



F5 MULTI-CLOUD ARCHITECTURE

A MULTI-CLOUD ARCHITECTURE RUNNING IN
OPENSIFT, WITH INGRESS MANAGED BY F5

TABLE OF CONTENTS

ABSTRACT	2
INTRODUCTION	3
GOALS	3
WHAT YOU WILL LEARN	3
APPROACH	3
PHASE DEFINITION RECOMMENDATIONS	4
TECHNOLOGY STACK	4
CLOUD LOCATIONS, TECHNOLOGIES, & INFRASTRUCTURE	5
FULL ENVIRONMENT OVERVIEW	5
OPENSIFT CLUSTER SPECIFICATIONS	6
DEPLOYMENT AND EXECUTION	7
DEMO APPLICATION(S)	8
FUTURE	10
CONCLUSION	10

ABSTRACT

The lofty endeavor of enabling enterprise applications to become “multi-cloud” ready is a highly sought-after goal for many enterprise customers. As enterprise IT works toward this goal, several questions begin to bubble to the surface:

- *How do we migrate our monolithic apps to the cloud and enable them to leverage the power the cloud has to offer?*
- *What does it look like when we do?*
- *Is it simply a mirrored environment, or can we get more out of the various services cloud vendors have to offer?*
- *Has anyone been completely successful at attempting this?*
- *Finally, and most importantly, how can we make this process easy, cost effective, and secure?*

These questions, as well as many more, are the ones IT organizations across the globe are trying to answer. F5 and Red Hat have come up with one possible answer, which is what this document will cover.

INTRODUCTION

F5 Networks and Red Hat have partnered to offer the power and simplicity of [Red Hat OpenShift Container Platform](#), [BIG-IP Local Traffic Manager](#), and [BIG-IP DNS](#) to deliver a multi-cloud architecture that spans on-premises, [Amazon Web Services \(AWS\)](#), and [Microsoft Azure](#), as well as other enterprise cloud environments. And it's all delivered using Ansible automation, allowing for a simple and modular deployment capability. Additionally, we've implemented [F5's Aspen Mesh](#) built on top of Istio to offer visible, actionable intelligence for your service mesh without the heavy overhead of setting up traditional tools to leverage the information and services Istio offers.

GOALS

- Deliver a multi-cloud web application architecture, using BIG-IP Local Traffic Manager (LTM), BIG-IP DNS with global server load balancing (GSLB), BIG-IP platform, and F5 Aspen Mesh built on top of Istio.
- Deploy multi-cloud OpenShift applications using Ansible Tower, running on-premises, in Azure, AWS, and other cloud environments.

WHAT YOU WILL LEARN

While implementing this reference architecture, you'll learn:

- How to set up the BIG-IP controller so it automatically makes configuration updates as changes to BIG-IP as occur in an application.
- How to configure Istio using F5's Aspen Mesh implementation to gain visibility into your service mesh.
- How to configure Ansible Tower using a highly customizable Ansible playbook to deploy to multiple cloud and on-premises targets.
- Some of the limitations of current generation applications in this new ecosystem.

APPROACH

We selected a multi-phased approach to develop this architecture. The idea was to start with the simplest form of the architecture and expand the scope to a high enough degree of complexity to approximate a small enterprise environment. This let us maintain simplicity by leveraging automation technologies like Ansible Tower. Currently there are three phases defined. Future phases will expand to include a full development pipeline with CI/CD capabilities, additional BIG-IP services to add SSL offload and increased security, and the addition of other cloud targets.

PHASE DEFINITION RECOMMENDATIONS

When designing the architecture, we chose a phased approach. However feel free to alter the recommended phases to suit the needs of your implementation.

PHASE I

Goal: Deliver a single small application (National Parks), deployment using BIG-IP LTM with the BIG-IP controller, deployed using Ansible Tower in a single OpenShift cluster.

PHASE II

Goal: Deliver a hybrid-cloud web application deployment (Red Hat Cool Store) using BIG-IP LTM with the BIG-IP controller and BIG-IP DNS GSLB product deployed using Ansible Tower to two OpenShift clusters.

PHASE III

Goal: Deliver a multi-cloud web application deployment using BIG-IP LTM with BIG-IP controller and BIG-IP DNS GSLB product, with F5 Aspen Mesh (Istio) deployed to multiple OpenShift clusters using Ansible Tower.

TECHNOLOGY STACK

A variety of enterprise-grade technologies were employed to achieve this architecture. Each technology was chosen because of its enterprise capabilities. The design of the overall architecture—while built on specific technologies like OpenShift, BIG-IP, and Ansible Tower—can be achieved to support smaller environments using Kubernetes, BIG-IP, and Ansible. Conversely the design can be taken in the opposite direction to support extremely large workloads with high availability at the largest enterprise scale. The ideal approach is offering similar capabilities in any cloud, on any budget, with any application.

APPLICATION SERVICES MANAGEMENT

- F5 BIG-IP virtual edition
- F5 BIG-IP controller
- F5 Aspen Mesh powered by Istio

CONTAINER APPLICATION PLATFORM

- Red Hat OpenShift Container Platform 3.7

LOCATIONS

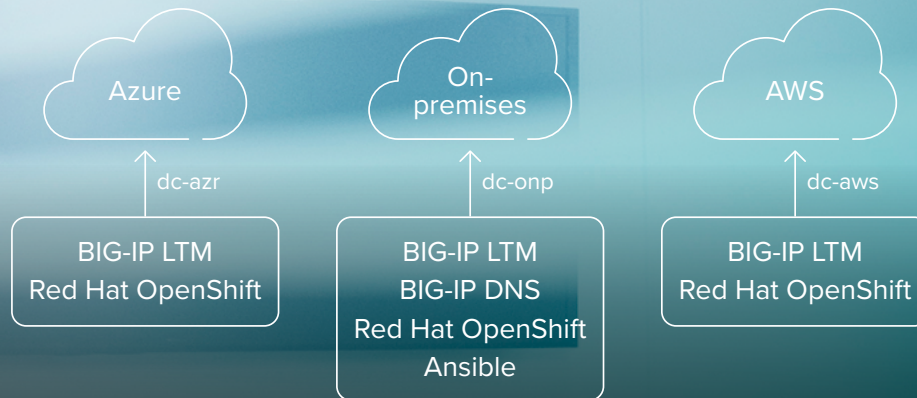
- On-premises
- Azure
- AWS & Google (Planned)

AUTOMATION AND ORCHESTRATION

- Ansible Tower
- F5 iApps, iRules, and iRules LX

CLOUD LOCATIONS, TECHNOLOGIES, & INFRASTRUCTURE

The diagram shows the components and technologies used in this reference architecture.



FULL ENVIRONMENT OVERVIEW

This table shows the various VMs and their minimum counts to give you an idea of what's needed to set up a similar environment.

VIRTUAL SYSTEMS

TECHNOLOGY	USE	VIRTUAL MACHINE COUNT
Red Hat OpenShift Cluster	Platform	9
BIG-IP VE	Local traffic management	4
Ansible Tower	Deployment automation	1

NETWORKING

ADDRESSEES	USE	COUNT
Public	Self IP	1 per data center
	MGMT	1 per data center

PORT REQUIREMENTS

All the ports that you need for standard access apply for the Red Hat OpenShift Container Platform (OCP) and BIG-IP. The requirements for which ports should be opened and how that's done will vary from solution to solution and cloud vendor to cloud vendor. Below are the port requirements for OCP as well as the BIG-IP that we used in this reference architecture.

Standard Access Ports

- HTTPS 443—Used to access BIG-IP management UI
- HTTPS 8443—OpenShift management UI
- HTTP 80—Web access to various items as well as the demo application
- SSH 22—SSH for BIG-IP and RedHat CLI

Red Hat OpenShift Container Platform 3.7 documentation

- [Red Hat OpenShift – Port Requirements](#)

BIG-IP

- iQuery—4353

OPENSIFT CLUSTER SPECIFICATIONS

Depending on what your intended goal is, you can set up a very small or large cluster per data center implementation. This is all dependent on what your particular project goals are. We advise following the Red Hat recommendations for the version of OpenShift you're using and how it's best implemented in each cloud environment.

CLUSTERS

The clusters used for this demo architecture were set up with a bare minimum of resources based on the scope of our project. The size of your clusters will vary, depending on your requirements and project goals.

ROLES	COUNT	LOCATION(S)
Primary	1	On-premises 1 & 2, AWS, Azure
Node	4	On-premises 1 & 2, AWS, Azure
NFS (optional)	1	On-premises 1 & 2

Note: Minimum Recommended version 3.7

BIG-IP Specifications

ROLES	COUNT	LOCATION(S)
LTM	3	On-premises, Azure, AWS
DNS / GSLB	1	On-premises

Note: recommended ver. 12.x or higher and Requires Best or equivalent license that enables SMB, DNS, and iRules ILX (optional.) For further information on licensing follow this link [Simplified Licensing](#)

AZURE CLUSTER

It's recommended that OpenShift is set up in the Azure cloud as per the documentation provided by Red Hat and Microsoft.

- [Browse the OpenShift and Azure setup guides](#)

For BIG-IP, use the template: F5 BIG-IP ADC+SEC BEST—BYOL, 1 Public IP serving both NATed MGMT and one Self IP

AWS CLUSTER

OpenShift could be set up in AWS in any number of ways. It's recommended that you follow the recommendation of Red Hat and AWS for using the cloud formation template described in this document.

- [OpenShift Container Platform on the AWS Cloud: Read the Quick Start Guide](#)

For BIG-IP Use the template: F5 BIG-IP ADC+SEC BEST—BYOL, 1 Public IP serving both NATed MGMT and one Self IP

DEPLOYMENT AND EXECUTION

This deployment is a high-level overview of the steps from start to finish. This is a very simplistic view of the design to give you a sense of what is required. Because this architecture is continually evolving, the full setup detail will be published to the web and made available, so the community can update and contribute to the project. Please see the reference links at the bottom of this section for additional detail and guides for each component.

Infrastructure Setup Note: All systems should be provisioned using their vendor's recommended requirements. This architecture assumes you have a basic knowledge of how to set up BIG-IP, Ansible Tower, and RedHat OpenShift.

HIGH-LEVEL SETUP STEPS

Please note that this part of the document is for setting up systems only. Automation for the multi-cloud demo is almost entirely done via the Ansible playbooks provided in the links at the end of this document. At a high level, all you'll need is OpenShift with a demo app of some kind, a licensed BIG-IP, and a single Ansible Tower server.

Note: The details of how we set up and configured the various parts of the infrastructure along with tips and tricks to make setup a bit easier can be found [here](#)

1. Build Ansible Tower Server
 - a. 1 Ansible Tower server ver. 2.4.2.0 or greater
2. On-premises
 - a. Build an OpenShift Cluster
 - i. 1 Main Cluster
 - b. 3 BIG-IPs VEs with licenses
 - i. 1 LTM per cluster for a total of 2
 - ii. 1 DNS with GSLB
3. Set up Aspen Mesh
4. Azure Cloud
 - a. Build an OpenShift Cluster
 - i. 1 Main Cluster
 - b. Provision one BIG-IP System
 - i. 1 BIG-IP System from an Azure template
5. AWS Cloud
 - a. Build an OpenShift Cluster
 - i. 1 Main Cluster
 - b. Provision one BIG-IP System
 - i. 1 BIG-IP System from an AWS template

ANSIBLE TOWER PREPARATIONS AND SETUP

Follow the documentation provided by Ansible to complete this part of setup. You only need to do this once.

- [Ansible Tower Quick Installation Guide](#)
- [Get setup guides for multiple use cases](#)

OPENSIFT PREPARATIONS AND SETUP

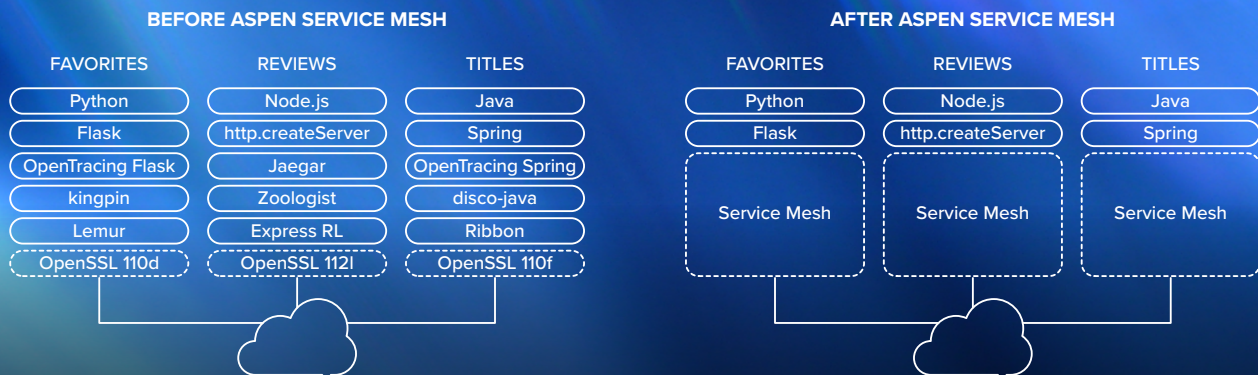
- [Follow the documentation provided by Red Hat](#) to complete this part of the setup, per cluster.

ASPEN MESH PREPARATIONS AND SETUP

Follow the documentation provided by Aspen Mesh to complete this part of the setup per OpenShift cluster.

The diagram shows the advantages of implementing Istio with Aspen Mesh in your OpenShift environment.

- [Try Aspen Mesh for free](#)



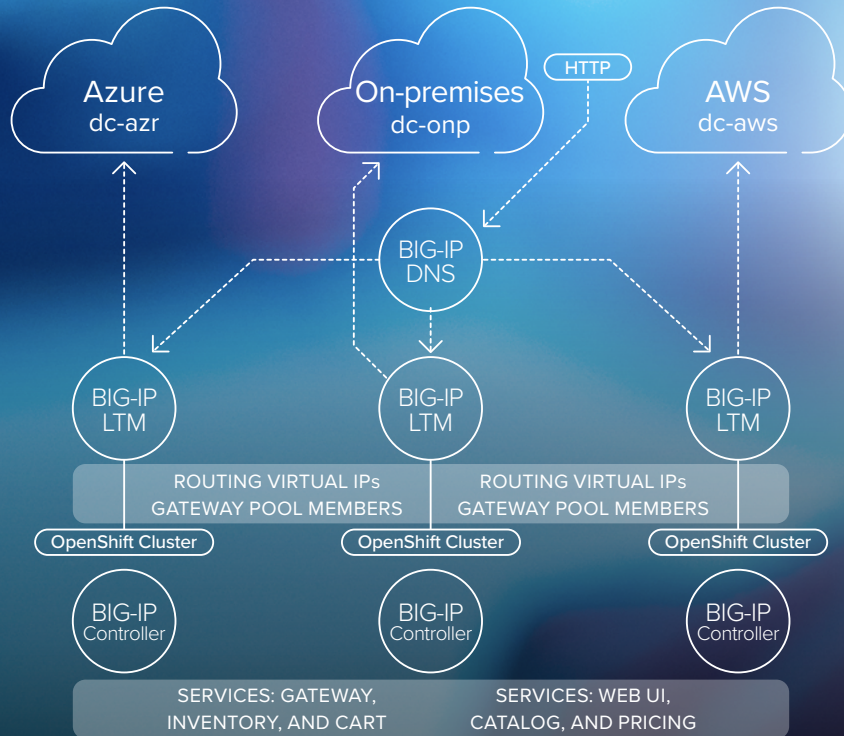
BIG-IP PREPARATIONS AND SETUP

Follow the instructions provided by F5 at the link below to stand up the BIG-IP systems required per cloud environment.

- [Post installation for BIG-IP LTM and DNS setup guide](#)

DEMO APPLICATION(S)

Feel free to choose any demo application you want to set up this architecture. To take advantage of some of the more advanced capabilities, it's recommended you use an application that makes use of several microservices. See below to understand which applications we picked and why, along with a diagram of where they live in relation to the overall architecture. Requirements for this design were to use of-the-shelf apps, ensuring that we didn't alter them in any way.



APPLICATIONS USED

As previously stated we had very specific requirements for the demo applications we chose. They needed to be truly off-the-shelf, cloned from GitHub, and deployed to OpenShift with no modifications to showcase what was possible with very little work. The two we picked fit the bill for demonstration purposes.

Red Hat Cool Store

Red Hat Cool store is an eCommerce demo application that showcases several Red Hat technologies and takes full advantage of several microservices. It's fully open sourced and deploys directly into OpenShift with a simple YAML payload at the command line.

- [GitHub Repo](#)

Istio Book Info

Istio's Book Info application was specifically designed to showcase the capabilities of the Istio platform on which Aspen Mesh is built. It has a simple microservices implementation with a light weight YAML payload, and it's perfect to show the basics of what Aspen Mesh can do.

- [GitHub Repo](#)

FUTURE

The F5 multi-cloud architecture is a living design. It will continue to evolve as time goes on and technologies change and advance. The team working on this has several upcoming changes:

- Google Cloud Platform implementation
- F5-designed demo application
- Aspen Mesh platform enhancements
- Full CI/CD pipeline

These changes will further demonstrate the power and versatility of building not only a multi-cloud architecture, but also an application that's designed from the ground up to take advantage of the advanced capabilities of F5 BIG-IP, Ansible Tower, and Red Hat OpenShift Container Platform.

CONCLUSION

Multi-Cloud is a challenging architecture to deliver. We've covered one possible path to build one out using this joint solution from Red Hat and F5. If you would like to learn more, please visit the links below or contact your territory account team to set up a call.

- github.com/f5devcentral/f5-bd-mcd
- clouddocs.f5.com/containers/v2/OpenShift/
- github.com/f5networks/k8s-bigip-ctlr
- github.com/f5devcentral/f5-tmsh2iapp
- github.com/aspenmesh
- <https://www.youtube.com/watch?v=3Etglg2lrGo&t=4s>

Learn more about the [F5](#) and [Red Hat](#) partnership



US Headquarters: 401 Elliott Ave W, Seattle, WA 98119 | 888-882-4447 // Americas: info@f5.com // Asia-Pacific: apacinfo@f5.com // Europe/Middle East/Africa: emeainfo@f5.com // Japan: f5j-info@f5.com

©2018 F5 Networks, Inc. All rights reserved. F5, F5 Networks, and the F5 logo are trademarks of F5 Networks, Inc. in the U.S. and in certain other countries. Other F5 trademarks are identified at f5.com. Any other products, services, or company names referenced herein may be trademarks of their respective owners with no endorsement or affiliation, expressed or implied, claimed by F5. GUIDE-CLOUD-554307498 | 11.20