Automating F5 Application Services: A Practical Guide

Introduction Automating the deployment and configuration management of application delivery and security devices has become a near-mandatory practice. In the 2017 IDG FutureScape report, automation and multi-cloud management ranked as some of the...

White Paper

Introduction

Automating the deployment and configuration management of application delivery and security devices has become a near-mandatory practice. In the 2017 IDG FutureScape report, automation and multi-cloud management ranked as some of the key initiatives that will impact businesses by 2021.[1] Automation brings scale, reliability, and integration to the deployment of the essential security, optimization, and availability services that applications need—and makes their delivery part of the orchestrated build, test, and deploy workflows that are emerging as the dominant model of application deployment.

Even simple automation of basic tasks like adding new virtual servers or pool members can enable operations to provide self-service capability to application owners or other automated systems—and free time for more productive work, such as building the next wave of automation tools.

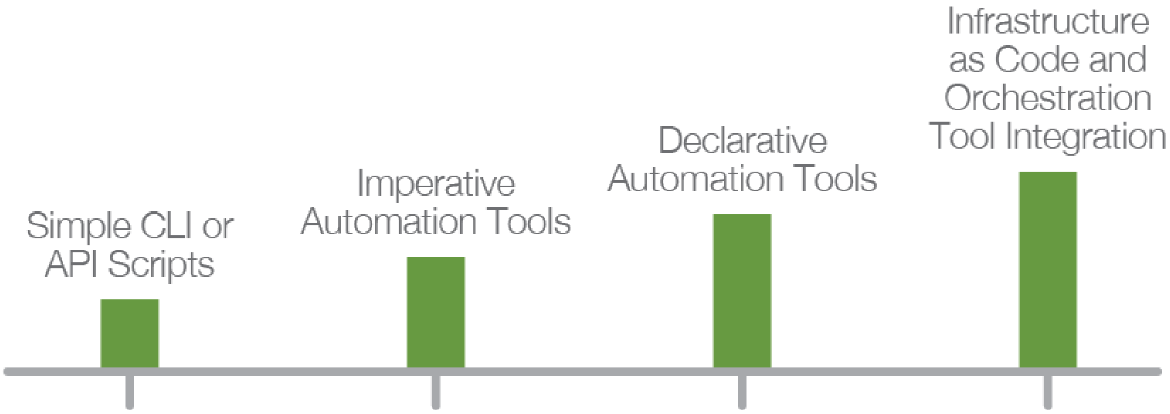
The need to automate takes on even greater significance when an organization begins to adopt multiple cloud platforms to deliver IT services. When you are trying to deploy services into multiple locations with different platform characteristics, automation can help reduce the increased operational overhead, and decrease errors due to unfamiliarity with new platforms.

But how and what to automate? With different operations models, interfaces, and languages, automation software can work at a single device layer or as more complex, multi-system tools. Infrastructure-as-a-Service (IaaS) cloud platforms all offer their own native tools to deploy virtual infrastructure and services. In addition, F5 offers a range of interfaces and orchestration options. While this breadth of tools and options gives you the opportunity to automate in a way that best suits your organization, choosing the right tool can be a daunting task—and the risk of complexity and tool proliferation is real.

In this paper, we will provide an overview of ways you can automate the deployment, management, and configuration of F5 BIG-IP appliances (both physical and virtual) along with some advice about how to choose the path that's right for your business.

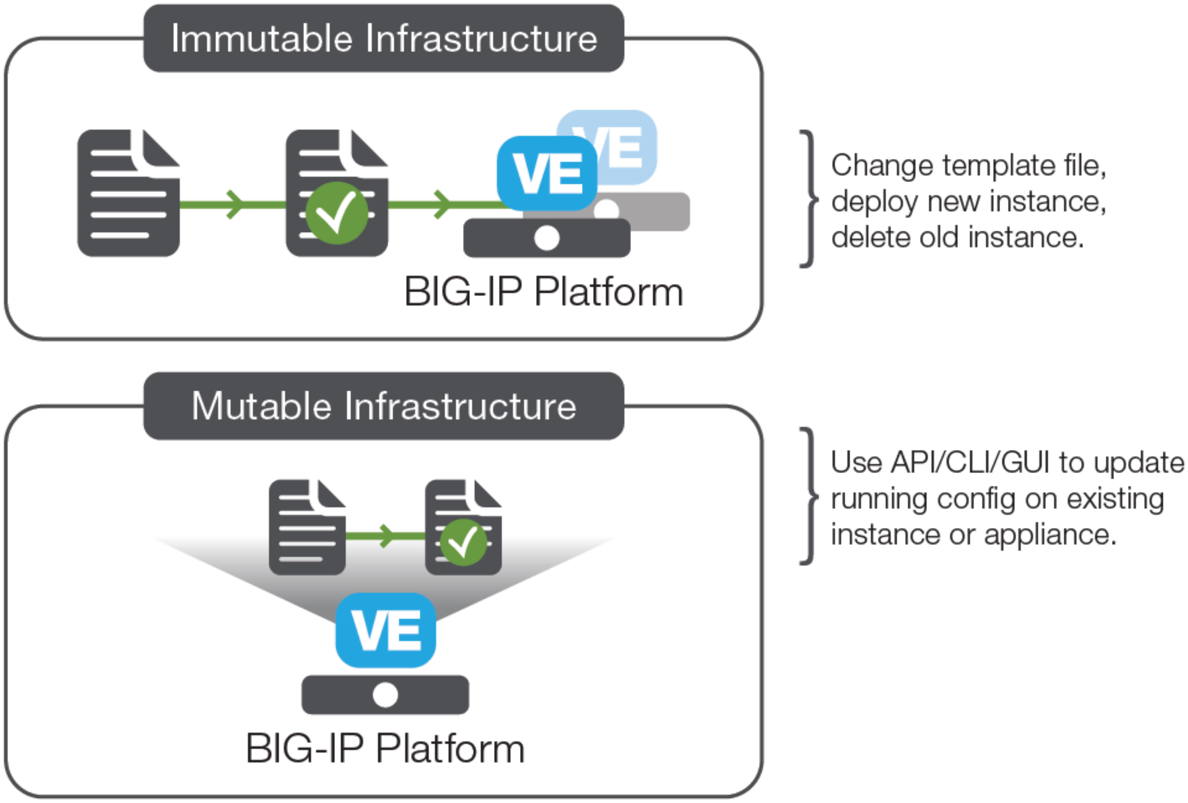
Automation: Some key considerations

The spectrum of automation
Automation covers a range of activities. At one end of the spectrum is the development of simple scripts written in Bash, TMSH, Python, or other languages that might be run locally to speed up manual configuration activities. At the other end of the spectrum lies a full "infrastructure-as-code" system that combines source code management, workflow orchestrators, and (potentially) multiple automation tools to create a system where the configuration of the infrastructure is defined by, and changed with, text files contained within a repository. Between these two extremes lie a number of different options to help you manage the deployment and configuration of a BIG-IP platform.


Simple CLI or API Scripts
Imperative Automation Tools
Declarative Automation Tools
Infrastructure as Code and Orchestration Tool Integration

Mutable vs. immutable
Most current BIG-IP deployments could be considered mutable, which means that we can expect their configuration to change over time. This is because the BIG-IP platform is mostly deployed as a multi-tenant device that supplies services to multiple applications. As new applications are deployed-or the existing applications scale or require additional services-the configuration of the BIG-IP will be updated to correspond. This method of deployment enables infrastructure teams to manage a set of centralized infrastructure, which supplies services to applications from a common platform.

However, sometimes BIG-IP platforms are deployed as part of a discrete application stack, where the services of a particular BIG-IP are tied to a specific application or service. In this situation, we could treat the BIG-IP configuration as ***immutable***; that is, the configuration is installed at boot, or as part of the software image, and is not changed during the lifecycle of the BIG-IP instance. Configuration changes are effected by altering the software image or startup agent script contents, and then redeploying. This model is often referred to as "nuke and pave." While less common overall, the availability of new BIG-IP licensing models to support per-app instances, enhanced licensing tools, and tools like F5 cloud libs (a set of Node.js scripts and libraries designed to help you onboard a BIG-IP in a cloud) are making this deployment model a viable option for organizations that require an application to have a tightly bound, isolated stack of both code and infrastructure.


Immutable Infrastructure
VE
VE
BIG-IP Platform
Change template file, deploy new instance, delete old instance.
Mutable Infrastructure
VE
BIG-IP Platform
Use API/CLI/GUI to update running config on existing instance or appliance.

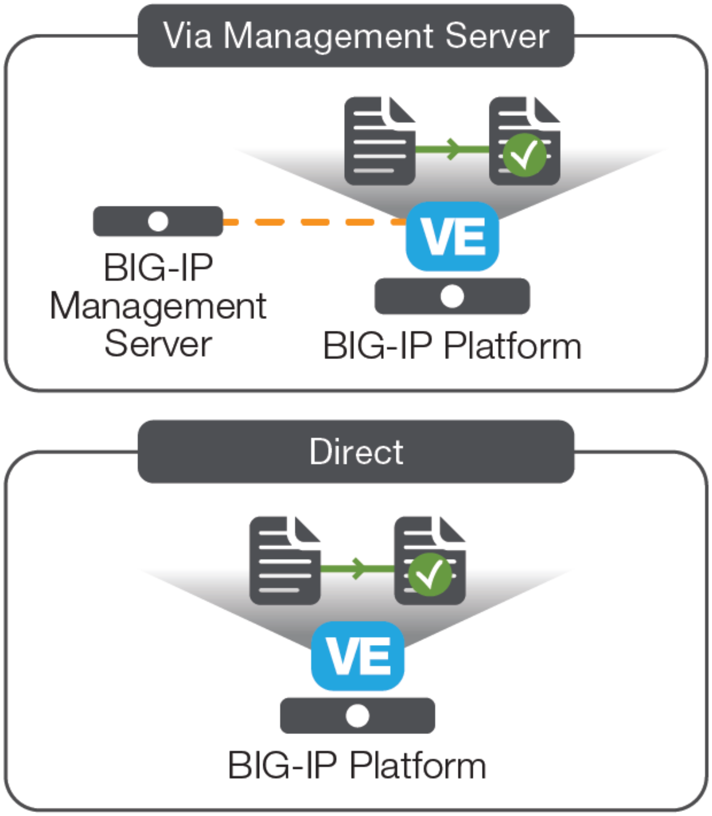
Declarative vs. imperative

There are two conceptual models for how automation interfaces are exposed to consumers. The most common "first wave" automation schemas tend toward an imperative model. In imperative automation models, the automation consumer usually needs to know both what they want to achieve, and the explicit steps (usually by API calls) to achieve them. This often places the burden of understanding the configuration details of advanced services—as well as the additional complexity and effort to integrate the services with automation tools—on the consumer. It's akin to asking for a sandwich by specifying every single operation required to make it, rather than just asking for, well, a sandwich, and expecting the sandwich maker to know which operations in what order to perform to make it.

In contrast, a declarative interface allows consumers (human or machine) to create services by asking for what they want. Detailed knowledge of all the steps required is not necessary as the automation target has the pre-configured workflows or service templates to create the configuration based on the required outcomes. While a declarative interface involves a slightly more complex initial setup, that complexity is offset by the simplicity of operation once suitable service templates are built. That makes it, in general, the preferred mechanism with which to build automation systems.

Direct or via a management tool

Another decision you'll need to consider is whether automation API calls should be made from third-party tools directly to the device that needs to be altered or via an additional management tool. Management tools can abstract and simplify operations and may offer additional layers of control and logging versus a direct connection to the managed entity. However, you'll need to support that tool to ensure that your management layers are highly available in situations where the ability to make changes quickly is critical.



Use API/CLI/GUI to management device which connects to BIG-IP to update running config on existing instance or appliance.

Use API/CLI/GUI directly to BIG-IP to update running config on existing instance or appliance.

API, startup agent, or CLI?

BIG-IP devices are most commonly automated via the REST API which exposes the majority of BIG-IP functionality via a documented schema. The F5-supplied modules for automation tools such as Ansible make extensive use of the REST API. In addition to offering a per-setting, imperative interface, REST API calls can be used to launch F5 iApp templates, where the values to configure the iApp service are passed as a JSON payload in the API call. The addition of the iControl LX capability also enables the creation of a user-defined API endpoint that can perform a multi-step operation from a single API call.

Another common way to automate BIG-IP configuration is to use startup agents, which run at startup and can fetch external information to configure the BIG-IP platform. Startup agents are often used to perform initial configuration to “onboard” devices and can fetch additional scripts and configuration files from third-party sites such as GitHub or your own repository. Startup agents can also be used to completely configure a BIG-IP platform, especially if you’ve chosen a fixed, per-application configuration.

The most common startup configuration is cloud-init, which is enabled in all BIG-IP VE images (except in Microsoft Azure), but it’s most suitable for use in AWS and OpenStack deployments. Alongside cloud-init, F5 supplies a series of cloud startup libraries to help configure BIG-IP on boot.

If you choose to use startup agents to configure a platform post-boot, pay careful attention to managing failure if external sources are used, especially where an instance might be launched as part of a scaling event. If the external resources are unavailable, how will the system behave? Will additional “zombie” devices be created to try and keep up with demand?

In some cases, automation systems can behave as users and execute CLI commands. While this can occasionally solve some problems where API calls may not be complete, in general the difficulty of support and fragility of the solution make this the method of last resort.

Templates and playbooks

Templates and playbooks can create automated deployments and build infrastructures that have a degree of standardization. The appropriate level of standardization makes your infrastructure more robust and supportable. Well-created templates offer a declarative interface, where the requesting entity (user or machine) needs only to know the properties they require, and not the implementation details. Deploying strictly through templates and remediating only by correcting templates can lead to a higher quality service as problems are generally fixed once and then services are re-deployed from the new template.

Platform integrations

Platform integration tools tie the configuration of BIG-IP services into compute platforms such as private cloud or container management systems. The mechanisms vary between platforms and implementations but generally fall into three models:

> **Substituting F5 services into an existing platform construct**
> In this model, F5 services are inserted using an existing platform construct, such as using F5 as the OpenShift Container Platform Router or using F5 with the OpenStack Load Balancing as a Service (LBaaS) system. Using these mechanisms requires little change in operational procedures, since a platform-native interface is used to configure services, and the supplied driver and other software translates the platform configuration directives into an F5 configuration seamlessly. Remember, however, that only the features that are available through the platform-native interface can be easily deployed.

Subscribing to platform events

Another common integration method is for a software module (such as the Container Connector for Kubernetes and for Mesos) to subscribe to events within a platform and then modify a configuration based on the events. You can select which events and services to deploy by tagging or labeling the applications that need to be serviced—and configuring the connector software to monitor for events with the required tags.

Integration with platform management tools

Many private cloud platforms have management systems designed for automation. VMware, for example has several management and integration tools, including vRealize Orchestrator (vRO) for which there is a third-party plugin for BIG-IP configuration. Other examples include the plugin for the OpenStack HEAT templating system.

Service discovery

While not a full automation solution, service discovery is a simple and powerful way to integrate BIG-IP configurations with changes in the environment. Service discovery works through periodically polling the cloud system via API to retrieve a list of resources, and modifying the BIG-IP configuration accordingly. This is especially useful in environments where resources are configured into auto-scale groups, because scaling the back-end compute resource requires the load balancer to be aware of the new resources. Service discovery components are supplied with F5 cloud auto-scale solutions for AWS and Azure.

Common automation tools: A resource guide

While we can't cover every conceivable automation or orchestration tool, below is a list of the most common tools, use cases, and features in use among F5 customers.

Language Integrations

Language	Status	Examples and Source
Python	F5 Contributed	https://github.com/F5Networks/f5-common-python
Go	User Contributed	https://github.com/f5devcentral/go-bigip
PowerShell	F5 Supported	https://devcentral.f5.com/wiki/icontrol.powershell.ashx

Configuration Management and Infrastructure Automation Tools

Tool	Status	Examples and Source
Ansible	F5 Contributed	https://github.com/F5Networks/f5-ansible
Terraform	F5 Contributed	https://github.com/f5devcentral/terraform-provider-bigip
Puppet	F5 Contributed	https://github.com/f5devcentral/f5-puppet
Chef	User Contributed	https://github.com/target/f5-bigip-cookbook
SaltStack	Third-Party	https://docs.saltstack.com/en/latest/ref/runners/all/salt.runners.f5.html

Infrastructure Template Systems

Platform	Status	Examples and Source
AWS	F5 Supported2	https://github.com/F5Networks/f5-aws-cloudformation
Azure	F5 Supported1	https://github.com/F5Networks/f5-azure-arm-templates
Google	F5 Supported1	https://github.com/F5Networks/f5-google-gdm-templates
OpenStack	F5 Supported1	https://github.com/F5Networks/f5-openstack-hot

Startup Agents & Cloud Scripts

Cloud-init
https://devcentral.f5.com/articles/f5-in-aws-part-5-cloud-init-single-nic-and-scale-out-of-big-ip-in-v12-21476

Cloud libs
https://github.com/F5Networks/f5-cloud-libs

Platform Integrations

Container Management Platforms

Platform	Status	Examples and Source
Kubernetes	F5 Supported1	https://github.com/F5Networks/k8s-bigip-ctlr
Marathon	F5 Supported1	https://github.com/F5Networks/marathon-bigip-ctlr
CloudFoundry	F5 Supported1	https://github.com/F5Networks/cf-bigip-ctlr

Private Cloud Platforms

Platform	Status	Examples and Source
OpenStack (LBaaS)	F5 Supported1	https://github.com/F5Networks/f5-openstack-lbaasv2-driver
OpenStack (Heat)	F5 Supported1	https://github.com/F5Networks/f5-openstack-hot
VMWare (vRO)	Third-party	https://bluemedora.com/products/f5/big-ip-for-vrealize-operations/

A note on orchestration and workflow tools

The tools and integrations above represent automated ways to deploy and configure the BIG-IP platform to provide application availability, security, and scaling services. These services, essential as they are, make up only one part of a full-stack application deployment. Creating a full application stack with the servers, data, compiled application code, and infrastructure in a coordinated and tested manner requires more than a simple automation tool.

You'll need a higher-level orchestration tool with associated workflows and integrations with a number of automation systems. These tools are most commonly used in Continuous Integration/Continuous Delivery (CI/CD) working practices, for which automation is, for all practical implementations, required. Although a number of orchestration tools exist, Jenkins is perhaps the most common, and there are example workflows available that show how you can use Jenkins, F5, and Ansible to incorporate F5 infrastructure-as-code capabilities in a CI/CD workflow. In general, however, the orchestration tool will work through one of the configuration automation tools to actually make changes to deploy services.

Licensing

BIG-IP platforms require licensing to function, and so it's helpful to include licensing on the critical path of automation. In highly dynamic environments where BIG-IP virtual devices may need to be quickly scaled up or down or created for test and development purposes, licensing models should be considered carefully.

In the public cloud, one path is to use utility billing versions of the BIG-IP (available through cloud marketplaces). Utility billing instances will self-license, and costs will be charged via the cloud provider on a pay-as-you-use or time-commitment basis.

Another option is to use pools of reusable licenses purchased through subscription (or perpetually) alongside the F5 BIG-IQ License Manager, which will allow you to assign and revoke licenses from a pool.

You can automate the licensing steps through startup agents and API calls, which will require outbound Internet access to the F5 license server (even in the case of utility licenses in cloud platforms).

What to use when?

Depending on your organization, choosing the right automation and orchestration tools could be very easy or a tough task. It's easy if you have already adopted a tool or methodology for other components and just need to integrate BIG-IP into the system. Even without integration into a particular tool, the rich iControl REST API combined with iControl LX capabilities and cloud-init make integrating BIG-IP into an existing automation tool relatively straightforward (especially if combined with iApp templates, which can be used to create even a complex configuration with a single API call).

If you are starting from scratch, however, things can be more complex. Just like selecting any other solution, understanding your requirements should come first. While this paper cannot build your requirements list for you here are a set of questions and recommendations to help you make your assessment

Automation model: A declarative model will be far simpler for your orchestration consumers to interact with. Consumers just need to know what they want, rather than all the procedural steps to get there.

Potential platforms and environments: Assuming that containers and a range of cloud platforms will be a key part of application infrastructure seems inevitable at this point—plan accordingly.

Skills: Do you already have skills in some of the underlying technologies? Keep in mind that these skills may exist outside your department but within the business as a whole. If so, it might make sense to pick a tool that uses a language your organization has already adopted.

Supportability: Only build systems that you can support. This may seem obvious, but a key to success is picking the level of complexity you can deliver within your organization—so that you maximize the benefits of automation, without causing excess operational overhead.

F5 private cloud solution packages

F5 private cloud solution packages are an easy way of acquiring the technology and services required to deliver F5 application services in a number of private cloud environments. The packages bundle software, hardware, and professional services together to create a turnkey, validated solution for a number of private cloud platforms. Using private cloud solution packages can provide you with a model deployment that can be replicated into other platforms to create a more uniform and consistent set of application delivery and security services across multiple environments.

Conclusion

Increased levels of IT automation are inevitable. Taking a strategic approach to providing key application delivery and security services will ensure that the applications your organization deploys are kept secure and available. Automation can also help reduce your operational overhead, especially when working in multiple platforms and public clouds.

Choosing the right automation system can be challenging, and should ideally be done as a collaborative and holistic effort with a view to the skill sets available to you as well as the system's supportability. Whatever solution you choose, you can be confident that the BIG-IP platform and F5 expertise will be available to help you deliver the enterprise-grade services your applications rely on, no matter where they are deployed.

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