Automation Concepts

General Terminology

Automation

Task(s) or function(s) that reduce human intervention in processes, mainly by predetermining decision criteria, subprocess relationships, and related actions

Orchestration

Multiple automated deployments done in a specific way, resulting in an optimized and consolidated *process* or *workflow*

Infrastructure as code (IaC)

The process of *managing* and *provisioning* computer data center resources through *machine-readable definition files*, rather than physical hardware configuration or interactive configuration tools

RESTful API

API

noun *Computing*

a set of functions and procedures that allow the creation of applications which access the features or data of an operating system, application, or **other** service.

RESTful API

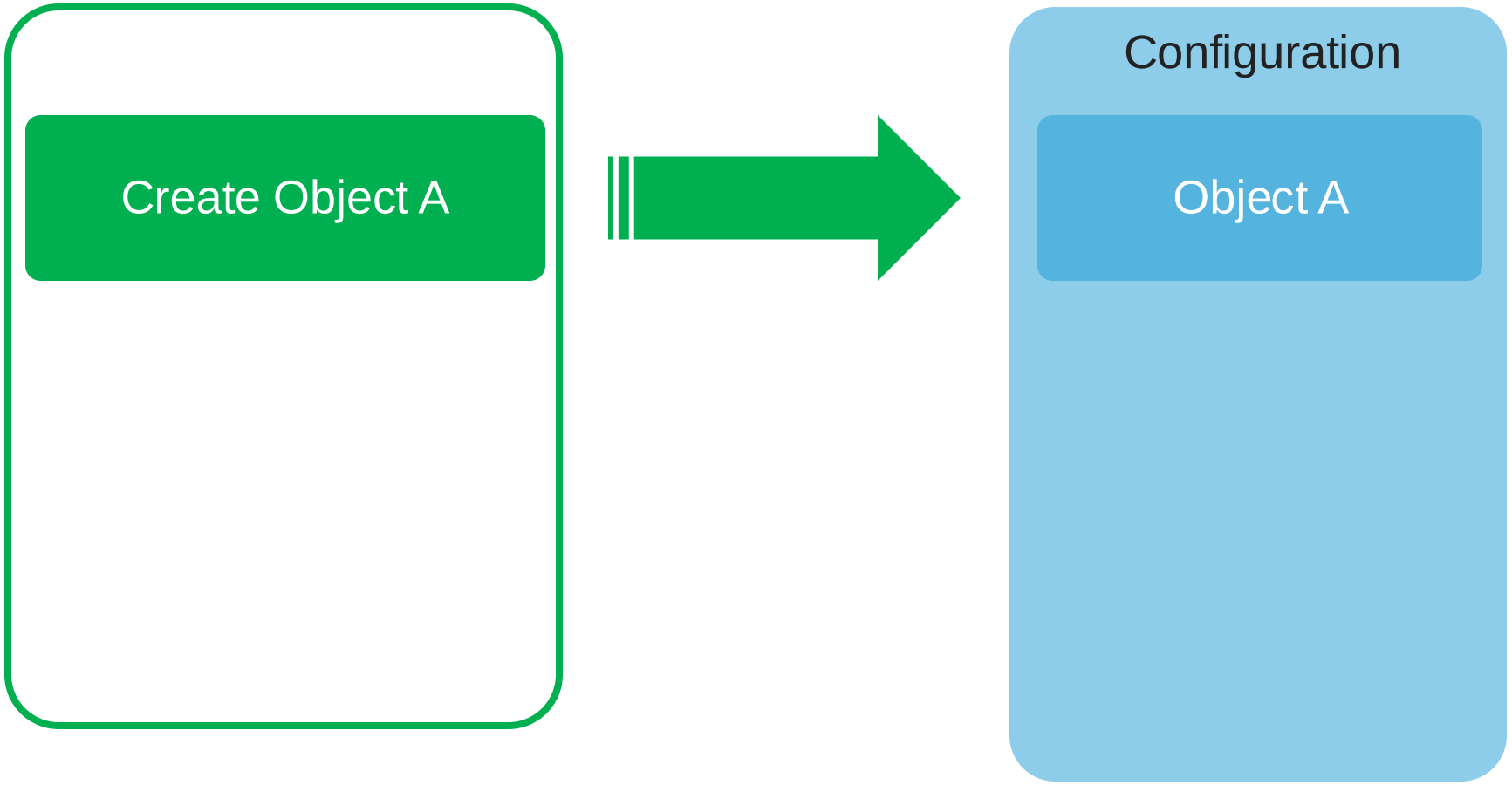
A Web API (or Web Service) conforming to the REST architectural style is called a REST API (or RESTful API).

REST API uses HTTP URIs and Methods (POST, GET, PUT DELETE) to Create, Read, Update, and Delete (CRUD)

GET to obtain /some/resource
POST to modify /some/other/resource

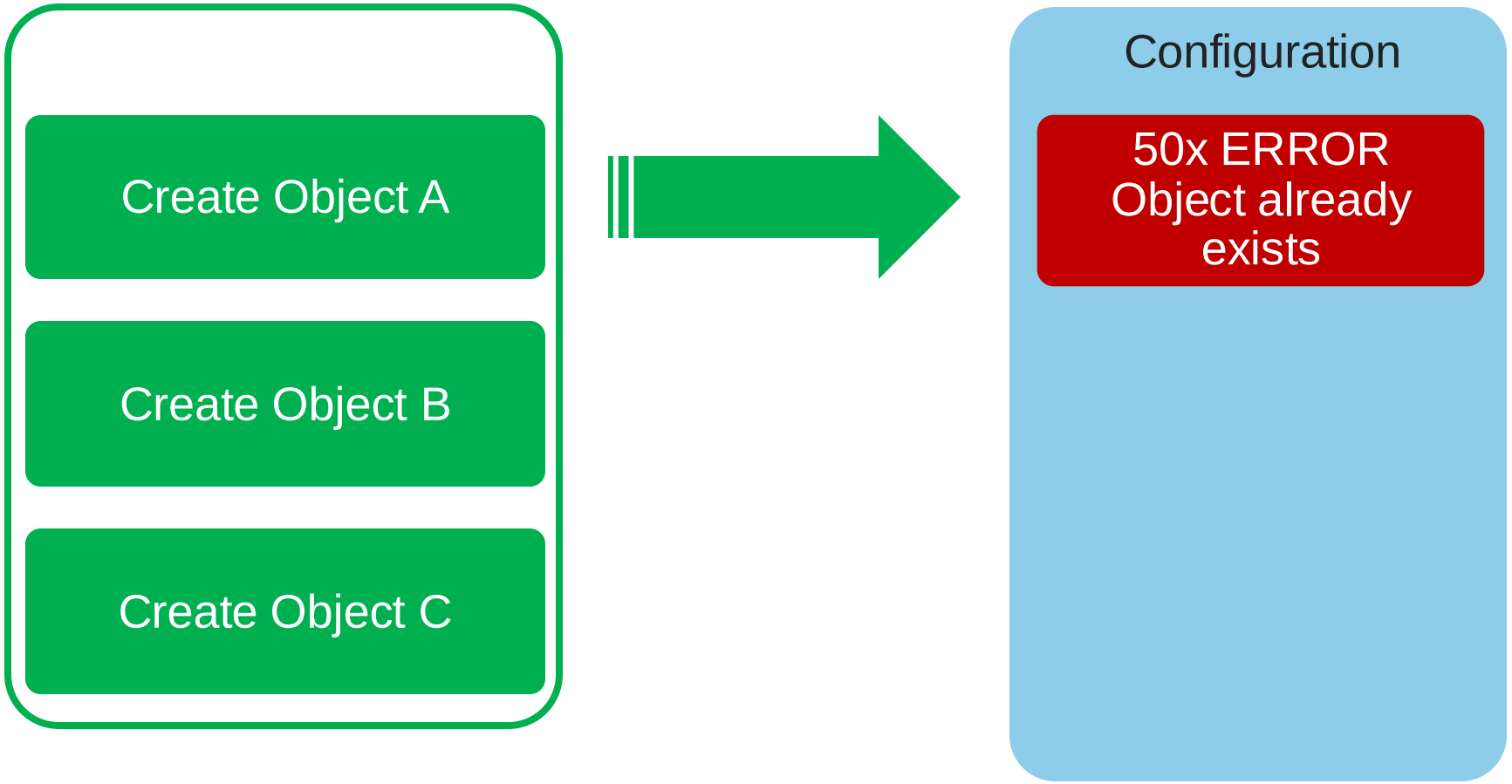
Non-Idempotent Operations

Multiple identical requests can produce errors…



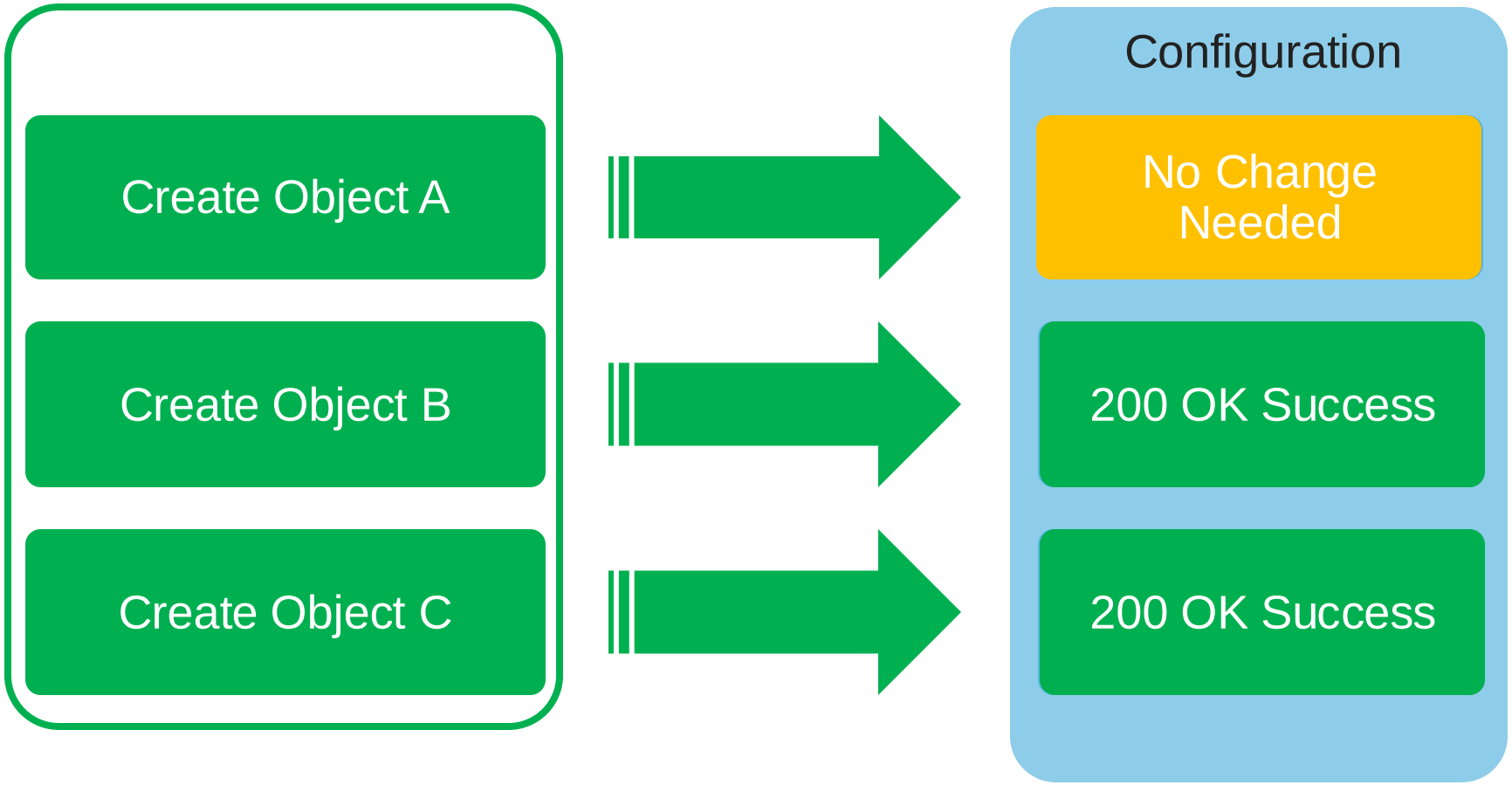
Non-Idempotent Operations

Multiple identical requests can produce errors…



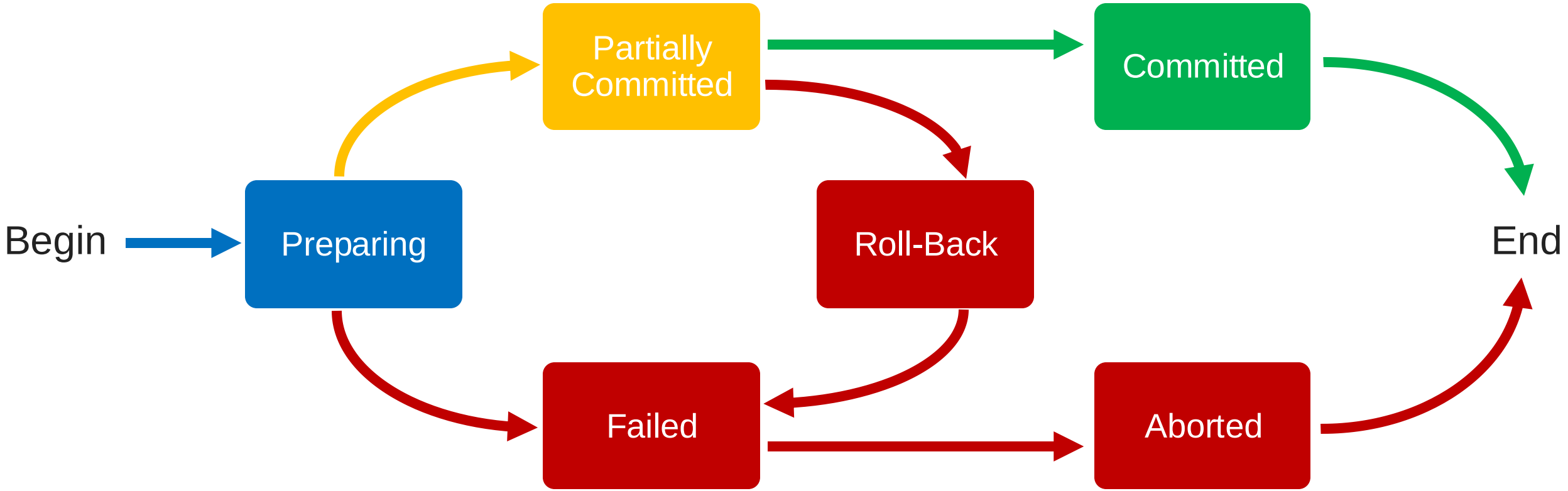
Idempotent Operations

Multiple identical requests are handled sensibly…



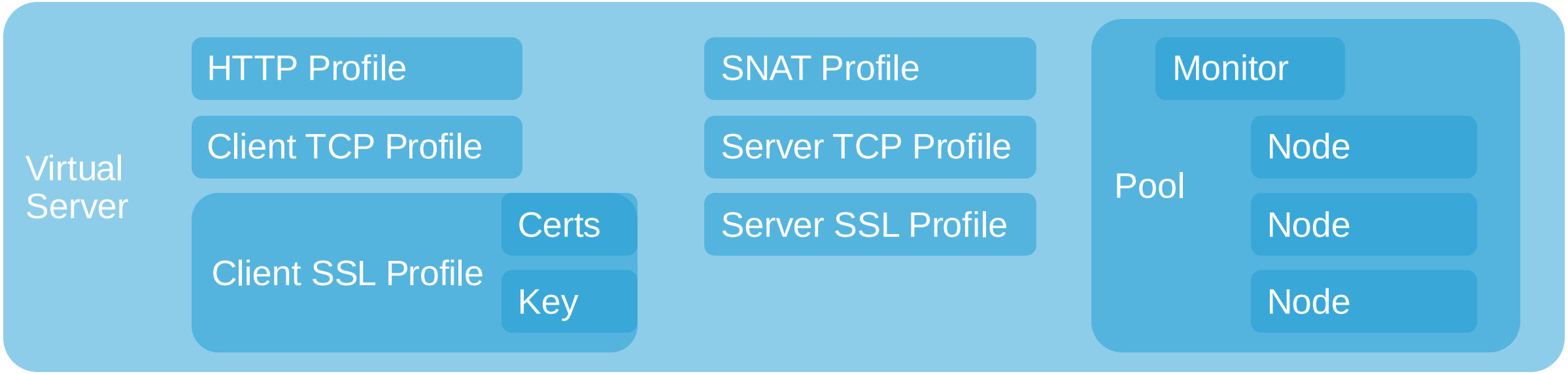
Atomic Transactions

Either all occur, or none occur



Imperative operations

Domain specific knowledge required to deploy a…



Declarative Operations

Easy to deploy…

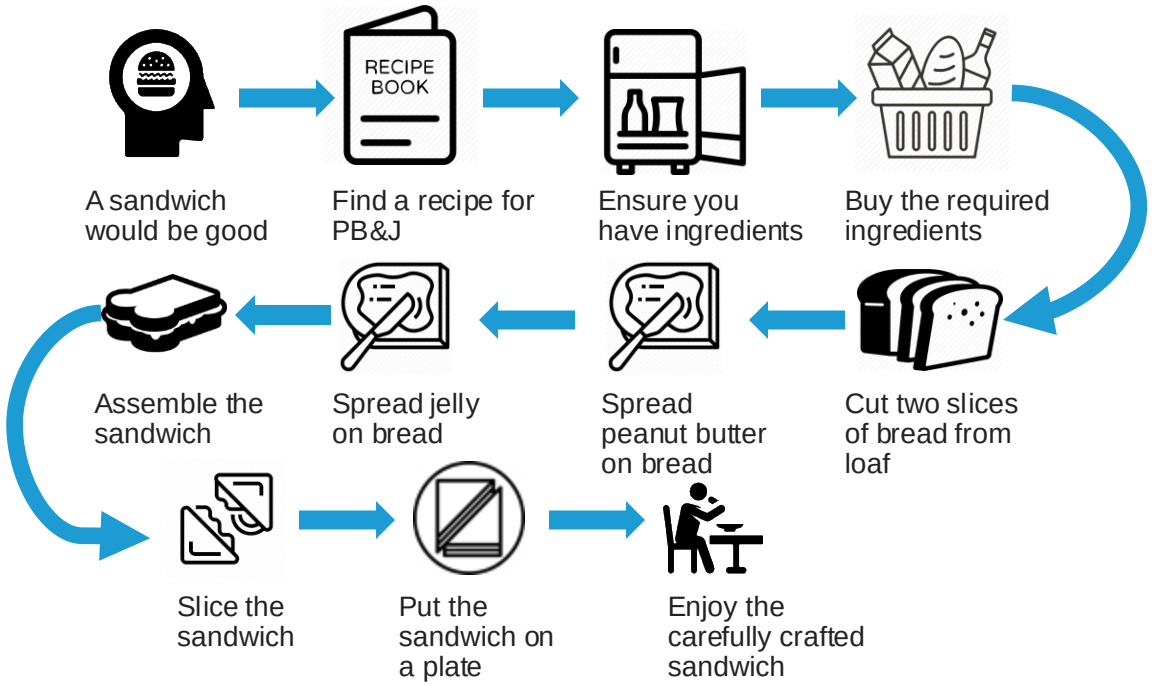
Virtual Server

You worry about this stuff… I don't care. Just give me a Virtual Server.

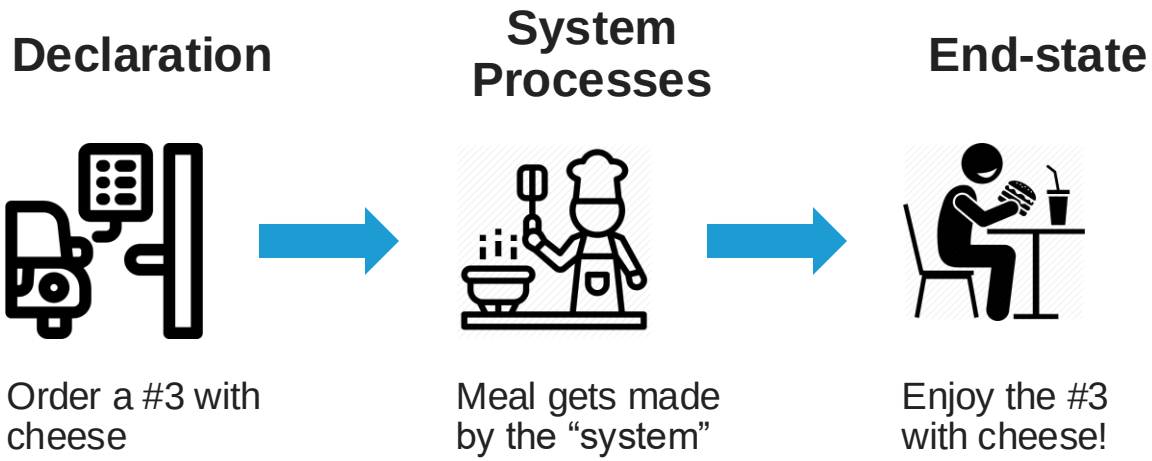
Understanding imperative vs. declarative

REAL WORLD EXAMPLE: EATING A SANDWICH

Imperative model: What everyone's done for years. Every step of a process is meticulously defined, resulting in the desired outcome.



- **Declarative model:** The model that F5 has aligned to. Just input the desired end-state and let the system figure out the rest.



F5 Automation Toolchain

F5 Automation Lifecycle

Bootstrap

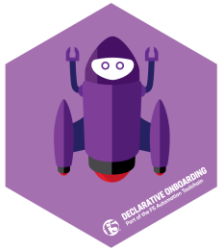
F5 BIG-IP IMAGE GENERATOR

Enables creation, customization and deployment of a BIG-IP disk image for a given private/public cloud.

Onboard

CLOUD SOLUTION TEMPLATES

Canonical examples using native templates for quickly deploying BIG-IP services on public cloud providers (AWS, Azure, Google)



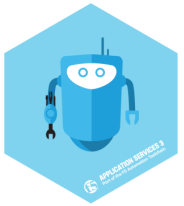
DECLARATIVE ONBOARDING EXTENSION

Initial Config of BIG-IP Instances

Cloud-Init
Handles early initialization of a cloud instance with user-data provided at instance launch time.

L1-L3

Deploy



APP SERVICES 3 EXTENSION

F5 APP SERVICES TEMPLATES (FAST)

Deploy Classic and Advanced Application Services on BIG-IP using Declarative REST APIs

L4-L7

Monitor

TELEMETRY STREAMING EXTENSION

Stream Telemetry, Events & Logs from BIG-IP to various Analytics and Logging solutions

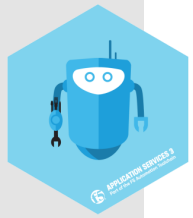
Example Declarations

Onboard

```
{
    "schemaVersion": "1.0.0",
    "class": "Device",
    "async": true,
    "webhook": "https://example.com/myHook",
    "label": "my BIG-IP declaration for declarative onboarding",
    "Common": {
        "class": "Tenant",
        "mySystem": {
            "class": "System",
            "hostname": "bigip.example.com",
            "cliInactivityTimeout": 1200,
            "consoleInactivityTimeout": 1200,
            "autoPhonehome": false
        },
        "myLicense": {
            "class": "License",
            "licenseType": "regKey",
            "regKey": "AAAAA-BBBBB-CCCCC-DDDDD-EEEEEEE"
        },
        "myDns": {
            "class": "DNS",
            "nameServers": [
                "8.8.8.8",
                "2001:4860:4860::8844"
            ],
            "search": [
                "f5.com"
            ]
        },
        "myNtp": {
            "class": "NTP",
            "servers": [
                "0.pool.ntp.org",
                "1.pool.ntp.org",
                "2.pool.ntp.org"
            ],
            "timezone": "UTC"
        },
        "root": {
            "class": "User",
            "userType": "root",
            "oldPassword": "default",
            "newPassword": "myNewPass1word"
        },
        "admin": {
            "class": "User",
            "userType": "regular",
            "password": "asdfjkl",
            "shell": "bash"
```

Deploy App Services

```
{
   "class": "AS3",
   "action": "deploy",
   "persist": true,
   "declaration": {
      "class": "ADC",
      "schemaVersion": "3.0.0",
      "id": "urn:uuid:33045210-3ab8-4636-9b2a-c98d22ab915d",
      "label": "Sample 1",
      "remark": "Simple HTTP application with RR pool",
      "Sample_01": {
         "class": "Tenant",
         "A1": {
            "class": "Application",
            "template": "http",
            "serviceMain": {
               "class": "Service_HTTP",
               "virtualAddresses": [
                  "10.0.1.10"
               ],
               "pool": "web_pool"
            },
            "web_pool": {
               "class": "Pool",
               "monitors": [
                  "http"
               ],
               "members": [{
                  "servicePort": 80,
                  "serverAddresses": [
                     "192.0.1.10",
                     "192.0.1.11"
                  ]
               }]
            }
         }
      }
   }
}
```



Monitoring/Telemetry

```
{
    "class": "Telemetry",
    "My_System": {
        "class": "Telemetry_System",
        "systemPoller": {
            "interval": 60
        }
    },
    "My_Listener": {
        "class": "Telemetry_Listener",
        "port": 6514
    },
    "My_Consumer": {
        "class": "Telemetry_Consumer",
        "type": "Splunk",
        "host": "192.0.2.1",
        "protocol": "https",
        "port": 8088,
        "passphrase": {
            "cipherText": "apikey"
        }
    }
}
```

L1-L3

L4-L7

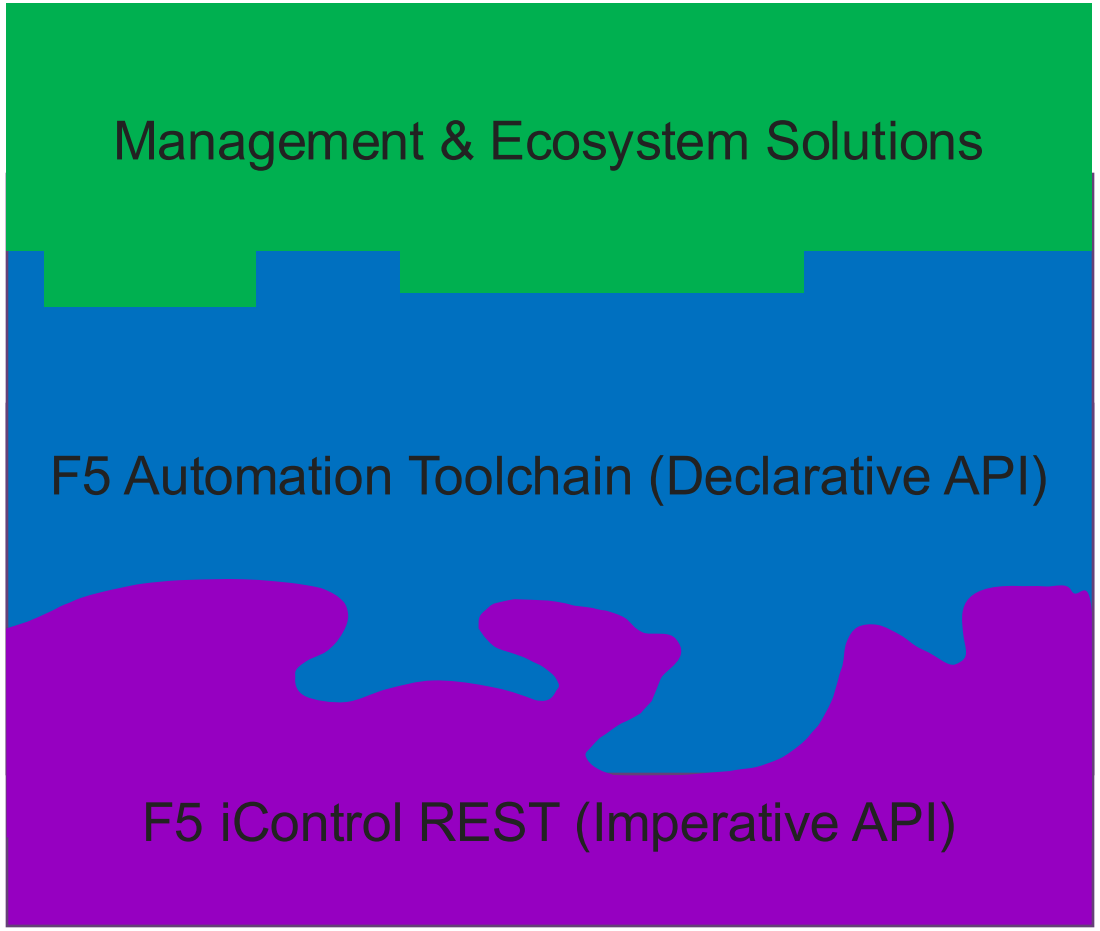
What is the App Services 3 Extension (AS3)?

- TMOS independent RPM package installed on BIG-IP*
- Provides a single declarative REST API endpoint…
  https://big-ip/mgmt/shared/appsvcs/declare
- Interface for common L4-7 app service use cases
- Atomic and idempotent
- Multi-tenant
- Accepts a single JSON document
- Omitted values assume defaults
- Guaranteed to be backward compatible
- Requires TMOS 12.1+
- Available now: https://github.com/F5Networks/f5-appsvcs-extension
- Free & supported by F5

What you can and can’t do with AS3?

- LTM (LB & L7 Full Proxy) almost full coverage
- ASM (WAF) – apply existing policy (can be downloaded from repo)
- APM (IAM) - apply existing policy (can be downloaded from repo)
- AFM (FW) – Firewall policies and auto-discovery
- DNS - basic GSLB

Automation Toolchain Within Customer Ecosystem



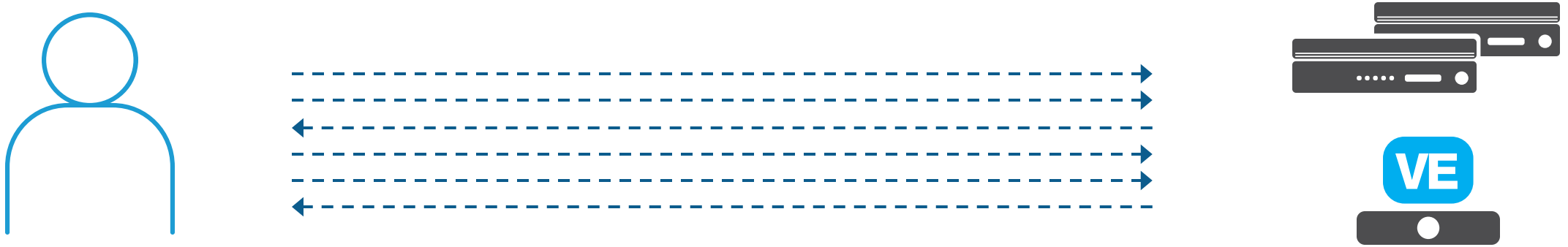
Source of Truth

Declarative - Tell the system **WHAT** you would like to happen and let it figure out **HOW** to do it

Imperative - Tell the system **HOW** to do something and as a result **WHAT** you want to happen

Automation WITHOUT F5 Automation Toolchain

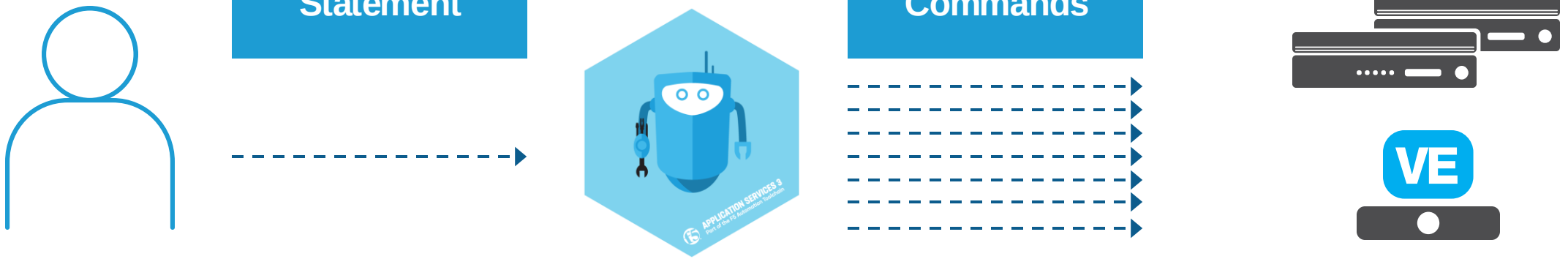
Configuring BIG-IP requires many REST API calls



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

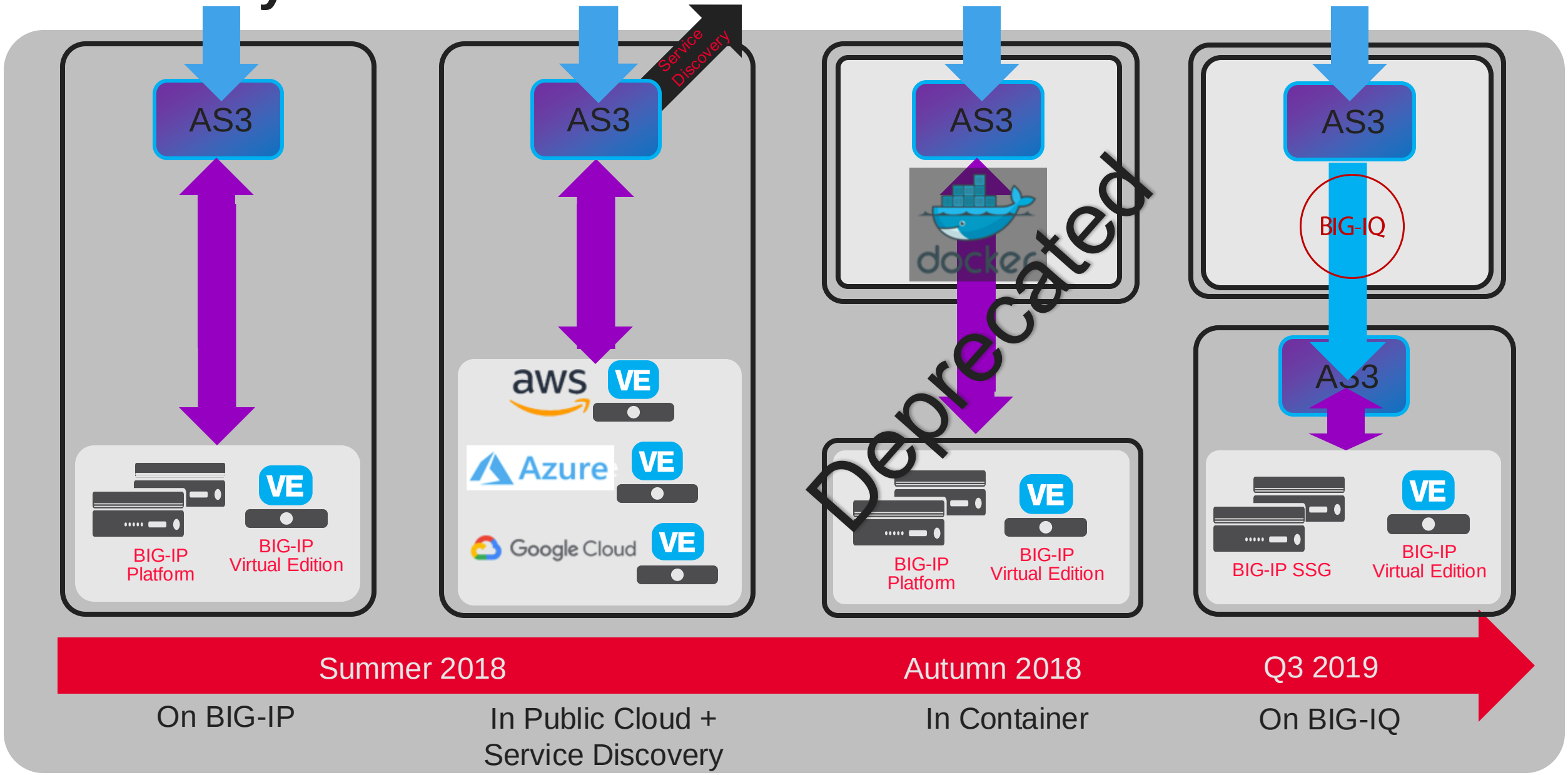
Automation WITH F5 Automation Toolchain

Configuring BIG-IP Services requires a SINGLE REST API call



- 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

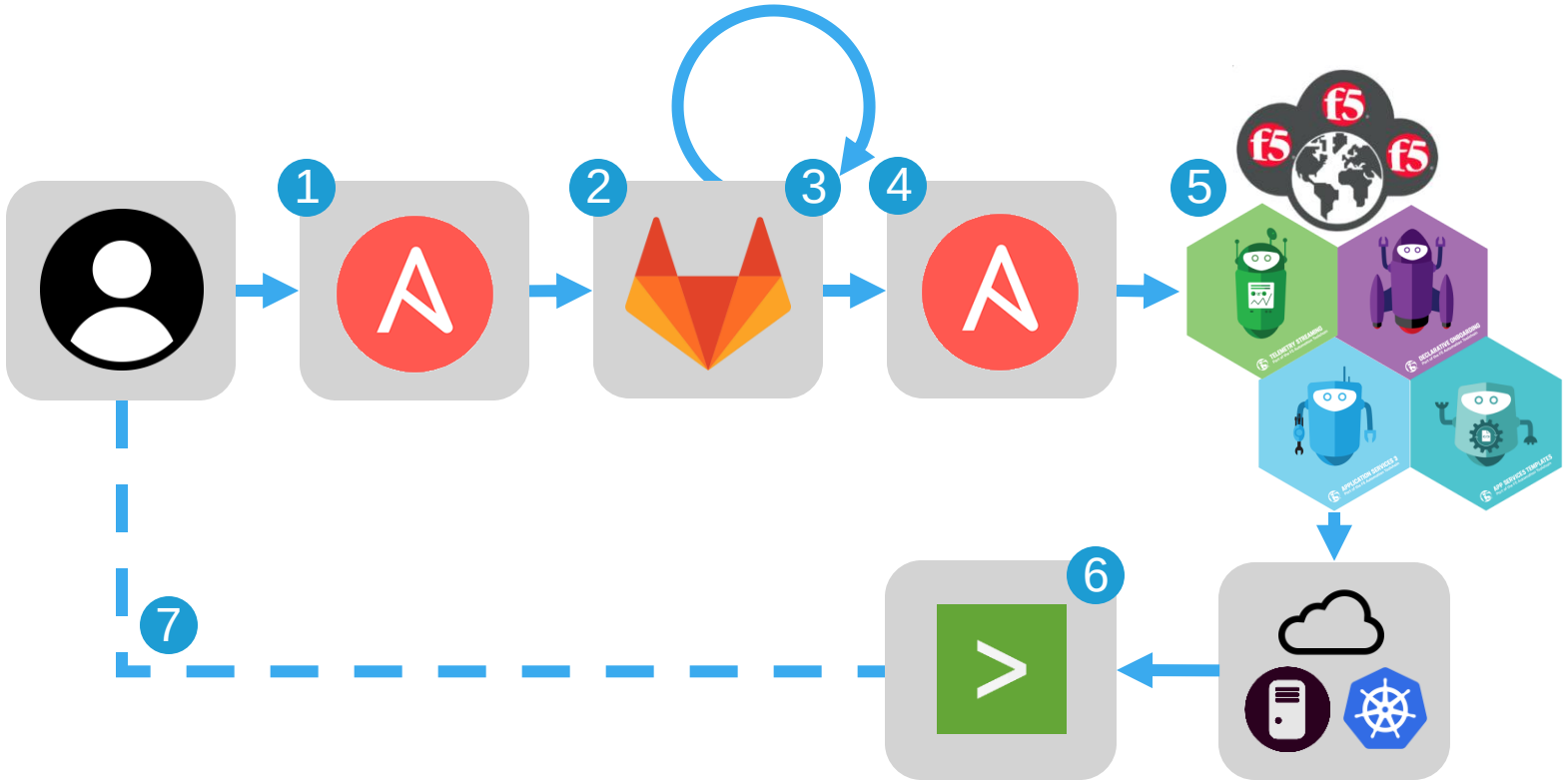
AS3 Everywhere – Consistent API



Automation Workflow Examples

Automation Toolchain and Ecosystems

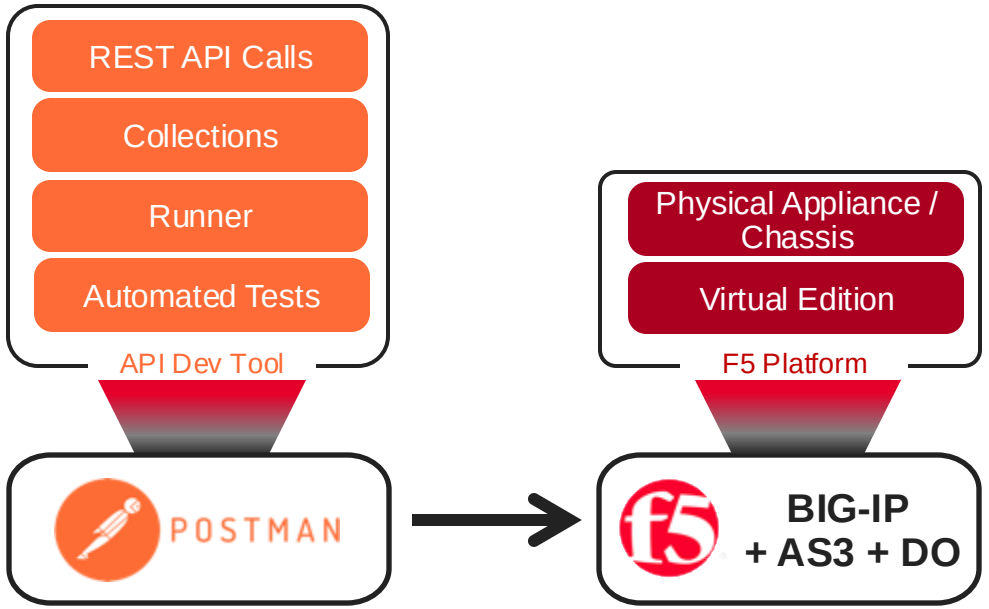
REAL WORLD WORKFLOW WITH POPULAR AUTOMATION, ORCHESTRATION, SOURCE CONTROL, AND ANALYTICS TOOLS



1. User logs into self-service Portal **(Tower)**
2. New app definition is created in Source Control Repo **(Gitlab)**
3. **CI** tool notices changes and runs a job on the orchestrator **(Gitlab)**
4. Orchestrator runs tasks against F5 API endpoints **(Ansible)**
5. **Automation Toolchain** & **F5CS GSLB** are consumed via Ansible to update environments (**VE**)
6. Analytic information is received from Telemetry Streaming (**Splunk**)
7. Automation or User will take action against Telemetry data and restart the cycle.

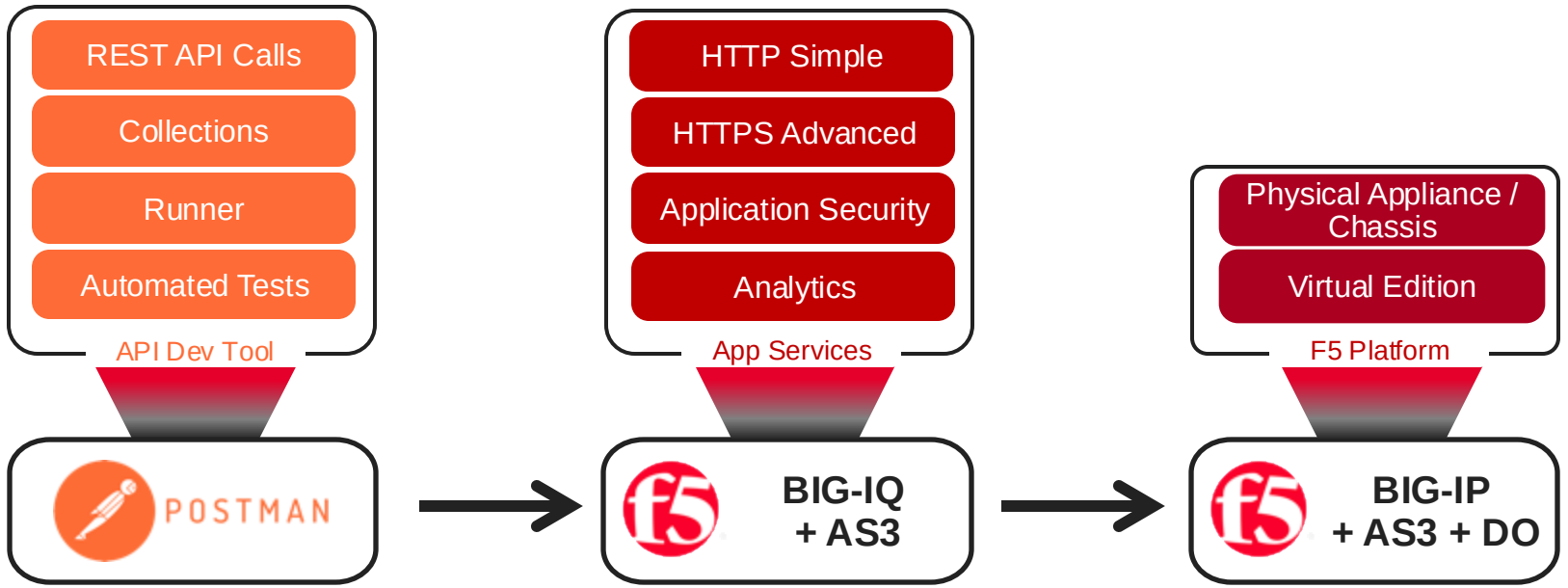
Workflow Example 0a

Learning about declarative onboarding and AS3 automation of BIG-IP



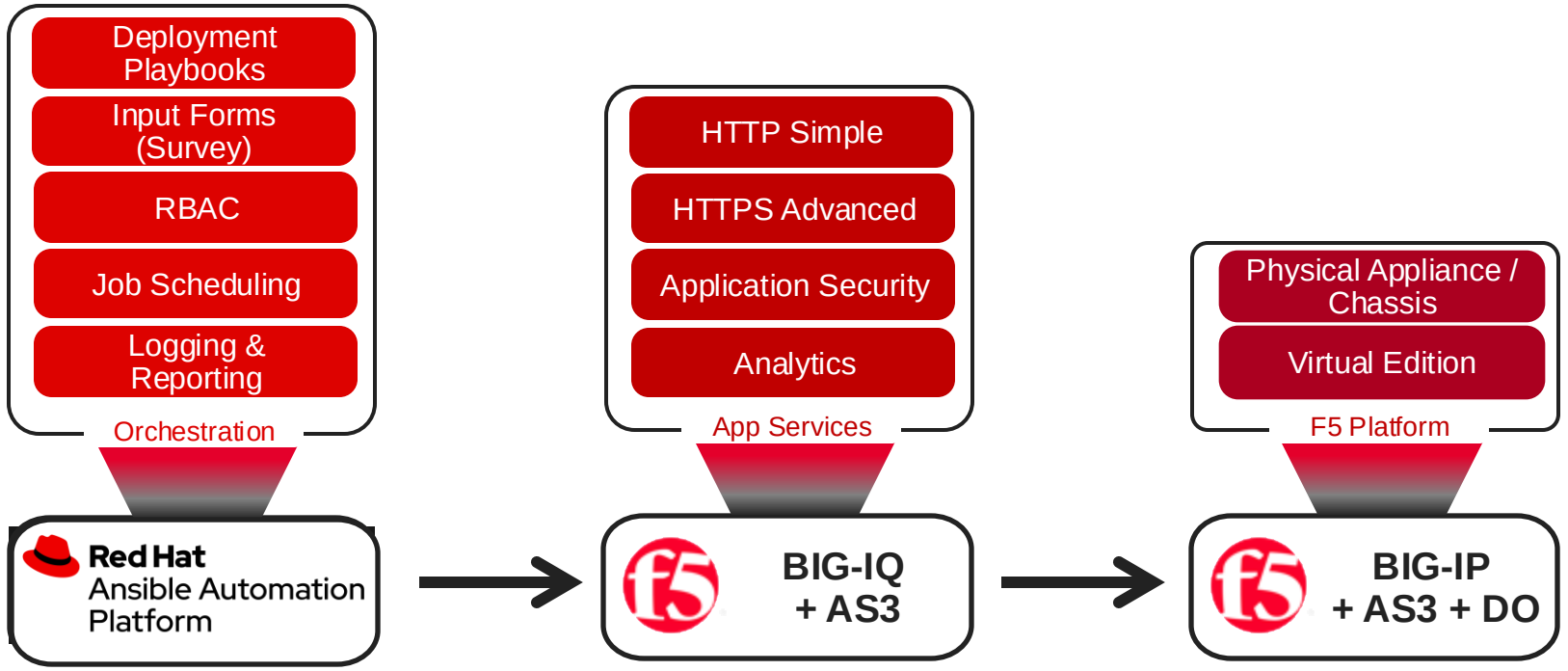
Workflow Example 0b

AS3 deployment via the BIG-IQ AS3 proxy to leverage BIG-IQ Analytics



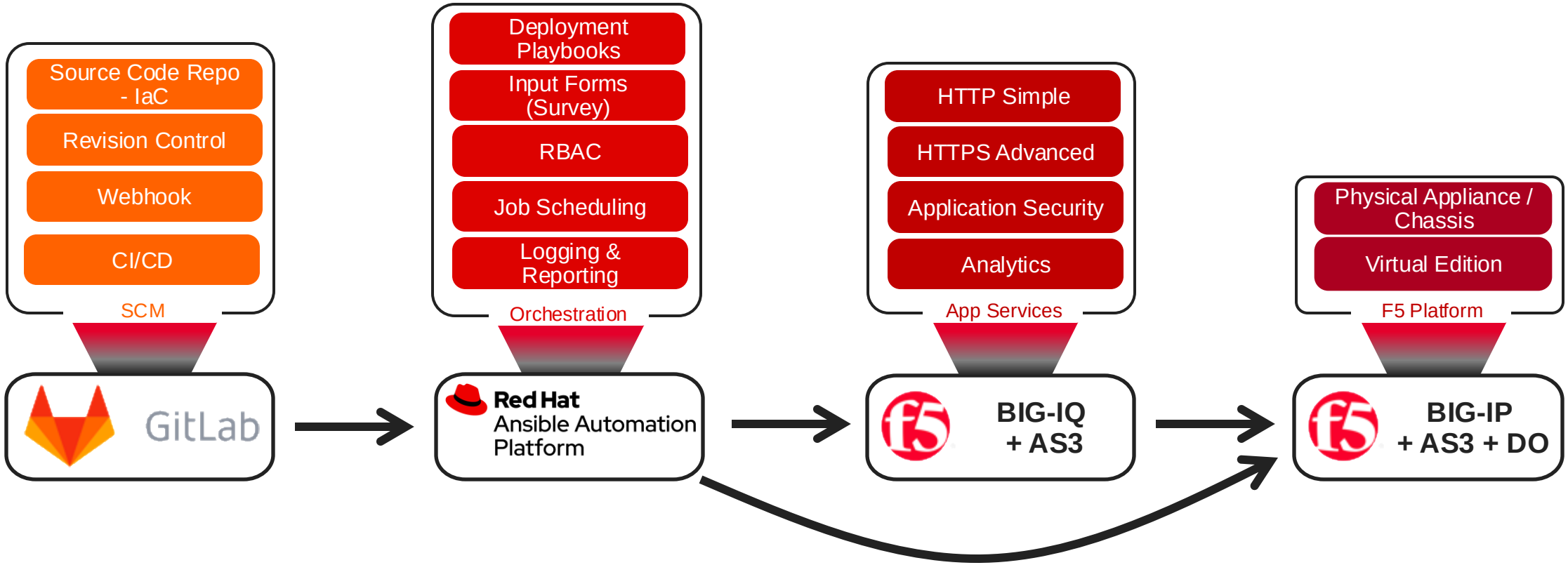
Workflow Example 1

Configuration Management using Ansible Automation Platform



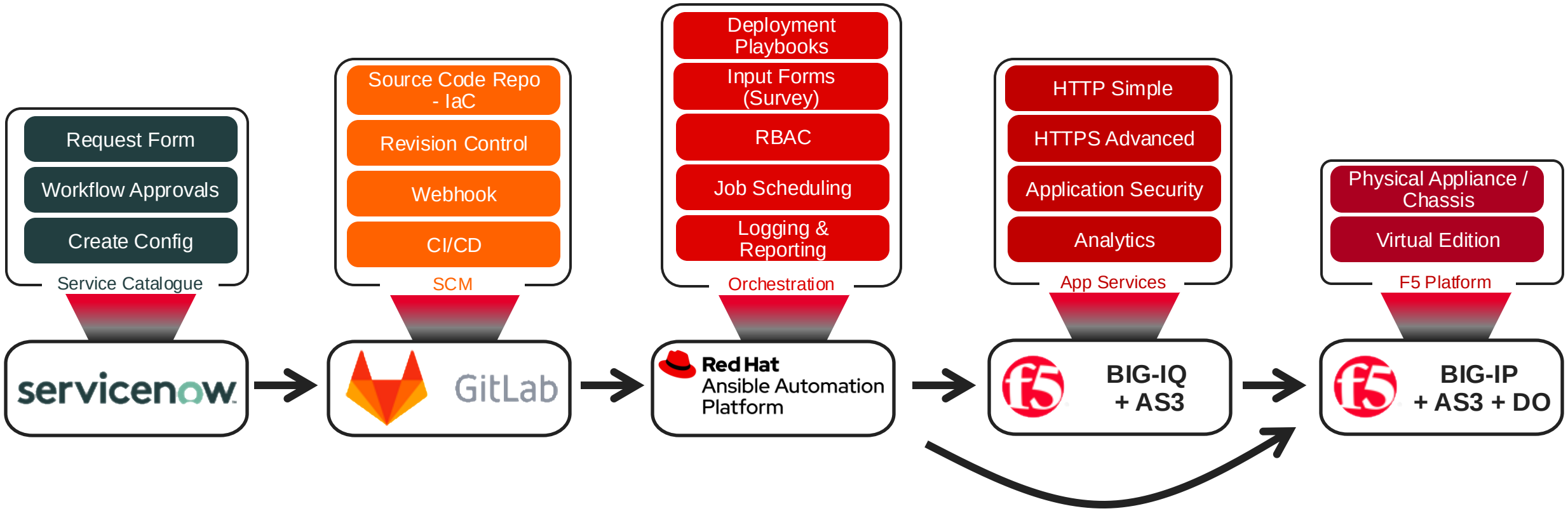
Workflow Example 2

Infrastructure as Code using a Source Code Management tool



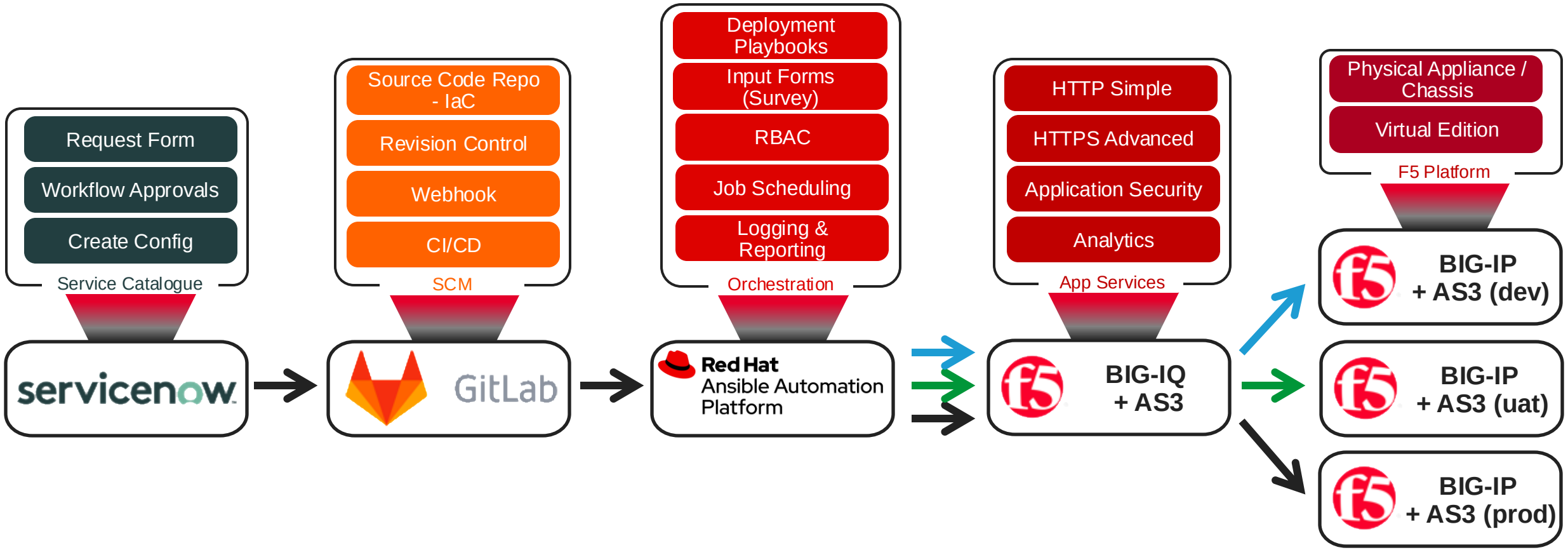
Workflow Example 3

Self-service provisioning via a service catalogue



Workflow Example 4

Automated staging and testing before deploying to production



AS3 Declaration

AS3 Methods

The AS3 API supports Create, Read, Update, and Delete (CRUD) actions…

POST

Deploy configuration;

Actions:

- deploy
- dry-run
- patch
- redeploy
- retrieve
- remove

GET

Retrieve previous declaration.

Select the data you want by appending to the AS3 path.

PATCH

Modify existing declaration:

- add
- remove
- replace
- move
- copy

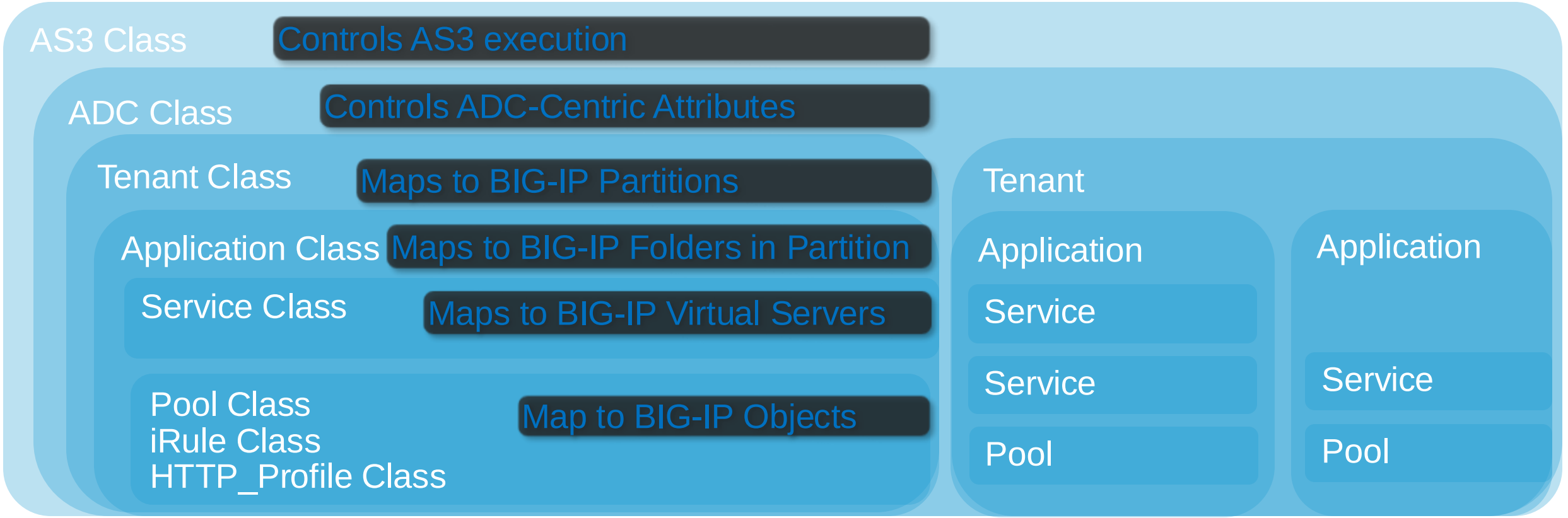
DELETE

Remove data for one or more tenants.

If no tenant specified, the entire AS3 configuration is removed.

AS3 JSON Schema

The simplest representation of an AS3 declaration…



AS3 Sample Declaration

Create a simple HTTP application service:

- HTTP Virtual Server
- Pool named web_pool with 2 members
- Monitored by default HTTP health monitor
- Partition (tenant) named Sample_01

Declaration not ordered, nor sequenced

POST this JSON text to AS3 at:

https://{big-ip}/mgmt/shared/appsvcs/declare

That's it!

my_first_as3.json

```
{
    "class": "AS3",
    "action": "deploy",
    "persist": true,
    "declaration": {
        "class": "ADC",
        "schemaVersion": "3.0.0",
        "id": "example-declaration-01",
        "label": "Sample 1",
        "remark": "Simple HTTP application with round robin pool",
        "updateMode": "selective",
        "Sample_01": {
            "class": "Tenant",
            "defaultRouteDomain": 0,
            "Application_1": {
                "class": "Application",
                "template": "http",
            "serviceMain": {
                "class": "Service_HTTP",
                "virtualAddresses": [
                    "10.0.1.10"
                ],
                "pool": "web_pool"
                },
                "web_pool": {
                    "class": "Pool",
                    "monitors": [
                        "http"
                    ],
                    "members": [
                        {
                            "servicePort": 80,
                            "serverAddresses": [
                                "192.0.1.10",
                                "192.0.1.11"
                            ]
                        }
                    ]
                }
            }
        }
    }
}
```

AS3 Sample Declaration

Add entries to a firewall rule based on AWS tags

- Instance tags
- Added and removed automatically

Declaration not ordered, nor sequenced

POST this JSON text to AS3 at:

https://{big-ip}/mgmt/shared/appsvcs/declare

That's it!

my_aws_as3.json

```
{
    "firewallPolicy": {
        "class": "Firewall_Policy",
        "remark": "A firewall policy",
        "rules": [{
            "use": "ruleList"
        }]
    },
    "ruleList": {
        "class": "Firewall_Rule_List",
        "remark": "A firewall rule list",
        "rules": [{
            "name": "rule",
            "remark": "A firewall rule list rule",
            "action": "drop",
            "protocol": "tcp",
            "source": {
                "addressLists": [{
                    "use": "addressList"
                }]
            },
            "loggingEnabled": true
        }]
    },
    "addressList": {
        "class": "Firewall_Address_List",
        "remark": "A firewall address list",
        "addresses": [{
            "servicePort": 80,
            "addressDiscovery": "aws",
            "updateInterval": 10,
            "tagKey": "application",
            "tagValue": "prod",
            "addressRealm": "private",
            "region": "ap-southeast-2",
            "accessKeyId": "XxXxX",
            "secretAccessKey": "XxXxX",
            "credentialUpdate": false
        }]
    }
}
```

AS3 Sample Declaration

The AS3 Class

```
class:     AS3 # mandatory
action:    deploy, dry-run, redeploy, retrieve,remove
persist:   true, false
```

The ADC Class

```
class:           ADC # mandatory
schemaVersion:   3.0.0, 3.1.0, 3.2.0
Id:              arbitrary (suggest using urn:
Label:           arbitrary (labels the declara
updateMode:      complete, selective
target:          Destination IP, via docker /
                 #optional
```

The Tenant Class

```
Sample_01:    Tenant name i.e. BIG-IP partition name
class:        Tenant # mandatory
defaultRouteDomain:  number
```

my_first_as3.json

```
{
    "class": "AS3",
    "action": "deploy",
    "persist": true,
    "declaration": {
        "class": "ADC",
        "schemaVersion": "3.0.0",
        "id": "example-declaration-01",
        "label": "Sample 1",
        "remark": "Simple HTTP application with round robin pool",
        "updateMode": "selective",
        "Sample_01": {
            "class": "Tenant",
            "defaultRouteDomain": 0,
            "Application_1": {
                "class": "Application",
                "template": "http",
                "serviceMain": {
                    "class": "Service_HTTP"
                                                    "servicePort": 80,
                                                    "serverAddresses": [
                                                        "192.0.1.10",
                                                        "192.0.1.11"
                                                    ]
                                                }
                                            ]
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

IMPORTANT

This will remove all tenants that AS3 has created …and replace with this declaration.

AS3 Sample Declaration

The Application Class

```
Application_1: The application name = TMSH folder in
partition.
class:     Application # mandatory
template:  http, https, tcp, udp, l4, generic, shared
```

The Service Class

```
class:      Service_HTTP, Service_HTTPS,
            Service_TCP, Service_UDP, Service_L4,
            generic, shared

virtualAddresses: IP Address
```

The Pool Class

```
web_pool:  Defines the name of the pool
class:     pool # mandatory
monitors:  not mandatory, but highly recommended!
members:   servicePort, serverAddresses # if
           servicePort is not included, it will
           choose a default related to the template
           e.g. http = 80
```

my_first_as3.json

```
{
    "class": "AS3",
                "web_pool": {
                    "class": "Pool",
                    "monitors": [
                        "http"
                    ],
                    "members": [
                        {
                            "servicePort": 80,
                            "serverAddresses": [
                                "192.0.1.10",
                                "192.0.1.11"
                            ]
                        }
                    ]
                }
            }
        }
    }
}
```

IMPORTANT

Aside from ‘generic’ and ‘shared’, values for template and virtualAddresses must correlate.

Generic: Doesn’t enforce required objects. Allows you to rename serviceMain

Shared: Holds objects other applications can use

AS3 Sample Declaration

The Application Class

```
Application_1: The application name = TMSH folder in
partition.

class:     Application # mandatory
```

The Service Class

```
class:     Service_HTTP, Service_HTTPS,
           Service_TCP, Service_UDP, Service_L4,
           generic, shared

virtualAddresses: IP Address
```

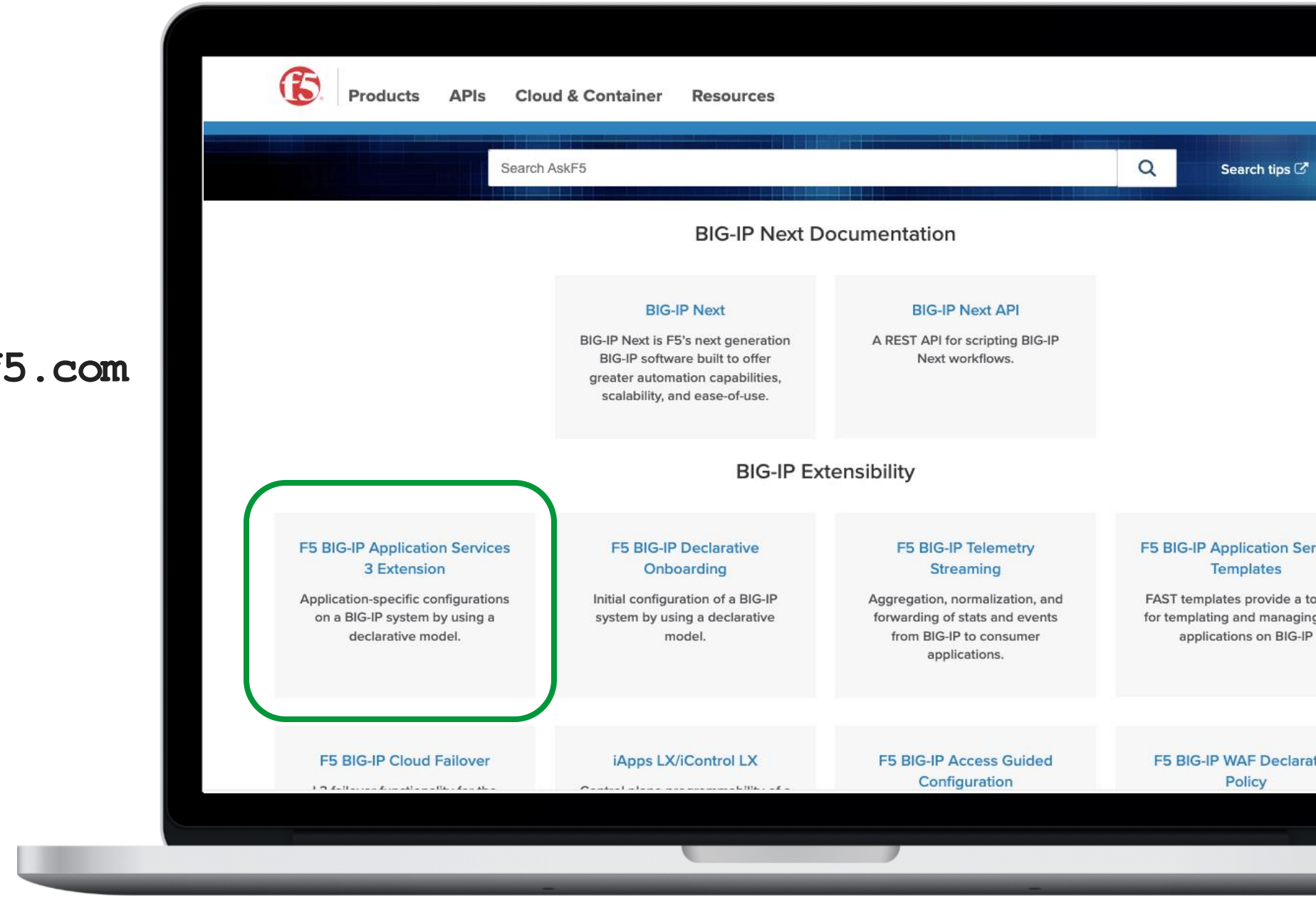
The Pool Class

```
web_pool: Defines the name of the pool
class:    pool # mandatory
monitors: not mandatory, but highly recommended!
members:  servicePort, serverAddresses # if
          servicePort is not included, it will
          choose a default related to the template
          e.g. http = 80
```

my_first_as3.json

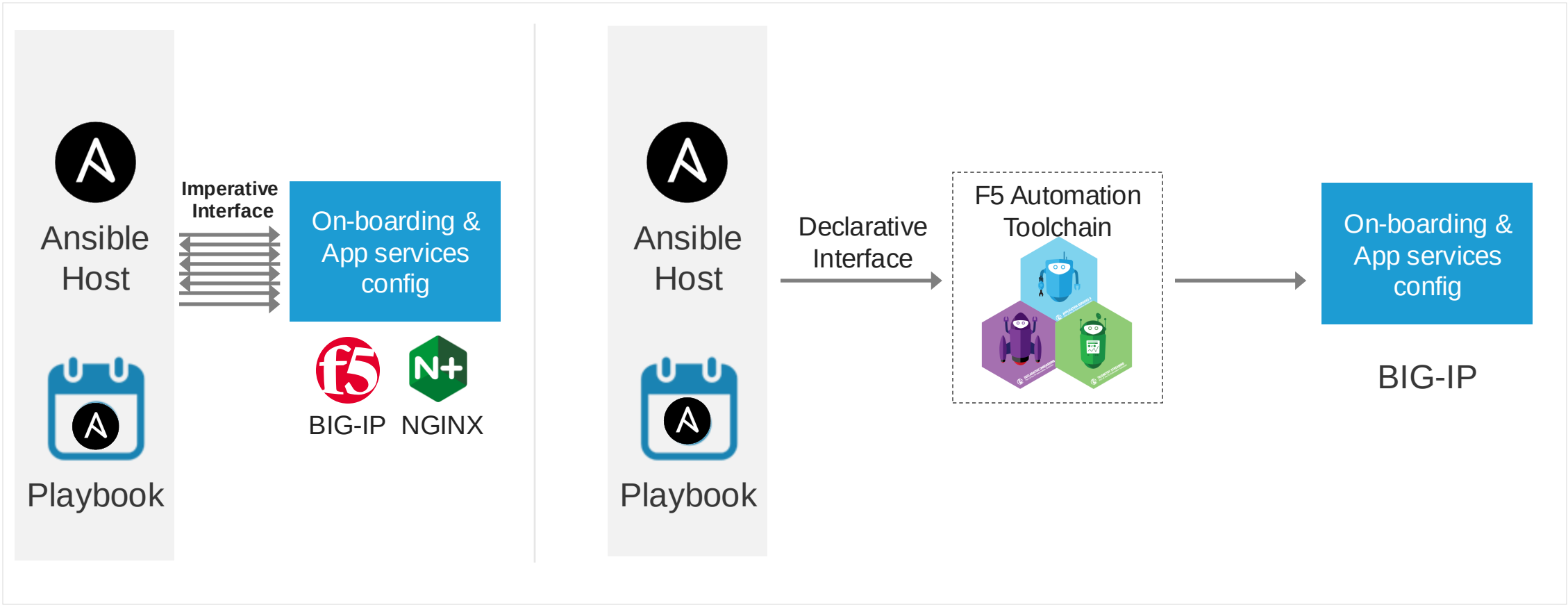
```
{
    "class": "AS3",
    "action": "deploy",
    "persist": true,
    "declaration": {
        "class": "ADC",
        "schemaVersion": "3.0.0",
        "id": "example-declaration-01",
        "label": "Sample 1",
        "remark": "Simple HTTP application with round robin pool",
        "updateMode": "selective",
        "Sample_01": {
            "class": "Tenant",
            "defaultRouteDomain": 0,
            "Application_1": {
                "class": "Application",
                "template": "http",
                "serviceMain": {
                    "class": "Service_HTTP",
                    "virtualAddresses": [
                        "10.0.1.10"
                    ],
                    "pool": "web_pool"
                    },
                    "web_pool": {
                        "class": "Pool",
                        "monitors": [
                            "http"
                        ],
                        "members": [
                            {
                                "servicePort": 80,
                                "serverAddresses": [
                                    "192.0.1.10",
                                    "192.0.1.11"
                                ]
                            }
                        ]
                    }
                }
            }
        }
    }
}
```


https://clouddocs.f5.com



Automating with Ansible

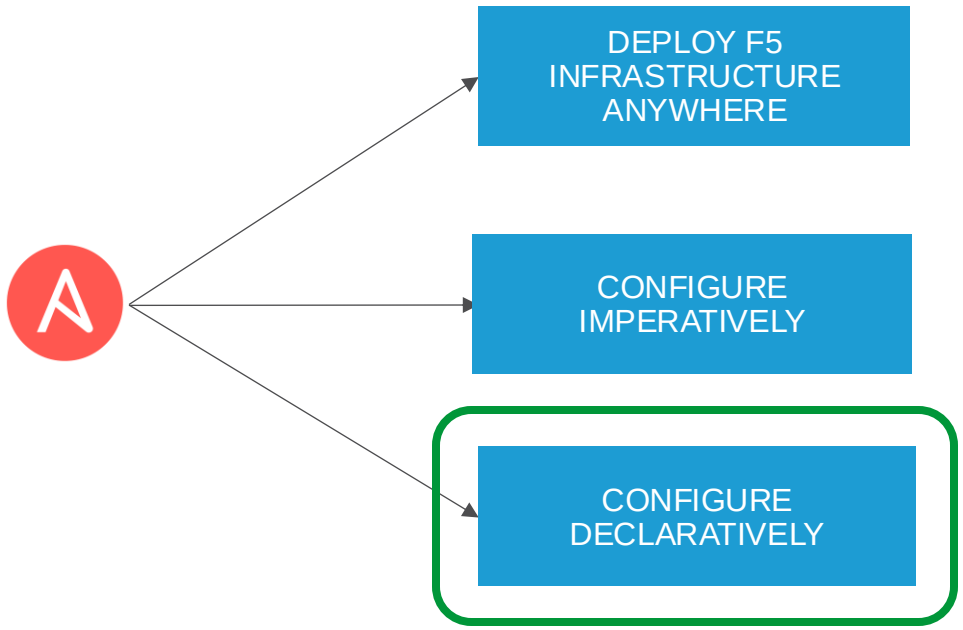
App Services Automation with Ansible and F5



Imperative Configuration

Declarative Configuration

AS3 Sample Ansible Playbook



my_first_as3_with_ansible.yaml

```
1  ---
2  - name: Deploy AS3 playbook
3    hosts: bigip
4    connection: local
5    gather_facts: no
6
7    vars:
8
9    vars_files:
10   - ../creds/autows_bigip_creds.yaml
11
12   tasks:
13   - name: Deploy AS3 declaration
14     uri:
15       url: https://{{bigip_host}}/mgmt/shared/appsvcs/declare
16       user: "{{bigip_user}}"
17       password: "{{bigip_pass}}"
18       method: POST
19       body: "{{ lookup('file','./as3.json') }}"
20       force_basic_auth: yes
21       body_format: json
22
23
24
```


Call to Action

If a task seems like it should be easier, it probably can be!

Commit	Start	Visit	Download and use
Commit to the shift in mindset and learning new things • Ansible / F5 Workshop	Start with something simple	www.f5.com/ansible • Learn more about F5/Ansible automation on our alliance pages. • Access Use cases, webinar recordings, Blogs	Supported F5 modules and roles on Ansible Galaxy • F5 on Ansible Galaxy

Where to Learn More

Component	Type	Resource	Link
Cloud Solution Templates	Documentation	Overview	https://www.f5.com/pdf/solution-profiles/boost-agility-and-automation-with-f5-cloud-solution-technologies.pdf
Application Services 3 (AS3)	Documentation	AS3 Extension Documentation	https://f5.com/AS3Docs
Declarative Onboarding (DO)	Documentation	DO Extension Documentation	https://f5.com/DODocs
Telemetry Streaming (TS)	Documentation	TS Extension Documentation	https://f5.com/TSDocs
Automation Toolchain	Video	Overview	https://community.f5.com/t5/technical-articles/lightboard-lessons-f5-automation-toolchain/ta-p/285368
F5 Community Training & Labs	Labs & Classes	Learning	https://clouddocs.f5.com/training/community/

Additional resources in notes