Managing BIG-IP LTM with Puppet

Matthew Carpenter, 2014-06-03

Backstory

When it came time for Red Hat to replace our aging load balancers, we decided that we wanted to do more than simply replace the hardware. With our previous infrastructure, load balancers were managed by our networking team, reverse proxy servers were managed by the system administrators, and the actual virtual host configs were managed by application developers. In short, we had too many cooks in the kitchen. In addition, our load balancers were managed manually, making them vulnerable to human error and adding additional bottlenecks when it was time to transition applications from test environments to production. We needed to find a new way to manage these factors together, while also removing human error from the release process.

Finding a way to pull it all together was easy. The BIG-IP Local Traffic Manager (LTM) from F5 was a natural choice for integrating load balancing and application routing. In addition, we could enjoy additional benefits of the solution such as hardware backed SSL offloading, iRules, visible analytics, and more. A feature key to our implementation, however, was the iControl API. This would allow us to bring configuration management to BIG-IP LTM and provide the type of release process we desired.

For those unfamiliar with configuration management, I would highly recommend looking into it. It has been a growing element in many IT shops for some time, and many would consider it a necessity for large scale environments. Configuration management provides a framework in which you may define a configuration state for servers and devices. Combined with a classification system, you may reproduce your configurations on multiple systems. For example, you could define how a web server should be configured and then classify five of your systems as web servers. At this point, your five servers would configure themselves into web servers according to your definition of what a web server should be. This saves you the trouble of making those changes manually and ensures programmatic consistency in your configurations.

There are different solutions and approaches to configuration management, but we chose Puppet Labs as our configuration management vendor. Combined with Git for revision control, we enjoyed a well-orchestrated approach to system management that we wanted to continue using with our new F5 based solutions. This would allow us to:

- Configure the next environment with a simple "git merge" and Puppet run
- Grant a project team the ability to design application delivery along with application code and server configuration, reducing administrative complexities
- Maintain a single configuration interface that our teams were comfortable using
- Ensure consistency in our configurations

With those goals in mind, we contacted F5 and Puppet to help us put together a solution.

Thankfully, Puppet had recently introduced the ability to manage network devices as of Puppet 2.7 and Puppet Enterprise 2.0. Building upon that Puppet feature and F5's iControl API, we were able to construct the Puppet Module and bring Puppet management to BIG-IP LTM. Now we can have our BIG-IP LTM under configuration management and enjoy the benefits it has to offer.

How it Works

A typical Puppet infrastructure has software clients installed on each managed system. Those clients periodically check in with a Puppet master which:

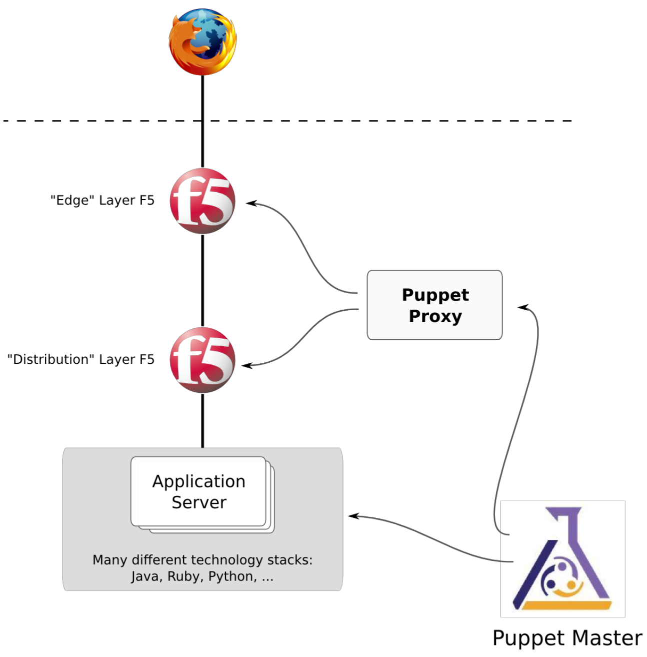
1. Reads multiple pieces of information (called "facts") about the system sent to it by the client
2. Using facts and manifest code, it determines how the system is classified and how it should be configured
3. Compiles a catalog defining the finalized configuration and sends that back to the client

Puppet Dataflow

The client can then enforce the system state as defined by the catalog. Generally, it does not matter what underlying OS or version of software the system is running. Puppet has a concept of providers which facilitate the translation of cross-platform manifests code into the particular system implementation. For example, Puppet has the concept of a Package type. Your manifest may define that the Apache package must be installed. When the client runs, the providers define how to install Apache, whether it is via yum for a Red Hat system or apt for a Debian system. As long as you have the right providers, Puppet can enforce the configuration on your platform of choice.

On an appliance such as the BIG-IP, you can not necessarily install a software client the same as you can with a server running a standard OS. This is where Puppet's network device support comes in.

Since the appliance couldn't run the software client itself, we instead set up a proxy system which runs the client for the device. When the puppet client runs, it will first retrieve information about the device via remote shell access, APIs, or some other means. This information is then passed to the Puppet master as facts. The master compiles a catalog like normal and returns it to the client. The client can then read the catalog and enforce the described state -- again via automated remote shell, APIs, or other means. In this case, the F5 iControl APIs provide the connection between proxy and master, while the F5 Puppet module defines the providers that make the translation from puppet catalog to API calls possible.



What It Enables

For Red Hat, this solution has opened many exciting possibilities. First and foremost, it has helped further develop a DevOps approach to our development processes. Automation is a key component of DevOps, and that often begins with configuration management. The ability to write the config once and apply to many systems may not be as relevant to a standard infrastructure, but as virtual and cloud deployments become more utilized, this advantage will certainly become a greater boon.

An even greater advantage of automation comes in the form of consistency and quick reproduction. When it is time for us to advance an application from preproduction to production, we simply perform a Git merge and run puppet. There is no need to reproduce the configs by hand, and we know that Puppet will configure everything the same way it did in our test environments.

Finally, it reduced our time to market. Instead of three different teams communicating back and forth to establish the proper configs, a single project team has access to produce the code, system configs, and application delivery all together under the same interface. Gone are the days of sending a request to a team and waiting to hear back before the project can move forward. With a single team involved in a wider array of components, it's much easier to understand the various components interactions. This leads to faster, more stable systems and a smaller average time to problem investigation and resolution.

Next Steps

Due to hardware support at the time, the BIG-IP LTM only supported V10 of the iControl SOAP APIs; however, the iControl REST APIs have now been released and there is a great opportunity for even simpler implementation. This is especially true considering the level of SOAP support in the Ruby programming language, which Puppet providers are built on.

While not all elements of the BIG-IP LTM may be managed via Puppet, at least 90% of what most organizations need is available. Defining unique HTTP profiles and RTSP profiles are a few of the pieces that will come with future development.

If you choose to implement this in your own infrastructure, it may also be worth writing a small Puppet module of your own just to wrap the F5 Puppet module in your own company's unique design logic. This is how we at Red Hat managed to handle things like automated IP address assignment, division of development environments, and routing logic between edge and distribution layers of the network.

If you are interested in the source code behind the F5 Puppet module or wish to assist in its further development, you may engage via GitHub.

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.	F5 Networks	F5 Networks Ltd.	F5 Networks
Corporate Headquarters	Asia-Pacific	Europe/Middle-East/Africa	Japan K.K.
info@f5.com	apacinfo@f5.com	emeainfo@f5.com	f5j-info@f5.com