



# F5 Automation via CI/CD

# Agenda

Why CI/CD?

Jenkins

Pipelines and Webhooks

Build Environment

F5 A&O Toolchain

Demo

Q&A Session

# Automation Series

A photograph of two individuals, a man and a woman, sitting at a desk in a dark room. They are looking intently at a computer monitor which displays a terminal window with multiple lines of command-line text. The woman has curly hair and is wearing a dark t-shirt. The man is partially visible behind her. A desk lamp is positioned above the monitor, casting light on the scene.

Webinar 1

Automating F5 BIG-IP with Ansible

Webinar 2

Infrastructure-as-Code

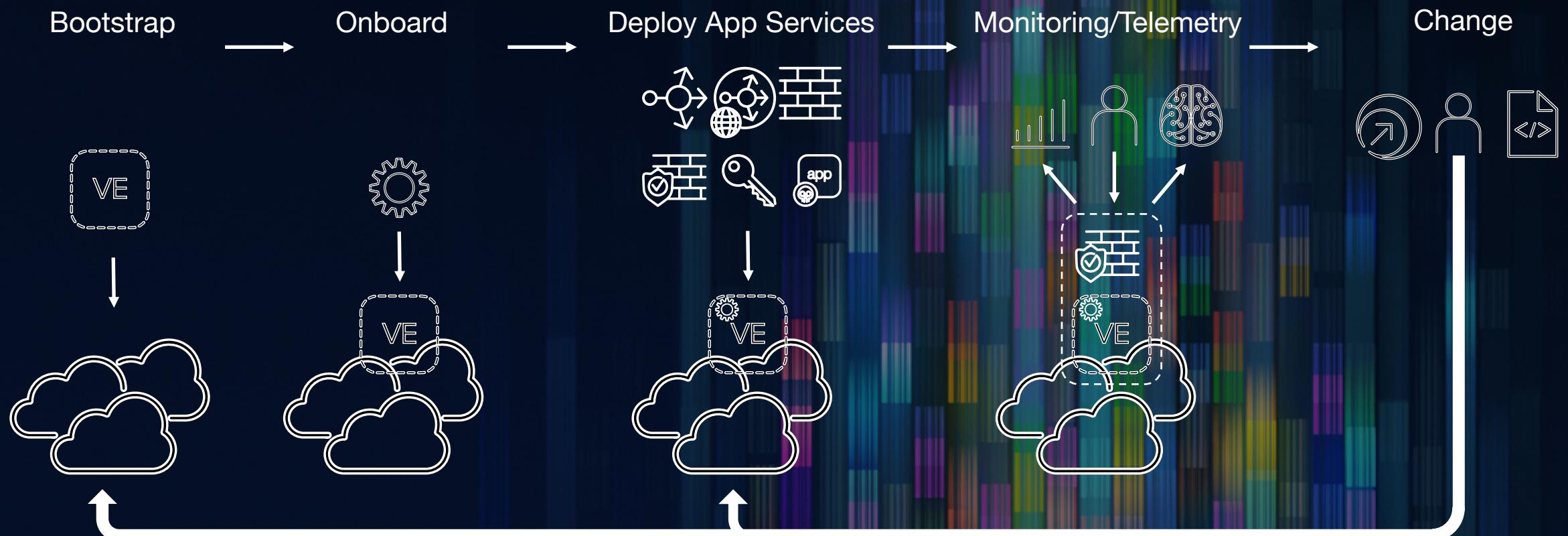
Webinar 3

Automation via CI/CD Pipelines

# Why CI/CD?

# Automation Lifecycle

DEPLOYING JUST AN APP IS NOT ENOUGH



# 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>



# Jenkins

# Jenkins

## Why Jenkins:

- Leading OSS automation server
- Hundreds of plugins
- Easy to Install and Configure

## Many other CI tools:

- Azure Pipeline
- Circle CI
- Travis CI



**Jenkins**

**cd**

**CD.FOUNDATION**

The screenshot shows a code editor interface with several tabs at the top: Jenkinsfile, main.tf, output.tf, config-bigip.yaml, pipeline, and demo-app.yaml. The Jenkinsfile tab is active. The main content area displays a Terraform configuration file for deploying Jenkins. It includes sections for ingress and egress rules, finding an Ubuntu AMI, creating IAM roles and profiles, and defining an AWS instance profile. The Groovy script part handles assume role policies and defines an AWS instance resource.

```
148 ingress {
149   from_port  = 443
150   to_port    = 443
151   protocol   = "tcp"
152   cidr_blocks = ["${chomp(data.http.myIP.body)}/32"]
153 }
154
155 egress {
156   from_port  = 0
157   to_port    = 0
158   protocol   = "-1"
159   cidr_blocks = ["0.0.0.0/0"]
160 }
161
162 # Find Ubuntu AMI
163
164 data "aws_ami" "compute" {
165   most_recent = true
166   owners      = ["amazon"] # Canonical
167
168   filter {
169     name  = "name"
170     values = ["amzn-ami-hvm*"] # Ubuntu Bionic
171   }
172
173   filter {
174     name  = "virtualization-type"
175     values = ["hvm"]
176   }
177 }
178
179
180 resource "aws_iam_instance_profile" "cicd" {
181   name = "lab_profile"
182   role = "${aws_iam_role.role.name}"
183 }
184
185 resource "aws_iam_role" "role" {
186   name = "lab_role"
187   path = "/"
188
189   assume_role_policy = <<EOF
190
191   "Version": "2012-10-17",
192   "Statement": [
193     {
194       "Action": "sts:AssumeRole",
195       "Principal": {
196         "Service": "ec2.amazonaws.com"
197       },
198       "Effect": "Allow",
199       "Sid": ""
200     }
201   ]
202 EOF
203 }
204
205 resource "aws_instance" "cicd" {
206   ami = "${data.aws_ami.compute.id}"
```

# Configure Jenkins

## AUTOMATION FOR THE WIN!

### Webinar GitHub Repository:

- [https://github.com/codygreen/Automation\\_Webinar](https://github.com/codygreen/Automation_Webinar)
- Look in side the code/3 – CICD folder/setup
- Terraform and Groovy script to build and deploy Jenkins in EC2
  - Terraform creates:
    - AWS user
    - VPC
    - security group
    - ec2 instance
    - Configure ec2 instance
  - Groovy script configures base security and settings for Jenkins

# Pipelines and Webhooks

# CD Pipelines

Pipelines allow you to automate the process of:

- Getting software from version control (GIT)
- Build required environment (IoC)
- Test environment
- Blue/Green or Canary Deployments





# Jenkins Pipeline

Collection of plugins that provide an automated expression of your processes.

Configurable through the GUI or via version control using Groovy script

Procedural or Declarative

- Ensure when looking for examples you stick with one or the other

# Webhooks

Provides a way to remotely trigger a pipeline

Typically used on master branch or unique pipelines are created for each branch

Jenkins can automate the creation of webhooks for you



# Build Environment

# Ansible & Terraform

**Best of both worlds:**

Terraform builds infrastructure and maintains state

- Webinar #2 covered Terraform basics

Ansible configures deployed instances and BIG-IP

- Webinar #1 covered Ansible basics





ANSIBLE

# Ansible

One of the most popular automation tools

Used primarily for configuration management

Agentless, does not require a master node/server

Modules make it easy to automate common tasks

# Terraform

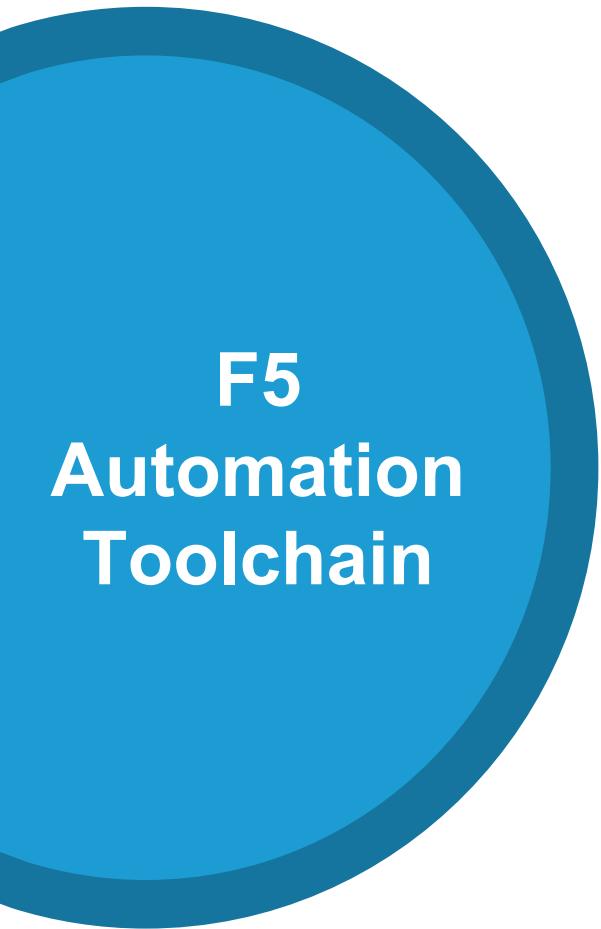
Ability to define desired infrastructure end state and manage the planning and deployment of said end state repeatably in multiple environments

- Easily manage infrastructure using code repositories
- Ability to see what changes will be made using plan method
- Powerful state management



# F5 Automation Toolchain

# F5 Automation Toolchain



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

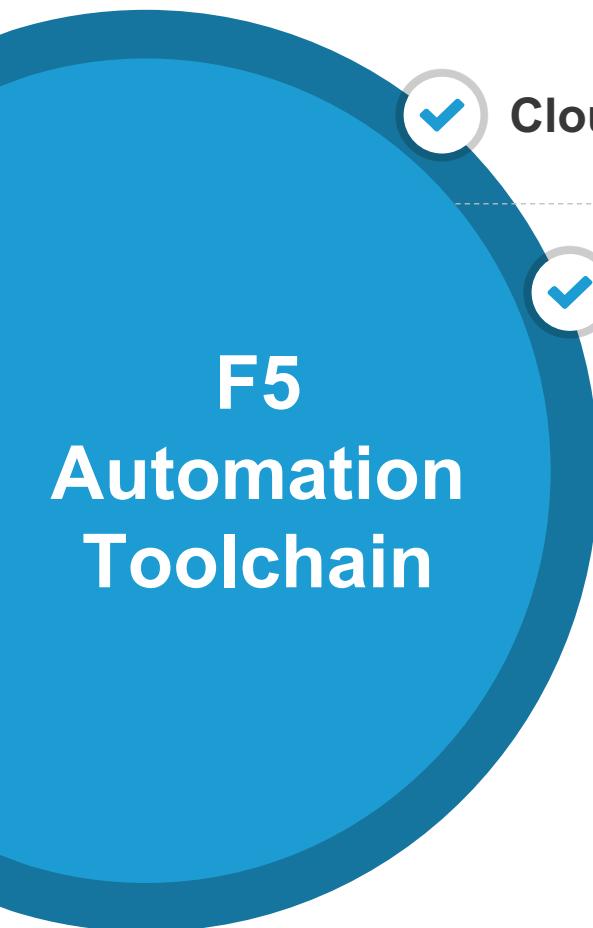


## F5 Cloud Templates

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

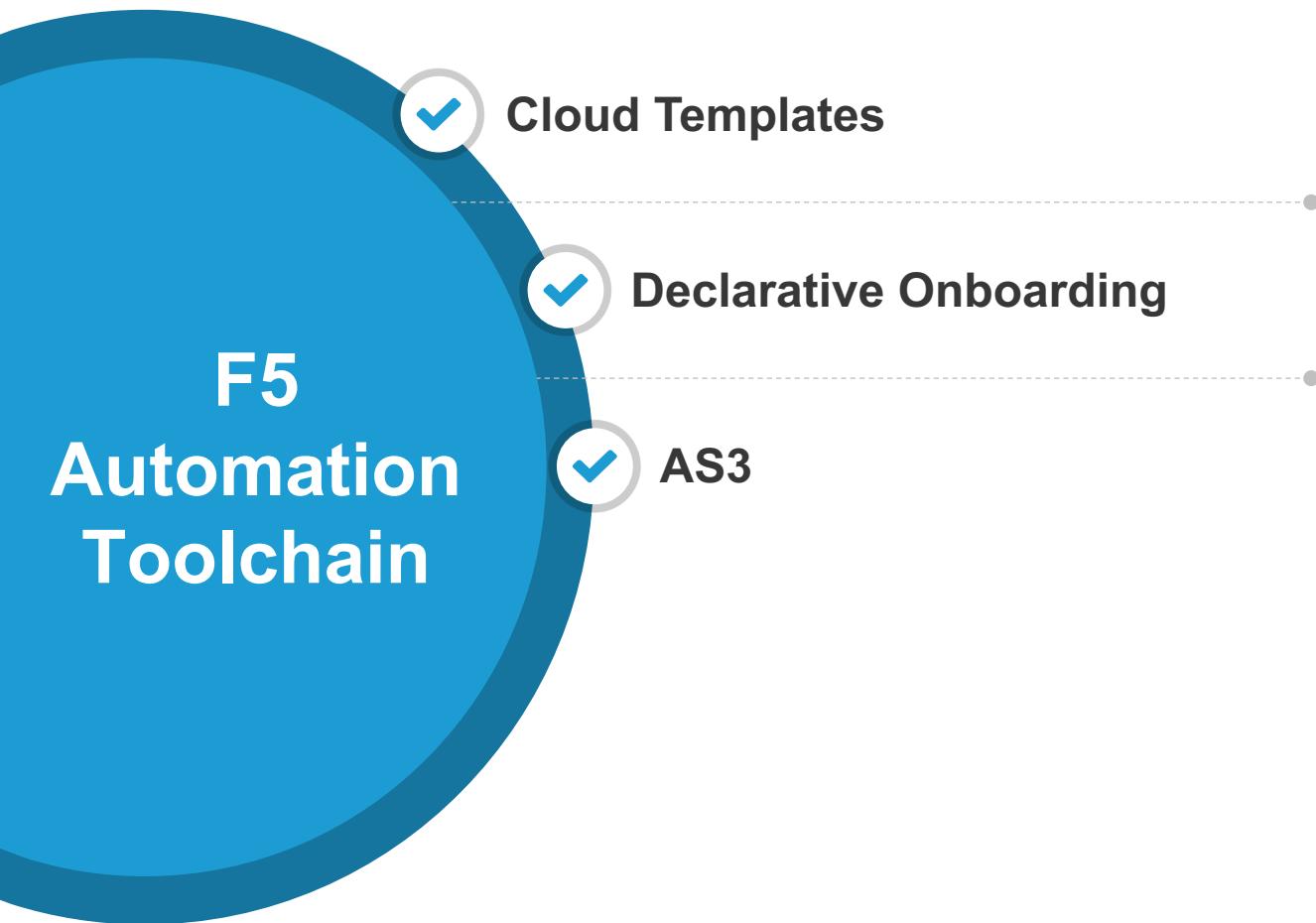
Leverage each cloud's native services to support a wide variety of BIG-IP VE use cases.

# F5 Automation Toolchain



...  
**F5 Declarative Onboarding**  
Provisions initial configuration of virtual BIG-IP devices through a declarative API.

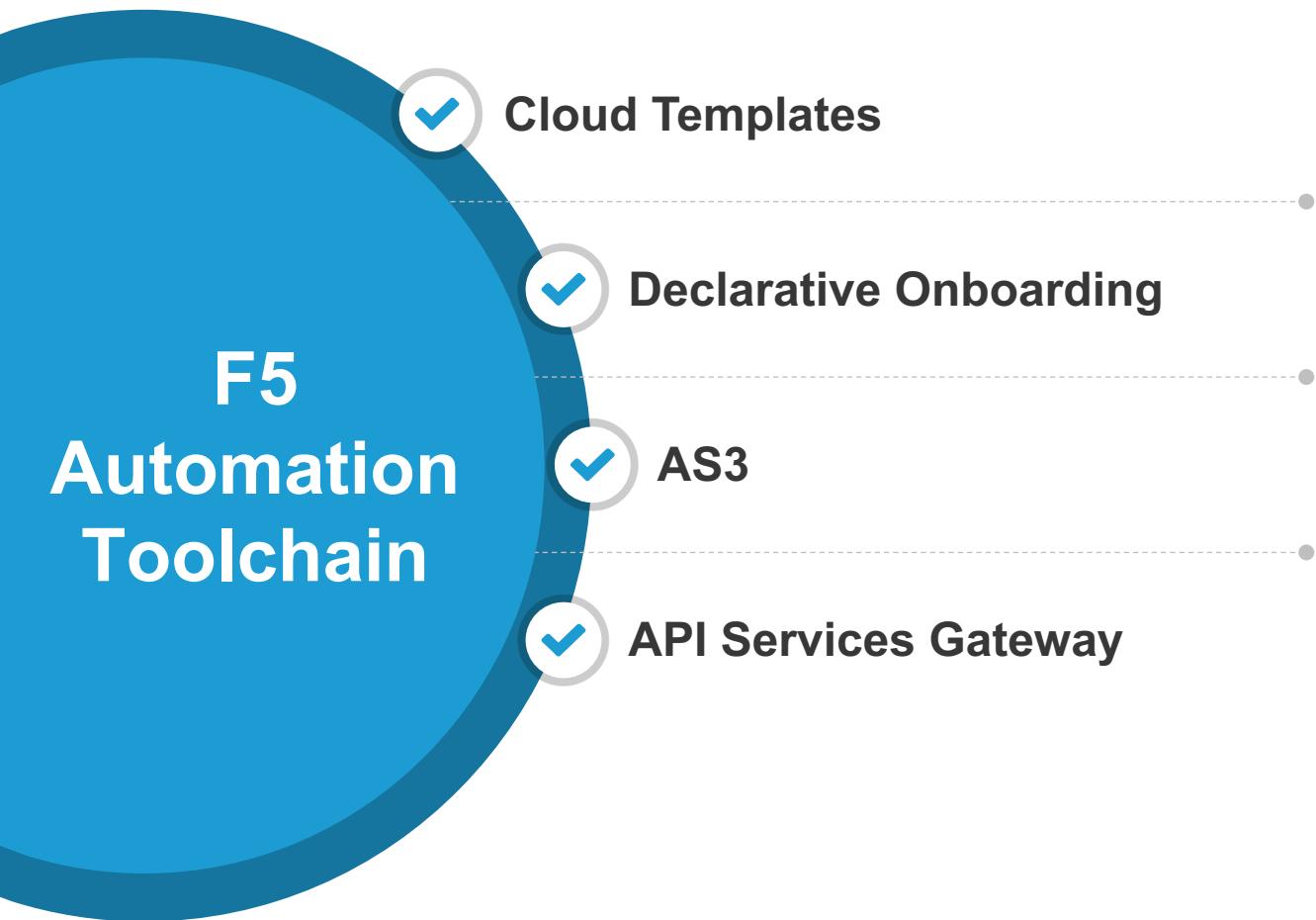
# F5 Automation Toolchain



## F5 Application Services 3 Extension

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

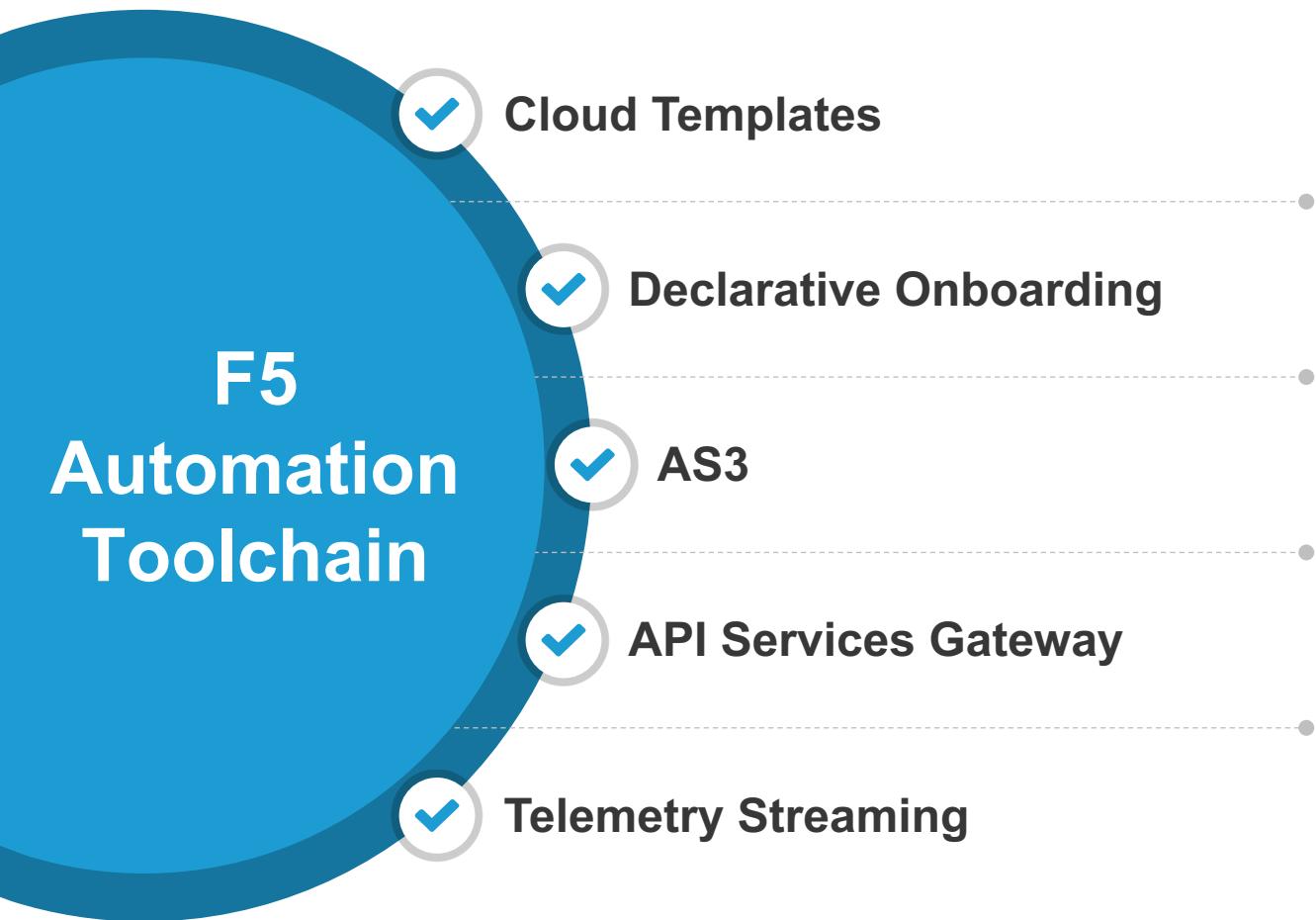
# F5 Automation Toolchain



## 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.

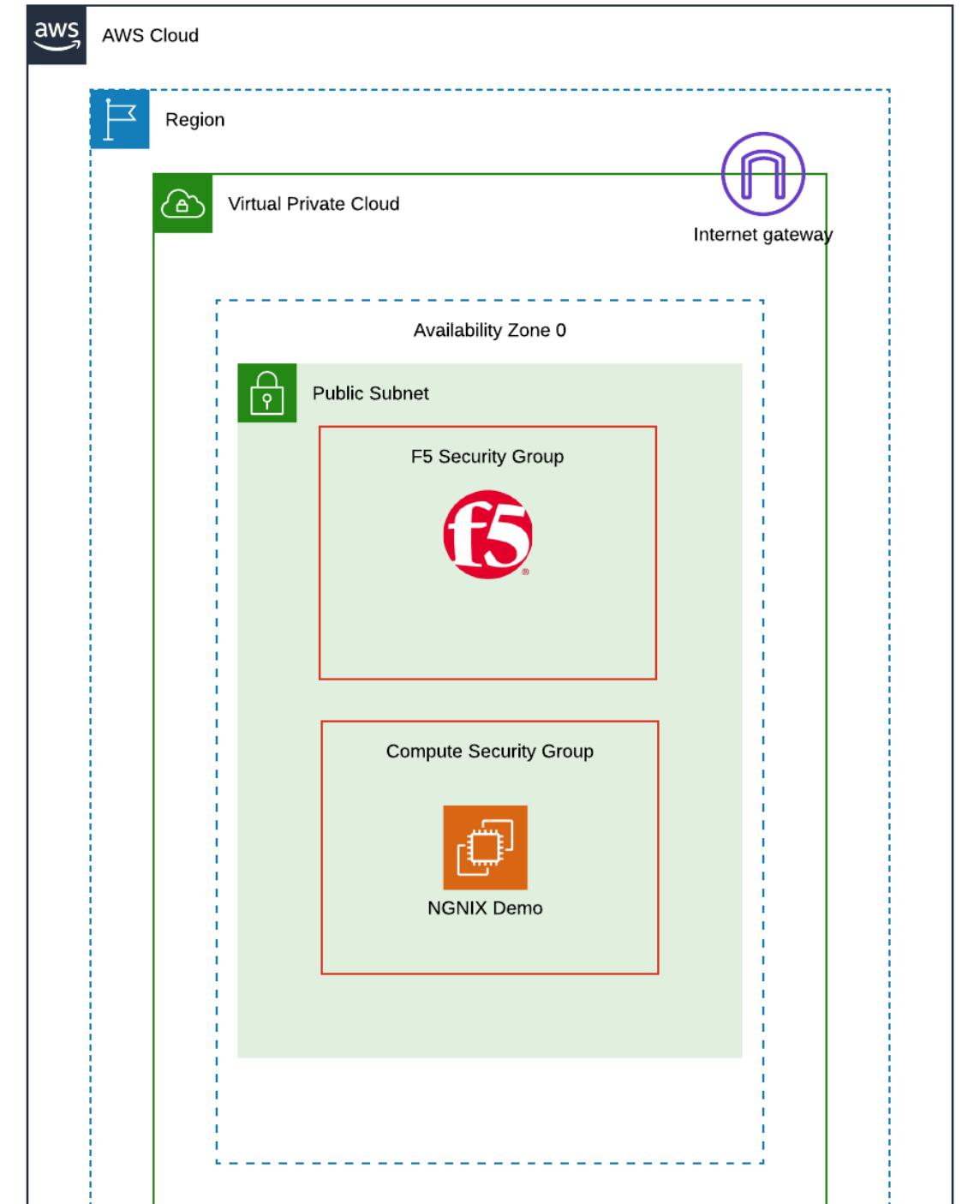
# Demo

# Environment

Simple lab setup

Built using Terraform

EC2 Instance and BIG-IP configured via  
Ansible



# Q&A

