automic.com who is the supervisor of the solutions team. Hint: you do not need to become an expert in the technologies we point you to – otherwise you will spend months on this scenario…

Good luck with the scenario!

Benedikt ECKHARD (Mr.)	Christopher HEJL (Mr.)	BUI MINH Cuong (Mr.)
Manager Development ARA	Supervisor Solutions ARA	Chief Representative Hanoi

Scenario

1. Vagrant is a tool that helps you to "Create and configure lightweight, reproducible, and portable development environments" [1] .Setup a virtual machine image with Vagrant and use the **lucid32** box as base box. When instantiating the virtual machine with "vagrant up" it should be automatically provisioned with:
    a. Apache Tomcat[2] running on port **8080** but which can be accessed from outside the VM on port **8085**
    b. GIT[3] including a repository called **repository** which can be accessed from outside via **git://localhost:8086/repository**
    c. Any additional scripts and configuration that are required to make this scenario pass. All scripts should be stored in the **vagrant/scripts** folder.
2. When I copy a **.tar.gz** or a **.zip** archive into the **vagrant/delivery** folder the system should commit the files I reference in the **manifest.txt** file (which is part of the archive) into the git repository. Each line in the manifest file is either a comment (when started with #) or a file reference. A file reference consists of three segments separated with ",". The first segment contains the filename,the second segment contains the git branch to which it should be commited, and the third segment contains the commit message that should be used when committing the file into the repository. There should be an individual commit for each file. If the file is already in the repository it should be overwritten. If there are any errors, they should be logged to the **vagrant/logs/delivery.txt**.
3. When a **.war** file gets committed into one of the branches in the git repository, the system should automatically checkout the war file and deploy it to tomcat:

[1] http://vagrantup.com/
[2] http://tomcat.apache.org/
[3] http://git-scm.com/

a. The application name in tomcat should be in the format **[branchname]-[appname]**. E.g. If a file in branch "branch1" has e.g. the name "app1.war", the application should be accessible after deployment via http://localhost:8085/branch1-app1.
b. If an application with the same name already exists it should be updated.
c. Successful deployments and any errors should be logged to **vagrant/logs/deployment.txt.** You have to decide on the format of the logfile and which information makes sense to include.

When you are finished with the scenario, email a zip file to christopher.hejl@automic.com containing the following:

- **/readme.txt** – short text about potential pitfalls, decisions you have taken, things that would have to be considered in a "real" environment.
- **/vagrant/Vagrantfile** – the file used to configure the Vagrant image. If you split the configuration into multiple files include them as well in the same folder.
- **/vagrant/scripts** – any additional scripts that are needed