F5 AND HASHICORP

INFRASTRUCTURE AS CODE - BIG-IP AND TERRAFORM BASICS

October 2020

WELCOME AND INTRODUCTIONS


F5 is the leader in
Multi-Cloud
Application Services
that enhance and secure your Application Capital.
49/50

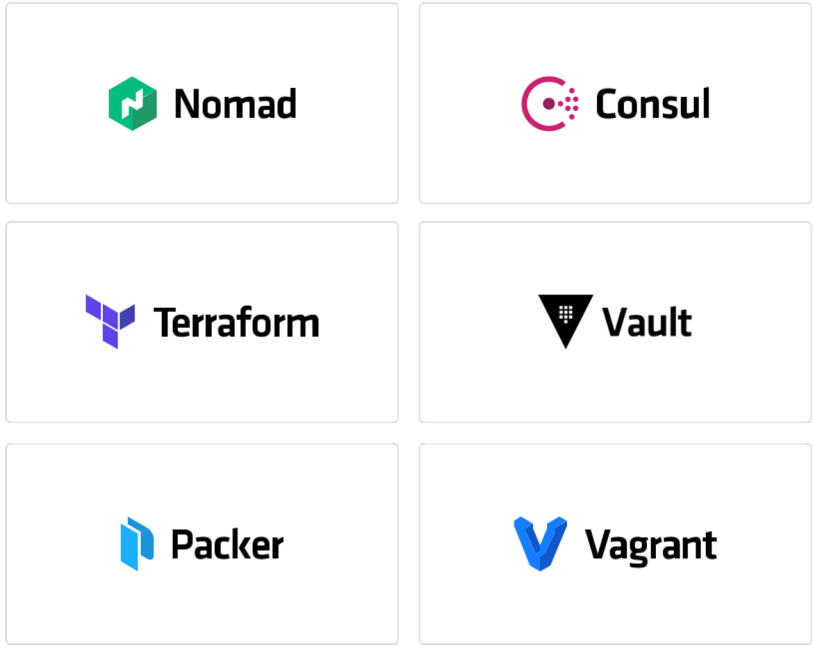
About HashiCorp

Leading Cloud Infrastructure Automation

Our software stack enables the provisioning, securing, connecting and running of apps and the infrastructure to support them.

We unlock the cloud operating model for every business and enable their digital transformation strategies to succeed.

Founded	Employees	Funding
2012	1000	349M



AUTOMATION & ORCHESTRATION

WHY SHOULD YOU CARE?

What's the Problem?

WHY DO ORGANIZATIONS USE AUTOMATION FRAMEWORKS?

REDUCE OPEX

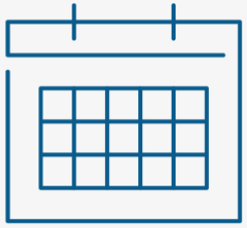
71%

SCALING TO MEET DEMAND

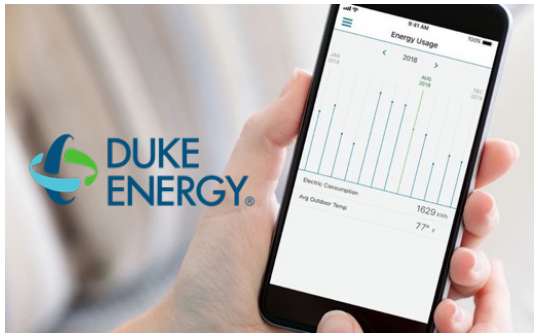
51%

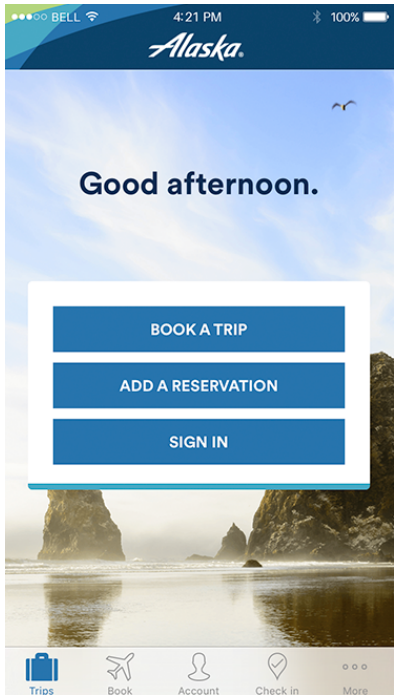
TIME TO MARKET

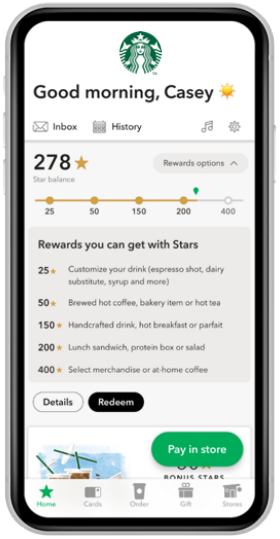
43%



Everyone is in the digital experience business







To overcome these challenges, we believe applications should have the capability to adapt

Our vision is that an application, like a living organism, will naturally adapt based on the environment, becoming an adaptive application.

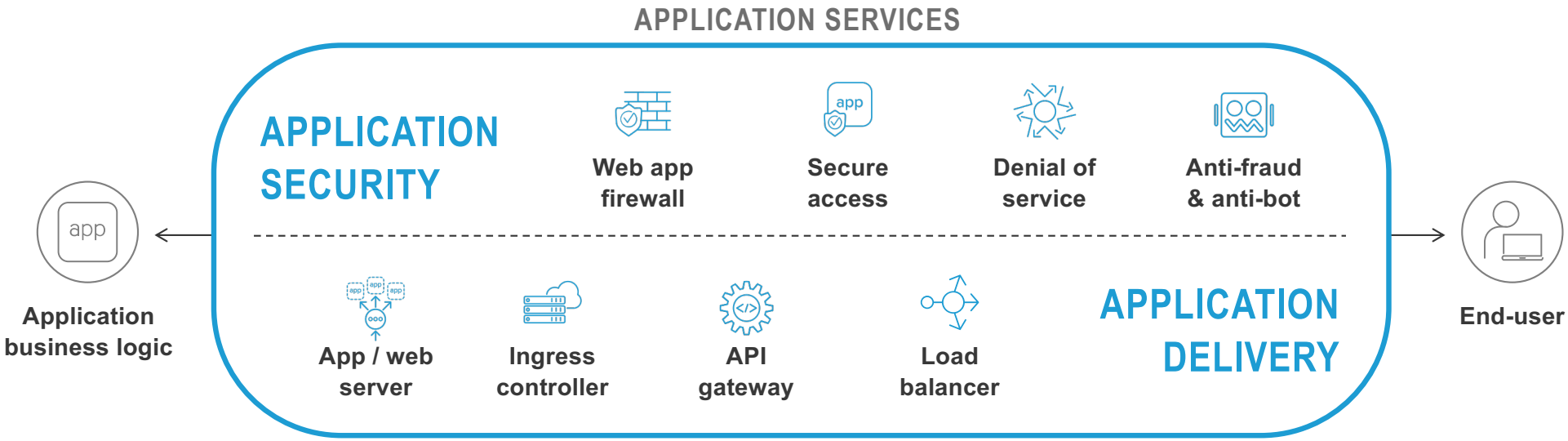
GROWS AS NEEDED

SHRINKS AS NEEDED

DEFENDS ITSELF

HEALS ITSELF

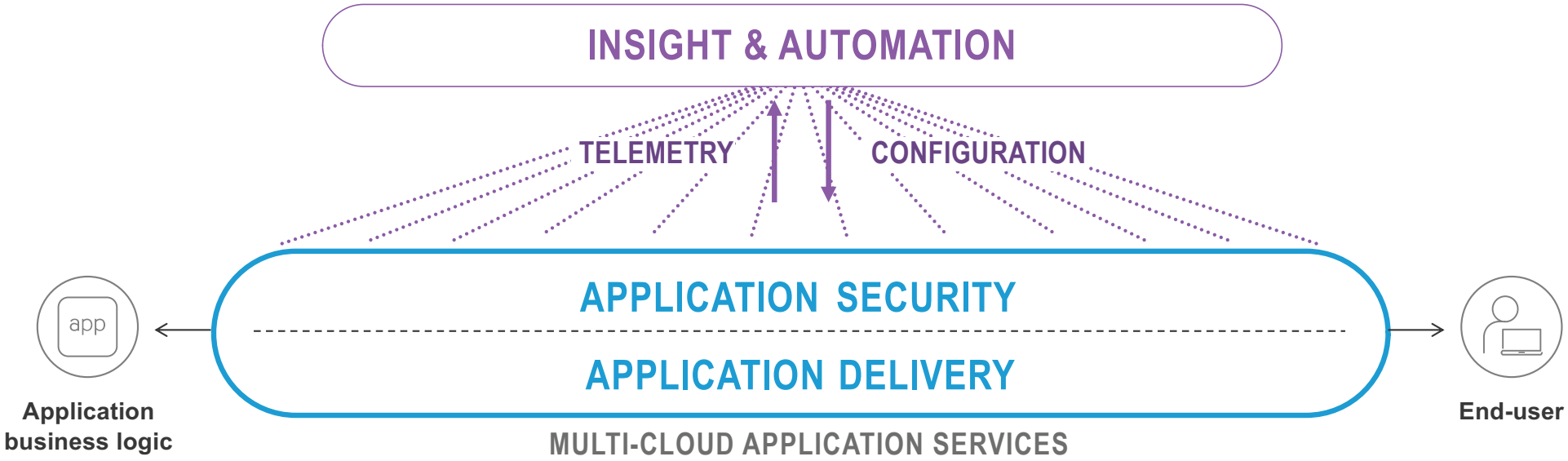
Application services are the foundation for fast and secure digital customer experiences



Ensure app availability and responsiveness

Secure app data, APIs, and traffic flows

Adaptive applications require that application services be automatable, with consistent policy, multi-cloud, and intelligent



Automation efficiency

Security consistency

Performance reliability in multi-cloud

Insight into your app portfolio

Hashicorp Terraform

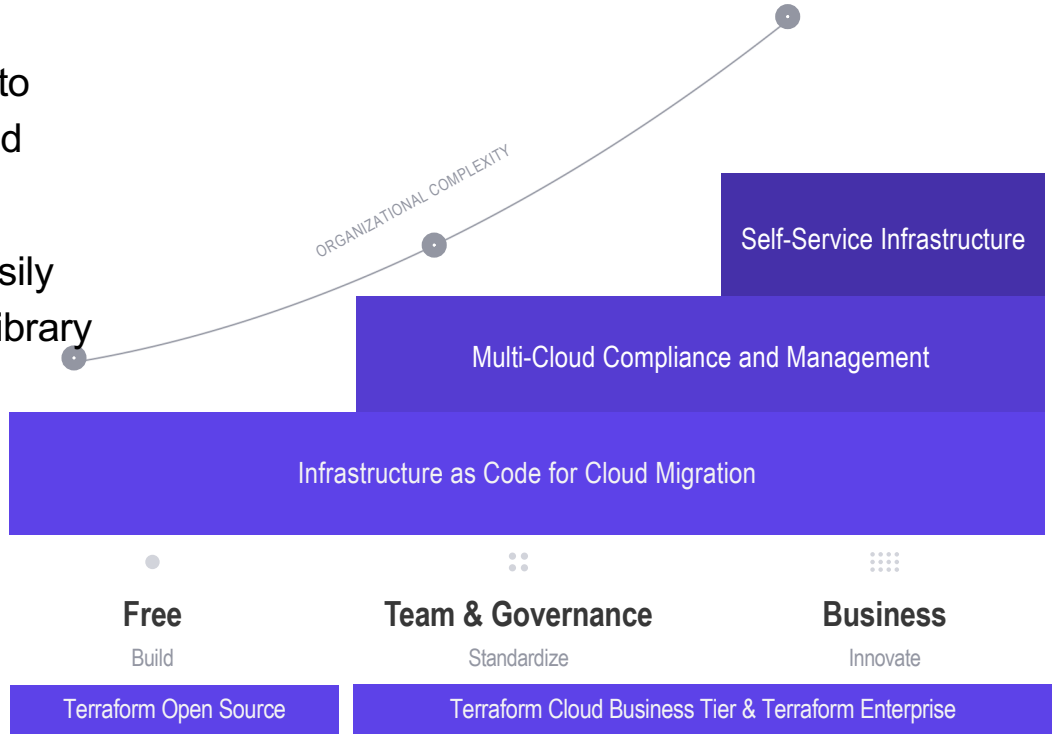
Provides foundation for infrastructure as code for provisioning and compliance in the cloud operating model

- **Multi-Cloud Compliance & Management** to provision and manage any infrastructure and compliance with one workflow
- **Self-Service infrastructure** for users to easily provision infrastructure on-demand with a library of approved infrastructure modules

200
Providers

100K+
Weekly D/Ls

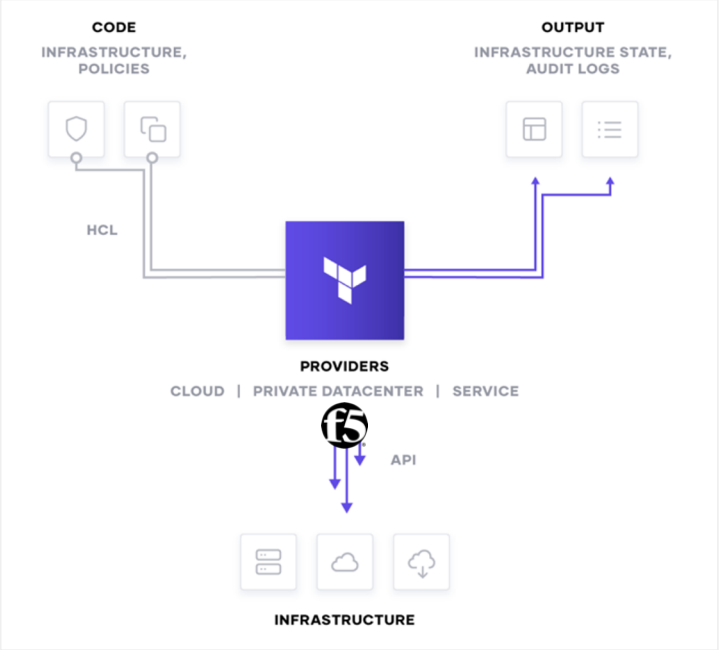
300+
Customers



Guiding Principle:

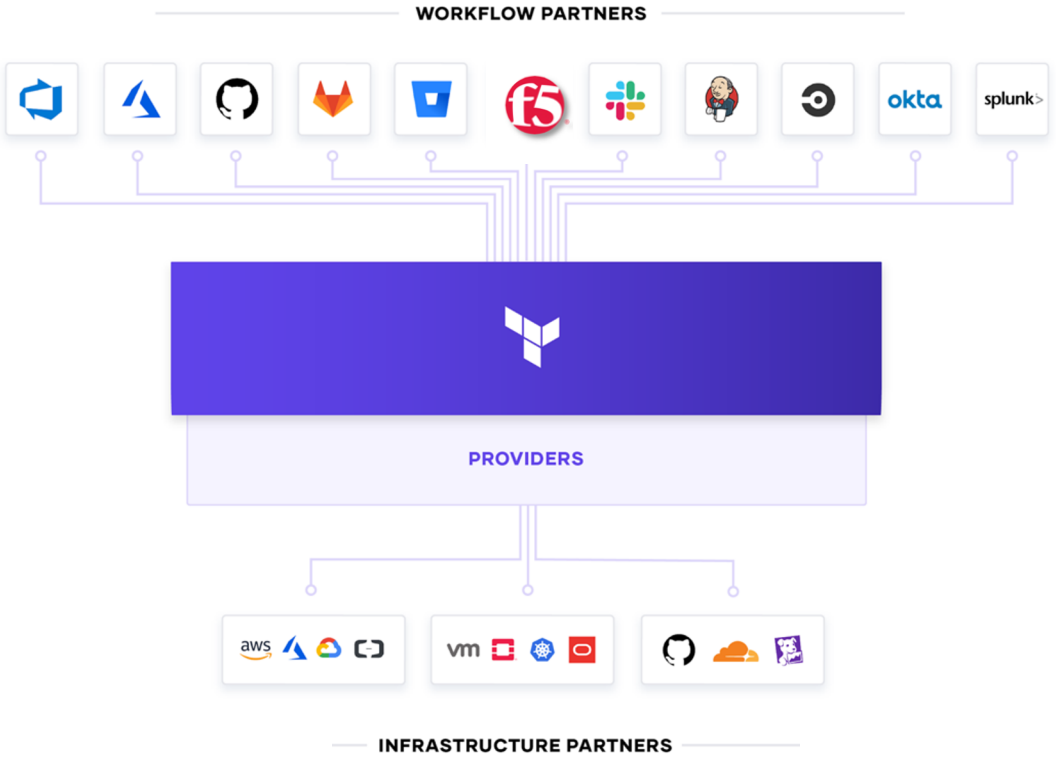
Infrastructure as Code

- Using version control and automation to reduce human error and failed builds
- Terraform infrastructure as code and policy as code to automate everything
- Open source providers allow rapid creation and support for any infrastructure



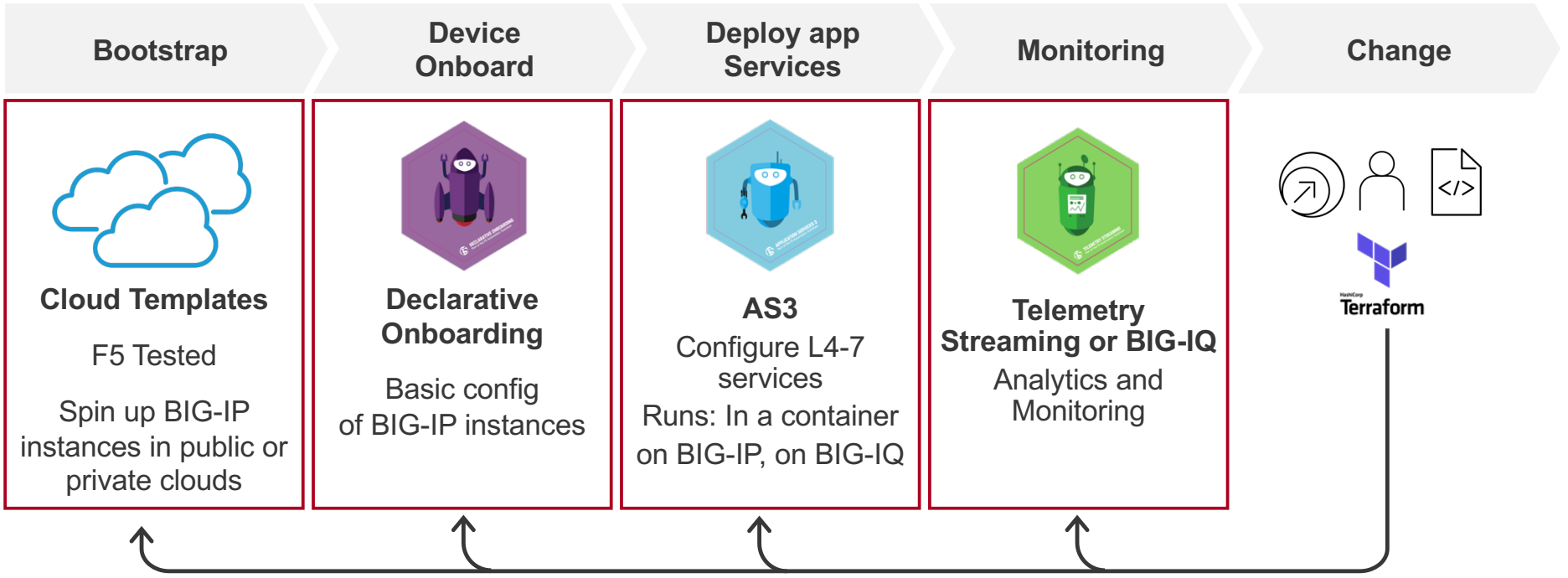
Broad Ecosystem

Open Source
and
Enterprise Integrations



The F5 Automation Toolchain

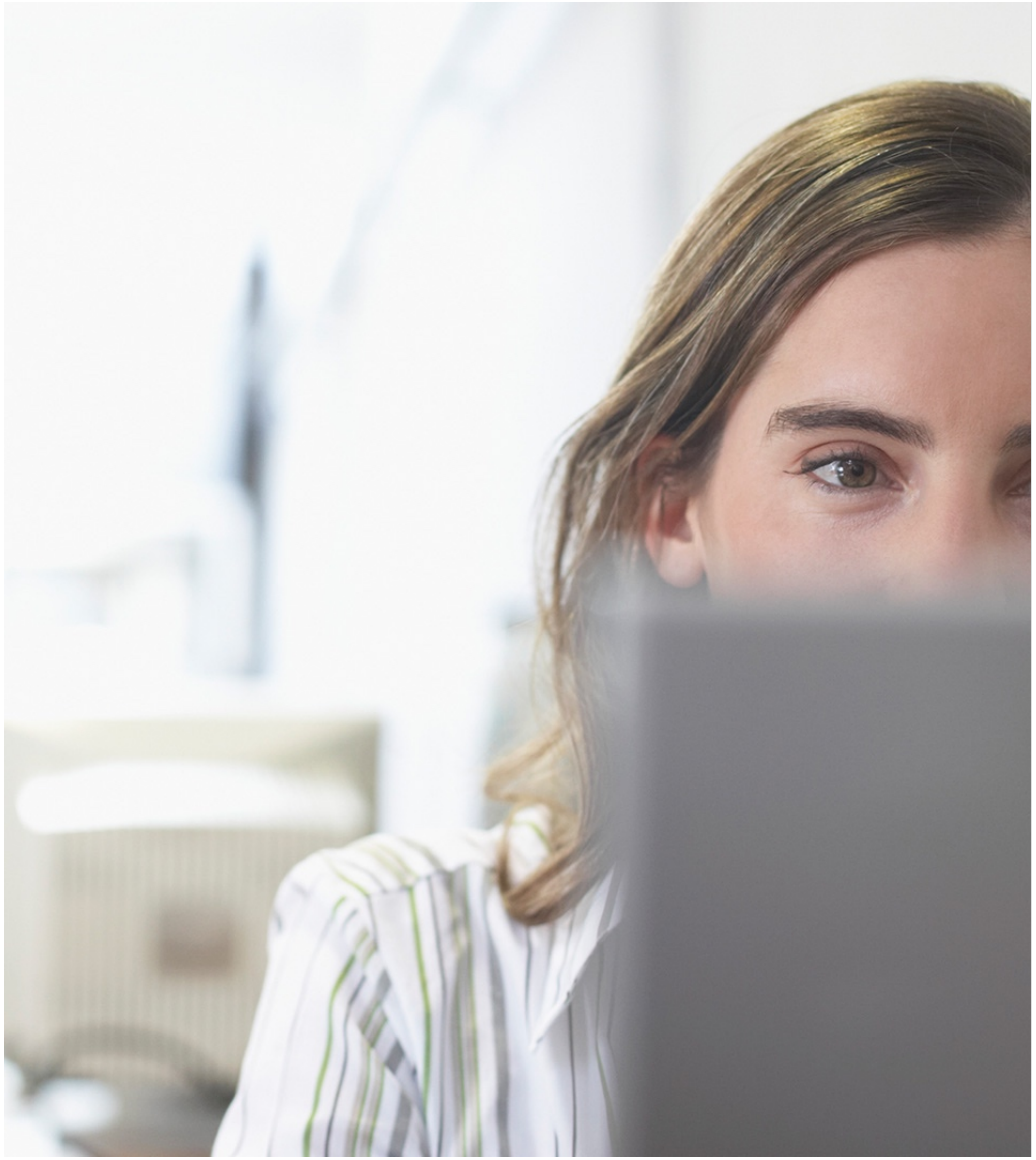
- Declarative API – Abstracts config complexity with a single API call



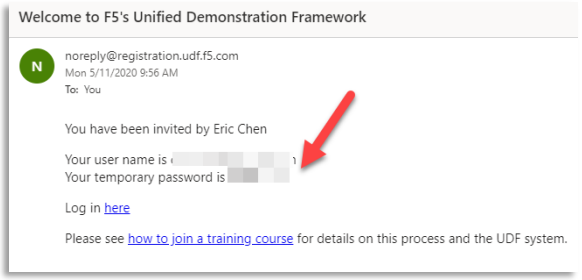
ACCESS THE LAB ENVIRONMENT

Access Lab

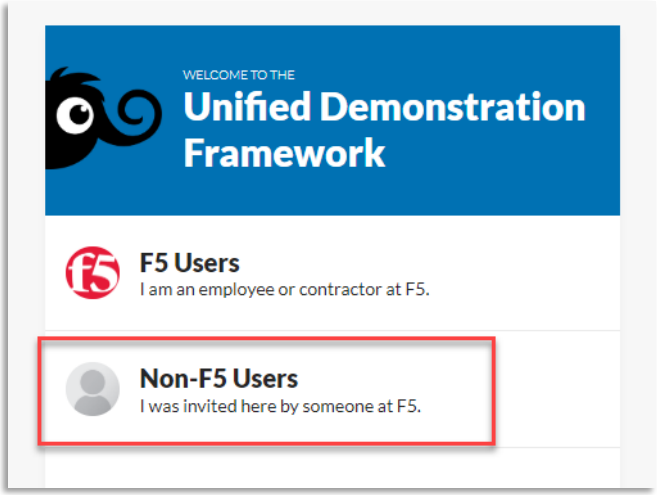
- Register with UDF
- Join Course
- Lab Guide



Access the lab environment



#1 Should have received email from UDF



#2 Login at https://udf.f5.com
(reset password, Accept T&C)

#3 Launch Course

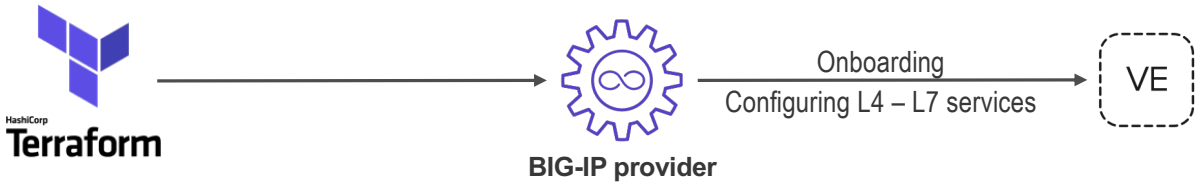
Thu 17 Sep	**12:00 PM** - 8:00 PM CDT Duration: 8 hours	F5 and Hashicorp Basics	Virginia, USA	Central	Ed Rabago & 6 oth…	UNREGISTER LAUNCH

LAB 1

BASIC BIG-IP ADMINISTRATION WITH TERRAFORM

F5 BIG-IP PROVIDER

A Terraform provider for F5 BIG-IP. Resources are currently available for LTM



Written in Go Lang, uses F5 GO SDK

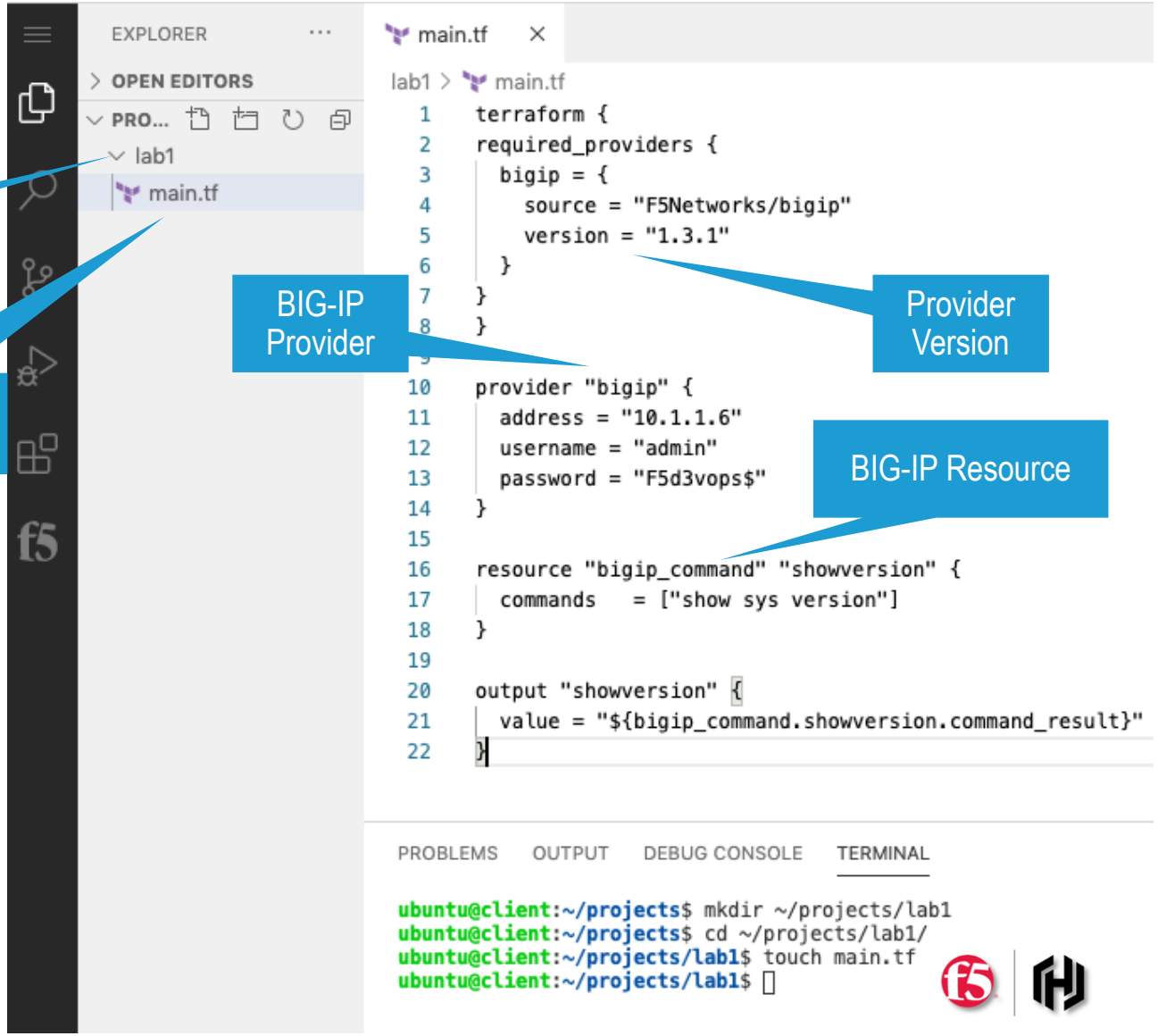
Uses the iControl REST API

All the resources are validated with TMOS v12.1.1

Imperative approach to configure BIG-IP with:

- Networks configurations
- Device group and HA configurations
- VIPs, Pools, Monitors, Profiles and other LTM configurations
- Data groups, iRules, SNMP

Terraform Module



Terraform init

terraform init initializes various settings and data, download provider plugin and install in a subdirectory of the current working directory

```
ubuntu@client:~/projects/lab1$ terraform  init

Initializing the backend...

Initializing provider plugins...
- Finding f5networks/bigip versions matching "1.3.1"...
- Installing f5networks/bigip v1.3.1...
- Installed f5networks/bigip v1.3.1 (signed by a HashiCorp partner, key ID 8

Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it her
https://www.terraform.io/docs/plugins/signing.html

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to se
any changes that are required for your infrastructure. All Terraform commanc
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, ot
commands will detect it and remind you to do so if necessary.
ubuntu@client:~/projects/lab1$
```

Provider Version

Terraform plan

Terraform plan is used to create an execution plan which is a way to check whether the set of changes matches our expectation without making any infrastructure

Add resource items

# of resouces to add

```
ubuntu@client:~/projects/lab1$ terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.


------------------------------------------------------------------------

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # bigip_command.showversion will be created
  + resource "bigip_command" "showversion" {
      + command_result = (known after apply)
      + commands       = [
          + "show sys version",
        ]
      + id             = (known after apply)
      + when           = "apply"
    }

Plan: 1 to add, 0 to change, 0 to destroy.

------------------------------------------------------------------------

Note: You didn't specify an "-out" parameter to save this plan, so Terraform
can't guarantee that exactly these actions will be performed if
"terraform apply" is subsequently run.

ubuntu@client:~/projects/lab1$
```

Terraform apply

terraform apply command is used to apply the changes required to reach the desired state.

terraform.tfstate file will be created to represent state after the apply.

```
ubuntu@client:~/projects/lab1$ terraform apply

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # bigip_command.showversion will be created
  + resource "bigip_command" "showversion" {
      + command_result = (known after apply)
      + commands       = [
          + "show sys version",
        ]
      + id             = (known after apply)
      + when           = "apply"
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

bigip_command.showversion: Creating...
bigip_command.showversion: Creation complete after 1s [id=apply]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

showversion = [
  "\nSys::Version\nMain Package\n  Product     BIG-IP\n  Version     16.0.0
     Tue Jun 23 18:31:26 PDT 2020\n\n",
]
ubuntu@client:~/projects/lab1$
```

# of resouces added

Start Lab 1

1. BIG-IP Infrastructure Onboarding - configure basic network configurations like NTP, DNS, and interface IP addresses

2. BIG-IP App Services Configuration - basic application delivery services such as nodes, pools, pool members and virtual servers

3. Modify and Destroy App Services Configuration - conduct basic add and delete administration

4. BIG-IP Onboarding wtih DO (Declarative Onbarding) – declaratively configure basic network configurations like NTP, DNS, and interface IP addresses with single API call

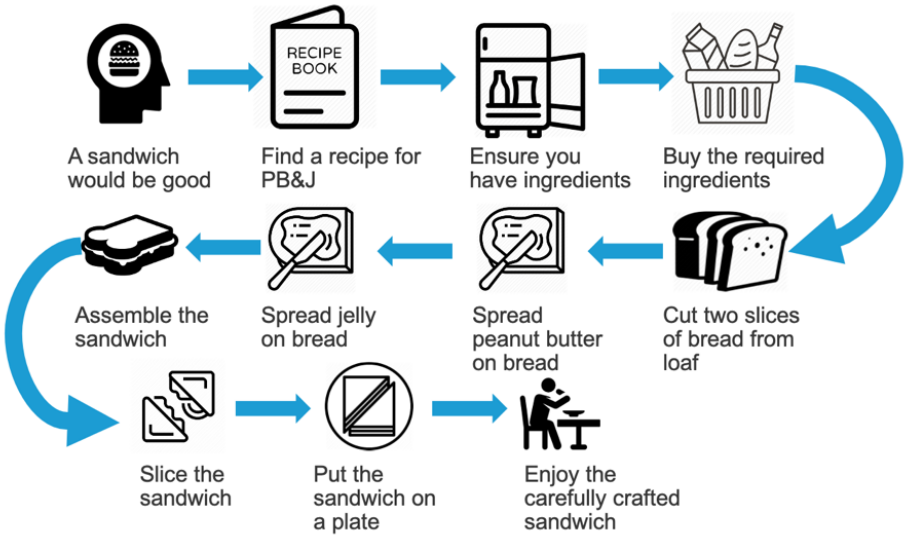
LAB 2

BASIC BIG-IP ADMINISTRATION WITH TERRAFORM

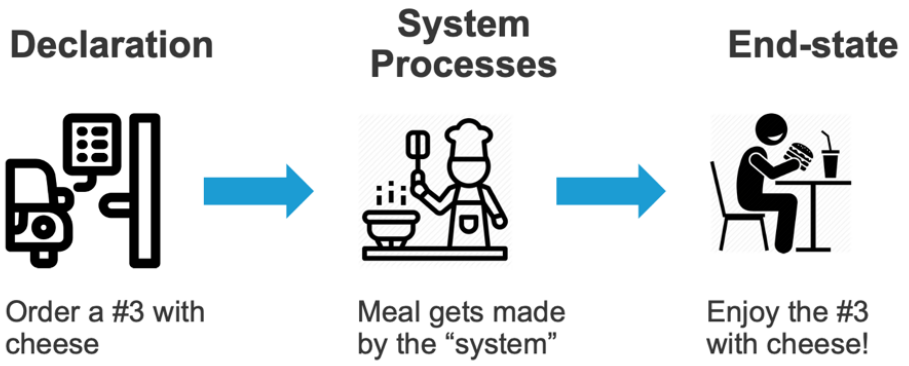
Understanding imperative vs. declarative

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.

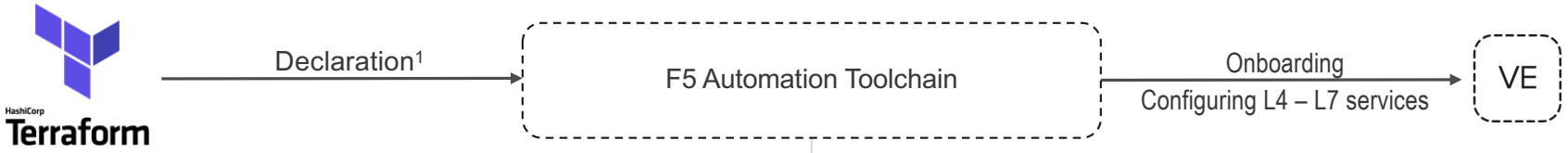


Declarative model: The model that F5 has aligned to. Just input the desired end-state and let the system figure out the rest.



F5 DO and AS3

Terraform will be able to onboard and configure ALL L4-L7 Services Declaratively



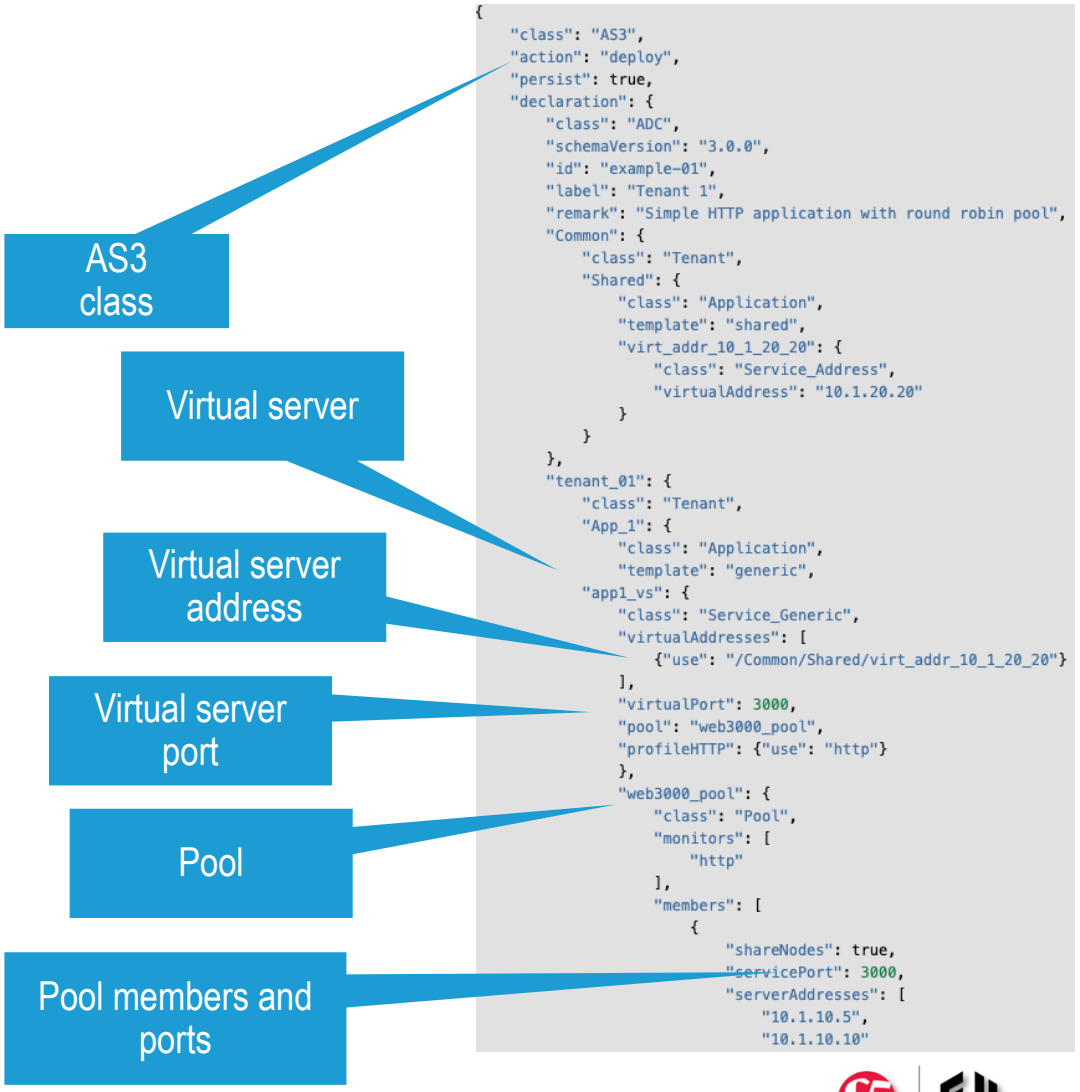
- Initial Config of BIG-IP Instances with DO
- Deploy Classic and Advanced Application Services on BIG-IP using Declarative REST APIs with AS3
- Service Discovery

- Declarative API – Abstracts config complexity with single API call
- User only needs to define the desired configuration end-state
- Requires no BIG-IP domain expertise
- Single resource simplifies automation and orchestrator integrations
- Accelerates app service deployment
- Declaration is reusable, ensures consistency, reduces errors

F5 Declarative API – AS3

Application Services 3 Extension (*AS3*) is a flexible, low-overhead mechanism for managing application-specific configurations on a BIG-IP system. AS3 uses a declarative model, meaning you provide a JSON declaration rather than a set of imperative commands.

Here are links to examples AS3 declarations and the BIG-IP LTM objects they create.



Start Lab 2

1. Deploy App Services via AS3- create json declaration to deploy app services using AS3

2. Deploy Canary Test Policy – create forwarding policy to direct test traffic to alternate pool when specified uri conditions is met

LAB 3

BASIC BIG-IP ADMINISTRATION WITH TERRAFORM

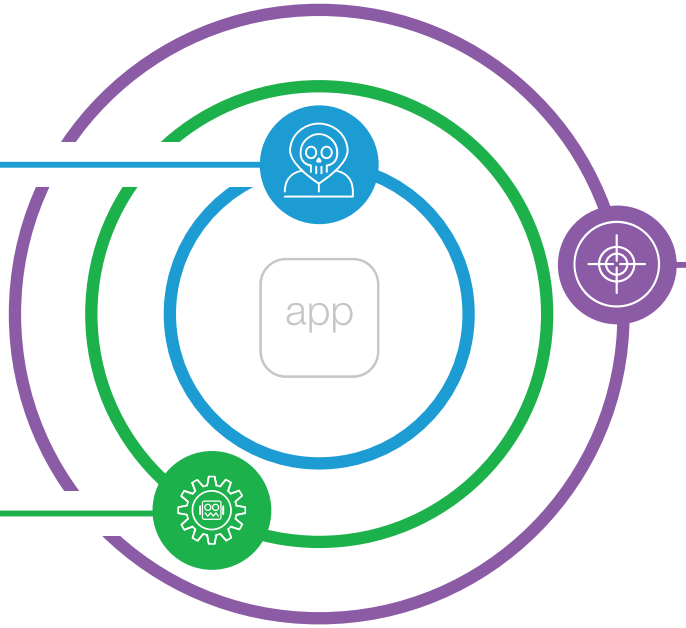
Application Layer Threats

FRAUD, ABUSE, & BUSINESS LOGIC ATTACKS

- Denial of Service (L7 and volumetric)
- Denial of Inventory
- Business Logic Attacks (fraud, intellectual property theft, abuse of service, abuse of brand)

TARGETED ATTACKS & ADVANCED THREAT ACTORS

- Application reconnaissance and profiling
- Site specific vulnerabilities
- Bypass of security services
- Development of exploits and methods used by bots and unwanted automation



SOFTWARE VULNERABILITIES & COMMON WEB EXPLOITS

- Known vulnerabilities (CVEs)
- OWASP Top 10 & common attack vectors
- Zero-day attacks

OWASP Top 10 Vulnerabilities

- Mostly the same vulnerabilities for last 20 years
- NetOps to DevOps deploy security
- Defense in depth
- Visibility is crucial
- WAF for **everyone**
- Layer 8: the human factor

2003 OWASP Top 10

1. Unvalidated parameters
2. Broken access control
3. Broken account and session management
4. Cross-site scripting (XSS)
5. Buffer overflows
6. Command injection flaws
7. Error handling problems
8. Insecure use of cryptography
9. Remote administration flaws
10. Web and application server misconfiguration

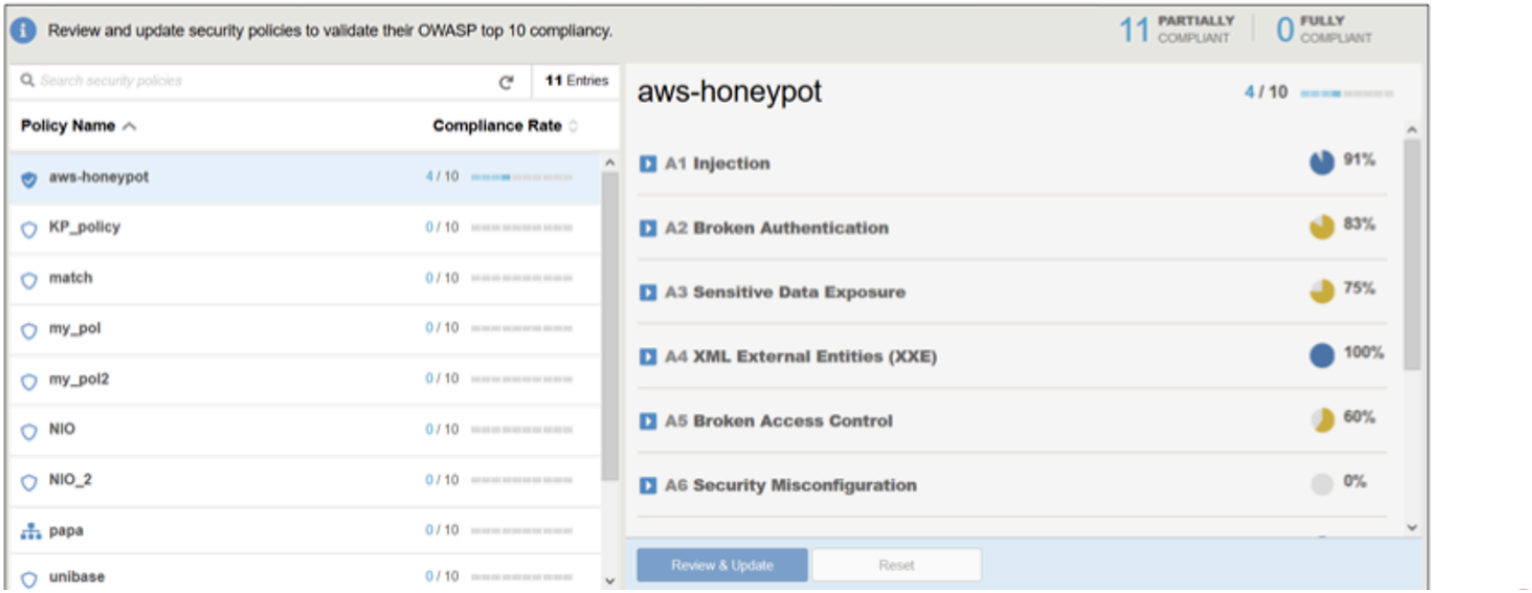
2017 OWASP Top 10

1. Injection
2. Broken authentication
3. Sensitive data exposure
4. XML external entities (XXE)
5. Broken access control
6. Security misconfiguration
7. Cross-site scripting (XSS)
8. Insecure deserialisation
9. Using components with known vulnerabilities
10. Insufficient logging and monitoring

F5 Advanced WAF (AWAF)

Threat Campaign | Advanced Bots | L7 BaDDoS | Credential Stuffing | Client Credentials | Device Based | API Security

Security Overview | **OWASP Dashboard** | Behavioral Enforcement | Deception Mitigation | Credentials Subscription



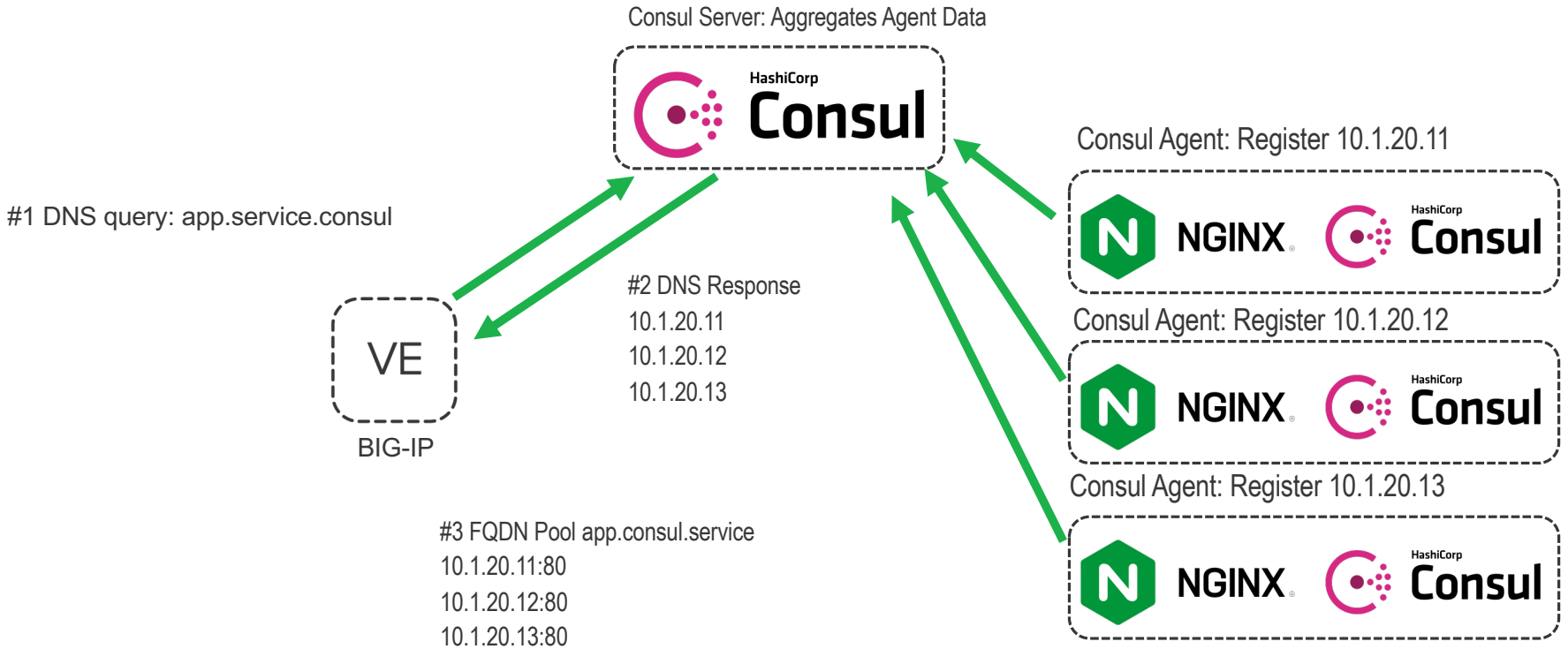
Start Lab 3

1. Deploy AS3 WAF Policy – use Terraform and AS3 to declaratively incorporate app security services

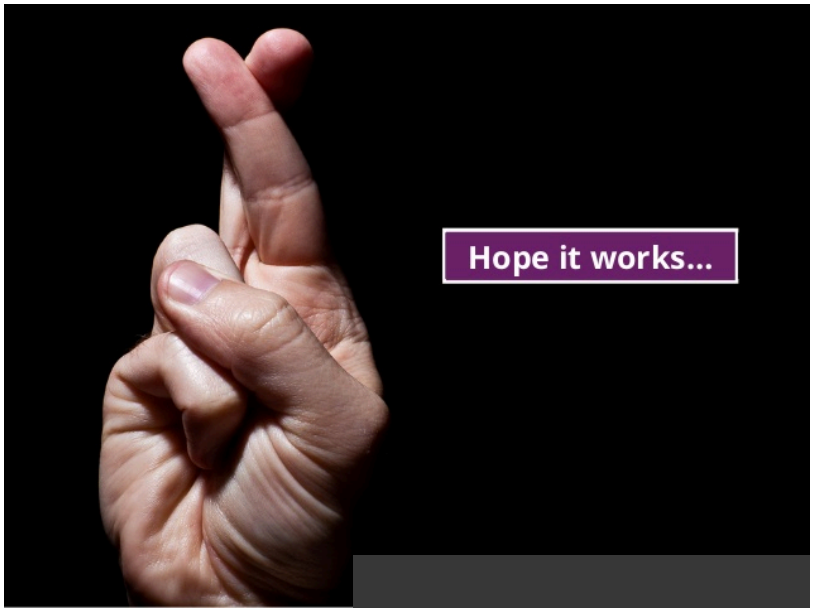
DEMO

APP SERVICES DISCOVERY WITH BIG-IP AND CONSUL

Service Discovery with Consul



Workshop 2 -
App Services Discovery – use Consul and AS3 to dynamically scale app delivery and security services in multi-cloud environments.



References

BIG-IP terraform Provider	https://www.terraform.io/docs/providers/bigip/index.html
Support issues	https://github.com/terraform-providers/terraform-provider-bigip/issues
Example workflow	https://github.com/terraform-providers/terraform-provider-bigip/blob/master/examples/aws/End_to_End_workflow.md
AS3 documentation	https://clouddocs.f5.com/products/extensions/f5-appsvcs-extension/latest/
Service Discovery using Consul	https://clouddocs.f5.com/products/extensions/f5-appsvcs-extension/latest/declarations/discovery.html?highlight=consul#service-discovery-using-hashicorp-consul
Webinar Asset	https://github.com/hashicorp/f5-terraform-consul-sd-webinar

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://devcentral.f5.com/s/articles/Lightboard-Lessons-F5-Automation-Toolchain
F5 in CI/CD Pipelines	Article	Thought Leadership	https://devcentral.f5.com/s/articles/f5-services-in-a-ci-cd-pipeline-34021
F5 Community Training & Labs	Labs & Classes	Learning	https://clouddocs.f5.com/training/community/