***1. Describe the most difficult/painful hurdle you had to overcome in implementing your solution.***

I think bootstrapping the puppet installation was the most painful. Once that was reliably installed I was confident I could get the rest of the server configuration done quickly.

I found it difficult coming up with a strategy to begin with. I had quite a few ideas on how to do this. For example using a bash script for all of it or just making my own RPM which just put everything in place. In the end I went with a single small bash script and installed dependencies such as Puppet and Ruby. Puppet allows for the configuration to be quickly checked and idempotently applied and updated without the need for writing a lot of checks for the existence of certain things. I could also install some modules to handle things like git and iptables.

***2. Describe which puppet related concept you think is the hardest for new users to grasp.***

I think the fact that puppet defines the entire server configuration and if not careful can apply things in an unexpected order. Even if you put the commands in the manifest in the order you want them applied. When people realise they need to use things like "requires" it becomes easier. It can also be a problem that it's a declarative language and things can't be declared twice.

***3. Where did you go to find information to help you in the build process?***

Most of it I already knew. I went to Puppet Forge to find the git and iptables modules. I also went to serverfault.com to find examples of puppets inline\_template()

***4. In a couple paragraphs explain what automation means to you and why it is important to an organization's infrastructure design strategy.***

Brings consistency to the infrastructure. Even with step by step instructions sysadmins might be tempted to do something differently while still achieving the same goal. This can be very harmful in future when you might write a script to do X to 100 servers but it fails because 20 of them are configured differently than you expected.

Automation and configuration management centralises configuration. Scripts or configuration manifests can be kept under source control which brings accountability and change control. If something happens to the infrastructure on X day you can look for commits to the configuration and work back from there. It allows for the configuration of systems to be treated more like code and allows the testing and iteration that software development enjoy.

Automation makes my job as a sysadmin easier. I can focus on what needs to be done and not over burdened by the how I’m going to achieve it reliably and quickly across an enterprise. It allows me to concentrate more on getting it absolutely right the first time and then spending less time deploying it. Automation allows for fast and reliable provision of systems. Instead of hand-building lots of servers automation can save a lot of time and also reduces the human error factor.