



**Red Hat**

**RED HAT FORUM | SEPTEMBER 10 2019**

**AGILE APP DEVELOPMENT AND SECURITY  
REQUIREMENTS IN BALANCE**

**Andrea Arquint | Systems Engineer | F5  
Networks**



# Agenda

**Host App on NGINX+ Docker instance**

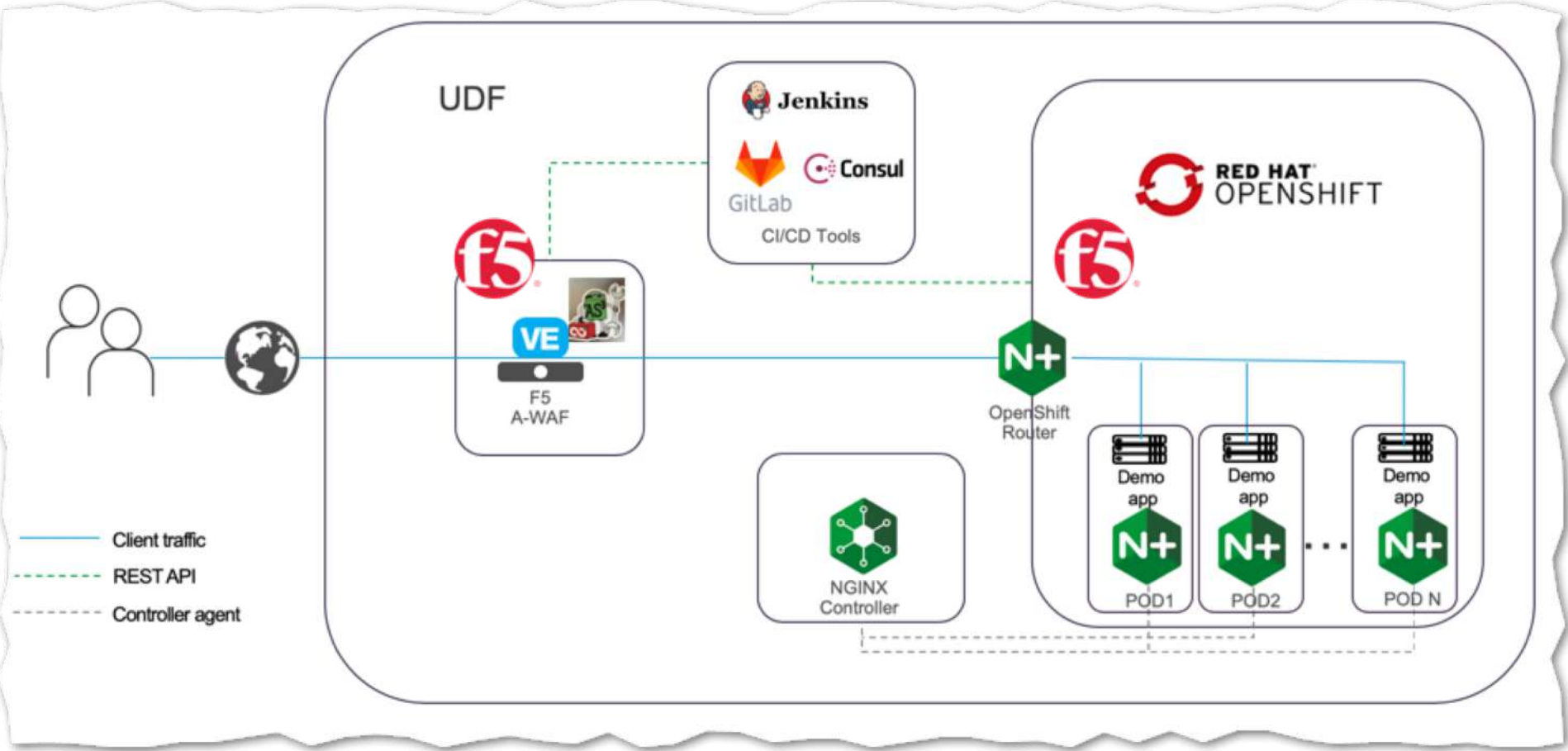
**Expose Apps outside OCP via NGINX+ OCP Router**

**BIG-IP AWAFF to protect containerized Apps**

**Ecosystem & Integrations**

**CI/CD  
Integration  
Dem  
Q&A**

# Architecture diagram (diff. use-cases)



# Why CI/CD ?

'Cos automation is KEY

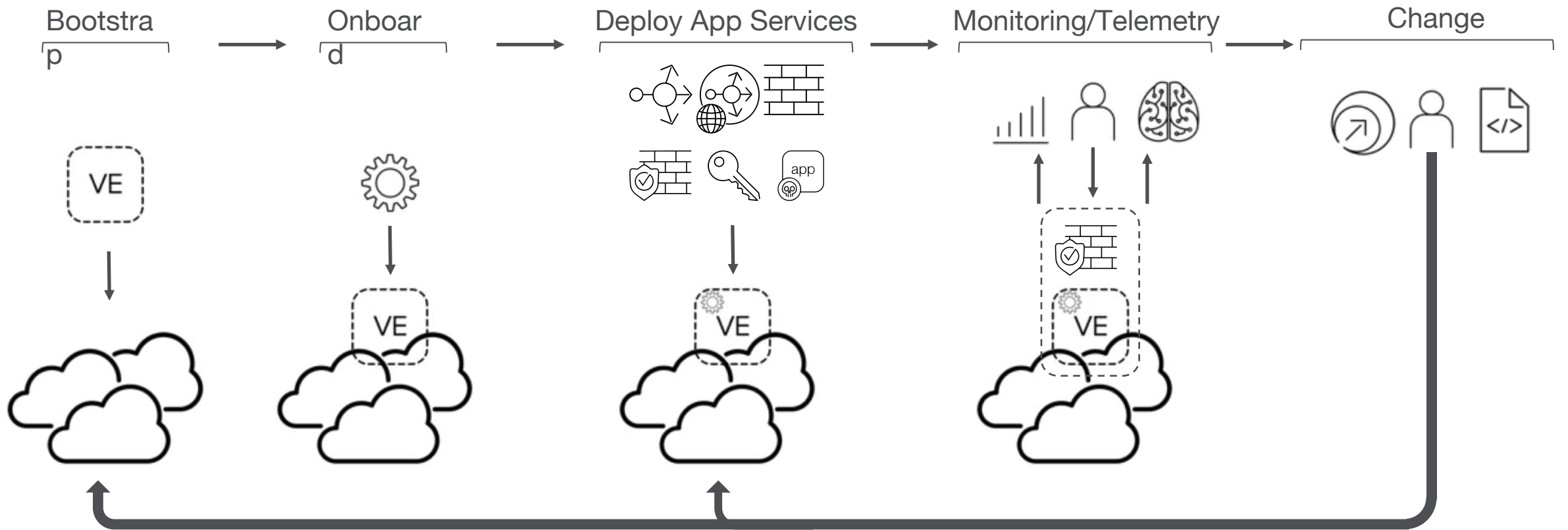
- helps reduce errors
- makes Devs more independent – IaC
- supports new methods and models – CI / CD
- Increases speed – better **time to market**

BUT – does it increase App Security ?

Would you trade Security for speed ?



# Application Services Lifecycle



# CI Overview

## CONTINUOUS INTEGRATION

**CI allows developers to work together on projects by:**

- Ensures code meets sprint requirements – test driven development
- Ensure developer does not break existing unit tests
- Ensuring developer does not introduce known bugs into existing code base

**Used heavily in agile development shops**







# CD Overview

CONTINUOUS DELIVERY/DEPLOYMENT

**Can stand for Delivery or Deployment:**

- Delivery example is building a Java WAR file for the application
- Deployment example is building a Blue/Green production environment

**Enable multiple release per-day**

**Relies on Infrastructure-as-Code**

# CI/CD

## BETTER TOGETHER

Allows developers to:

- Move at a greater velocity
- Ensure deployment is tested in production like environment (ideally)
- Gives App owner and developer freedom to deploy on their own

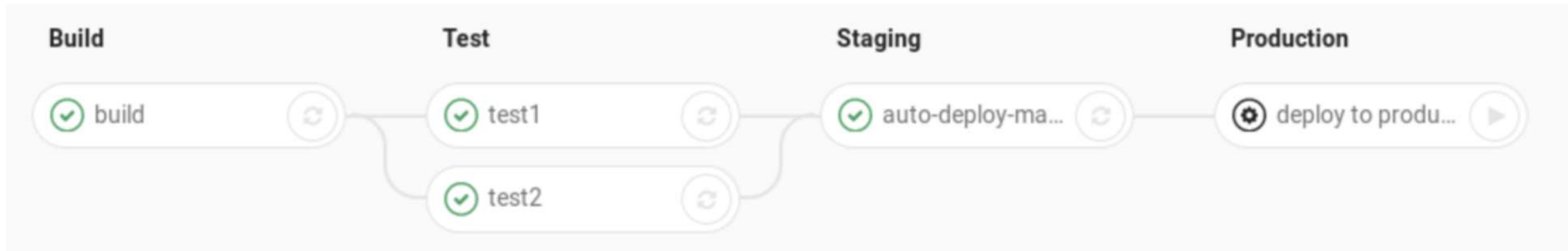
Ideal state for most teams I work with – but automation has to be solved 1<sup>st</sup>





# What is a pipeline ?

A pipeline is a group of **jobs** that get executed in **stages**. All of the jobs in a stage are executed in parallel, and if they all succeed, the pipeline moves on to the next stage. If one of the jobs fails, the next stage is not (usually) executed.



# Bespoke vs. Pipeline

## WHY DO WE NEED A PIPELINE?

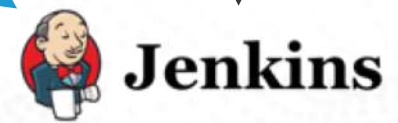
### BESPOKE

1. App Team Creates App
  - a. Service Request (SR) #1 to Systems
2. App Throws App Over Wall to Systems
  - a. SR #2 to Networking
3. Systems Deploys Apps Throws to Networking
  - a. SR #3 to Security
4. Networking Creates ADC Services Throws to Security
  - a. Change Advisory Board (CAB) Meeting
5. Security Enables Access
  - a. Application Deployed?

### PIPELINE

1. App Team commits code
  - a. Trigger Systems Build
2. Systems pipeline deploys code
  - a. Trigger Networking Build
3. Networking pipeline deploys network access with Security Policy
  - a. Fetches Security Policy from Security Repository

# DESIGN



NGINX+

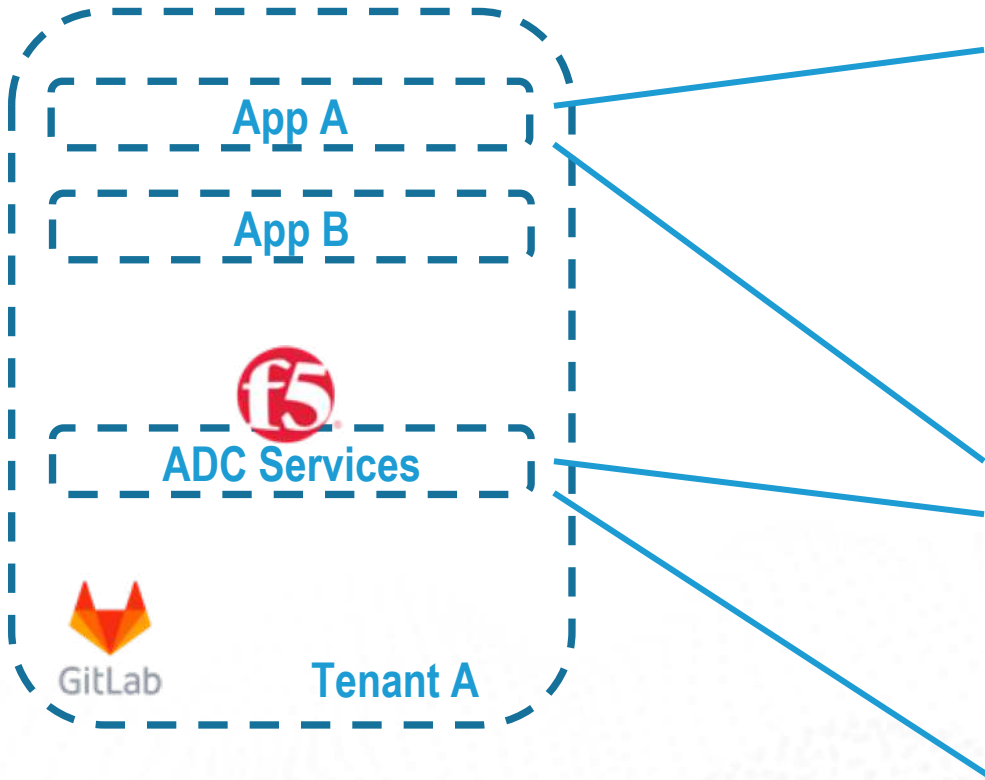


BIG-IP  
Virtual  
Edition(s)



NGINX  
Controller



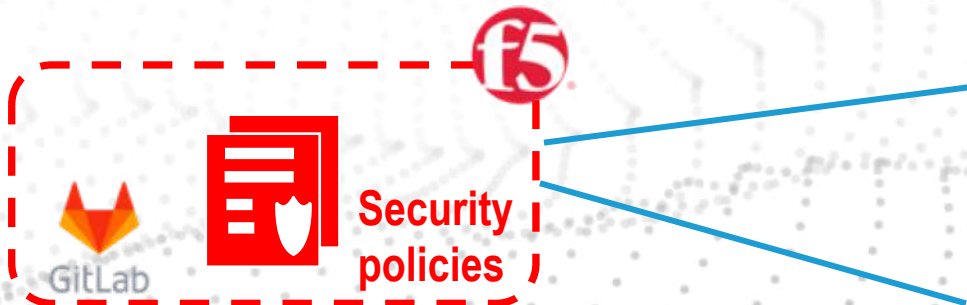


It contains:

- App definition: deployment, service, route definition for minishift
- ADC service: app definition using as3 declaration model, BIG-IP cluster to use

It contains:

- BIG-IP clusters “owned” by this tenant
- Each cluster contains a list of apps hosted on it



It contains:

- Security policies to protect apps





Consul is used to:

- Store "secrets" : BIG-IP cluster credentials
- Store infrastructure information:
  - BIG-IP IPs related to a cluster name
  - Minishift IP
  - Minishift API Token



Jenkins is used to:

- handle pipeline (as a CI server)
- Trigger a pipeline when changes happen in gitlab repos (Webhooks)
  - App CRUD
  - ADC Services CRUD



## Red Hat Openshift (Minishift):

- Will host application based on the gitlab
- Deployed by Jenkins Server



## F5 BIG-IP:

- Will host application ADC-Services (**A**pplication **D**elivery **C**ontrolling) based on the repo on gitlab
- Configured by Jenkins Server (pipeline)

# WORKFLOW – App update



CI server will:

1. Check App B repo
2. Deploy APP B
3. Test availability of App B
4. Retrieve APP B servers IP
5. Create AS3 app definition
6. Update the ADC Services repo with this app definition



# WORKFLOW – ADC Services Update



NGINX+



Jenkins



BIG-IP  
Virtual  
Edition(s)

CI server will:

1. Check ADC services repo
2. Find the cluster to update (based on commit message)
3. Aggregate all apps in this cluster in a single AS3 Declaration
4. Push AS3 declaration on relevant BIG-IPs (use Consul)





# Demo Time

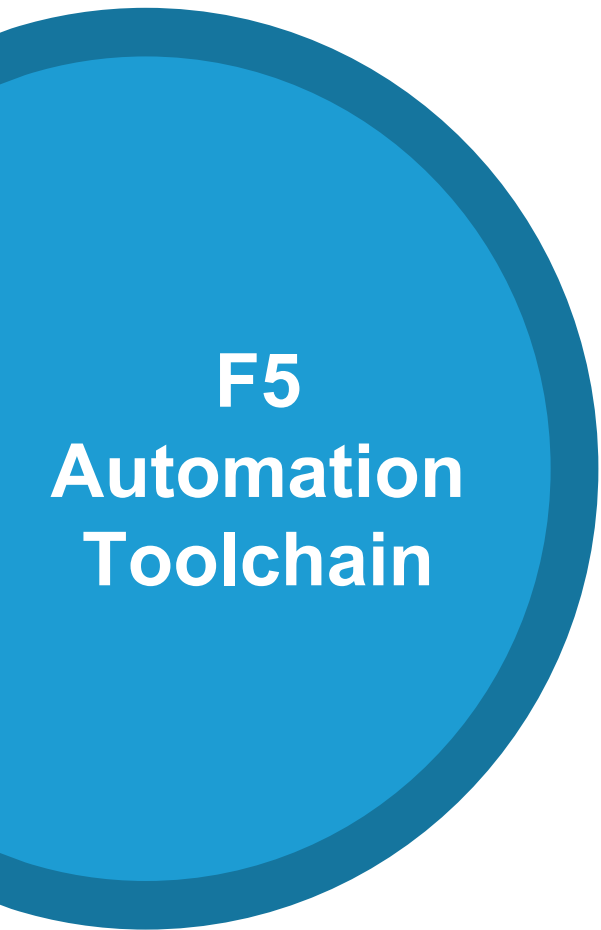


# Backup Slides

# F5 Automation Toolchain



# F5 Automation Toolchain

A blue circular graphic with a white border, containing the text 'F5 Automation Toolchain' in white.

**F5  
Automation  
Toolchain**

## The Toolchain

A set of free tools for automating deployment and configuration of F5 devices and services through declarative APIs.

Use for automation and integration of F5 solutions into automation and orchestration systems.

# F5 Automation Toolchain



Cloud Templates

**F5  
Automation  
Toolchain**

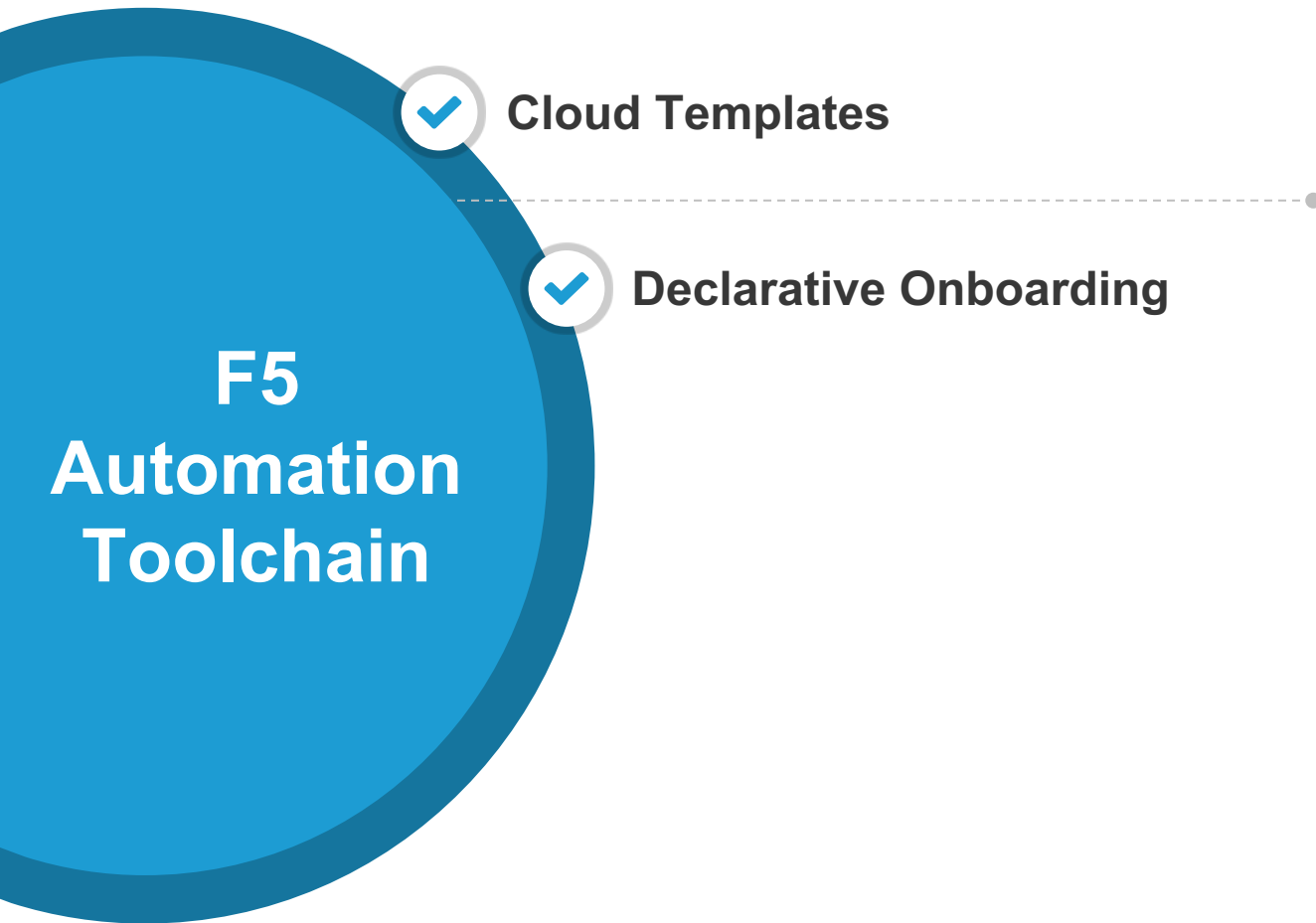


## **F5 Cloud Templates**

Easily deploy F5 fully configured, operational F5 BIG-IP VEs into public and private clouds.

Leverage each cloud's native services to support a wide variety of BIG-IP VE use cases.

# F5 Automation Toolchain

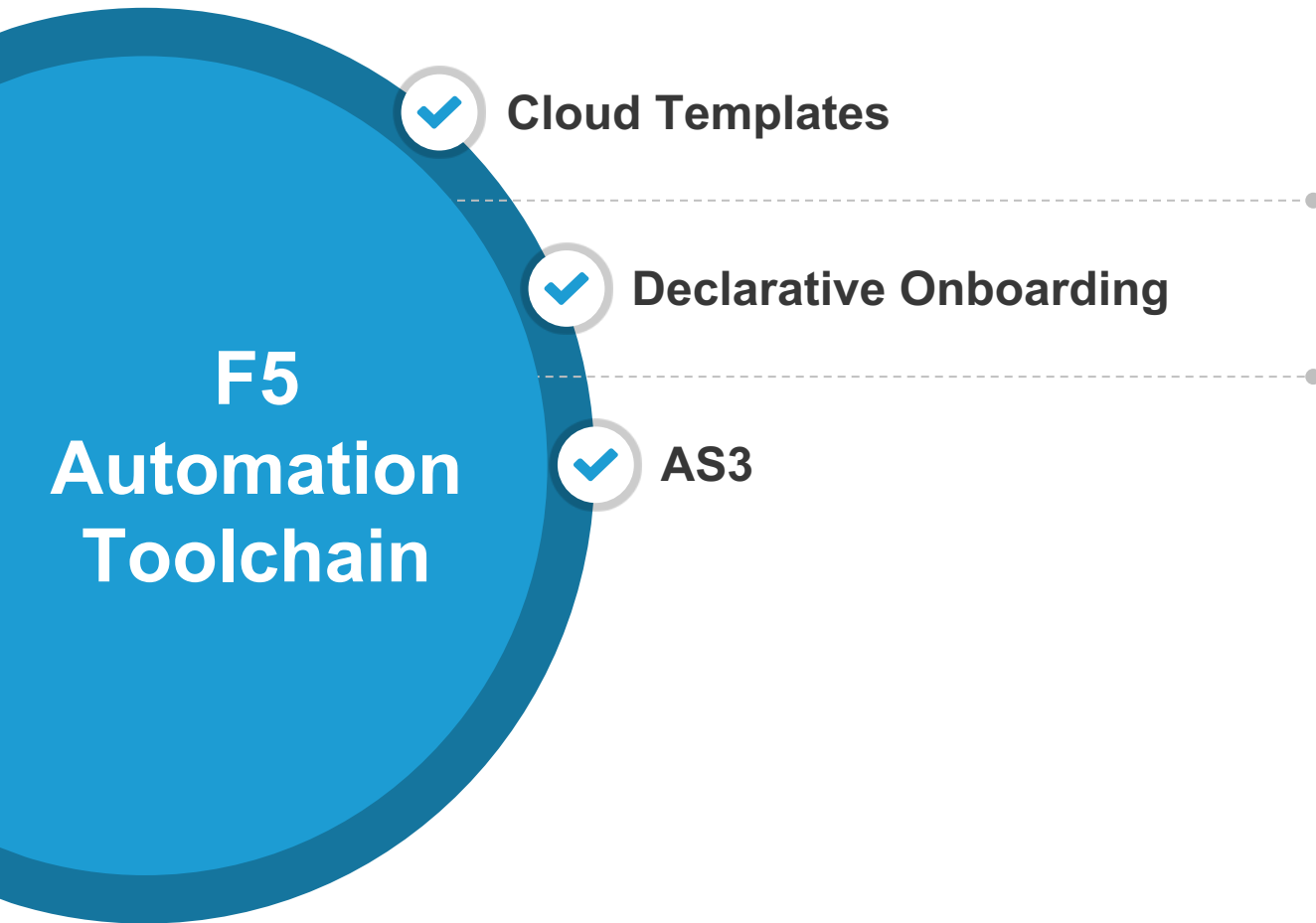


•••••

## F5 Declarative Onboarding

Provisions initial configuration of virtual BIG-IP devices through a declarative API.

# F5 Automation Toolchain



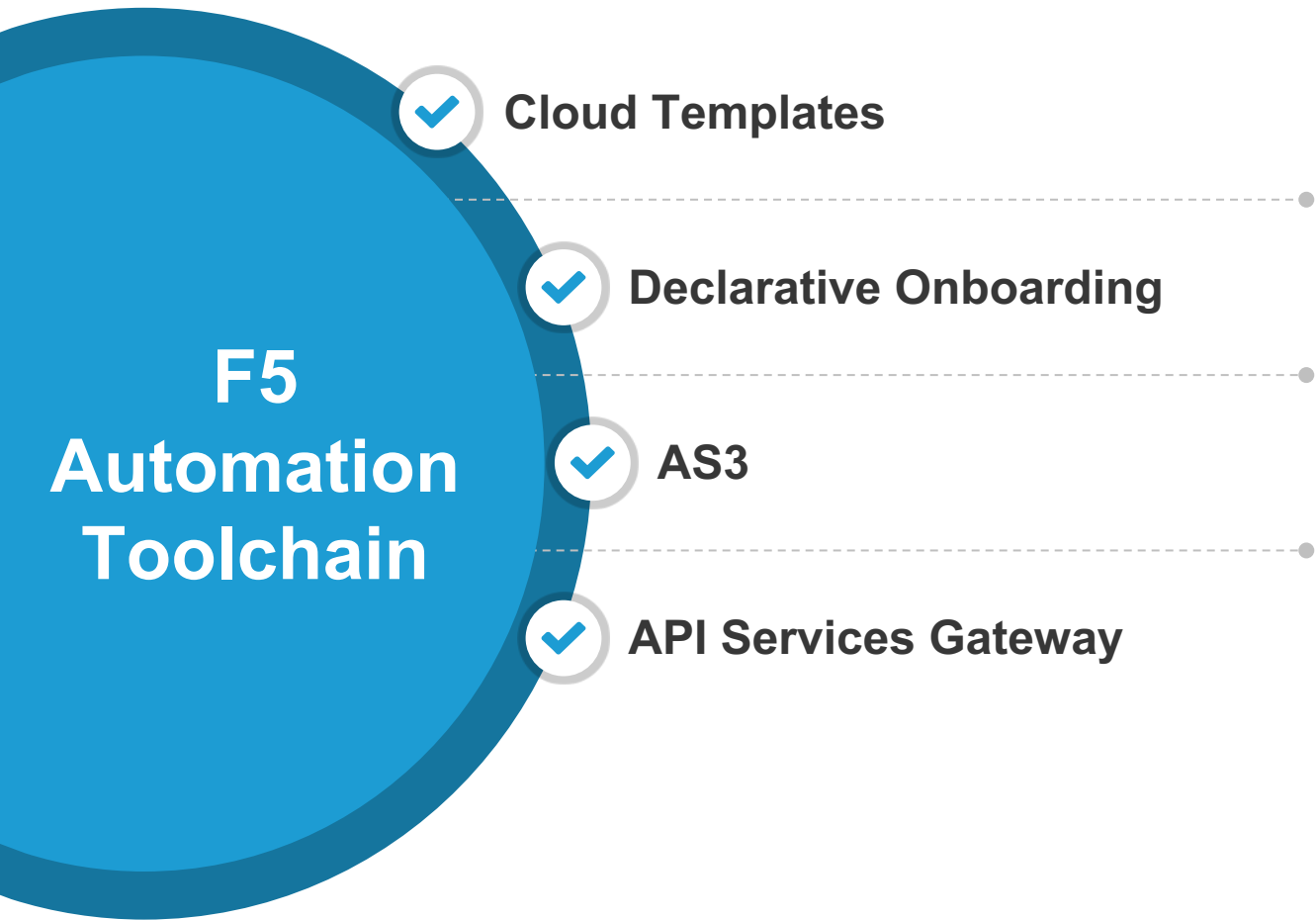
...

## F5 Application Services 3 Extension

Configure F5 modules using declarative APIs. Runs on TMOS, as a container, or in BIG-IQ 6.1+



# F5 Automation Toolchain



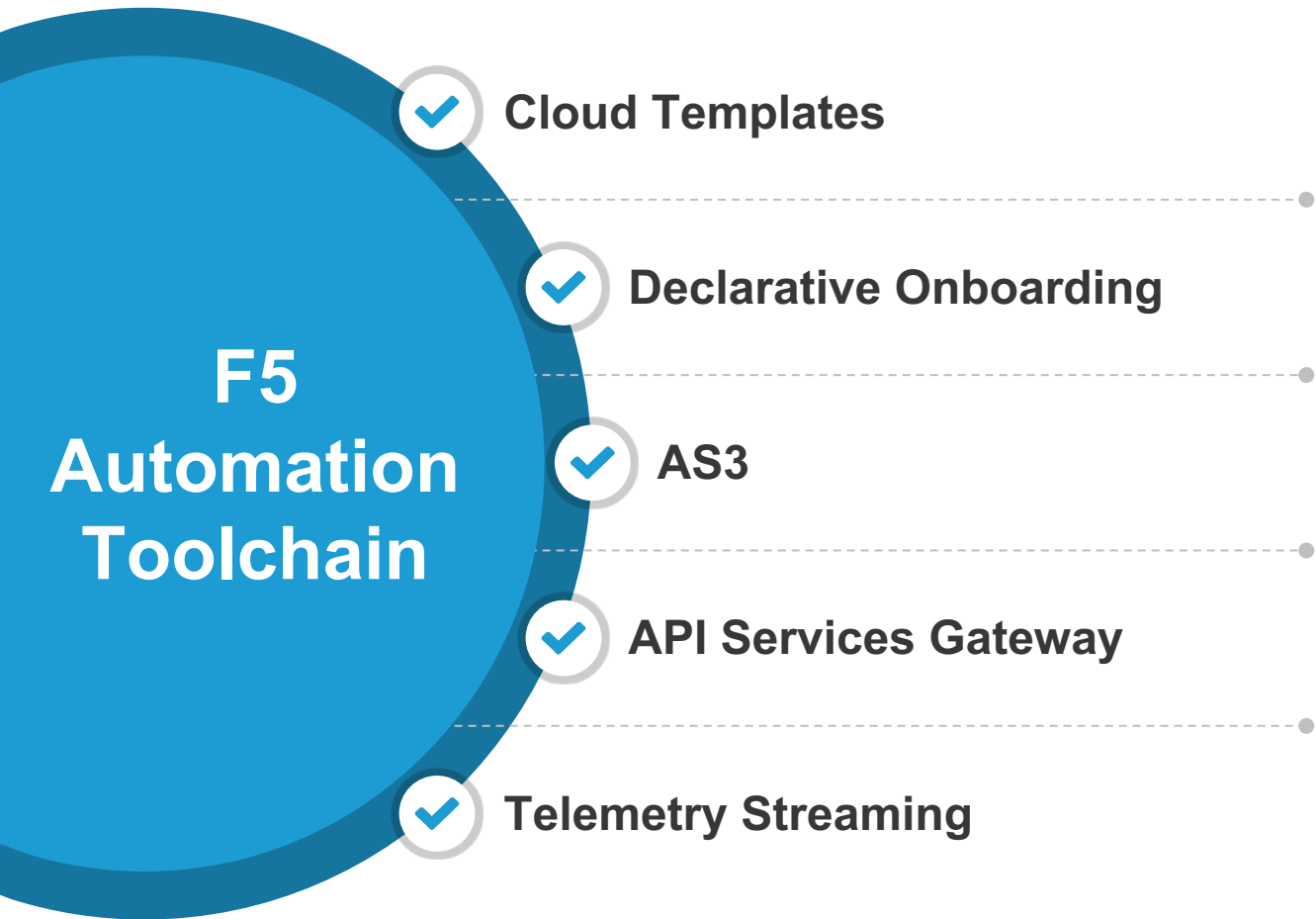
....

## F5 API Services Gateway

Runs F5 & Customer BIG-IP API Extensions in a container or VM platform.

---

# F5 Automation Toolchain



## F5 Telemetry Streaming

Stream BIG-IP telemetry for analytics and automation.

Telemetry Streaming is an BIG-IP API extension that will send client/server stats from the BIG-IP in Kafka format.