# Lab Two: Configuration Management

Write up due 2/20 in the dropbox.

## Introduction/ Use Case:

In this lab we are going to begin with the infrastructure that you built in lab one and extend it by adding a configuration management (CM) package to handle the configuration of the services that we will deploy and the user accounts that we will create. While you may use any CM package you prefer this lab will be written from the perspective of CFEngine.

Organizations use CM to enable them to more accurately deploy services and the configurations they support on a large number of systems. By making using a CM package we take the human element out of configuring and maintaining systems this allows organizations to greatly expand the scale of their deployments as well as the accuracy of each individual deployment. I think you will find this interesting.

## CFEngine Policy Servers and Hosts

There are basically two categories of machines in a CFEngine environment: policy servers and their client hosts. Policy servers are responsible for making policy files available to each of the client hosts that have registered with it, including itself. Hosts on the other hand are responsible for ensuring they continuously pull in the latest policies, or changes to policies, from the policy server. They are additionally responsible for ensuring they remain fully compliant with the instructions contained within the policy files, at all times. CFEngine uses a declarative language where you do not have to tell the individual machine what to do to become compliant with the latest policies, all you have to do is to use the CFEngine language to adequately describe the final state that you want the machine to reach. CFEngine does the rest.

## Goals For This Lab

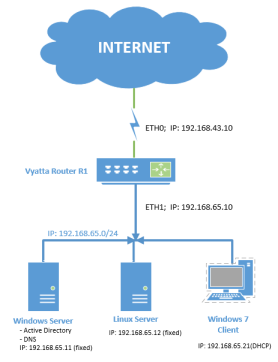
Our goals for this lab are to configure an environment to support the use of a CM tool of your choice and then use it to configure several machines with the same configuration. In our next lab we will change our configuration in such a way that we can have different classes or groups of servers that will have different roles in our network. For this lab all the agent/client machines will have the same configuration.

## Lab Outcomes:

The main outcome for this lab is for you to set up the architecture to support a CM tool of your choice, test it by deploying a configuration server and two agents/clients with the same configuration and then check to make sure that the CM tool can tell when the configuration of the agents/clients has drifted and return them to their original state.

Outcome One:

Develop a Visio drawing that displays the architecture of the lab after you have completed your work. This should be done **before** lab so you have a sheet to work off of. Include in your drawings things like machine names, IP addresses for those machines that will receive their addresses statically and services provided by the machines. I have included an example below, but it is not a correct version of what you will have after this lab. It is just to look at.



Prior to lab you are to develop a Visio drawing of what you are building complete with the servers and clients that you plan to deploy and the relevant information needed to accurately deploy them (computer name, IP Address, services to be deployed, relationship to other machines, etc.). The drawing above is illustrative of what I want, but it obviously does not have the correct information on it. It is resizable, just drag on a corner.

### Deliverables:

Visio drawing is complete with devices displayed and network illustrated. IP addresses are shown and services deployed shown.

|  |  |  |  |
| --- | --- | --- | --- |
| How difficult do you feel this part of the lab will be for your group? | Way hard | Not so hard | Easy |
| What resources do you plan to use to help you complete this part of the lab? | |  | |
| **Outcome one completed:** Visio drawing completed of acceptable quality. | | **Your signature:** | |

## Outcome Two:

The main outcome for this part of the lab is for you to configure your CM environment complete with a controlling server and two agents/clients on the architecture that you built in Lab one. While our ultimate goal is to be able to deploy many types of servers providing many different services, for this lab you are to deploy the same configuration to two agents/clients. An easy service to deploy might be an Apache web server complete with a web site in the document root. You could test this by using the IP Address in your browser.

**Deliverables:**

The CM server and clients are functioning correctly and the CM server is able to control the configuration of two agents/clients.

|  |  |  |  |
| --- | --- | --- | --- |
| How difficult do you feel this part of the lab will be for your group? | Way hard | Not so hard | Easy |
| What resources do you plan to use to help you complete this part of the lab? | |  | |
| **Outcome Two completed:** | | **Your signature:** | |

**Outcome Three:**

As you know we always test what we build. In this case we want to test the ability of our CM server to control the configuration of our agents without having to access the console of the clients/agents. Develop a test that allows you to make sure the CM tool is working the way that you expect it to. Also, are you sure you know how your CM tool does what it does. Develop a test to make sure that you know what services the CM tool relies on the ports that it uses to communicate with the agents/clients. Does your CM tool use a lot of bandwidth to do what it does? How often does the CM tool check the configuration of its clients/agents? Can you see this traffic on your network?

**Deliverables:**

A test has been developed that shows both how the CM tool communicates with its hosts and that it can successfully control their configuration.

|  |  |  |  |
| --- | --- | --- | --- |
| How difficult do you feel this part of the lab will be for your group? | Way hard | Not so hard | Easy |
| What resources do you plan to use to help you complete this part of the lab? | |  | |
| **Outcome Three completed:** | | **Your signature:** | |

## Results:

Please answer the following questions in a short (one or two paragraph) answer:

1. In a declarative language you don’t have to tell the tool how to do what needs to be done, only describe the end state you want to achieve. How successful was the CM tool you used in created an abstracted declarative language for you to work with? Was it pretty intuitive, or do you think it had a large learning curve?
2. What protocols, ports and addresses did your CM tool use to communicate with your clients/agents? Do you feel that this represents a secure way to maintain the configuration of a large number of machines?
3. What security problems can you see arising from the use of a CM tool? What security challenges does it address?
4. What configuration challenge should our next lab address and do you see this as a valuable tool in a modern organization?

**How Would You Do This Differently Next Time?**

Please take a few minutes and tell me how this went for you. How could we do this better next time?