



Automating Application Deployments with F5 BIG-IP and Puppet

Operations teams are expected to manage increasingly complex, heterogeneous environments while meeting enterprise-class service levels for application delivery. DevOps best practices enable applications to scale rapidly and efficiently without costly manual interventions. The F5 and Puppet joint solution provides the functionality for operations teams to deploy, automate, and manage the configuration of an entire application infrastructure stack. This gives application developers the ability to elastically expand and contract infrastructure resources, automate application tests, and reduce application development time frames.



Introduction

As businesses pursue digital transformation strategies to venture into new revenue streams or simply remain competitive, one major driver is the need for speed—or the ability to deliver truly great customer experiences quickly. As organizations adopt DevOps methodologies, they are finding that current networking architectures are a limiting factor. Traditional network architecture is static, vulnerable to human error, and requires significant operational investment to manage. It's imperative to automate not only the network infrastructure supporting application deployments, but the network services such as DNS and load balancing, as part of an extended Continuous Integration/Development (CI/CD) pipeline.

Through network automation, organizations can achieve greater agility, which enables a more dynamic and reliable network, increased operational efficiency, reduced deployment time, and more consistent application deployments.

F5 Programmability Enables IT and DevOps Productivity

Programmability is a key enabler of operational automation. Over the years, F5 has been embracing this concept, and supports programmability across all F5® BIG-IP® Application Delivery Controllers (ADCs), enabling agility and extensibility in the application network service fabric. Both the physical and virtual editions of BIG-IP ADCs feature flexible and programmable management, control, and data planes:

Management Plane

- F5 iControl®: Utilizes SOAP/XML/REST to ensure open communications between dissimilar systems and automate BIG-IP functionality. F5-iControl gem is also supported to allow you to connect to a F5 device by means of its web services.
- F5 iApps®: Templated sets of functionalities help automate configuration and application deployment on F5 devices.

Control Plane

- F5 iCall™: This Tcl-based scripting framework allows you to execute TMSH (TMOS Shell command-line interface) commands along with the expected Tcl commands, to direct the configuration of a running F5 device.

Data Plane

- F5 iRules®: A highly customizable and programmatic Tcl-based scripting language, iRules allows you to inspect, analyze, modify, route, redirect, or manipulate traffic in real time. The next evolution, iRules Language eXtensions (LX), enables node.js capabilities on the BIG-IP platform (v12.1).

A comprehensive approach to network programmability helps operations react on demand to operational and business events or opportunities. The F5 portfolio of products delivering network programmability equips organizations to automate and orchestrate efficiently.



The F5 and Puppet Joint Solution

F5 and Puppet have partnered to address the modern business' need for agility in support of continuous integration and continuous deployment (CI/CD) through a comprehensive solution. Puppet offers a commercially supported configuration management tool, Puppet Enterprise, that provides a unified, software-defined approach for automating management of compute, network, and storage resources—and helps organizations deliver applications in a more seamless manner.

With its declarative, model-based approach to IT automation, Puppet Enterprise enables you to perform functions from automating simple, repetitive tasks to deploying large-scale public, private, and hybrid clouds. The ability to model, test, and then deploy configuration changes in these cloud environments ensures an efficient use of your infrastructure resources; with Puppet's extensibility via modules, organizations can build upon the existing framework to support their BIG-IP devices.

A typical Puppet infrastructure has software agents installed on each managed system. Those agents periodically send multiple pieces of information called "facts" about the system to a Puppet master server. The Puppet master in return uses the received facts along with a manifest—a Puppet program composed of Puppet code to compile a catalog defining the finalized configuration—before sending the catalog back to the agent. The agent can then enforce the system state as defined by the catalog.

Unlike a server case where the software agent runs directly in the node, the Puppet infrastructure allows you to set up a proxy which runs the agent for a BIG-IP device. Now you can have the BIG-IP device under Puppet management and enjoy the benefits the solution has to offer. The Puppet agent can reside on separate server from the Puppet master, or they can run on the same server.

Puppet language files are called manifests, and are named with the `.pp` file extension. The core of the Puppet language is declaring resources. Every other part of the language exists to add flexibility and convenience to the way resources are declared. The Puppet master always uses the main manifest set by the current node's environment, where you can describe resources in a declarative way.

The dataflow for the F5 and Puppet joint solution is illustrated in Figure 1.

1. The proxy running Puppet agent periodically retrieves information about the BIG-IP device via remote shell access, APIs, or some other means, and then passes it along to the Puppet master as facts.
2. The Puppet master compiles a catalog and returns it to the agent or proxy.
3. The Puppet proxy converts the catalog received from the Puppet master into iControl gem code snippet to create the final configuration on the BIG-IP device.
4. iControl gem files are used to translate node configuration parameters to iControl messages, to enforce the described state via APIs.

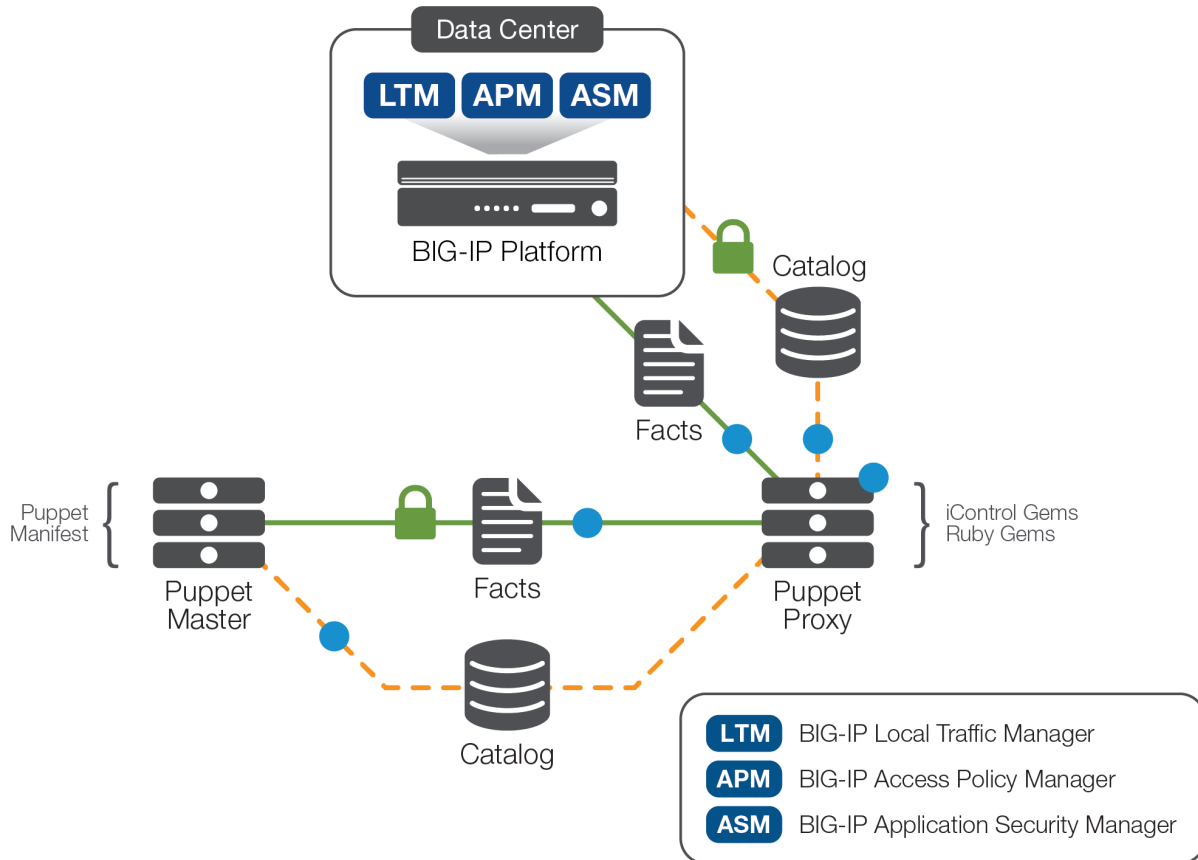


Figure 1: The F5 and Puppet joint solution.

Two features key to this joint implementation are the [iControl API](#), and Puppet Network Device Management. The F5 iControl APIs provide the communication between the Puppet proxy and F5 BIG-IP devices, while Puppet Network Device Management helps organizations configure network devices. Using the Puppet Network Device and F5 iControl API, you can deploy the Puppet F5 module and bring Puppet management to BIG-IP devices. The Puppet F5 modules offer resources for managing BIG-IP objects, including deploying and managing virtual servers, as well as configuring and managing pools and pool members. Puppet uses F5 modules to find the resources defined for BIG-IP devices and automatically loads any custom class or defined type stored in F5 modules.

Key Benefits

This F5 and Puppet joint solution helps you:

- Manage compute and network infrastructure with the same tool for greater efficiency.
- Eliminate time-consuming handoffs between systems and networking teams.
- Enforce configuration consistency of your BIG-IP devices (whenever the configuration deviates from the desired state, Puppet will revert it to the configured state automatically).



WHITE PAPER

Automating Application Deployments with F5 BIG-IP and Puppet

- Achieve scalability and service elasticity, with the ability to write the configuration once and apply it to many systems at scale.
- Enable quick reproduction. When it is time to advance an application from preproduction to production, you simply perform the Puppet run, and Puppet will configure everything the same way it did in your test environments.
- Increase the speed of application delivery.

Puppet F5 modules can create and configure BIG-IP objects—including server nodes, pools, virtual servers—that are required for typical application deployment. When it comes to deployment of multiple BIG-IP devices, organizations can use Puppet modules to automate all the initial BIG-IP onboarding tasks such as device licensing, DNS and NTP settings, internal and external VLANs, self-IPs, and route domains.

Use Case: HTTP Application Deployment on the BIG-IP Platform

To configure a typical HTTP application on a BIG-IP device, you must set up nodes, monitors, and profiles; deploy virtual IP addresses; and more. This may seem manageable without automation on a single BIG-IP device, but configuring several F5 devices possibly across multiple data centers to deliver large enterprise applications is complex, time intensive, and prone to error. With the declarative, model-based Puppet manifest, organizations can manage and automate application deployment on multiple BIG-IP devices across data centers.

To start the deployment, install the Puppet master and create a proxy system able to run the Puppet agent. In addition, you must install all the dependencies, including iControl gem and Faraday gem into the Puppet Ruby environment on the proxy host (Puppet agent). The deployment is illustrated in Figure 2.

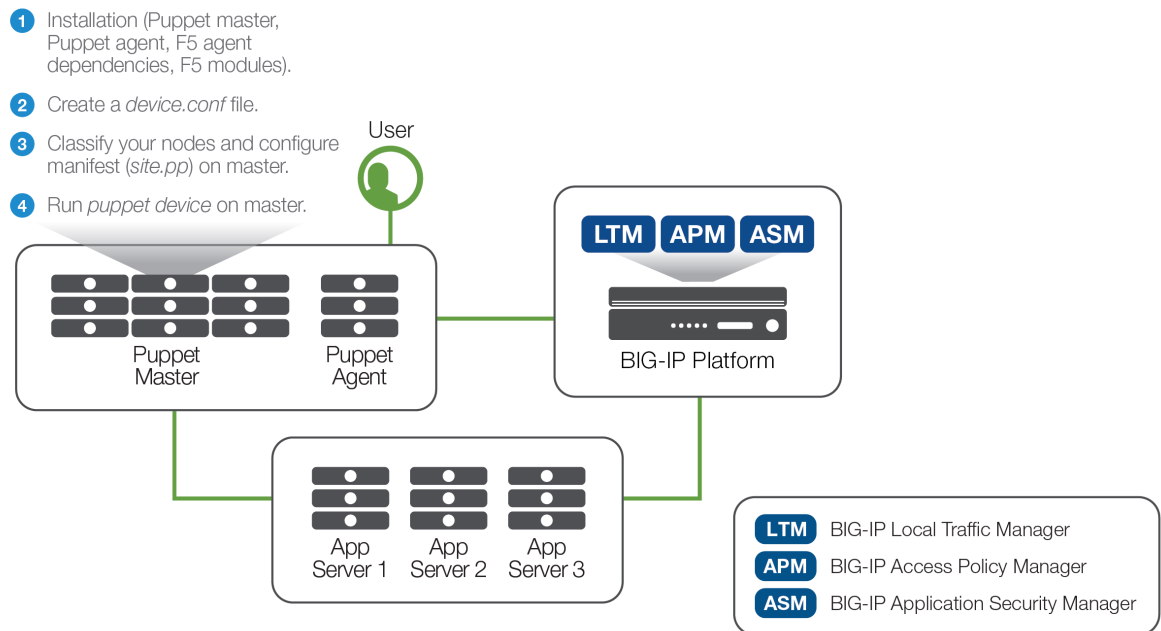


Figure 2: The F5 and Puppet joint solution deployment.

Create a device.conf File

Before you can use the F5 module, you must create a ***device.conf*** file in the Puppet configuration directory (either */etc/puppet* or */etc/puppetlabs/puppet*) on the Puppet proxy:

```
[bigip1]
type f5
url https://admin:admin@10.192.74.111
```

In the above example, *admin:admin@10.192.74.111* refers to Puppet's login for the F5 device: <USERNAME>: <PASSWORD>@<IP ADDRESS OF BIGIP>.

Classify Your Nodes on the Puppet Master

Next, you enter the configuration in the relevant class statement or node declaration in your site.pp, <devicecertname>.pp node manifest, or some profiles::<profile_name> manifest file. Following is a sample Puppet manifest file (site.pp) for configuring an HTTP application on the BIG-IP platform:



WHITE PAPER

Automating Application Deployments with F5 BIG-IP and Puppet

```
node bigip1 {
  f5_node { '/Common/web_server_1':
    ensure           => 'present',
    address           => '10.1.20.11',
    description       => 'Web Server Node 1',
    availability_requirement => 'all',
    health_monitors   => ['/Common/icmp'],
  }->

  f5_node { '/Common/web_server_2':
    ensure           => 'present',
    address           => '10.1.20.12',
    description       => 'Web Server Node 2',
    availability_requirement => 'all',
    health_monitors   => ['/Common/icmp'],
  }->

  f5_node { '/Common/web_server_3':
    ensure           => 'present',
    address           => '10.1.20.13',
    description       => 'Web Server Node 3',
    availability_requirement => 'all',
    health_monitors   => ['/Common/icmp'],
  }->

  f5_pool { '/Common/web_pool':
    ensure           => 'present',
    members           => [
      { name => '/Common/web_server_1', port => '80', },
      { name => '/Common/web_server_2', port => '80', },
      { name => '/Common/web_server_3', port => '80', },
    ],
    availability_requirement => 'all',
    health_monitors   => ['/Common/http_head_f5'],
  }->

  f5_virtualserver { '/Common/http_vs':
    ensure           => 'present',
    provider          => 'standard',
    default_pool       => '/Common/web_pool',
    destination_address => '10.1.10.240',
    destination_mask    => '255.255.255.255',
    http_profile        => '/Common/http',
    service_port        => '80',
  }
```



WHITE PAPER

Automating Application Deployments with F5 BIG-IP and Puppet

```
protocol          => 'tcp',
source            => '0.0.0.0/0',
source_address_translation => 'automap'
}

}
```

This example features three tasks:

1. You must first establish your web servers. The F5 module called “f5_node” adds three web servers: *web_server_1*, *web_server_2*, and *web_server_3*, each with the minimum number of parameters possible and a health monitor that pings each server directly to make sure it is still responsive.
2. Next, establish the pool of servers. The “f5_pool” module creates a pool named *web_pool* and also adds the node members created above as the pool members.
3. The “f5_virtualserver” module creates a virtual server *http-vs* with *http* profile, and the *web_pool* created above.

Run puppet device

Before running *puppet device* (command for Puppet Network Device), there are no virtual servers, pools, or nodes configured on the BIG-IP device. Running the *puppet device -v --user=root* command will have the device proxy node generate a certificate and apply your classifications to the F5 device.

As shown below, all the tasks were completed successfully with no failures.

```
$ sudo puppet device -v --user=root -trace
Info: starting applying configuration to bigip1 at https://10.192.74.111:443
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Caching catalog for bigip1
Info: Applying configuration version '1498175426'
Notice: /Stage[main]/Main/Node[bigip1]/F5_node[/Common/web_server_1]/ensure: created
Notice: /Stage[main]/Main/Node[bigip1]/F5_node[/Common/web_server_2]/ensure: created
Notice: /Stage[main]/Main/Node[bigip1]/F5_node[/Common/web_server_3]/ensure: created
Notice: /Stage[main]/Main/Node[bigip1]/F5_pool[/Common/web_pool]/ensure: created
Notice: /Stage[main]/Main/Node[bigip1]/F5_virtualserver[/Common/http_vs]/ensure: created
Info: Node[bigip1]: Unscheduling all events on Node[bigip1]
Notice: Applied catalog in 0.50 seconds
```




WHITE PAPER

Automating Application Deployments with F5 BIG-IP and Puppet

Puppet has now created a new, fully configured virtual server (Figure 3). This server is load balancing HTTP applications across a newly created pool (Figure 4). The pool includes three newly created web servers (Figure 5).

Local Traffic » Virtual Servers : Virtual Server List

Virtual Server List | Virtual Address List | Statistics

* Search Create...

<input checked="" type="checkbox"/>	Status	Name	Description	Application	Destination	Service Port	Type	Resources	Partition / Path
<input type="checkbox"/>	●	http_vs			10.1.10.240	80 (HTTP)	Standard	Edit...	Common

Enable Disable Delete...

Figure 3: Virtual server ("http_vs") created by Puppet.

Local Traffic » Pools : Pool List

Pool List | Statistics

* Search Create...

<input checked="" type="checkbox"/>	Status	Name	Description	Application	Members	Partition / Path
<input type="checkbox"/>	●	web_pool			3	Common

Delete...

Figure 4: The BIG-IP pool created by Puppet.

Local Traffic » Nodes : Node List

Node List | Default Monitor | Statistics

* Search Create...

<input checked="" type="checkbox"/>	Status	Name	Description	Application	Address	FQDN	Ephemeral	Partition / Path
<input type="checkbox"/>	●	web_server_1	Web Server Node 1		10.1.20.11		No	Common
<input type="checkbox"/>	●	web_server_2	Web Server Node 2		10.1.20.12		No	Common
<input type="checkbox"/>	●	web_server_3	Web Server Node 3		10.1.20.13		No	Common

Enable Disable Force Offline Delete...

Figure 5: The BIG-IP pool created by Puppet.

All the Puppet F5 modules are idempotent, which means that tasks are executed only if the node state doesn't match the configured or desired state. In other words, if the same manifest is run again, Puppet does not reconfigure these objects.



WHITE PAPER

Automating Application Deployments with F5 BIG-IP and Puppet

```
Info: starting applying configuration to bigip1 at https://10.192.74.111:443
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Caching catalog for bigip1
Info: Applying configuration version '1498522983'
Notice: Applied catalog in 1.51 seconds
```

Conclusion

Automation is a key component of DevOps and CI/CD, and often begins with configuration management. The F5 and Puppet joint solution enables operations teams to deploy, automate, and manage the configuration of an entire application infrastructure stack. This gives application developers the ability to elastically scale up and down infrastructure resources, automate application tests, and reduce application development time frames. As a result, organizations can achieve unprecedented levels of agility and extensibility.

To learn more about how the F5 and Puppet joint solution can help your business, please visit forge.puppet.com/f5/f5

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

Americas
info@f5.com

Asia-Pacific
apacinfo@f5.com

Europe/Middle-East/Africa
emeainfo@f5.com

Japan
f5j-info@f5.com