



F5 Demo Labs – Self Service Guide – LTM and ASM in AWS Public Cloud

Date: June 2018

Ver 1.0

Gagan Delouri, g.delouri@f5.com

Table of Contents

TABLE OF CONTENTS	2
<IN PROGRESS ITEMS>	3
1. LAB SETUP REQUIREMENTS	3
2. USE CASES TESTED	6
3. SETUP AWS INFRASTRUCTURE	6
4. ACCEPT F5 VE IN AWS MARKETPLACE	6
5. INSTALL A LINUX FLAVOUR OS.....	7
6. INSTRUCTIONS IN DOCKER INSTANCE	7
6.1. ENVIRONMENT VARIABLES	7
6.2. COPY LATEST CODE FROM GITHUB	8
6.3. INSTALL AWS CLI AND CONFIGURE AWS KEYS.....	8
6.4. LAUNCH TERRAFORM SCRIPTS TO CREATE AWS ENVIRONMENT.....	9
6.5. VERIFY THE ENVIRONMENT CREATED.....	10
6.6. HA & AUTO SCALE USE CASE FOR LTM AND ASM MODULES	11
6.6.1.UC-A - TEST HA DEPLOYMENT ACROSS AZS	12
6.6.2.UC-B - TEST AUTO SCALE.....	12
7. ASM – WAF USE CASES	13
7.1. UC-1 ASM – BOT DETECTION AND BLOCKING TEST	15
7.2. UC-2 ASM – GEOLOCATION BLOCKING TEST	16

7.3. UC-3 ASM – SQL INJECTION / XML PROTECTION TEST	16
7.4. UC-5 ASM – FILE BLOCKING TEST.....	17
7.5. ASM USE CASES VIDEOS	17
8. ADVANCED WAF (AWAF) USE CASES.....	19
8.1. AWAF – UC-1 – APP LEVEL ENCRYPTION.....	19
8.2. AWAF – UC-2 – LAYER 7 DOS PROTECTION	19
8.3. AWAF – UC-3 – TBC.....	19
9. VERY IMPORTANT – CLEAN UP ENVIRONMENT	19
10. APPENDIX	20
10.1. POSTMAN SCRIPTS	20
10.2. OTHER IMPORTANT LINKS.....	22

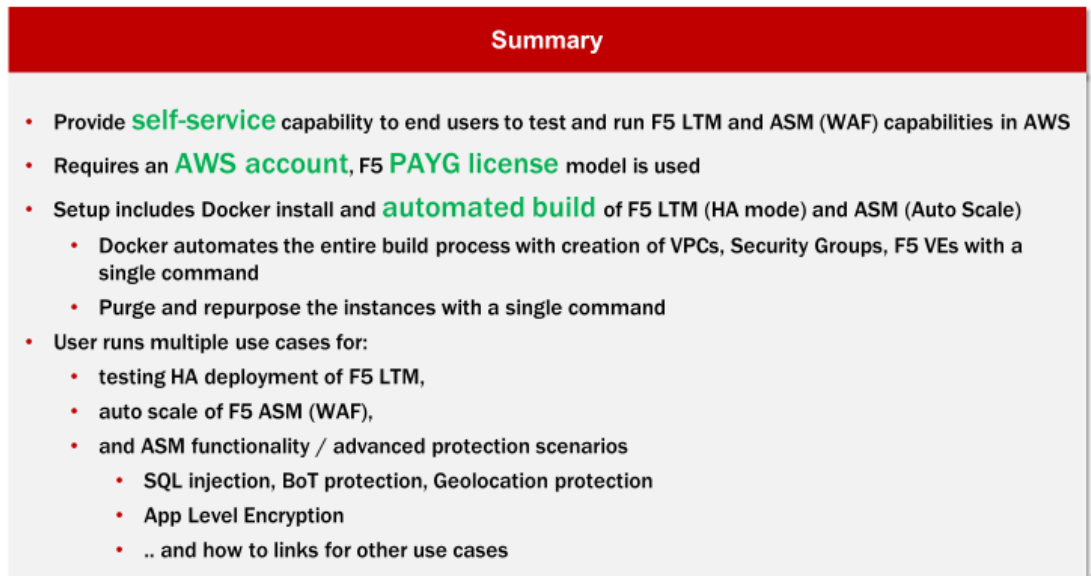
<In progress Items>

- Further automation with CI/CD pipeline through Jenkins to be added
- To be able to run BYOL licenses

1. Summary about the Labs and Environment

1.1. Quick Summary

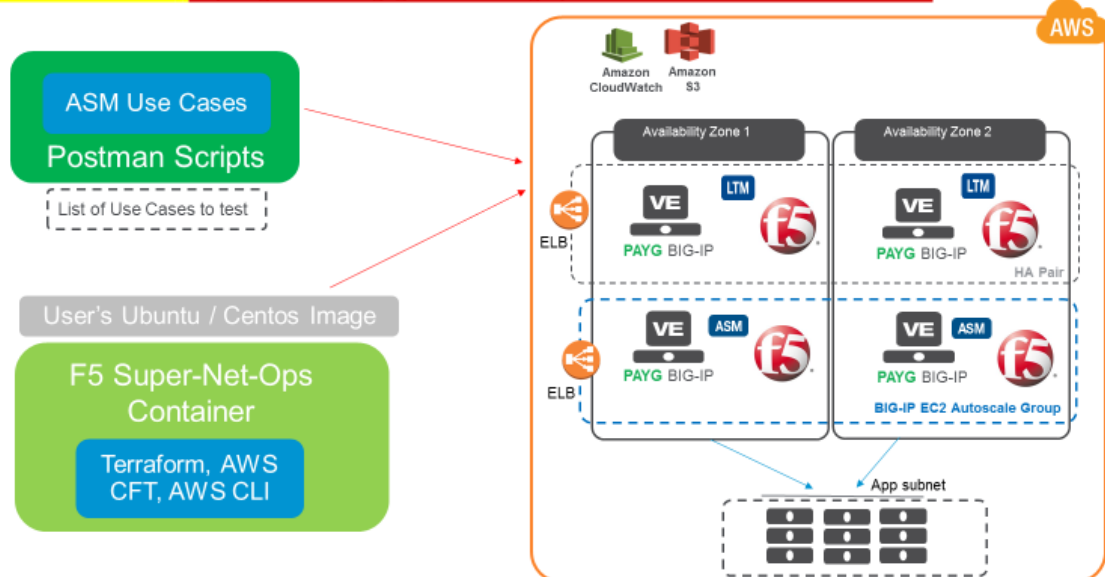
Quick Summary : About the Labs



1.2. About the Environment

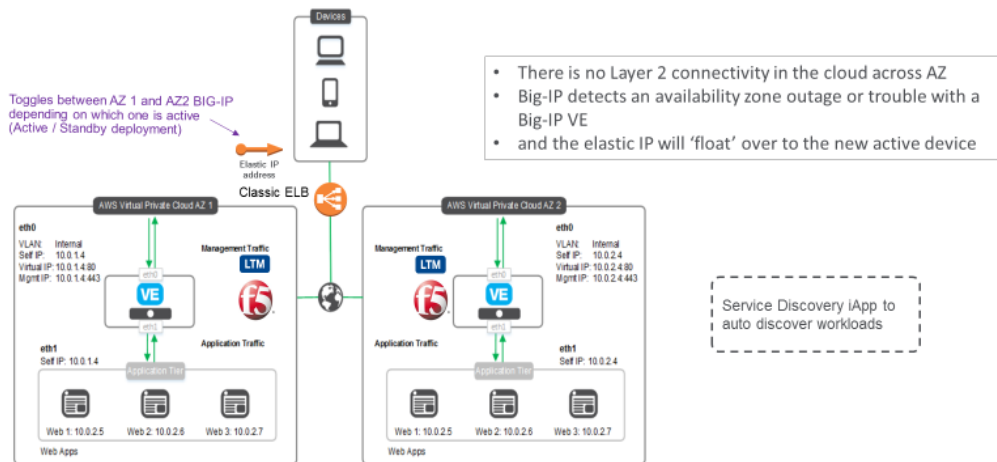
About the environment

Lab Guide available at - <https://github.com/gagandelouri/f5-gd-public-cloud-labs-waf/tree/master/docs>



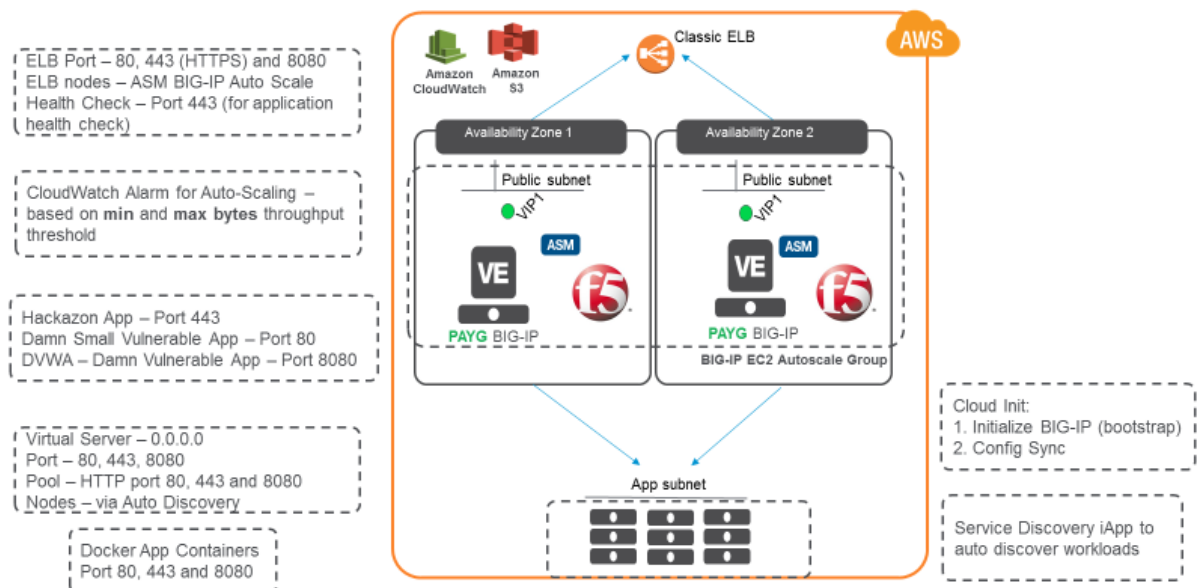
1.3. Test HA Deployment Across AZs in AWS

Architecture – UC-1 Test HA Deployment Across AZs in AWS



1.4. Test auto scale F5 ASM in AWS

Architecture – UC-2 Test auto scale F5 ASM in AWS



2. Lab Setup Requirements

- Requires an AWS account
- This lab will be using F5 PAYG license of GOOD and BEST model
- The labs have only been tested for the following regions, and other regions – **will also be gradually tested**
 - Sydney – ap-southeast-2
 - Singapore – ap-southeast-1
 - US East (Virginia) – us-east-1

3. Use Cases Tested

UC-A	Test HA deployment of F5 LTM across AZs in AWS (including auto discovery of workloads)
UC-B	Test auto scale F5 ASM in AWS
UC-1	BoT detecting and block
UC-2	Geolocation blocking
UC-3	SQL Injection and XML protection
UC-5	File Blocking Test
UC-6	Brute Force Attack – tbc
UC-7	Using and Enforcing Attack Signatures – tbc
UC-8	Brute Force and Credential Stuffing – tbc
AWAF-UC-1	AWAF – DataSafe – App Level Encryption
AWAF-UC-2	AWAF – Layer 7 Proactive DoS Protection

4. Setup AWS Infrastructure

You can run this lab on your own, on any AWS account, but you will need to create your own admin account and tweak the startup steps slightly.

1. Login to your AWS account and create an admin user and access keys.
2. Services -> Security, Identity & Compliance -> IAM -> Groups -> Create New Group. Name this group "terraform-admin" Attach Policy: AdministratorAccess.
3. Services -> Security, Identity & Compliance -> IAM -> Groups -> Create New Group. Name this group "terraform-lab-user" Attach Policy: AdministratorAccess.
4. Services -> Security, Identity & Compliance -> IAM -> Users -> Create New User. Name this user "terraform-lab-admin".
5. Access type: "Programmatic access" only. Click "Next: Permissions."
6. Add user to group: terraform-admin. click "Next: Review."
7. Click "Create user"
8. Copy the Access key ID and Secret access key before clicking "Close." You will need both of these keys to configure the AWS command line tool.

5. Accept F5 VE in AWS Marketplace

You also need to make sure you accept the license agreement by going to AWS market place and Subscribe to it and accept the terms and conditions. We are using F5 BIG-IP Virtual Edition - GOOD - (Hourly, 25Mbps) and F5 BIG-IP Virtual Edition - BEST - (Hourly, 25Mbps) in this instance for our demo.

Click continue to subscribe to accept the license agreements. It may take a minute to process your request. One completed, click continue to configuration and make sure you can see the AMI images for your region.

F5 BIG-IP **Good** Edition 25 Mbps (PAYG)

https://aws.amazon.com/marketplace/pp/B079C44MFH?qid=1525391390565&sr=0-6&ref=srh_res_product_title

F5 BIG-IP **Best** Edition 25 Mbps (PAYG)

https://aws.amazon.com/marketplace/server/configuration?productId=3e567b08-20a9-444f-a72a-7e8da3c2cbdf&ref=psb_cfg_continue

Amazon Linux AMI

https://aws.amazon.com/marketplace/fulfillment?productId=f37c8255-1ff9-48bd-b5da-b5046f4fee68&ref=dtl_psb_continue

Do not proceed any further on the site, just close your browser.

6. Install a Linux flavour OS

1. Install a Linux (Ubuntu or CentOS) flavour of your choice with outbound internet access. This can be installed on Vmware Workstation or on bare metal system
2. In our example, we have taken Ubuntu flavour. Once installed, login as root user and install docker – following the steps below:

<https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-16-04>

3. Confirm docker is installed, by running the command

```
docker ps
```

4. If output is shown with no container running it means, docker install is successful
5. Execute the command below:

```
docker run --dns=4.2.2.1 -p 8080:80 -p 2223:22 -it -e SNOPS_AUTOCLONE=0  
f5devcentral/f5-super-netops-container:base
```

6. This will open up the docker shell.

7. Instructions in Docker Instance

7.1. Environment Variables

The following will store the lab variables that will be used later in the lab. The commands need to be executed in the Docker instance.

The **emailid** and **shortUrl** environment variables must still be exported, but they can be anything you want. Emailid must be in valid email format, but does not have to be a real email address that can be validated.

Below shown just for example:

```
export emailid=gdtterra5@f5.com
export shortUrl=gdtterra5
printenv
```

7.2. Copy latest code from Github

- Change to your home directory.
- Clone the git repository for this lab.
- Change to the working directory.

Copy and paste the commands below to accomplish the steps above.

```
cd ~
git clone https://github.com/gagandelouri/f5-gd-public-cloud-labs-waf
cd ~/f5-gd-public-cloud-labs-waf/
```

7.3. Install AWS CLI and Configure AWS Keys

- Install and configure aws cli with admin keys previously created, region and output format.
- Run the start script.

```
pip install --upgrade --user awscli

export PATH=~/.local/bin:$PATH
export AWS_CONFIG_FILE=~/.aws/config
```

Execute the command:

```
aws configure
```

Access Key and **Secret Key**

should be of user terraform-lab-admin generated at the AWS console.

Also, use the value: **us-east-1 (N. Virginia)** or **ap-southeast-1 (Singapore)** or **ap-southeast-2 (Sydney)**.

Below shown as example:

```
aws configure
```



```
AWS Access Key ID [None]: XXXXXXXXXXXXXXXXXXXX
AWS Secret Access Key [None]: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Default region name [None]: us-east-1
Default output format [None]: json
```

Example output below:

```
#aws configure
AWS Access Key ID [None]: XXXXXXXXXXXXXXXXXXXX
AWS Secret Access Key [None]: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Default region name [None]: us-east-1
Default output format [None]: json
```

Execute the command below:

```
source start
```

7.4. Launch Terraform Scripts to create AWS Environment

Execute the command below:

Invoke terraform.

```
terraform plan
```

This will output the changes that terraform will apply. Now run.

```
terraform apply
```

Note – Once the above command is run, please wait for at least **15-20 minutes** for the environment to spin up, be configured etc. with ELB in AWS.

Check the output and the result should be in green.

Once “terraform apply” completes, you can test your web server instances and ELB are up:

```
curl -I `terraform output elb_dns_name`
```

You should see a reply with status code “HTTP/1.1 200 OK”. Hit <ctrl>+C to stop.

When ‘terraform apply’ completes, note the ****aws_alias**** and vpc-id values. Open up your ****aws_alias**** link in a browser and login to the AWS console with the email address and password you created during the install. You can always get these values by invoking terraform output with the variable name:

```
terraform output **aws_alias**  
terraform output vpc-id
```

7.5. Verify the Environment Created

Execute the command below:

```
lab-info
```

The output should be similar to the below, note the Username and password is shown, same as the environment variables that were set.

```
[root@f5-super-netops] [~/f5-gd-public-cloud-labs-waf] # lab-info  
  
AWS Console  
  
URL: https://.signin.aws.amazon.com/console?us-east-1  
Username: xxxx@f5.com / Password: xxxxxx  
  
WAF ELB  
  
URL: https://waf-gdterra5f5com-1735694819.us-east-1.elb.amazonaws.com  
  
web-az1.0: gdterra5f5com  
PRIVATE IP: 10.0.1.77  
  
web-az2.0: gdterra5f5com  
PRIVATE IP: 10.0.2.229  
  
BIG-IP Autoscale Instance: waf-gdterra5f5com  
MGMT IP: 34.229.141.64  
STATUS: MCPD is down, System Ready  
  
Big-IP2: ha-gdterra5f5com-vpc-b07245cb  
MGMT IP: 18.204.155.160  
STATUS: MCPD is down, System Ready  
VIP IP: 10.0.2.109  
  
Big-IP1: ha-gdterra5f5com-vpc-b07245cb  
MGMT IP: 34.232.220.59  
STATUS: MCPD is down, System Ready  
VIP IP: 10.0.1.30  
Elastic IP: 52.1.30.59
```

- We have an application behind an F5 autoscale WAF that can be reached by the WAF ELB URL.
- The web-az1.0 and webaz2.0 PRIVATE IP addresses will soon be configured as pool members for our Big-IP HA cluster.
- Big-IP1 and Big-IP2 are configured as a high availability cluster across two separate availability zones. Only the active Big-IP will have an Elastic IP address assigned. Configuration changes to the active unit will automatically propagate to the

standby unit. During an outage, even one affecting an entire availability zone, the Elastic IP will 'float' over to the unit that is not affected.

- BIG-IP Autoscale Instance is a single NIC deployment WAF with the MGMT IP address identified.

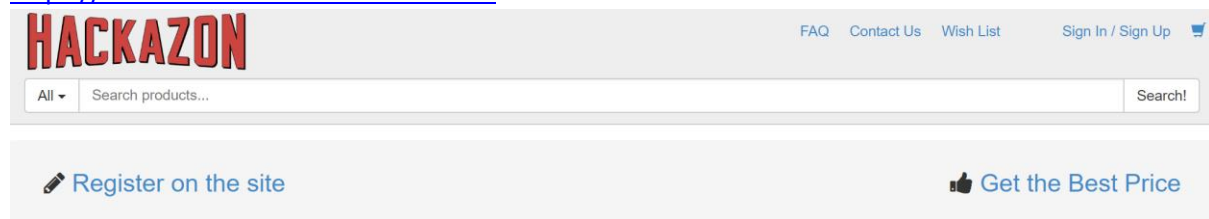
From the f5-super-netops container test out application behind the auto-scale waf is up. Replace the example https url with the one specific to your lab. See lab-info.

```
curl -kI https://waf-user01f5io-499431932.us-east-1.elb.amazonaws.com
```

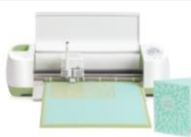
The HTTP/1.1 200 OK status code is a sign that things went well. You can hit the example site (Hackazon) behind the F5 WAF with a web browser.

At the browser, go to the URL – replace with URL with the output from lab-info command.


<https://waf-xxxxxx.elb.amazonaws.com>




Special selection





Cricut Explore Electronic Cutting Machine with...
Cuts the widest variety of materials (50+) Upload and cut...

 **\$250**




Martha Stewart Crafts Garland, Pink Pom Pom Small
Fun, festive party decorative pop poms. Perfect for any...

 **\$9**



Diesel Men's Sleenker Skinny-Leg Jean 0608D
98% Cotton/2% Elastane Imported Hand Wash Super...

 **\$238**

Top 3 most popular

- SpaSilk Unisex-Baby Newborn 3 Pack 100% Cotton Burp Cloths
- Jack Black True Volume Thickening Shampoo, 12 fl. oz.
- Apple iPhone 5s, Gold 16GB (Unlocked)

Top 3 best selling

- Hakko CHP-170 Micro Clean Cutter, 16 Gauge Maximum Cutting Capacity

7.6. HA & Auto Scale Use Case for LTM and ASM Modules

IMPORTANT NOTE → Please follow the instructions from section 2.3.1 ONLY till 2.3.5. These steps will complete the auto-scale demonstration of LTM and ASM in HA environment in AWS. The remaining section will cover the WAF and Advanced WAF use cases.

Continue with the steps **Configuration Utility (Web UI)** access located at 2.3.2. **Configuration Utility (Web UI)** access – at the link below:

- [2.3.1. Verify a healthy F5 environment](#)

```
http://f5-agility-labs-public-cloud.readthedocs.io/en/latest/class1/module3/lab2.html
```

- [2.3.2. Configuration Utility \(Web UI\) access](#)

```
http://f5-agility-labs-public-cloud.readthedocs.io/en/latest/class1/module3/lab2.html
```

7.6.1.UC-A – Test HA Deployment Across AZs

Use Case 1 – Test HA deployment of F5 LTM across AZs in AWS (including auto discovery of workloads)

Note – For these demos, the actual page displayed may differ from what is shown in the links – as the nodes and application is changed for the ASM (WAF) use cases.

- [2.3.3. Deploy the Service Discovery iApp on a BigIP Cluster across two Availability Zones](#)

```
http://f5-agility-labs-public-cloud.readthedocs.io/en/latest/class1/module3/lab3.html
```

- [2.3.4. Deploy an AWS High-Availability-aware virtual server across two Availability Zones](#)

Note – **At Step 13**, Instead of running the curl command, just browse to the application to confirm that it is working ok

```
http://f5-agility-labs-public-cloud.readthedocs.io/en/latest/class1/module3/lab4.html
```

7.6.2.UC-B – Test Auto Scale

Use Case 2 – Test auto scale F5 ASM in AWS

- [2.3.5. Autoscale WAF](#)

Follow the instructions below the caveat mentioned in **Note** section:

```
http://f5-agility-labs-public-cloud.readthedocs.io/en/latest/class1/module3/lab5.html
```

Note – For this use case, sometimes the auto scaling may not trigger by itself because of amount of load generated may be lower, in which case run the ab command, multiple times by example below (by adding **&** and repeating the commands) as shown below:

(Run only once)

```
base64 /dev/urandom | head -c 3000 > payload
```

```
ab -t 120 -c 2000 -c 500 -T 'multipart/form-data; boundary=1234567890'
-p payload https://waf-gdterra12f5com-1726238542.ap-southeast-1.elb.ama
zonaws.com/product/view?id=1
```

to run multiple times to generate auto scaling:

(Run multiple times)

```
base64 /dev/urandom | head -c 3000 > payload
ab -t 120 -c 2000 -c 500 -T 'multipart/form-data; boundary=1234567890'
-p payload https://waf-gdterra12f5com-1726238542.ap-southeast-1.elb.ama
zonaws.com/product/view?id=1 & ab -t 5000 -c 90000 -c 900 -T 'multipa
rt/form-data; boundary=1234567890' -p payload https://waf-gdterra12f5co
m-1726238542.ap-southeast-1.elb.amazonaws.com/product/view?id=1 & ab
-t 120 -c 2000 -c 500 -T 'multipart/form-data; boundary=1234567890' -p
payload https://waf-gdterra12f5com-1726238542.ap-southeast-1.elb.amazon
aws.com/product/view?id=1
```

Note – Before proceeding with the next steps, please reset the password for the new auto scale BIG-IP instance by running the `lab-info` command and then make sure the configuration has synced between the 2 devices before proceeding any further.

8. ASM – WAF Use Cases

There are 3 virtual servers created for the test case that allow to test the use cases, each running on separate port as 80, 443, 8080.

Import the Postman scripts located in the `document appendix`. Environment files and JSON files

Once imported, go to the environment file and update the following to your lab-environment variables.

Replace 1 and 2 with the Mgmt IP and Password fields that you have set.

Regarding 3 -7, please make sure the Username that you used during the setup is defined in this format. For example: `gdterra5@f5.com` username as shown below.

Notice the / and ~ symbols, make sure they are left as it is only replace the values marked in `colour` with your environment variables.

1	big_ip_a_mgmt	BIG-IP Autoscale Instance MGMT IP
2	big_ip_a_password	Password
3	VS_AllCases	waf-gdterra5f5com.app/waf-gdterra5f5com_vs
4	VS_Hackazon	waf-gdterra5f5com-443.app/waf-gdterra5f5com-443_vs
5	VS_Hackazon_BOT	waf-gdterra5f5com-443.app~waf-gdterra5f5com-443_vs

6	VS_DVWA	waf-gdterra5f5com-8080.app~waf-gdterra5f5com-8080_vs
7	VS_DVWA_ASM	waf-gdterra5f5com-8080.app/waf-gdterra5f5com-8080_vs

MANAGE ENVIRONMENTS



Edit Environment

E2E LTM-ASM Demo

	Key	Value	Bulk Edit
<input checked="" type="checkbox"/>	big_ip_a_mgmt	westcon.f5demo.com	
<input checked="" type="checkbox"/>	port	8443	
<input checked="" type="checkbox"/>	big_ip_a_username	admin	
<input checked="" type="checkbox"/>	big_ip_a_password	gdterra5	
<input checked="" type="checkbox"/>	VS_addr	0.0.0.0	
<input checked="" type="checkbox"/>	VS_AllCases	waf-gdterra5f5com.app/waf-gdterra5f5com_vs	
<input checked="" type="checkbox"/>	ASM_BOT_Policy	ASMDemo_Policy_BOT	
<input checked="" type="checkbox"/>	VS_Hackazon	waf-gdterra5f5com-443.app/waf-gdterra5f5com-443..	
<input checked="" type="checkbox"/>	VS_Hackazon_BOT	waf-gdterra5f5com-443.app~waf-gdterra5f5com-44...	
<input checked="" type="checkbox"/>	VS_DVWA_ASM	waf-gdterra5f5com-8080.app/waf-gdterra5f5com-80..	
<input checked="" type="checkbox"/>	VS_DVWA	waf-gdterra5f5com-8080.app~waf-gdterra5f5com-8...	
<input checked="" type="checkbox"/>	ASMDemo_Policy_Hackazon_IPGeolocation-Allow	ASMDemo_Policy_Hackazon_IPGeolocation-Allow	
<input checked="" type="checkbox"/>	ASMDemo_Policy_Hackazon_IPGeolocation-Block	ASMDemo_Policy_Hackazon_IPGeolocation-Block	
<input checked="" type="checkbox"/>	ASMDemo_Policy_AllCases_SQL_XML_IPGeolocation	ASMDemo_Policy_AllCases_SQL_XML_IPGeolocation	
<input checked="" type="checkbox"/>	ASM_AppEncryption_Policy	ASMDemo_Policy_AppEncryption	

Cancel

Update

The following test cases are going to be run and tested.

All Virtual Servers are on the same VIP - just on different ports – 80, 443 and 8080

ASM (WAF) Attack Types	Status	ELB Port	Backend Pool	Behaviour Shown
SQL Injection attack	Done	80	80 (DSVW)	WAF blocked
Malformed XML Request	Done	80	80 (DSVW)	WAF blocked
IP geolocation Alerting / Blocking	Done	443	443 (Hackazon)	WAF blocked
BoT detection Alerting / Blocking	Done	443	443 (Hackazon)	Connection reset

Application Layer Protection	Done	8080	8080 (DVWA)	WAF encrypts the fields
Blocking File Types	Done	8080	8080 (DVWA)	WAF blocked
Layer 7 DoS Protection	Done	8080	8080 (DVWA)	WAF blocked

Once the files are imported in Postman, do the following steps:

3 Virtual Servers are created running on 443, 80 and 8080. Below are just examples

- Hackazon App – Port 443 – <https://waf-gdterra5f5com-638791433.us-east-1.elb.amazonaws.com/>
- Small Vulnerable App – Port 80 – <http://waf-gdterra5f5com-638791433.us-east-1.elb.amazonaws.com/>
- Vulnerable App – Port 8080 – <http://waf-gdterra5f5com-811173699.ap-southeast-1.elb.amazonaws.com:8080/login.php>

For Port 8080 site – **please go to the link for your DNS and click on**

Click here to setup the database, Create/Reset Database to create your application. Click on Home, the username/password for login is admin / password. **This is intentionally running on only 1 node as opposed to 2 nodes.**

Note – When running any of the test cases below, please give it a few seconds for the changes to be replicated to the 2 cluster nodes of BIG-IP (if the auto scale exercise has been completed already). If there is only 1 instance of auto-scale WAF running, there is no wait required.

8.1. UC-1 ASM – BoT Detection and Blocking Test

Run the Postman script **1. Hackazon 443 Test - ASM - DoS, BoT, BADoS Revoke/Apply**

1. 1st script creates the BoT policy and assigns to the Hackazon Virtual Server running on 443
2. 2nd adds the BoT objects with no blocking policy
3. Assign BoT policy to HTTPS – Hackazon, this is just further updating the policy
4. Run the command below, the site is still reachable

curl -kl https://waf-gdterra5f5com-638791433.us-east-1.elb.amazonaws.com/contact

```
HTTP/1.1 200 OK
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Content-Length: 64784
Content-Type: text/html; charset=utf-8
Date: Mon, 21 May 2018 06:21:00 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Pragma: no-cache
Server: Apache/2.4.7 (Ubuntu)
Set-Cookie: PHPSESSID=peqsucej8a4dgkdujo8edh05v3; path=/
Set-Cookie: BIGipServerwaf-gdterra5f5com-443=1040252938.47873.0000; path=/;
HttpOnly
X-Powered-By: PHP/5.5.9-1ubuntu4.11
Connection: keep-alive
```

5. Run the Postman script – Update Objects to block the BoT objects (curl command gets blocked)
6. Run the command and now see the gateway timeout

curl -kl https://waf-gdterra5f5com-638791433.us-east-1.elb.amazonaws.com/contact

HTTP/1.1 504 GATEWAY_TIMEOUT

Connection: keep-alive

8.2. UC-2 ASM – Geolocation Blocking Test

Run the Postman script 2. Hackazon 443 - GeoLocation Allow / Block Policy

1. 1st script creates allows you to access the HTTPS site with no issues
2. Update the Blocking Geo
3. You can confirm the locations are blocked in the ASM policy
4. Run the Apply ASM Policy - Block Geo for policy to come in effect and see the site is now blocked, when you try to access

<https://waf-gdterra5f5com-638791433.us-east-1.elb.amazonaws.com/>

8.3. UC-3 ASM – SQL Injection / XML Protection Test

Run the Postman script 3. All Use Cases - SQL Injection / File Block - Revoke/Apply

1. Try to access the site on port 80 – not the HTTP in the link
<http://waf-gdterra5f5com-638791433.us-east-1.elb.amazonaws.com/>
2. Click on some the attacks and the application is currently vulnerable

Attacks:

- Blind SQL Injection (*boolean*) - **vulnerable|exploit|info**
- Blind SQL Injection (*time*) - **vulnerable|exploit|info**
- UNION SQL Injection - **vulnerable|exploit|info**
- Login Bypass - **vulnerable|exploit|info**
- HTTP Parameter Pollution - **vulnerable|exploit|info**
- Cross Site Scripting (*reflected*) - **vulnerable|exploit|info**
- Cross Site Scripting (*stored*) - **vulnerable|exploit|info**
- Cross Site Scripting (*DOM*) - **vulnerable|exploit|info**
- Cross Site Scripting (*JSONP*) - **vulnerable|exploit|info**
- XML External Entity (*local*) - **vulnerable|exploit|info**
- XML External Entity (*remote*) - **vulnerable|exploit|info**
- Server Side Request Forgery - **vulnerable|exploit|info**
- Blind XPath Injection (*boolean*) - **vulnerable|exploit|info**
- Cross Site Request Forgery - **vulnerable|exploit|info**
- Frame Injection (*phishing*) - **vulnerable|exploit|info**
- Frame Injection (*content spoofing*) - **vulnerable|exploit|info**
- Clickjacking - **-|exploit|info**
- Unvalidated Redirect - **vulnerable|exploit|info**
- Arbitrary Code Execution - **vulnerable|exploit|info**
- Full Path Disclosure - **vulnerable|exploit|info**
- Source Code Disclosure - **vulnerable|exploit|info**
- Path Traversal - **vulnerable|exploit|info**
- File Inclusion (*remote*) - **vulnerable|exploit|info**
- HTTP Header Injection (*phishing*) - **vulnerable|exploit|info**
- Component with Known Vulnerability (*pickle*) - **vulnerable|exploit|info**
- Denial of Service (*memory*) - **vulnerable|exploit|info**

Result(s) :

id	username	name	surname
2	dricci	dian	ricci

3. Apply the policy **Assign ASM Policy to HTTP - All Use Cases** to the Virtual Server and now you can see the attacks will be blocked by ASM
4. You can revoke the policy again by running the **Revoke ASM Policy**. After which the application again becomes vulnerable

8.4. UC-5 ASM – File Blocking Test

1. Access the file below and access to the file is allowed, replacing the server URL with one provided in the **lab-info** output
`http://waf-gdterra12f5com-1726238542.ap-southeast-1.elb.amazonaws.com:8080/php.ini`
2. This file shows internal configuration and sensitive details of the application that need to be blocked by a WAF policy that blocks ini files and password files etc.
3. Run the Postman script **4. UC-4. DVWA - File Block - Revoke/Apply - Assign ASM Policy to HTTP - File Block**
4. Access the file, and is now blocked on the above URL
5. Run the revoke **Revoke ASM Policy to HTTPS VS** and the file access is now allowed

8.5. ASM Use Cases Videos

Please refer to the Youtube link covering 40 Videos [7-12 min each] to show the test cases that you can use for further review:

<https://www.youtube.com/playlist?list=PLZmbPz-KgDtgJLfsdLmSHIXyv0TIQ-CJj>



F5 Networks WW Field Enablement - WWFE

Subscribe 137

Home Videos **Playlists** Channels Discussion About



BIG-IP ASM/WAF Demo Series

F5 Networks WW Field Enablement - WWFE • 40 videos • 1,571 views • Last updated on Apr 3, 2018

▶ Play all

◀ Share

+ Save

ASM Demo 1: Simple Security Policy and L7 DoS Protection Deployments with F5 BIG-IP ASM
ASM Demo 2: Blocking Common Web Vulnerabilities with BIG-IP ASM
ASM Demo 3: Blocking SQL Injection Attacks with F5 BIG-IP ASM
ASM Demo 4: Blocking Cross-Site Scripting Attacks with F5 BIG-IP ASM
ASM Demo 5: DataGuard and PCI Compliance with F5 BIG-IP ASM
ASM Demo 6: Building Security Policies Using Rapid Deployment with F5 BIG-IP ASM
ASM Demo 7: Using Manual Policy Building with F5 BIG-IP ASM
ASM Demo 8: Using Automatic Policy Building with F5 BIG-IP ASM
ASM Demo 9: Enforcing File Types with F5 BIG-IP ASM
ASM Demo 10: Applying and Enforcing Global File Type Settings with F5 BIG-IP ASM
ASM Demo 11: Applying and Enforcing Global Parameter Settings with F5 BIG-IP ASM
ASM Demo 12: Building F5 BIG-IP ASM Security Policies using Trusted vs Untrusted Requests
ASM Demo 13 (Exercise): Understanding File Type and Parameter Learning and Enforcement
ASM Demo 14: Working with Parameters in F5 BIG-IP ASM
ASM Demo 15 (Exercise): Understanding Entity Enforcement with F5 BIG-IP ASM
ASM Demo 16: Protecting Against Cookie Modification with F5 BIG-IP ASM
ASM Demo 17: Using Security Logging and Reporting with F5 BIG-IP ASM
ASM Demo 18: Updating Security Policies Manually with F5 BIG-IP ASM
ASM Demo 19: Updating Security Policies Automatically with F5 BIG-IP ASM
ASM Demo 20: Using and Enforcing Attack Signatures with F5 BIG-IP ASM
ASM Demo 21: Using CSRF Protection with F5 BIG-IP ASM
ASM Demo 22: Using Layered Security Policies with F5 BIG-IP ASM
ASM Demo 23 (Exercise): Understanding Parent and Child Security Policies with F5 BIG-IP ASM
ASM Demo 24: Using Custom Data Guard Settings with F5 BIG-IP ASM
ASM Demo 25: Using Custom Attack Signatures with F5 BIG-IP ASM
ASM Demo 26: Blocking Brute Force Attacks with F5 BIG-IP ASM
ASM Demo 27: Blocking Web Scraping Attacks with F5 BIG-IP ASM
ASM Demo 28: Using Login Page Enforcement with F5 BIG-IP ASM
ASM Demo 29: Stabilizing Security Policies with F5 BIG-IP ASM
ASM Demo 30 (Exercise): Advanced Security Policy Building Options with BIG-IP ASM
ASM Demo 31: Using Layer 7 DoS Bot Protection with BIG-IP ASM
ASM Demo 32: Blocking Suspicious Web Browsers with BIG-IP ASM
ASM Demo 33: Using IP Geolocation Enforcement with BIG-IP ASM
ASM Demo 34 (Exercise): Using Layer 7 DoS Protection with F5 BIG-IP ASM
ASM Demo 35: Using Session Cookie Hijacking Protection with F5 BIG-IP ASM
ASM Demo 36: Using ASM Cookie Hijacking Protection with F5 BIG-IP ASM
ASM Demo 37: Using Client-Side Integrity Defense with F5 BIG-IP ASM
ASM Demo 38: Using Session Tracking and Violation Detection by Usernames with F5 BIG-IP ASM
ASM Demo 39: Using Session Tracking and Violation Detection by Device ID with F5 BIG-IP ASM
ASM Demo 40: Using WebSocket Protection with F5 BIG-IP ASM

UC-6	Brute Force Attack – tbc
UC-7	Using and Enforcing Attack Signatures – tbc
UC-8	Brute Force and Credential Stuffing – tbc

9. Advanced WAF (AWAF) Use Cases

9.1. AWAF – UC-1 – App Level Encryption

Please contact your F5 Pre-Sales team for more details on this use case and how to run this advanced use case for WAF.

9.2. AWAF – UC-2 – Layer 7 DoS Protection

Run the command below in the docker container:

```
curl -A "Hacker Browser" http://waf-gdterra5f5com-1469136514.ap-southeast-1.elb.amazonaws.com:8080/login.php?[1-100]
```

Run the Postman script to create the Layer 7 DoS Protection profile - AWAF - UC-2 - DVWA 8080 - Layer 7 Proactive DoS Protection.

Create the Proactive DoS Profile, Update the policy and Assign Proactive DoS to DVWA.

Once the Postman scripts are run, execute the above command to see javascript message is being challenged in the output and Layer 7 DoS is activated, not allowing requests to reach the ASM or App Server for any processing. The DoS Layer 7 takes precedence and protects the application.

```
<script type="text/javascript" src="/TSPD/080b8ad399ab2000db38995c8a380b475e53721fc36629858f73bbb1134b4ad178f2bab127c5e90f?type=10"></script>
<noscript>Please enable JavaScript to view the page content.<br/>Your support ID is: 11429344193009421980.</noscript>
</head><body>
</body></html>
```

Further information on the lab is at the video below:

<https://www.youtube.com/watch?v=A7DkzDtPp9o&list=PLZmbPz-KgDtGJLfsdLmSHIXyv0TIQ-CJj&index=34>

9.3. AWAF – UC-3 – tbc

Please contact your F5 Pre-Sales team for more details on this use case and how to run this advanced use case for WAF.

10. Very Important – Clean up environment

1. Run the lab-cleanup script.

```
lab-cleanup
```

2. Followed by “terraform destroy”. Confirm destroy with ‘yes’ when prompted.

```
terraform destroy
```

Make sure the VPC, Security Groups, ELB – etc. all objects are removed, if not delete them manually – else you may be charged for the usage of these instances.

11. Appendix

11.1. Postman Scripts

Postman Scripts are located at - <https://github.com/gagandelouri/f5-gd-postman-scripts>

Import the E2E LTM-ASM Demo.postman_environment as environment variable file in Postman and [E2E.ltm.apm.asm - Demo - Detailed.postman_collection](#) as the JSON collection file. Screenshots shown below.

Import environment and select the environment file:


MANAGE ENVIRONMENTS



Environments are a group of variables & values, that allow you to quickly switch the context for your requests and collections.

[Learn more about environments](#)

E2E LTM-ASM Demo

 Share

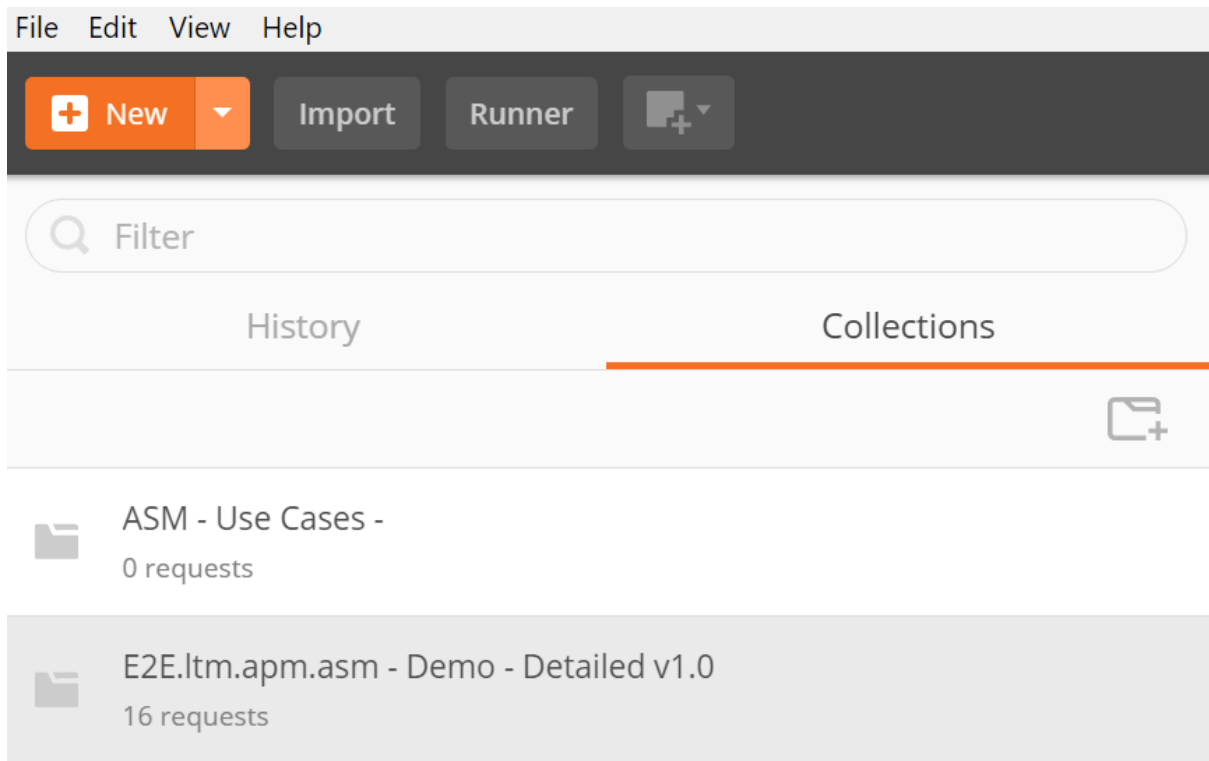


Globals

Import

Add

After this, click on Import for the JSON file and import the file



11.2. Other Important Links

Github	https://github.com/gagandelouri/f5-gd-public-cloud-labs-waf
Cloud Formation Templates	https://s3.amazonaws.com/gd-f5-public-cloud/v1/gd-f5-autoscale-bigip.template https://s3.amazonaws.com/gd-f5-public-cloud/v1/gd-f5-existing-stack-across-az-cluster-hourly-2nic-bigip.template
Docker Images for ASM workloads	https://hub.docker.com/r/gaganld/dvwa-gagan/ https://hub.docker.com/r/gaganld/dsvw-gagan/
Postman Scripts	https://github.com/gagandelouri/f5-gd-postman-scripts

