15 October 2020

Programmability 101 - Introduction to BIG-IP REST Structure and Concepts

PRESENTED BY:

Saad Khalid, Solutions Engineer, F5

In this lab we will…

- Review Automation & Orchestration Concepts
- Learn Basics of REST
- Module 1: Imperative Automation via the BIG-IP REST API
- Explore Declarative APIs and AS3
- Module 2: Abstracting Services using the App Services 3 Extension

Elevate your knowledge of automation basics so you can support and drive modern workflows in your organization

Automation & Orchestration Concepts

Automation and Orchestration

- Automation is about codifying tasks.
- Orchestration is about codifying processes.
- Orchestration takes advantage of automation by reusing these basic building blocks.

Why Organizations Use Automation Frameworks

REDUCE OPEX

SCALING TO MEET DEMAND

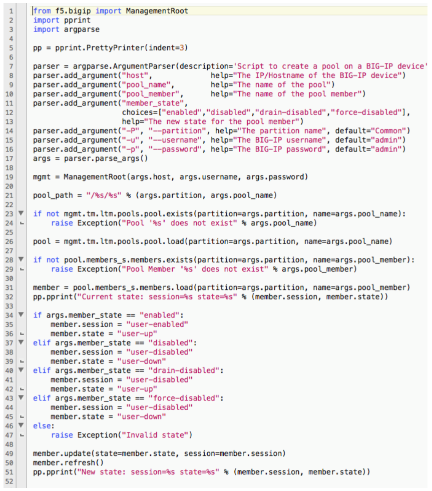
TIME TO MARKET

Automation makes your life easy!

- Use it!
- Saves a valuable resource—*time*—allowing you to focus on more important tasks!

BIG-IP Platform

+

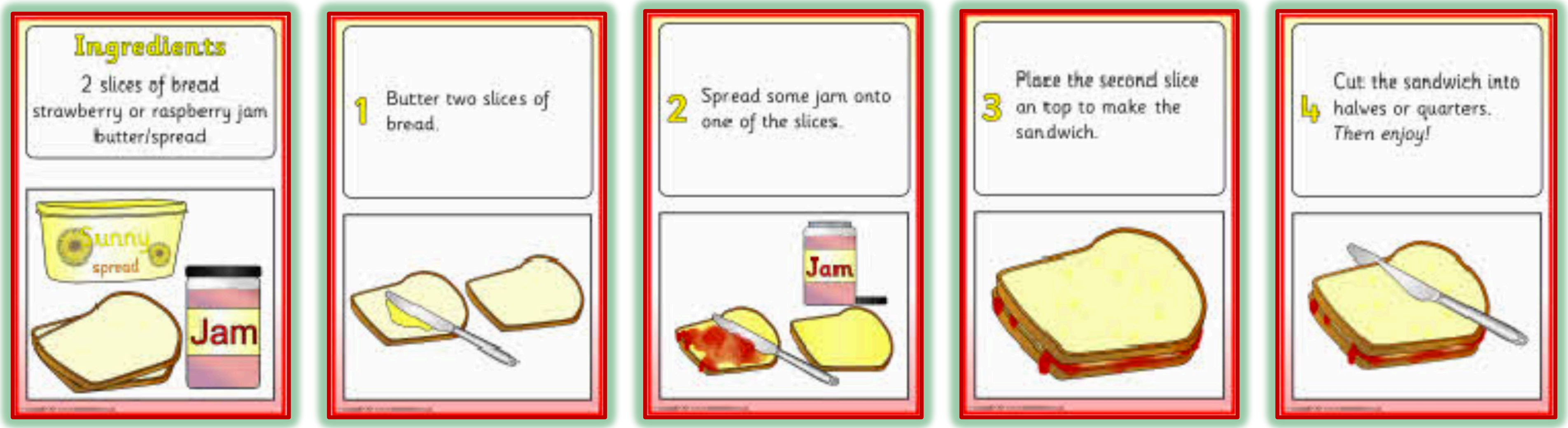


```
from f5.bigip import ManagementRoot
import pprint
import argparse

pp = pprint.PrettyPrinter(indent=3)

parser = argparse.ArgumentParser(description='Script to create a pool on a BIG-IP device'
parser.add_argument("host",              help="The IP/Hostname of the BIG-IP device")
parser.add_argument("pool_name",         help="The name of the pool")
parser.add_argument("pool_member",       help="The name of the pool member")
parser.add_argument("member_state",
                    choices=["enabled","disabled","drain-disabled","force-disabled"],
                    help="The new state for the pool member")
parser.add_argument("-P", "--partition", help="The partition name", default="Common")
parser.add_argument("-u", "--username", help="The BIG-IP username", default="admin")
parser.add_argument("-p", "--password", help="The BIG-IP password", default="admin")
args = parser.parse_args()

mgmt = ManagementRoot(args.host, args.username, args.password)

pool_path = "/%s/%s" % (args.partition, args.pool_name)

if not mgmt.tm.ltm.pools.pool.exists(partition=args.partition, name=args.pool_name):
    raise Exception("Pool '%s' does not exist" % args.pool_name)

pool = mgmt.tm.ltm.pools.pool.load(partition=args.partition, name=args.pool_name)

if not pool.members_s.members.exists(partition=args.partition, name=args.pool_member):
    raise Exception("Pool Member '%s' does not exist" % args.pool_member)

member = pool.members_s.members.load(partition=args.partition, name=args.pool_member)
pp.pprint("Current state: session=%s state=%s" % (member.session, member.state))

if args.member_state == "enabled":
    member.session = "user-enabled"
    member.state = "user-up"
elif args.member_state == "disabled":
    member.session = "user-disabled"
    member.state = "user-down"
elif args.member_state == "drain-disabled":
    member.session = "user-disabled"
    member.state = "user-up"
elif args.member_state == "force-disabled":
    member.session = "user-disabled"
    member.state = "user-down"
else:
    raise Exception("Invalid state")

member.update(state=member.state, session=member.session)
member.refresh()
pp.pprint("New state: session=%s state=%s" % (member.session, member.state))
```

=

Understanding imperative vs. declarative

Imperative – What we’ve done for years (scripting, iRules, etc.) Imperative methodology implies that you define the flow of an operation implicitly. It also implies that domain-specific knowledge is required to interact with the system.



Ingredients

2 slices of bread
strawberry or raspberry jam
butter/spread

1 Butter two slices of bread.

2 Spread some jam onto one of the slices.

3 Place the second slice on top to make the sandwich.

4 Cut the sandwich into halves or quarters. *Then enjoy!*

Understanding imperative vs. declarative

Declarative – What we're evolving to. Declarative methodology implies that you define the desired outcome and depend on underlying mechanisms to deliver that outcome. This methodology tries to reduce or eliminate the need for domain specific knowledge.

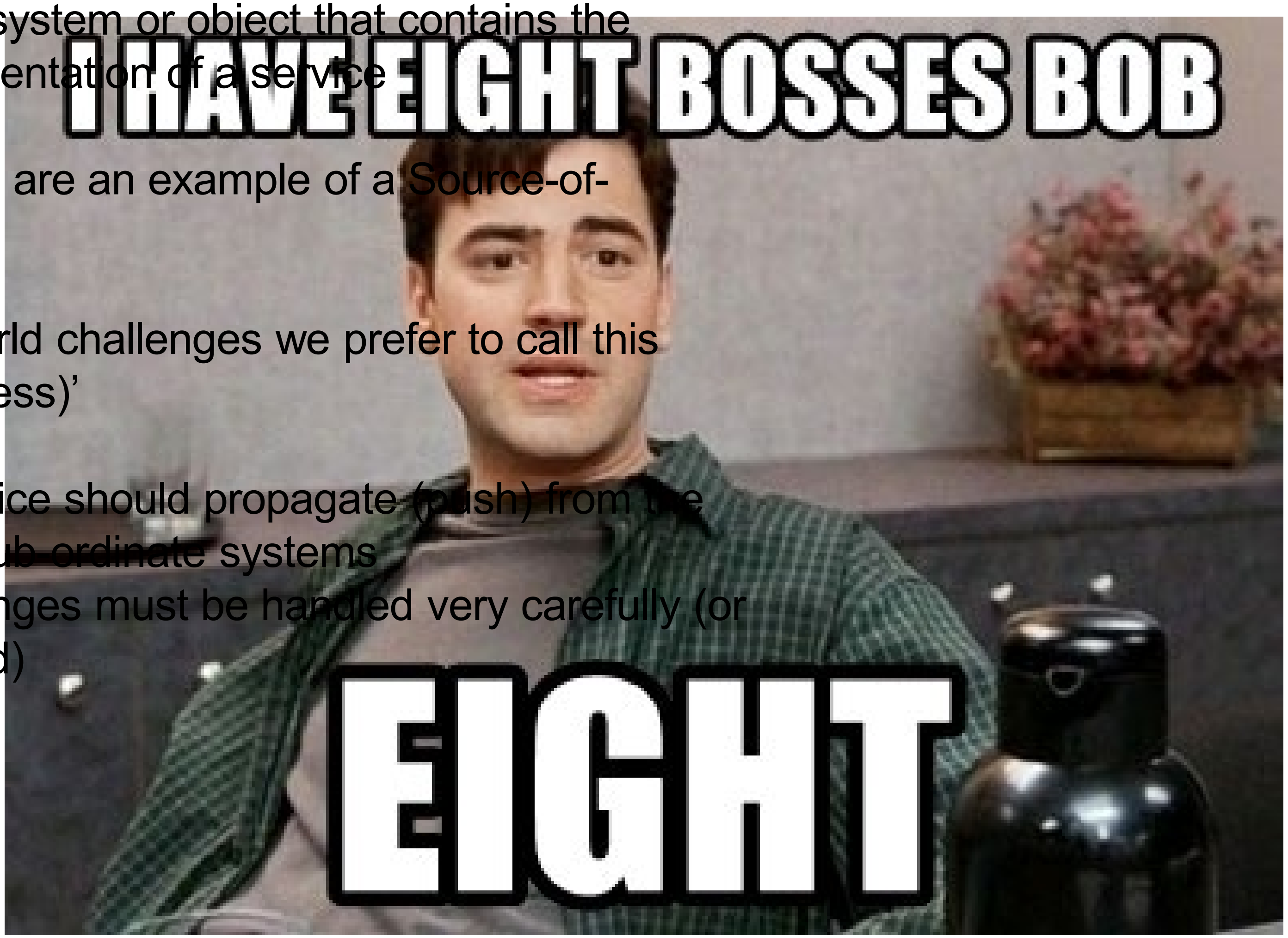
Concepts – Source of Truth(iness)

- Source of Truth: A system or object that contains the authoritative representation of a service

- GitHub repositories are an example of a Source-of-Truthiness

- To address real-world challenges we prefer to call this 'Source-of-Truth(iness)'

- Changes for a service should propagate (push) from the source of truth to sub-ordinate systems
  - Out-of-band changes must be handled very carefully (or be totally avoided)



Introduction to REST

Introduction to REST

- Based on HTTP and JSON
- Uses HTTP methods (GET, POST, PUT, PATCH, DELETE)
- Data is sent using the Javascript Object Notation format

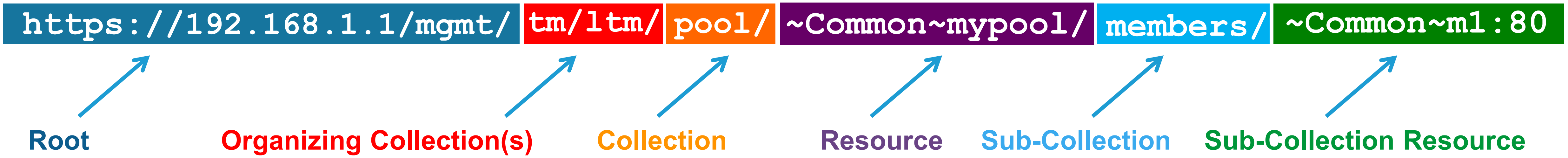
```
{
  “attribute1”:”value1”,
  “attribute2”:[“array”,”of”,”values”],
  “attribute3”: [ { “nested1”:”value1”, “nested2”:”value2” }, {“nested3”:”value3”} ]
}
```

REST APIs and HTTP Verbs

- **What action do I want to perform?**
- HTTP methods (verbs) are used to create, read, update, and delete (CRUD) resources
- APIs must use HTTP verbs in a manner described in the table below

URI	POST	GET	PUT	DELETE	PATCH
Collection	Create resources.	Get representation of all resources in the collection.	Fully update all resources in a collection.	Delete all resources in a collection.	Partially update all resources in a collection.
Resource	Used for non-idempotent controller resources.	Get a resource's representation.	Fully update the resource if it exists.	Delete a resource.	Partially update a resource.

Anatomy of a REST URI



`https://192.168.1.1/mgmt/` `tm/ltm/` `pool/` `~Common~mypool/` `members/` `~Common~m1:80`

Root

Organizing Collection(s)

Collection

Resource

Sub-Collection

Sub-Collection Resource

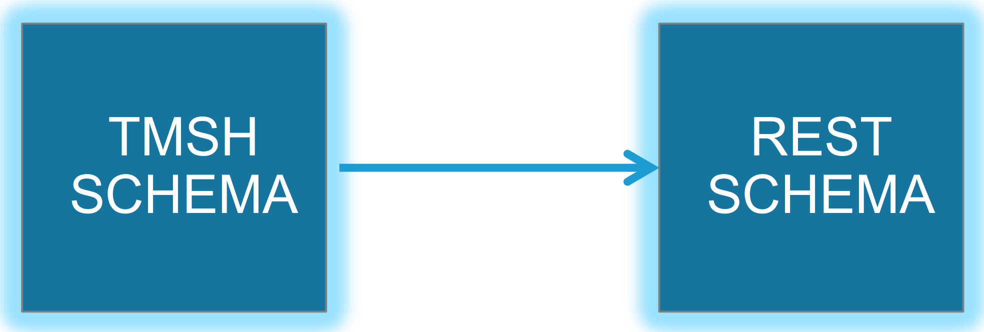
NOTE: Resource names map ‘~’ to ‘/’ (e.g. ~Common~mypool is really /Common/mypool)

Response Codes

- **What was the result of my action?**
- APIs must make use of HTTP response codes where appropriate
- The following table describes the required success response codes

Response Code	Applicable Verbs	Notes
200 OK	• All	Return on most positive responses including DELETE.
201 Created	• POST	HTTP Location header contains link to newly created resource.
202 Accepted	• POST • PUT • PATCH • DELETE	Return when a request will take a long time; server should return a Location header for client to get state updates.
404	• GET	The resource does not exist.
500	• All	Check /var/log/restjavad.0.log

How the REST API is Implemented on TMOS



REST API attributes are derived from TMSH schema

Generally, if the attribute/option is available in TMSH it’s available in REST

list ltm pool pool1 members {10.1.20.1:80}

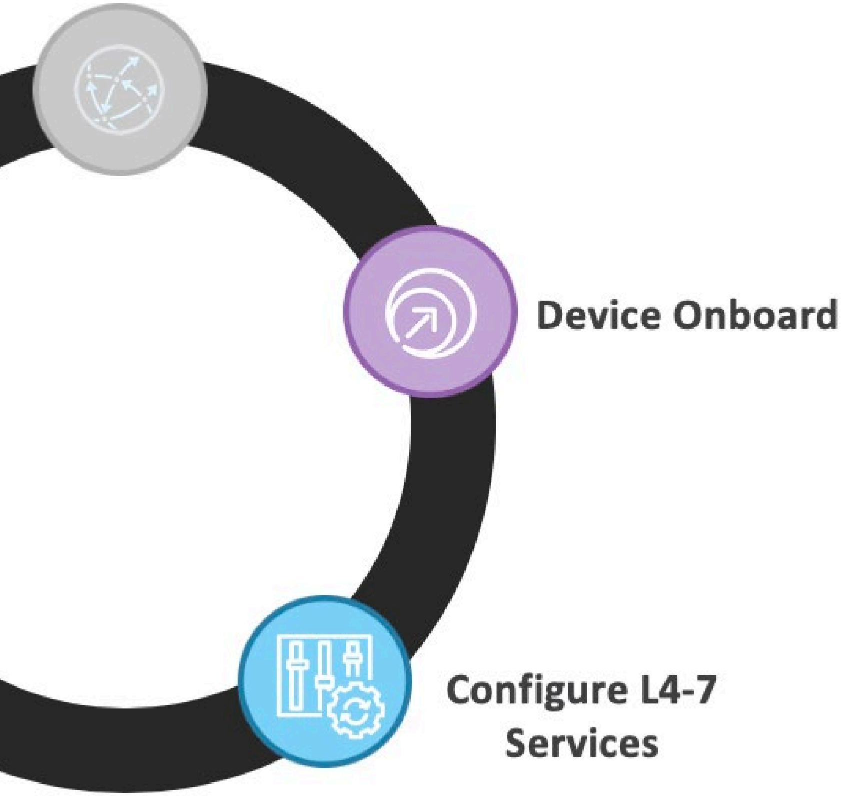
GET: https://10.1.1.245/mgmt/tm/ltm/pool/pool1/members/~Common~10.1.20.1:80

Got it? Now let's…

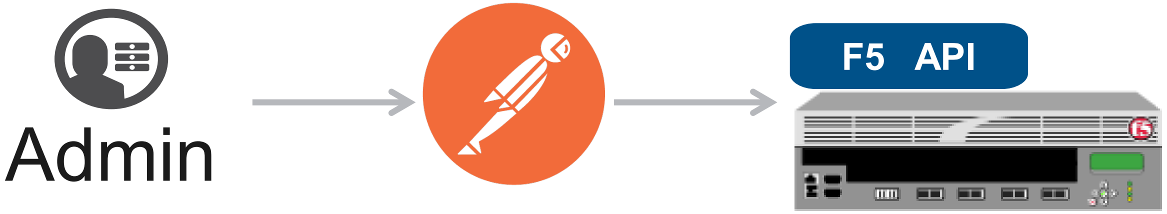
UDF Status

Module 1

Imperative Automation via the BIG-IP REST API



1.1: Exploring the iControl REST API
1.2: REST API Authentication & example Templates
1.3: Review/Set Device Settings
1.4: Basic Network Connectivity
1.5: Build a BIG-IP Cluster using a Collection
1.6: Build a Basic LTM Config using REST Transactions



Imperative Automation via the BIG-IP REST API

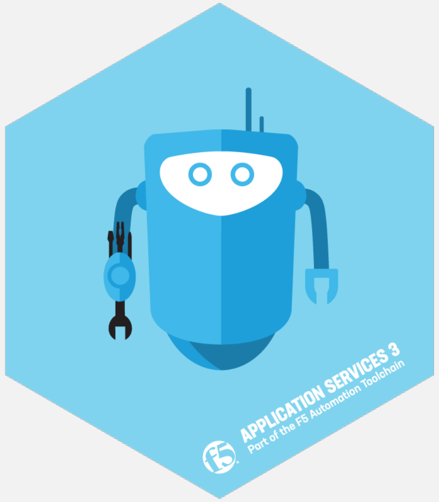
AS3 - Declarative API

F5 Automation Toolchain

Bootstrap	Onboard	Deploy App Services	Monitoring/Telemetry
CLOUD TEMPLATES	**DECLARATIVE ONBOARDING EXTENSION**	**APP SERVICES 3 EXTENSION**	**TELEMETRY STREAMING EXTENSION**
Start BIG-IP Instances in Public & Private Clouds	Initial Config of BIG-IP Instances	Deploy Classic and Advanced Application Services on BIG-IP using Declarative REST APIs	Stream Telemetry, Events & Logs from BIG-IP to various Analytics and Logging solutions
	L1-L3	**L4-L7**	

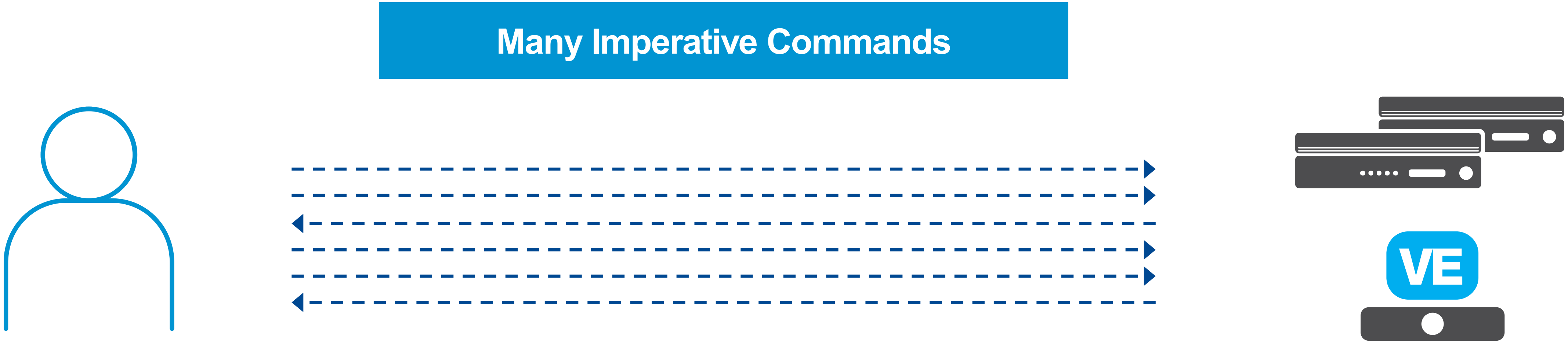
Simplify Automation with AS3

- Application Services 3 Extension – BIG-IP API extension that accepts a declarative API call
- Configures BIG-IP L4-7 services
- Minimizes need for BIG-IP domain specific knowledge
- Minimizes deployment errors
- Makes it easy to integrate F5 automation into orchestration systems
- Runs on BIG-IP, on BIG-IQ or in a container
- Create application dashboards / monitoring / alerts when used with BIG-IQ



Without AS3

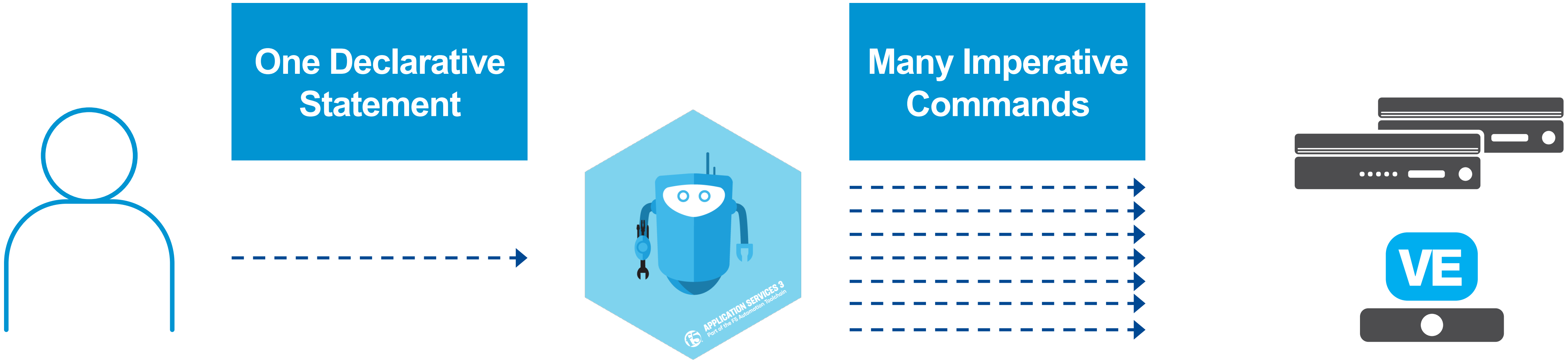
Configuring BIG-IP requires many REST API calls



- Requires BIG-IP domain expertise
- Dozens of REST API calls
- Costly to automate and integrate with orchestration systems
- Time-consuming
- Error-prone

With AS3

Configuring BIG-IP requires a SINGLE REST API call

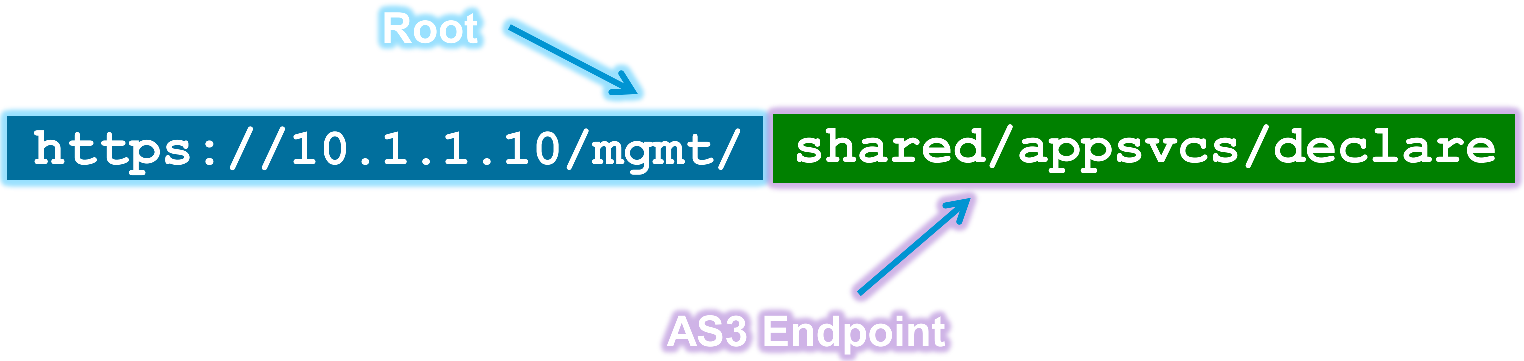


One Declarative Statement

Many Imperative Commands

- AS3 abstracts away all BIG-IP configuration complexity
- User only needs to define the desired configuration end-state
- Requires no BIG-IP domain expertise
- Single REST API call simplifies automation and orchestrator integrations
- Accelerates app service deployment
- Declaration is reusable, ensures consistency, reduces errors

AS3 API Endpoint



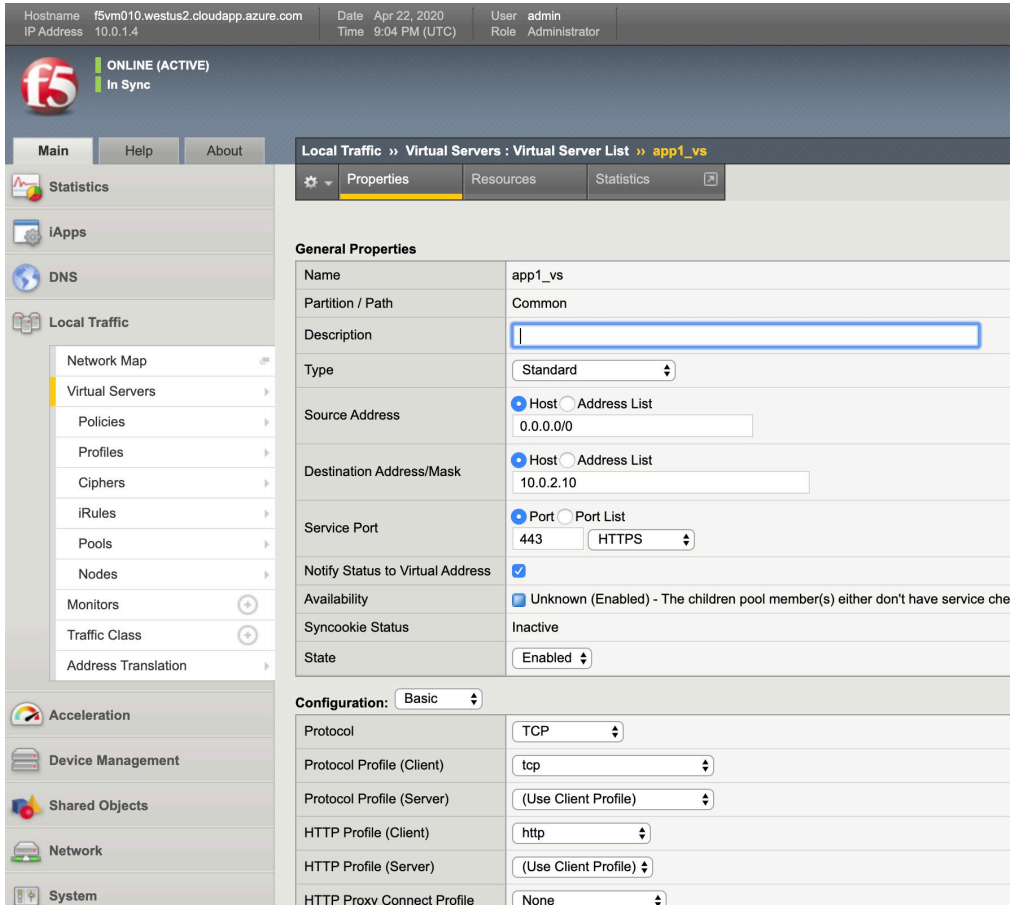
Root

https://10.1.1.10/mgmt/

shared/appsvcs/declare

AS3 Endpoint

Enabling Automation & Orchestration



Hostname f5vm010.westus2.cloudapp.azure.com Date Apr 22, 2020 User admin
IP Address 10.0.1.4 Time 9:04 PM (UTC) Role Administrator

ONLINE (ACTIVE)
In Sync

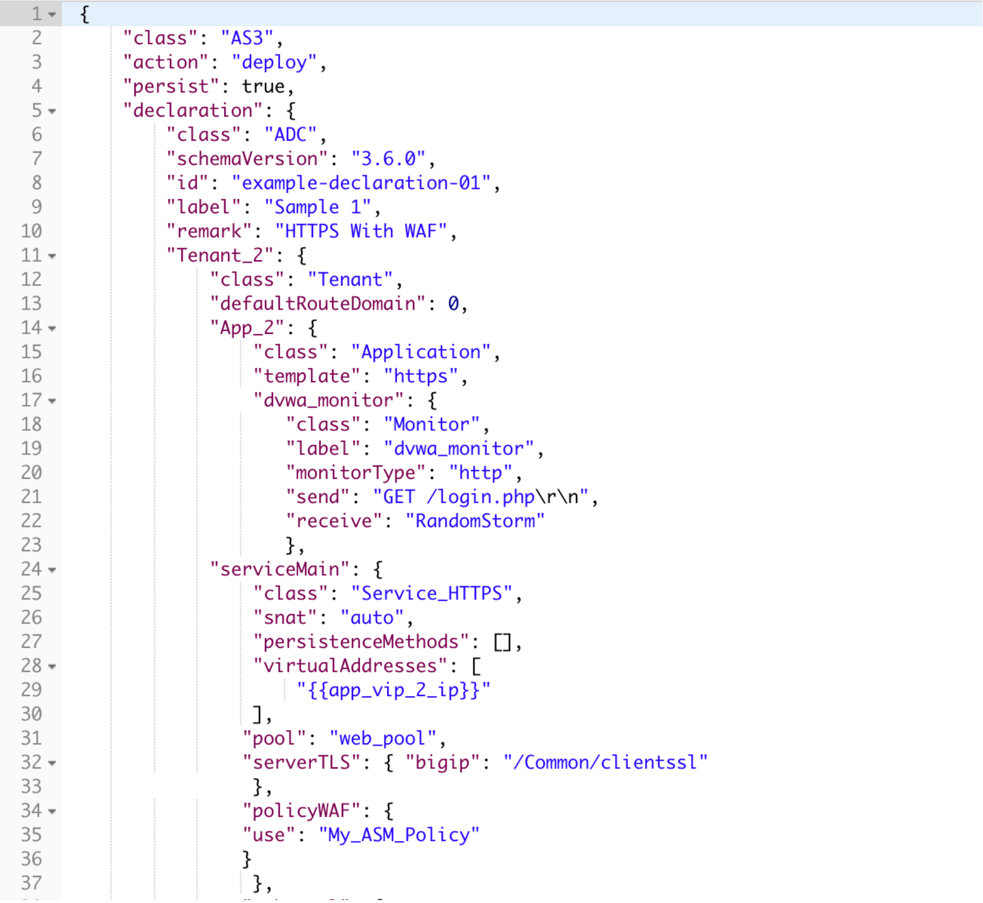
Local Traffic » Virtual Servers : Virtual Server List » app1_vs

General Properties

Name	app1_vs
Partition / Path	Common
Description	
Type	Standard
Source Address	● Host ○ Address List 0.0.0.0/0
Destination Address/Mask	● Host ○ Address List 10.0.2.10
Service Port	● Port ○ Port List 443 HTTPS
Notify Status to Virtual Address	☑
Availability	Unknown (Enabled) - The children pool member(s) either don't have service che
Syncookie Status	Inactive
State	Enabled

Configuration: Basic

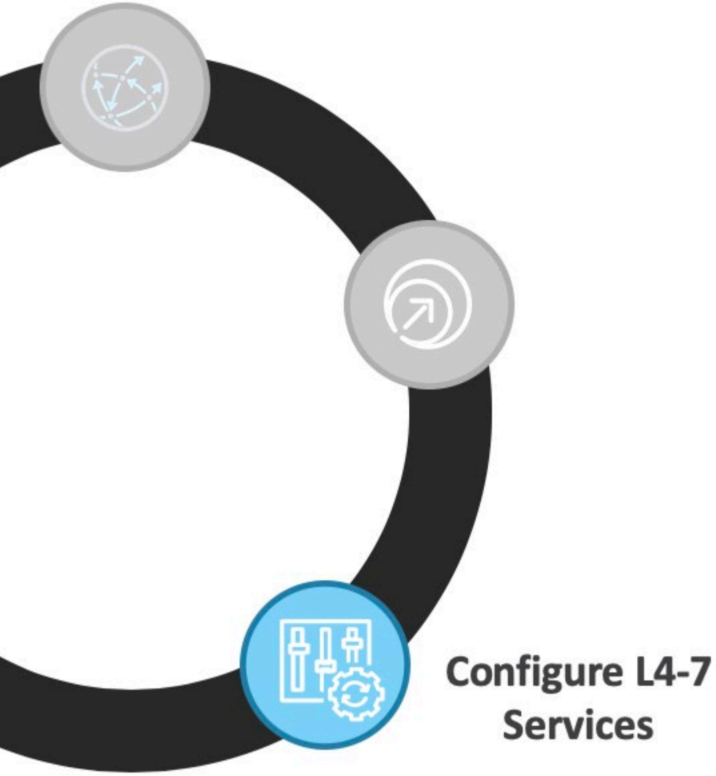
Protocol	TCP
Protocol Profile (Client)	tcp
Protocol Profile (Server)	(Use Client Profile)
HTTP Profile (Client)	http
HTTP Profile (Server)	(Use Client Profile)
HTTP Proxy Connect Profile	None



```
{
  "class": "AS3",
  "action": "deploy",
  "persist": true,
  "declaration": {
    "class": "ADC",
    "schemaVersion": "3.6.0",
    "id": "example-declaration-01",
    "label": "Sample 1",
    "remark": "HTTPS With WAF",
    "Tenant_2": {
      "class": "Tenant",
      "defaultRouteDomain": 0,
      "App_2": {
        "class": "Application",
        "template": "https",
        "dvwa_monitor": {
          "class": "Monitor",
          "label": "dvwa_monitor",
          "monitorType": "http",
          "send": "GET /login.php\r\n",
          "receive": "RandomStorm"
          },
      "serviceMain": {
        "class": "Service_HTTPS",
        "snat": "auto",
        "persistenceMethods": [],
        "virtualAddresses": [
          "{{app_vip_2_ip}}"
        ],
       "pool": "web_pool",
       "serverTLS": { "bigip": "/Common/clientssl"
        },
       "policyWAF": {
       "use": "My_ASM_Policy"
       }
        },
```

Module 2

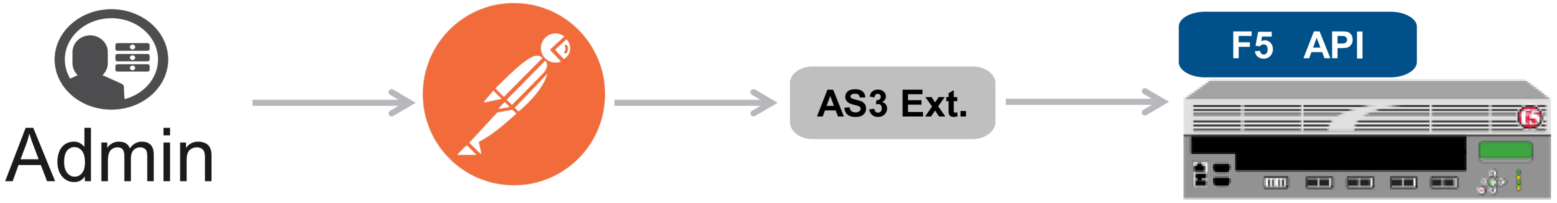
Abstracting Services using the App Services 3 Extension



2.1: Exploring AS3

2.2: Install the AS3 Extension

2.3: Application Service Deployments with AS3



Conclusion

In this lab we…

- Learned the basic concepts required to interact with the BIG-IP iControl REST API
- Walked through a typical Device Onboarding workflow and deployed a fully functional BIG-IP Active/Standby pair
- Deployed an application using the Imperative approach to Automation
- Learned about and deployed an application with F5’s Declarative interface App Services 3 Extension (AS3)

Elevated your knowledge of automation basics so you can support and drive modern workflows in your organization

Thank You

AGILITY

F5 DoD Virtual Enablement Schedule – Fall Sessions

Date	F5 DoD Virtual Enablement Schedule – Q1	Registration Link
Oct 15 - 9-11AM PST	BIG-IQ	https://www.cvent.com/d/07q36p
Oct 20 - 9-11AM PST	A&O Tool Chain 111: BIG-IP Deployments with Declarative Onboarding (DO)	https://www.cvent.com/d/37qpc6
Oct 22 - 9-11AM PST	F5 + Ansible - Basic Workshop	https://www.cvent.com/d/y7q361
Oct 27 – 11-1:30PM EST	F5 Certification Bootcamp – 101 Certification	https://www.f5.com/c/amer-2020/virtual-meetup/channel-certification-prep-Oct?utm_medium=email&utm_source=f5sv&utm_campaign=amer-operational&utm_content=vm
Oct 29 – 1-3PM EST	NGINX DOD Workshop	https://carahevents.carahsoft.com/Event/Details/181477-f5
Oct 29 – 11-5PM EST	F5 Certification Bootcamp – 201 Certification	https://www.f5.com/c/amer-2020/virtual-meetup/channel-certification-prep-Oct?utm_medium=email&utm_source=f5sv&utm_campaign=amer-operational&utm_content=vm
Nov 4 - 9-11AM PST	WAF 111: Protecting Against the OWASP Top 10	https://www.cvent.com/d/47qpch
Nov 5 - 9-11AM PST	BIG-IQ	https://www.cvent.com/d/07q36p
Nov 10 - 9-11AM PST	F5 + Ansible - Advanced Workshop	https://www.cvent.com/d/y7q361
Nov 10 - 9-2:30 PST	F5 Security Summit	https://www.f5.com/c/amer-2020/event/cec-security-summitNov10utm_medium=email&utm_source=f5sv&utm_campaign=amer-ap_ap&utm_content=ev
Nov 17 - 9-11AM PST	BIG-IQ	https://www.cvent.com/d/07q36p
Nov 18 - 9-11AM PST	WAF 141: Getting Started with WAF, Bot Defense & Threat Campaigns	https://www.cvent.com/d/w7qpcy
Dec 1 - 9-11AM PST	BIG-IQ	https://www.cvent.com/d/07q36p
Dec 2 - 9-11AM PST	SSLO 101: Essential SSL Visibility with SSL Orchestrator	https://www.cvent.com/d/57qpc6
Dec 15 - 9-11AM PST	BIG-IQ	https://www.cvent.com/d/07q36p