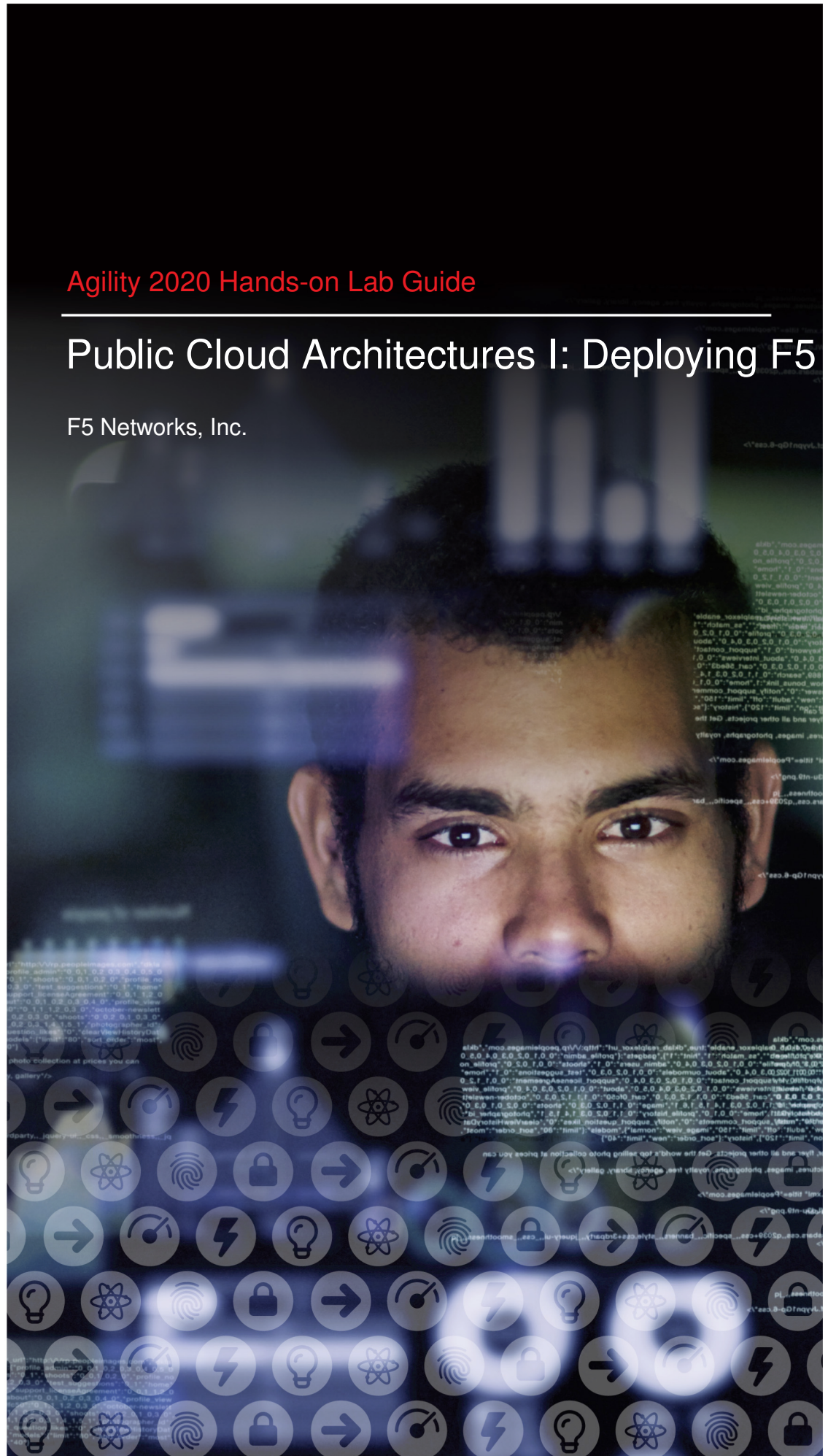




Agility 2020 Hands-on Lab Guide

Public Cloud Architectures I: Deploying F5

F5 Networks, Inc.



Contents:

1	Getting Started	5
1.1	Connecting to the Lab Environment	5
2	Class - Public Cloud Architectures I: Deploying F5 BIG-IP Virtual Edition in AWS	7
2.1	Infrastructure As Code	7
2.2	Exploring AWS	15
2.3	Explore the F5 Big-IP Virtual Editions Deployed	29
2.4	Extending and Securing your Cloud	36
2.5	Logging to CloudWatch	52
2.6	Autoscale WAF	71
2.7	Clean Up Environment	74

Getting Started

Your instructor will provide a URL where you can access your lab environment.

Note: All work for this lab can be performed exclusively from the Linux jumphost. No installation or interaction with your local system is required.

1.1 Connecting to the Lab Environment

Your instructor will provide directions on how to connect to the Ravello Portal.

Class - Public Cloud Architectures I: Deploying F5 BIG-IP Virtual Edition in AWS

This class covers the following topics:

- Deploying AWS environments with CloudFormation Templates and Terraform
- Service Discovery iApp for dynamically populating pool members using instance tags
- Cross Availability Zone HA with F5
- Autoscale WAF
- Logging to Cloudwatch

2.1 Infrastructure As Code

This lab will use HashiCorp Terraform and AWS CloudFormation templates to deploy two common F5 use cases:

- Cross Availability Zone High Availability
- Autoscale WAF

The CloudFormation templates used in this lab are hosted in the official F5 Github repository:

<https://github.com/F5Networks/f5-aws-cloudformation>

2.1.1 Connecting to the Lab

Important: Your student account, and shortUrl value will be announced at the start of the lab.

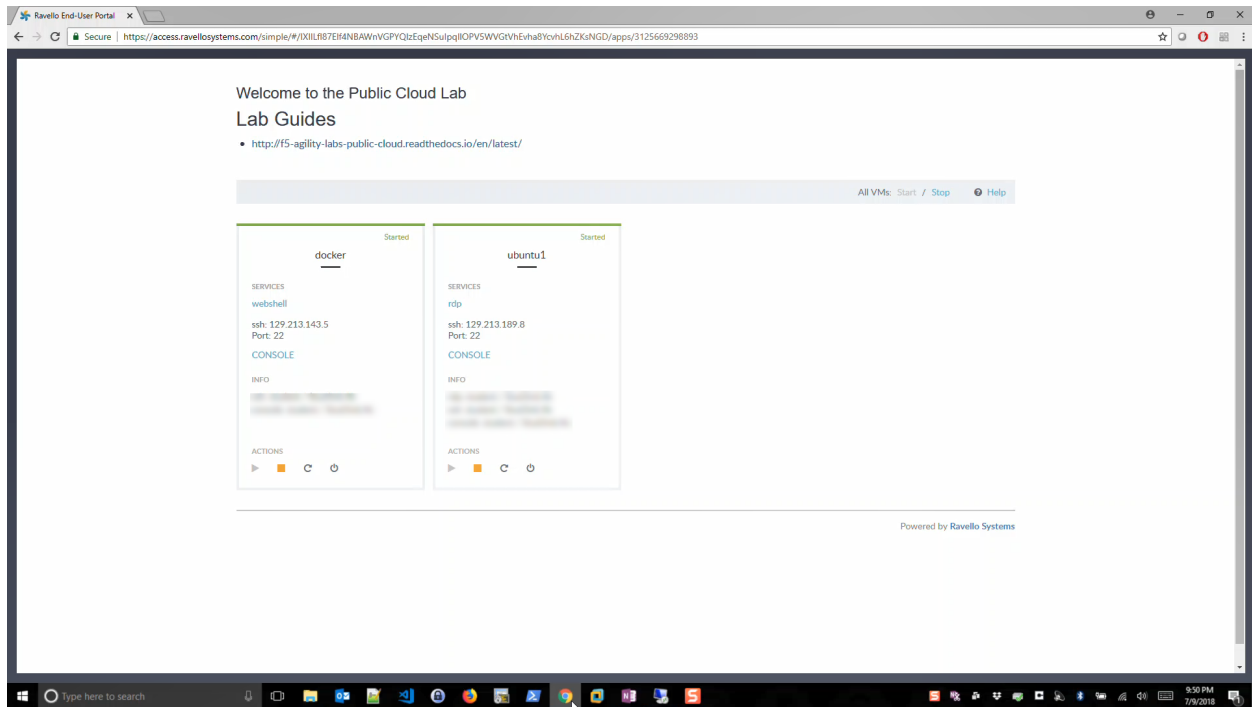
- For this lab, a Linux Remote Desktop jump host will be provided as a consistent starting point.
- Though the public cloud environment runs on a shared AWS account, every student will build and work in a dedicated AWS VPC.
- A convenient way to work through the lab is to split your screen in half: one side for the lab environment, the other side for the lab guide.

2.1.2 Lab Variables

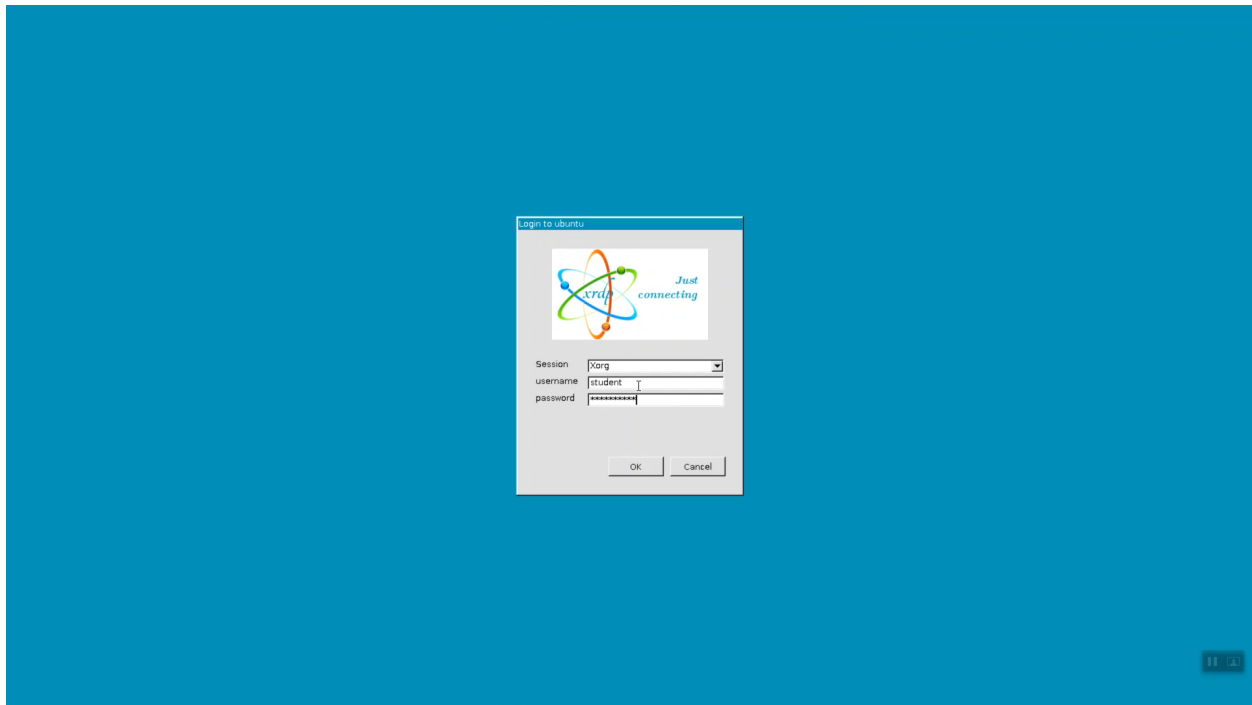
The lab will make use of unique variables to provide access to the lab and isolate student environments.

Variable Name	Variable Value
shortUrl	Unique key that provides access to this lab (i.e. abc123)
emailId	Account name for each student (i.e. user01@f5lab.com)

2.1.3 Launch Remote Desktop Session to Linux

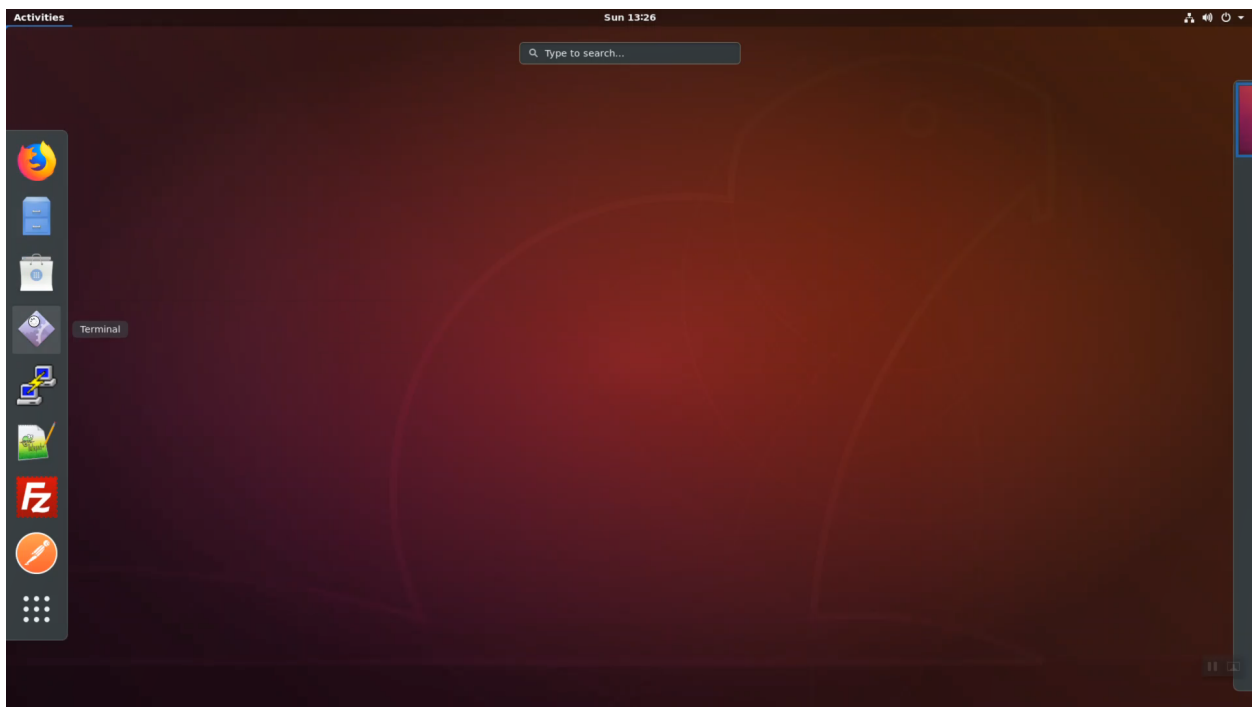


- Look for ubuntu1. Note the username / password. Click on *rdp* link. Download the rdp file. Click on the rdp file to launch a Remote Desktop Session to your client.
- Alternatively, you can copy and paste the ubuntu1 IP address into your Remote Desktop client to modify settings.
 - Local Resources => Keyboard => Apply Windows key combinations: On the remote computer. This will allow you to quickly toggle (ALT + TAB) between windows inside the Remote Desktop Session.
- Login with username / password



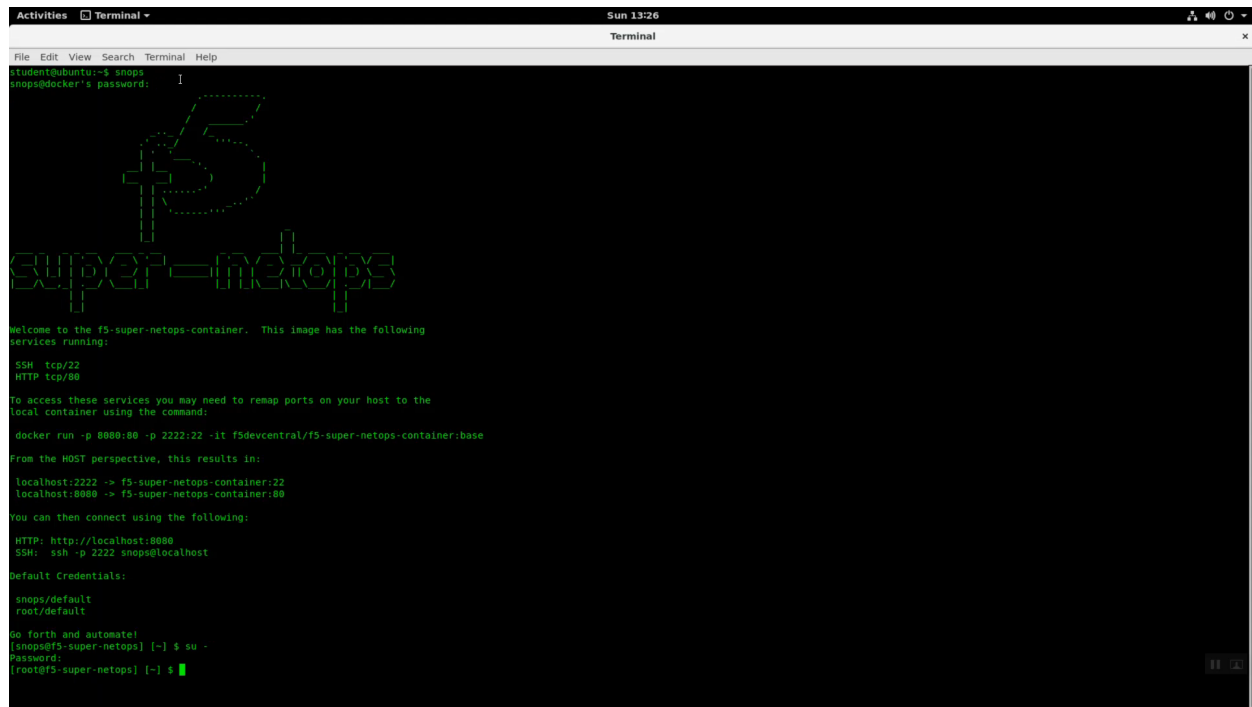
2.1.4 SSH to the F5-Super-NetOps docker container

From the Linux desktop, click on the upper-left-hand corner “Activities” to reveal the application Dock. Click to launch the terminal application.



From the terminal, invoke the ‘snops’ command alias to ssh to the f5-super-netops docker container. Substitute user (su) to root.

```
snops
default
su -
default
```



```
Activities Terminal - Sun 13:26
Terminal
File Edit View Search Terminal Help
student@buntu:~$ snops
snops@docker:~$ password:
f5
super-netops
Welcome to the f5-super-netops-container. This image has the following
services running:
SSH tcp/22
HTTP tcp/80
To access these services you may need to remap ports on your host to the
local container using the command:
docker run -p 8080:80 -p 2222:22 -it f5devcentral/f5-super-netops-container:base
From the HOST perspective, this results in:
localhost:2222 -> f5-super-netops-container:22
localhost:8080 -> f5-super-netops-container:80
You can then connect using the following:
HTTP: http://localhost:8080
SSH: ssh -p 2222 snops@localhost
Default Credentials:
snops/default
root/default
Go forth and automate!
[snops@f5-super-netops] [-] $ su -
Password:
[root@f5-super-netops] [-] $
```

2.1.5 Set Variables

Export your student account and short URL path variables.

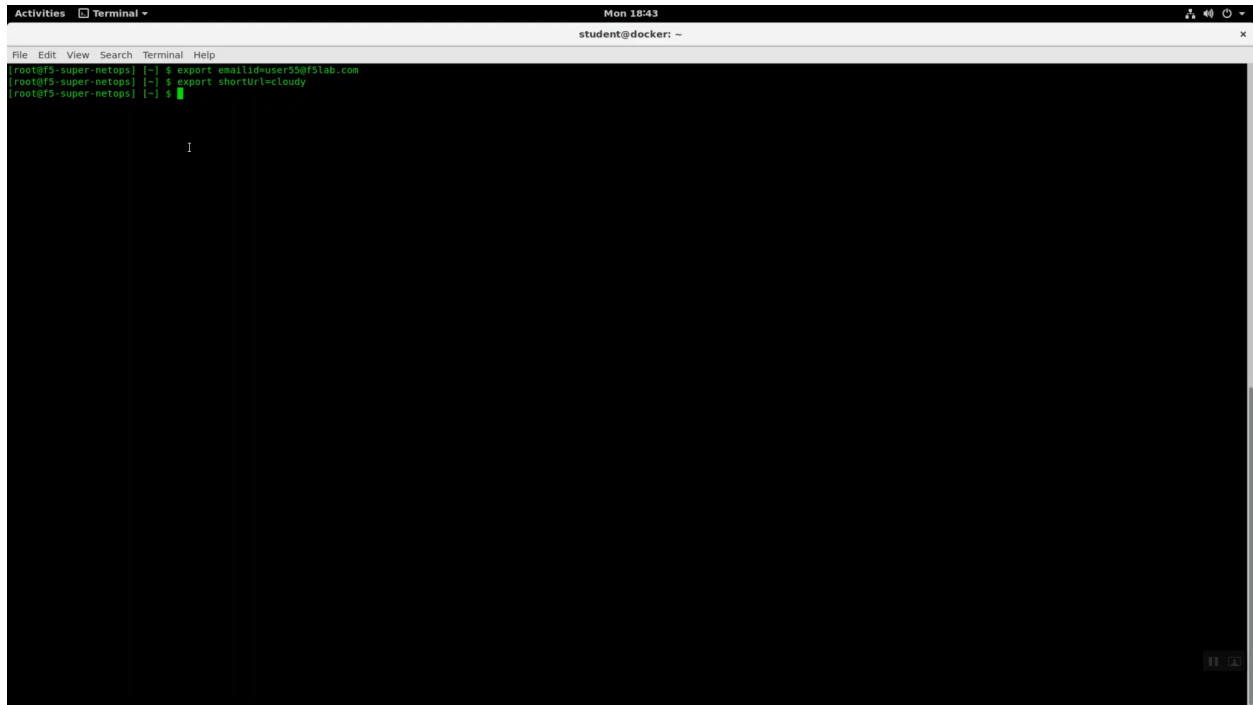
Your student account will be used to create an AWS console login and provide unique names for infrastructure that you create in the shared AWS account.

The short URL path will be used to grant access to the shared AWS account both via the AWS API and as the password for the AWS web console. Replace the emailid and shortUrl values below with the student account name and short URL assigned to you at the start of the lab.

Attention: REPLACE THE EXAMPLE VALUES WITH THE VALUES PROVIDED TO YOU BY YOUR INSTRUCTOR.

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

```
export emailid=user55@f5lab.com
export shortUrl=abc123
printenv
```


A terminal window titled 'Terminal' with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (student@docker: ~). The terminal shows a root shell prompt [root@f5-super-netops: ~] where two environment variables are exported: emailid=user55@f5lab.com and shortUrl=cloudy. The prompt returns to [root@f5-super-netops: ~] \$ after each command. A large 'I' is visible in the background of the terminal window.

```
File Edit View Search Terminal Help
student@docker: ~
[root@f5-super-netops: ~] $ export emailid=user55@f5lab.com
[root@f5-super-netops: ~] $ export shortUrl=cloudy
[root@f5-super-netops: ~] $
```

The `printenv` command will echo all your environment variables. Look for `emailid` and `shortUrl`. Confirm the exported variables are correct.

2.1.6 Initialize your Lab Environment

This will create AWS credentials that you will use to access the shared AWS account.

You will:

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

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

```
cd ~
git clone -b dev https://github.com/TonyMarfil/marfil-f5-terraform
cd ~/marfil-f5-terraform/
source start
```

```
Activities Terminal Mon 18:43 student@docker: -
File Edit View Search Terminal Help
[root@fs-super-netops] [~] $ export emailid=user55@fslab.com
[root@fs-super-netops] [~] $ export shortUrl=cloudy
[root@fs-super-netops] [~] $ cd -
[root@fs-super-netops] [~] $
[root@fs-super-netops] [~] $ git clone -b dev https://github.com/TonyMarfil/marfil-f5-terraform
Cloning into 'marfil-f5-terraform'...
remote: Counting objects: 999, done.
remote: Compressing objects: 100% (102/102), done.
remote: Total 999 (delta 83), reused 95 (delta 41), pack-reused 855
Receiving objects: 100% (999/999), 371.00 KiB | 0 bytes/s, done.
Resolving deltas: 100% (600/600), done.
[root@fs-super-netops] [~] $
[root@fs-super-netops] [~] $ cd ~/marfil-f5-terraform/
[root@fs-super-netops] [~/marfil-f5-terraform] $
[root@fs-super-netops] [~/marfil-f5-terraform] $ source start
Collecting awscli
  Downloading https://files.pythonhosted.org/packages/d8/13/496330abdd2ebf1f091275fb68620b29e1fbc24d59f844c8be0782a01/awscli-1.15.55-py2.py3-none-any.whl (1.3MB)
    100% |#####| 1.3MB 6.3MB/s
Requirement not upgraded as not directly required: docutils==0.10 in /usr/lib/python2.7/site-packages (from awscli) (0.12)
Requirement not upgraded as not directly required: pyyaml<3.12,>=3.10 in /usr/lib/python2.7/site-packages (from awscli) (3.12)
Requirement not upgraded as not directly required: s3transfer<0.2.0,>=0.1.12 in /usr/lib/python2.7/site-packages (from awscli) (0.1.13)
Collecting colorama<0.3.9,>=0.2.5 (from awscli)
  Downloading https://files.pythonhosted.org/packages/db/c8/7dcf9dcb22429512788fe3a547f8b6181cd0d2137bcb892505aee5adf/colorama-0.3.9-py2.py3-none-any.whl
Collecting rsa<3.5.0,>=3.1.2 (from awscli)
  Downloading https://files.pythonhosted.org/packages/e1/ae/baed9cb175552e95f3395c43055a6a5e125ae4d48a1d7a924baca83e92e/rsa-3.4.2-py2.py3-none-any.whl (46KB)
    100% |#####| 51KB 6.0MB/s
Collecting boto-core==1.10.54 (from awscli)
  Downloading https://files.pythonhosted.org/packages/2c/84/calc66a4c87afdac3ca0dc72be3907e94526947d5094faf8704c0eedaa67/botocore-1.10.54-py2.py3-none-any.whl (4.4MB)
    100% |#####| 4.4MB 1.5MB/s
Requirement not upgraded as not directly required: futures==4.0.0,>=2.2.0; python_version == "2.6" or python_version == "2.7" in /usr/lib/python2.7/site-packages (from s3transfer<0.2.0,>=0.1.12->awscli) (3.2.0)
Requirement not upgraded as not directly required: python-dateutil<3.0.0,>=2.1; python_version == "2.7" in /usr/lib/python2.7/site-packages (from boto-core==1.10.54->awscli) (2.7.3)
Requirement not upgraded as not directly required: jmespath<1.0.0,>=0.7.1 in /usr/lib/python2.7/site-packages (from boto-core==1.10.54->awscli) (0.9.3)
Requirement not upgraded as not directly required: six==1.5 in /usr/lib/python2.7/site-packages (from python-dateutil<3.0.0,>=2.1; python_version == "2.7"->boto-core==1.10.54->awscli) (1.10.0)
Installing collected packages: colorama, rsa, botocore, awscli
  The scripts pyrsa-decrypt, pyrsa-decrypt-bigfile, pyrsa-encrypt, pyrsa-encrypt-bigfile, pyrsa-keygen, pyrsa-priv2pub, pyrsa-sign and pyrsa-verify are installed in '/root/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed awscli-1.15.55 botocore-1.10.54 colorama-0.3.9 rsa-3.4.2
+sh: development: unknown operand
--2018-07-10 01:43:37-- https://pyndc37yn0.execute-api.us-east-1.amazonaws.com/p/cloudy
Resolving pyndc37yn0.execute-api.us-east-1.amazonaws.com... 13.33.148.153, 13.33.148.131, 13.33.148.95, ...
Connecting to pyndc37yn0.execute-api.us-east-1.amazonaws.com[13.33.148.153]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 548 [application/json]
Saving to: 'config'

config 100%[#####] 548 ---KB/s in 0s

2018-07-10 01:43:39 (30.6 MB/s) - 'config' saved [548/548]

/root/marfil-f5-terraform
[ ] [ ]
```

Git clone completes successfully.

```
Activities Terminal Mon 18:44 student@docker: -
File Edit View Search Terminal Help
config 100%[#####] 548 ---KB/s in 0s

2018-07-10 01:43:39 (30.6 MB/s) - 'config' saved [548/548]

/root/marfil-f5-terraform
aws - version
aws-cli/1.15.55 Python/2.7.14 Linux/4.4.0-130-generic botocore/1.10.54

{
  "User": {
    "UserName": "user55@fslab.com",
    "Path": "/",
    "CreateDate": "2018-07-10T01:43:41.714Z",
    "UserId": "AIDA35XS45ZM7XT0RT6E",
    "Arn": "arn:aws:iam:457112751961:user:user55@fslab.com"
  }
}
{
  "AccessKey": {
    "UserName": "user55@fslab.com",
    "Status": "Active",
    "CreateDate": "2018-07-10T01:43:43.040Z",
    "SecretAccessKey": "rnfXAdYmB4MCHNoFJkzHL+Tj+ZBR8kPrQz1PEPx",
    "AccessKeyId": "AKIAJBBLT3DJETPMV13Q"
  }
}
{
  "LoginProfile": {
    "UserName": "user55@fslab.com",
    "CreateDate": "2018-07-10T01:43:43.715Z",
    "PasswordResetRequired": false
  }
}
{
  "User": {
    "UserName": "user55@fslab.com",
    "Path": "/",
    "CreateDate": "2018-07-10T01:43:41Z",
    "UserId": "AIDA35XS45ZM7XT0RT6E",
    "Arn": "arn:aws:iam:457112751961:user:user55@fslab.com"
  }
}
{
  "PolicyNames": []
}
Generating a 2048 bit RSA private key
.....+++
writing new private key to 'user55fslabcom.key'
-----
emailid exists and is user55@fslab.com
TF_VAR_bigIqLicenseManager value is 18.205.205.53.
[root@fs-super-netops] [~/marfil-f5-terraform] $
```

Attention: For a smooth ride, always invoke commands from inside the cloned git repository (marfil-f5-terraform). To check you're in the right place you can run the command `pwd` and the output should read `/root/marfil-f5-terraform`

2.1.7 Launch Terraform

Now that we have created credentials to access the AWS account, we will use Terraform to deploy our lab environment.

Initialize terraform.

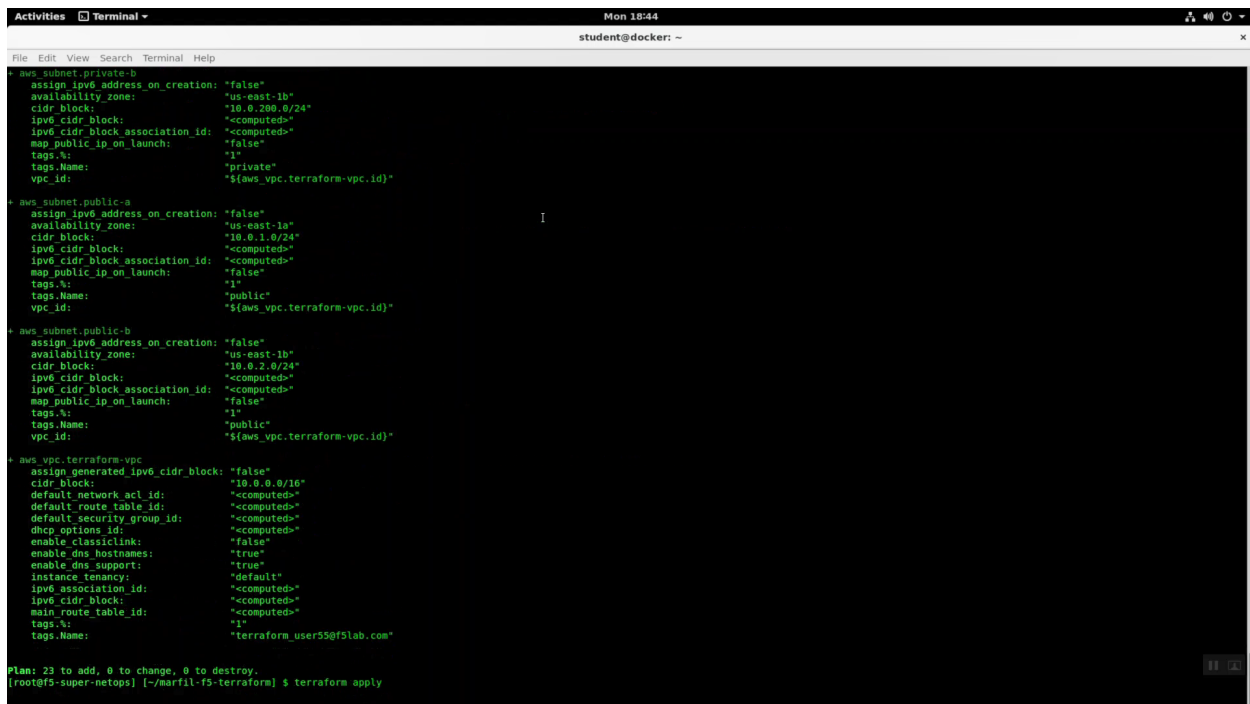
```
terraform init
```

Invoke terraform plan. This will output the changes that terraform will apply.

```
terraform plan
```

Terraform apply.

```
terraform apply
```



```
Activities Terminal - Mon 18:44 student@docker: -
File Edit View Search Terminal Help
+ aws_subnet.private-b
  assign_ipv6_address_on_creation: "false"
  availability_zone: "us-east-1b"
  cidr_block: "10.0.200.0/24"
  ipv6_cidr_block: "<computed>"
  ipv6_cidr_block_association_id: "<computed>"
  map_public_ip_on_launch: "false"
  tags.%: "1"
  tags.Name: "private"
  vpc_id: "${aws_vpc.terraform-vpc.id}"
+ aws_subnet.public-a
  assign_ipv6_address_on_creation: "false"
  availability_zone: "us-east-1a"
  cidr_block: "10.0.1.0/24"
  ipv6_cidr_block: "<computed>"
  ipv6_cidr_block_association_id: "<computed>"
  map_public_ip_on_launch: "false"
  tags.%: "1"
  tags.Name: "public"
  vpc_id: "${aws_vpc.terraform-vpc.id}"
+ aws_subnet.public-b
  assign_ipv6_address_on_creation: "false"
  availability_zone: "us-east-1b"
  cidr_block: "10.0.2.0/24"
  ipv6_cidr_block: "<computed>"
  ipv6_cidr_block_association_id: "<computed>"
  map_public_ip_on_launch: "false"
  tags.%: "1"
  tags.Name: "public"
  vpc_id: "${aws_vpc.terraform-vpc.id}"
+ aws_vpc.terraform-vpc
  assign_generated_ipv6_cidr_block: "false"
  cidr_block: "10.0.0.0/16"
  default_network_acl_id: "<computed>"
  default_route_table_id: "<computed>"
  default_security_group_id: "<computed>"
  dhcp_options_id: "<computed>"
  enable_classiclink: "false"
  enable_dns_hostnames: "true"
  enable_dns_support: "true"
  instance_tenancy: "default"
  ipv6_association_id: "<computed>"
  ipv6_cidr_block: "<computed>"
  main_route_table_id: "<computed>"
  tags.%: "1"
  tags.Name: "terraform_user55@f5lab.com"

Plan: 23 to add, 0 to change, 0 to destroy.
[root@f5-super-netops] [~/marfil-f5-terraform] $ terraform apply
```

```
Activities Terminal Mon 18:48 student@docker: -
File Edit View Search Terminal Help
aws_cloudformation_stack.f5-cluster-cross-az-ha-bigip: Still creating... (2m20s elapsed)
aws_cloudformation_stack.f5-autoscale-waf: Still creating... (2m10s elapsed)
aws_cloudformation_stack.f5-cluster-cross-az-ha-bigip: Still creating... (2m30s elapsed)
aws_cloudformation_stack.f5-autoscale-waf: Still creating... (2m20s elapsed)
aws_cloudformation_stack.f5-cluster-cross-az-ha-bigip: Still creating... (2m40s elapsed)
aws_cloudformation_stack.f5-autoscale-waf: Still creating... (2m30s elapsed)
aws_cloudformation_stack.f5-cluster-cross-az-ha-bigip: Still creating... (2m50s elapsed)
aws_cloudformation_stack.f5-autoscale-waf: Still creating... (2m40s elapsed)
aws_cloudformation_stack.f5-cluster-cross-az-ha-bigip: Still creating... (3m0s elapsed)
aws_cloudformation_stack.f5-autoscale-waf: Still creating... (2m50s elapsed)
aws_cloudformation_stack.f5-cluster-cross-az-ha-bigip: Still creating... (3m10s elapsed)
aws_cloudformation_stack.f5-autoscale-waf: Still creating... (3m0s elapsed)
aws_cloudformation_stack.f5-cluster-cross-az-ha-bigip: Still creating... (3m20s elapsed)
aws_cloudformation_stack.f5-autoscale-waf: Still creating... (3m10s elapsed)
aws_cloudformation_stack.f5-cluster-cross-az-ha-bigip: Still creating... (3m30s elapsed)
aws_cloudformation_stack.f5-autoscale-waf: Still creating... (3m20s elapsed)
aws_cloudformation_stack.f5-cluster-cross-az-ha-bigip: Still creating... (3m30s elapsed)
aws_cloudformation_stack.f5-autoscale-waf: Still creating... (3m40s elapsed)
aws_cloudformation_stack.f5-cluster-cross-az-ha-bigip: Still creating... (3m50s elapsed)
aws_cloudformation_stack.f5-autoscale-waf: Still creating... (4m0s elapsed)
aws_cloudformation_stack.f5-autoscale-waf: Creation complete (ID: arn:aws:cloudformation:us-east-1:457112...8/cb45ffdb-83e2-11e8-b711-5044763dbb7b)

Apply complete! Resources: 23 added, 0 changed, 0 destroyed.

The state of your infrastructure has been saved to the path
below. This state is required to modify and destroy your
infrastructure, so keep it safe. To inspect the complete state
use the 'terraform show' command.

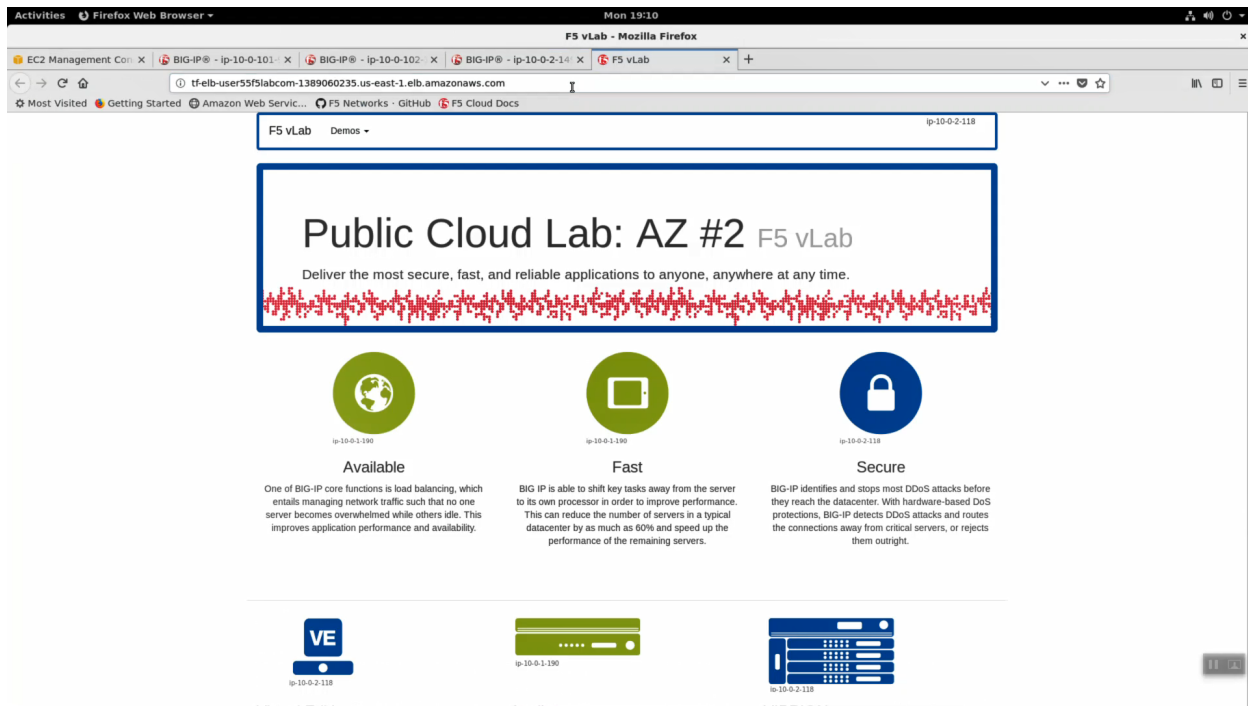
State path:

Outputs:

**aws_alias** = https://f5agility2018.signin.aws.amazon.com/console
bigipLicenseManager = 18.205.205.53
bigipExternalSecurityGroup = sg-fe4257b5
bigipManagementSecurityGroup = sg-864f3acd
elb_dns_name = tf-elb-user55f5labcom-1389060235.us-east-1.elb.amazonaws.com
managementSubnetA1 = subnet-fd7805b7
managementSubnetA2 = subnet-45db7519
restrictedIpAddress = 9.0.0.0/0
sshKey = MyKeyPair-user55f5lab.com
ssl_certificate_id = arn:aws:iam:457112751961:server-certificate/elb_cert_user55f5lab.com
vpc-id = vpc-7252a108
vpc-private-a = 10.0.100.0/24
vpc-private-a-id = subnet-37017c7d
vpc-private-b = 10.0.200.0/24
vpc-private-b-id = subnet-61c46a3d
vpc-public-a = 10.0.1.0/24
vpc-public-a-id = subnet-e27a07a8
vpc-public-b = 10.0.2.0/24
vpc-public-b-id = subnet-9ad779c6
[robert@super-netops] ~/marfil-f5-terraform $
```

2.1.8 F5 AWS Lab Test application

Note the elb_dns_name value in terraform output. HTTP to this site from any browser to see the example lab application.



2.1.9 What just happened?

This is the TL;DR version of the steps completed.

When you clone the git repository, you are pulling down a current version of the files you need to get started. These files are hosted on Github, the most popular online revision control repository, and include:

- Onboarding scripts that create your AWS account and other prerequisites: *f5-super-netops-install.sh*, *addUser.sh*, *export.sh*.
- Terraform configuration files—a declarative, comprehensive representation of our entire application stack:
 - *main.tf* - Every terraform configuration has a main.tf. This contains all of the AWS specific (non-F5) environment configuration, including web instances
 - *f5-cloudformation.tf* files - A terraform file that takes the officially supported CloudFormation template hosted in the official F5 github repo: <https://github.com/F5Networks/f5-aws-cloudformation> and stuffs all of the prerequisite parameters so we don't have to do it manually.
 - *outputs.tf* - Any variable in the outputs.tf file can be rendered to the console with 'terraform output' and is exposed to other command line tools.
 - *vars.tf* - Variables for terraform.
- Handy utilities to help move the lab along with minimum fuss: *lab-info*, *password-reset*.

The start script takes care of all of the prerequisites to standing up an AWS environment. Precisely:

- Installs all of the necessary software, including: terraform, the aws cli, and various other command line tools.
- Creates your AWS console login and api account and stores the keys locally for use by the AWS command line.
- Creates SSH keys for use by all of your EC2 instances: web servers and Big-IP virtual editions.
- Creates a self-signed SSL certificate for use in deploying https services.
- Sets the default region: us-east-1 (Virginia), ap-southeast-1 (Singapore), etc.

The terraform files go into effect when you invoke *terraform apply*. This step makes use of all of the prerequisites from the step before to build the environment in AWS.

2.2 Exploring AWS

This lab will examine the AWS Lab Environment created previously.

2.2.1 Explore the F5 / AWS lab environment

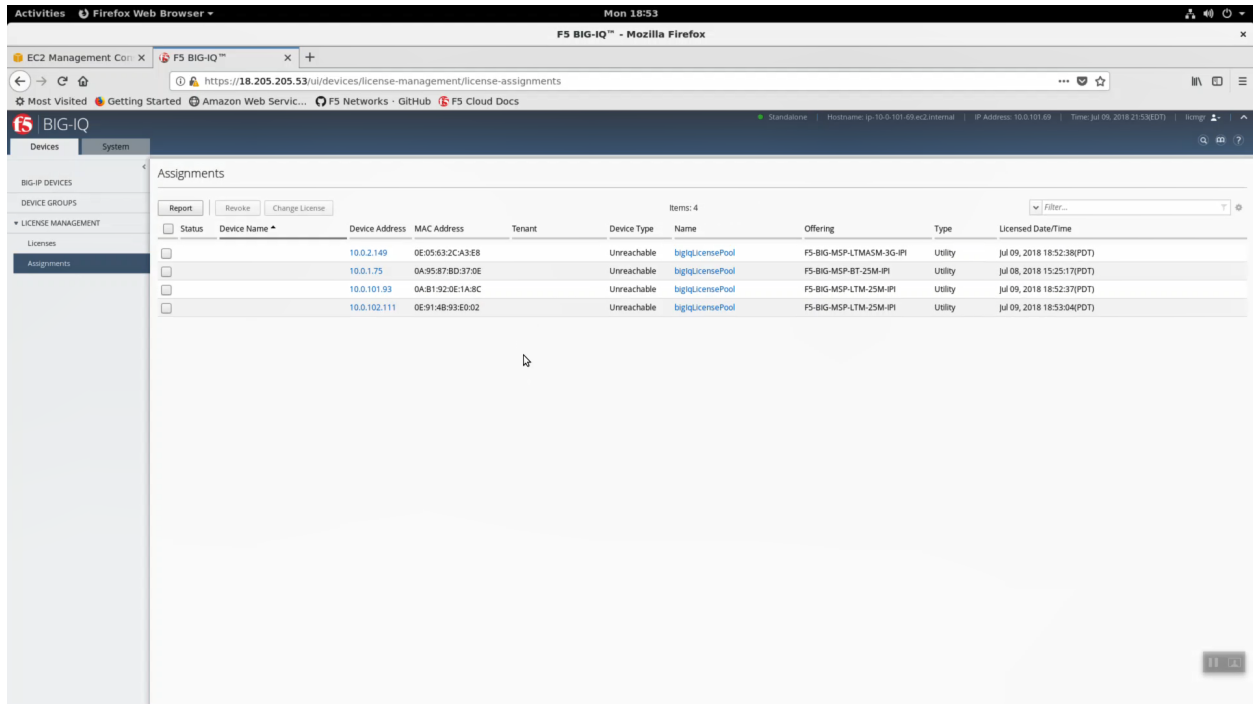
Your instructor will share a view of the Big-IQ License Manager hosted on AWS. The class will see all of the instances dynamically licensed through Big-IQ.

When deploying to AWS you have flexible licensing options:

- Bring Your Own License (BYOL) - Can be transferred from one Virtual Edition environment to another (i.e. VMWare => AWS)
- Hourly - Launch an instance from the AWS self-service Marketplace portal and pay only for metered hourly use.

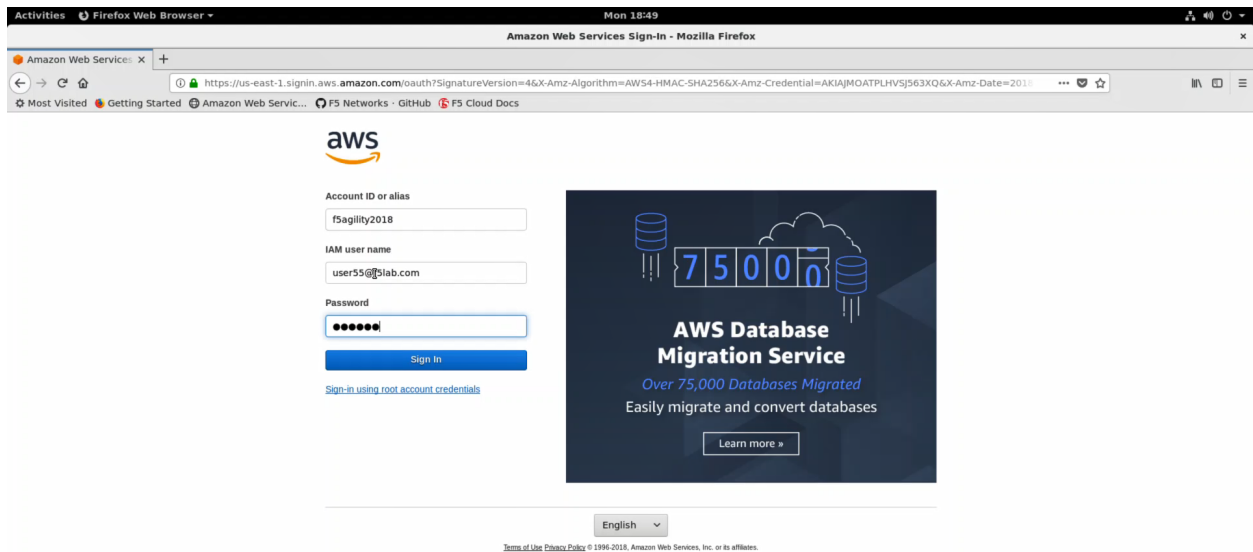
- Subscription - **This is the option used in this lab.** Every Big-IP launched will query the Big-IQ License Manager for a license. From Big-IQ we can revoke licenses as well.
- Enterprise License Agreement

Attention: Below is a snapshot of the Big-IQ License Manager dynamically licensing devices in AWS. You're instructor can show this to the class during a lab session.

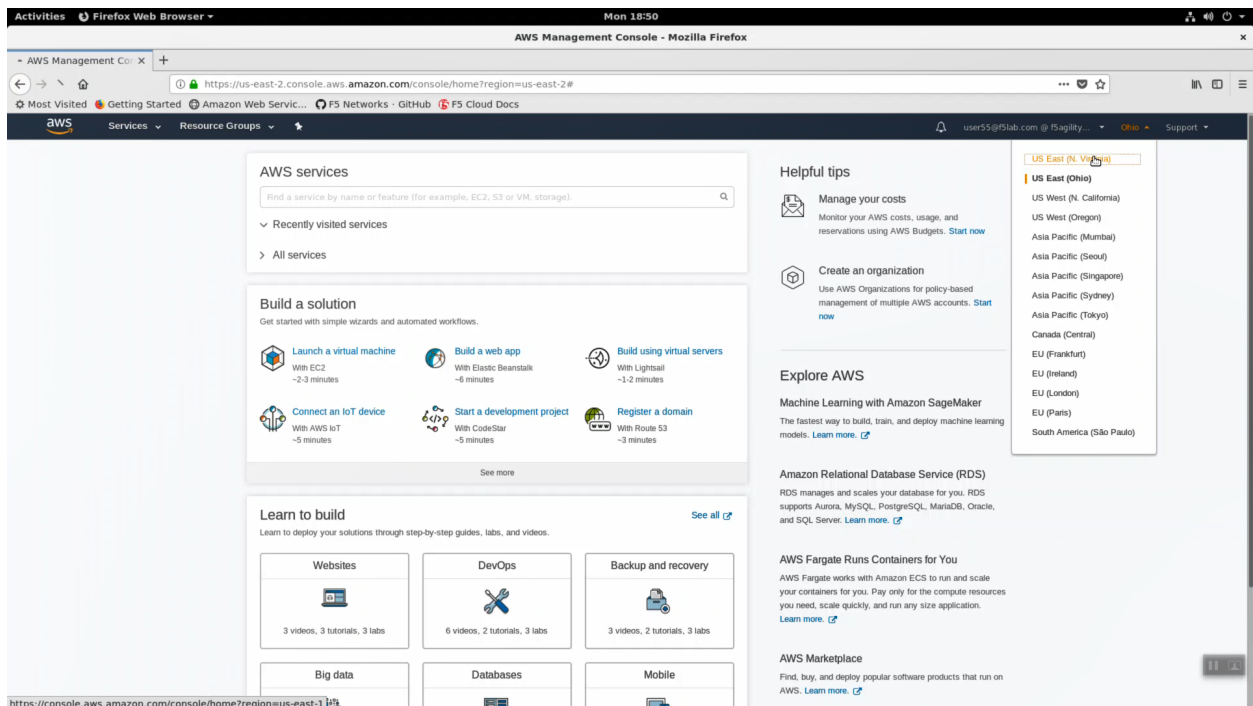


Launch the Firefox browser. Click on the bookmark for the Amazon AWS Console link in the upper-left-hand corner. Login with emailid as the username and shortUrl as password.

Parameter	value
Account:	f5agility2018
User Name:	userxx@f5lab.com, change xx to your student number
Password:	sames as shortUrl / echo \$shortUrl

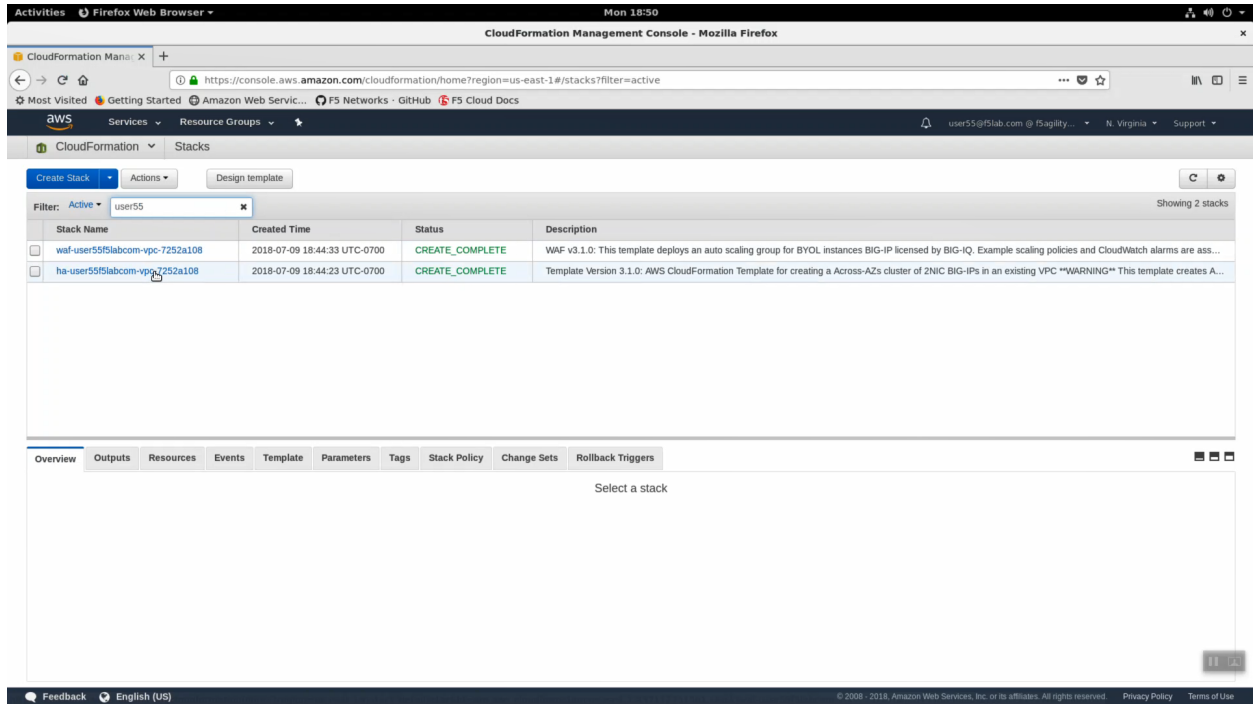


Attention: In the upper right-hand corner, ensure you are in the correct region. For example: N. Virginia region (us-east-1) is the default.

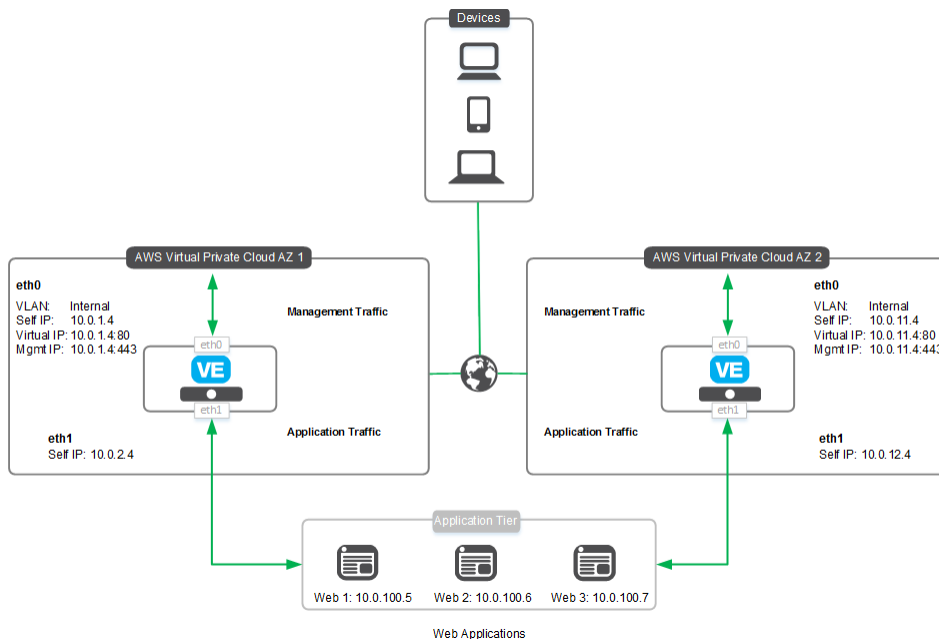


2.2.2 CloudFormation

Navigate to Services => Management Tools => CloudFormation. In the search field type your user account name (i.e user99). You should see your CloudFormation deployment details. You launched two CloudFormation templates.

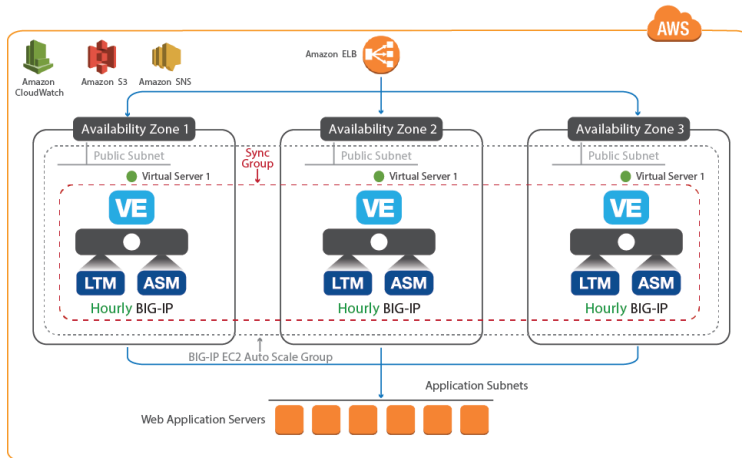


- ha-userxxf5labcom-vpc-xxxxxxx - Is the Cross-Availability-Zone deployment well documented in the F5 Github repository: <https://github.com/F5Networks/f5-aws-cloudformation/tree/master/supported/failover/across-net/via-api/3nic/existing-stack/bigip>



- waf-userXXf5labcom-vpc-xxxxxxx - Is the Autoscale WAF deployment well documented in the

F5 Github repository: <https://github.com/F5Networks/f5-aws-cloudformation/tree/master/supported/autoscale/waf/via-lb/1nic/existing-stack/bigip>



- Click the Events tab. The F5 CloudFormation template records every successful or failed event here. Look for the final “CREATE_COMPLETE” at the top. This indicates all went well.

Stack Detail - Mozilla Firefox

Stack Detail

Physical ID: am:aws:cloudformation:us-east-1:457112751961:stack/ha-user55flabcom-vpc-7252a108/c54c9c60-83e2-11e8-aa55-500c28b236fd

Physical ID	Status	Type	Logical ID	Status Reason
18:47:59 UTC-0700	CREATE_COMPLETE	AWS::CloudFormation::Stack	ha-user55flabcom-vpc-7252a108	
18:47:54 UTC-0700	CREATE_COMPLETE	AWS::EC2::Instance	Bigip2Instance	
18:47:36 UTC-0700	CREATE_IN_PROGRESS	AWS::EC2::Instance	Bigip2Instance	Resource creation Initiated
18:47:35 UTC-0700	CREATE_IN_PROGRESS	AWS::EC2::Instance	Bigip2Instance	
18:47:31 UTC-0700	CREATE_COMPLETE	AWS::EC2::Instance	Bigip1Instance	
18:47:14 UTC-0700	CREATE_IN_PROGRESS	AWS::EC2::Instance	Bigip1Instance	Resource creation Initiated
18:47:13 UTC-0700	CREATE_IN_PROGRESS	AWS::EC2::Instance	Bigip1Instance	
18:47:10 UTC-0700	CREATE_COMPLETE	AWS::IAM::InstanceProfile	bigipServiceDiscoveryProfile	
18:45:20 UTC-0700	CREATE_COMPLETE	AWS::EC2::EIPAssociation	Bigip1ManagementEipAssociati	
18:45:20 UTC-0700	CREATE_COMPLETE	AWS::EC2::EIPAssociation	Bigip2ManagementEipAssociati	
18:45:10 UTC-0700	CREATE_COMPLETE	AWS::EC2::EIPAssociation	Bigip1subnet1Az1SelfEipAssoci	
18:45:10 UTC-0700	CREATE_COMPLETE	AWS::EC2::EIPAssociation	Bigip1VpEipAssociation	
18:45:09 UTC-0700	CREATE_COMPLETE	AWS::EC2::EIPAssociation	Bigip2subnet1Az2SelfEipAssoci	
18:45:09 UTC-0700	CREATE_IN_PROGRESS	AWS::IAM::InstanceProfile	bigipServiceDiscoveryProfile	Resource creation Initiated
18:45:08 UTC-0700	CREATE_IN_PROGRESS	AWS::IAM::InstanceProfile	bigipServiceDiscoveryProfile	
18:45:05 UTC-0700	CREATE_IN_PROGRESS	AWS::EC2::EIPAssociation	Bigip1ManagementEipAssociati	Resource creation Initiated
18:45:04 UTC-0700	CREATE_COMPLETE	AWS::IAM::Role	bigipServiceDiscoveryAccessR	
18:45:04 UTC-0700	CREATE_IN_PROGRESS	AWS::EC2::EIPAssociation	Bigip2ManagementEipAssociati	Resource creation Initiated

- Click on the Outputs tab. When CloudFormation deployments complete successfully, they can export key value pairs you can use to integrate other automation tools. For example, you can query these CloudFormation outputs to find out to which region, availability zone, private IPs, public IPs your F5 Big-IP Virtual Edition instance has been assigned.

Activities Firefox Web Browser Mon 18:51 CloudFormation Management Console - Mozilla Firefox

CloudFormation Management Console - Mozilla Firefox

Most Visited Getting Started Amazon Web Services F5 Networks GitHub F5 Cloud Docs

CloudFormation Stacks

Overview **Outputs** Resources Events Template Parameters Tags Stack Policy Change Sets Rollback Triggers

Key	Value	Description	Export Name
Bigip2subnet1A22SelfEipAddress	54.174.2.152	IP Address of the External interface attached to BIG-IP	
Bigip1ManagementInterface	eni-3d8e2e59	Management interface ID on BIG-IP	
Bigip1subnet1A21Interface	eni-4c8d2d28	External interface ID on BIG-IP	
bigipExternalSecurityGroup	sg-f24752b9	Public or External Security Group	
Bigip1ManagementInterfacePrivateIp	10.0.101.93	Internally routable IP of the management interface on BIG-IP	
Bigip1ExternalInterfacePrivateIp	10.0.1.160	Internally routable IP of the public interface on BIG-IP	
availabilityZone1	us-east-1a	Availability Zone	
availabilityZone2	us-east-1b	Availability Zone	
Bigip2ManagementEipAddress	35.170.139.232	IP address of the management port on BIG-IP	
Bigip1Uih	https://18.208.30.193	BIG-IP Management GUI	
Bigip2Uih	https://35.170.139.232	BIG-IP Management GUI	
Bigip2Instanceid	i-0914900cd5f3d51e	Instance ID of BIG-IP in Amazon	
Bigip2ManagementInterfacePrivateIp	10.0.102.111	Internally routable IP of the management interface on BIG-IP	
Bigip1VipPrivateIp	10.0.1.150	VIP on External Interface Secondary IP 1	
Bigip1VipEipAddress	http://18.209.138.129.80	EIP address for VIP	
Bigip2ManagementInterface	eni-387acc09	Management interface ID on BIG-IP	
Bigip2subnet1A22Interface	eni-327acc03	External interface ID on BIG-IP	
Bigip1subnet1A21SelfEipAddress	18.209.73.87	IP Address of the External interface attached to BIG-IP	
Bigip1ManagementEipAddress	18.208.30.193	IP address of the management port on BIG-IP	
Bigip2ExternalInterfacePrivateIp	10.0.2.97	Internally routable IP of the public interface on BIG-IP	
Bigip1Instanceid	i-043e748cb06eef79	Instance ID of BIG-IP in Amazon	

Feedback English (US) © 2008 - 2018 Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- Click on the Resources tab. Here we see a map (resource type to unique id) of all the AWS resources that were deployed from the CloudFormation template.

Activities Firefox Web Browser Mon 18:51 CloudFormation Management Console - Mozilla Firefox

CloudFormation Management Console - Mozilla Firefox

Most Visited Getting Started Amazon Web Services F5 Networks GitHub F5 Cloud Docs

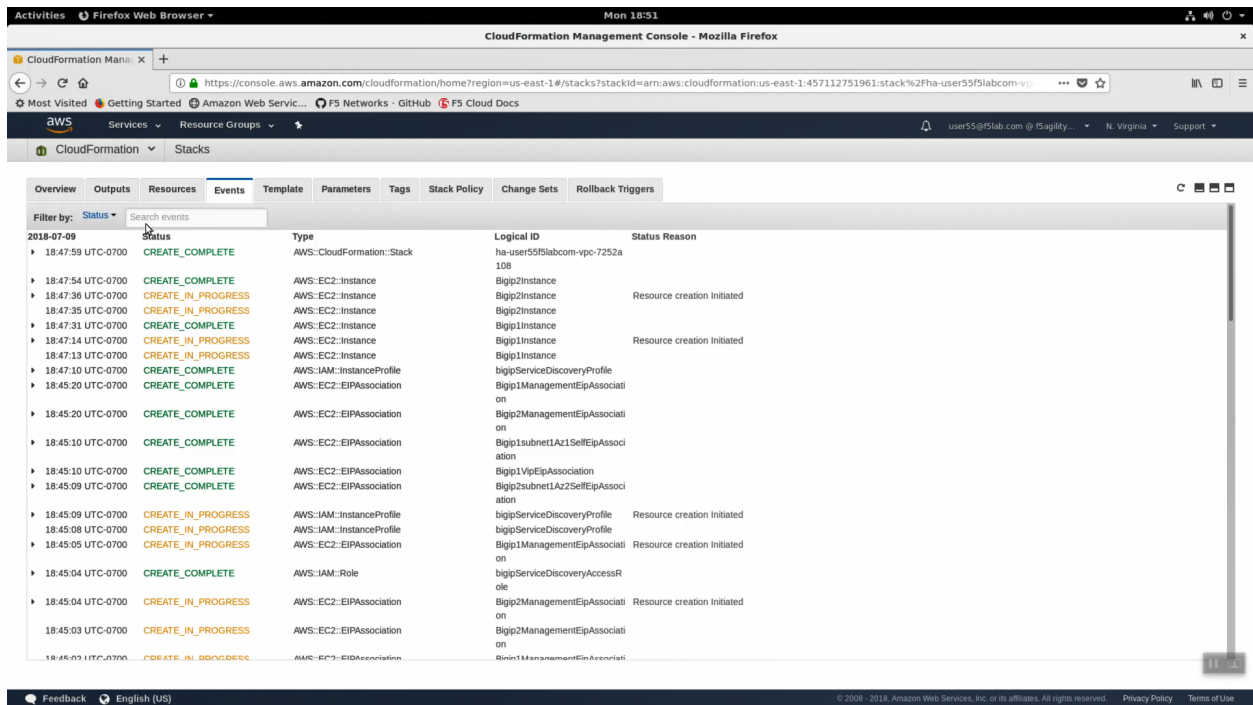
CloudFormation Stacks

Overview Outputs **Resources** Events Template Parameters Tags Stack Policy Change Sets Rollback Triggers

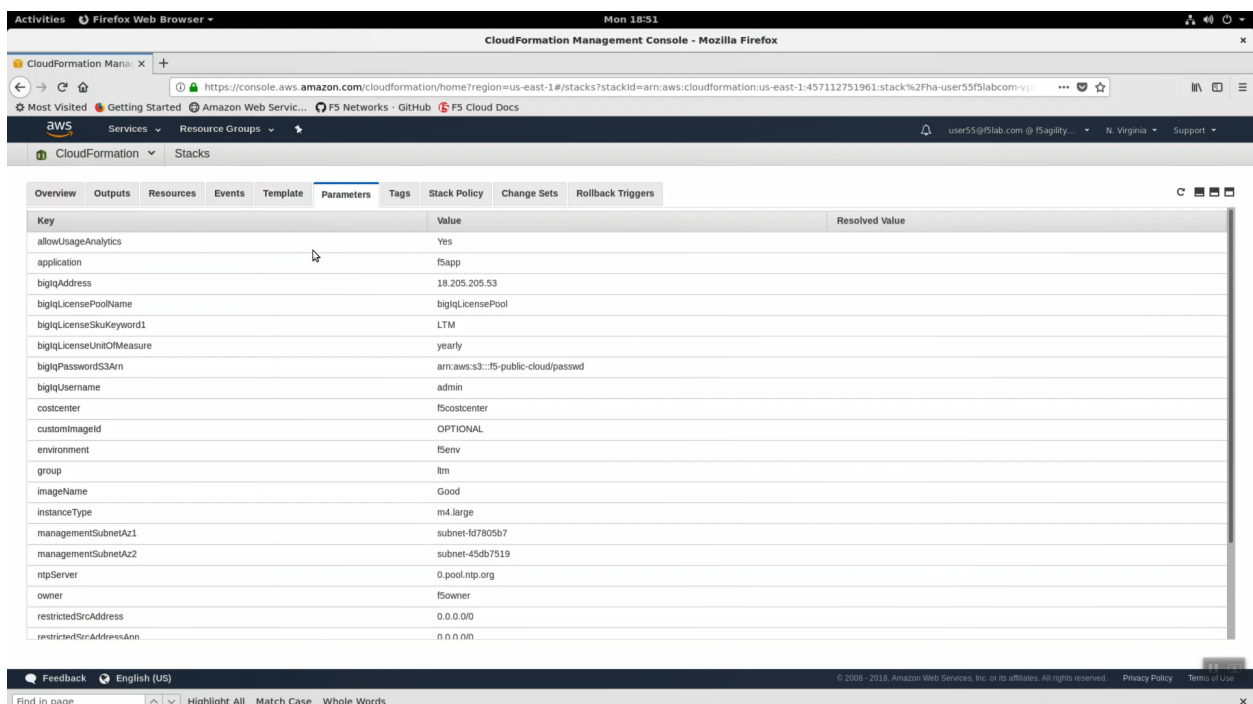
Logical ID	Physical ID	Type	Status	Status Reason
Bigip1Instance	i-043e748cb06eef79	AWS::EC2::Instance	CREATE_COMPLETE	
Bigip1ManagementEipAddress	18.208.30.193	AWS::EC2::EIP	CREATE_COMPLETE	
Bigip1ManagementEipAssociation	eipassoc-17c2f9cd	AWS::EC2::EIPAssociation	CREATE_COMPLETE	
Bigip1ManagementInterface	eni-3d8e2e59	AWS::EC2::NetworkInterface	CREATE_COMPLETE	
Bigip1VipEipAddress	18.209.138.129	AWS::EC2::EIP	CREATE_COMPLETE	
Bigip1VipEipAssociation	eipassoc-70c5faa	AWS::EC2::EIPAssociation	CREATE_COMPLETE	
Bigip1subnet1A21Interface	eni-4c8d2d28	AWS::EC2::NetworkInterface	CREATE_COMPLETE	
Bigip1subnet1A21SelfEipAddress	18.209.73.87	AWS::EC2::EIP	CREATE_COMPLETE	
Bigip1subnet1A21SelfEipAssociation	eipassoc-56c6f8bc	AWS::EC2::EIPAssociation	CREATE_COMPLETE	
Bigip2Instance	i-0914900cd5f3d51e	AWS::EC2::Instance	CREATE_COMPLETE	
Bigip2ManagementEipAddress	35.170.139.232	AWS::EC2::EIP	CREATE_COMPLETE	
Bigip2ManagementEipAssociation	eipassoc-592874f2	AWS::EC2::EIPAssociation	CREATE_COMPLETE	
Bigip2ManagementInterface	eni-387acc09	AWS::EC2::NetworkInterface	CREATE_COMPLETE	
Bigip2subnet1A22Interface	eni-327acc03	AWS::EC2::NetworkInterface	CREATE_COMPLETE	
Bigip2subnet1A22SelfEipAddress	54.174.2.152	AWS::EC2::EIP	CREATE_COMPLETE	
Bigip2subnet1A22SelfEipAssociation	eipassoc-d92c707d	AWS::EC2::EIPAssociation	CREATE_COMPLETE	
S3Bucket	ha-user55f5labcom-vpc-7252a108-s3bucket-1jndyn7ty4mz	AWS::S3::Bucket	CREATE_COMPLETE	
bigipExternalSecurityGroup	sg-f24752b9	AWS::EC2::SecurityGroup	CREATE_COMPLETE	
bigipManagementSecurityGroup	sg-6d435626	AWS::EC2::SecurityGroup	CREATE_COMPLETE	
bigipSecurityGroupingressBigipLic	bigipSecurityGroupingressBigipLic	AWS::EC2::SecurityGroupingress	CREATE_COMPLETE	
bigipSecurityGroupingressConfigSync	bigipSecurityGroupingressConfigSync	AWS::EC2::SecurityGroupingress	CREATE_COMPLETE	

Feedback English (US) © 2008 - 2018 Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- Click the Events tab. The F5 CloudFormation template records every successful or failed event here. Look for the final "CREATE_COMPLETE" at the top. This indicates all went well.



- Click on the Parameters tab. We used terraform to stuff all of the necessary parameters into the CloudFormation template. Here you can see the CloudFormation parameter name and value provided.



2.2.3 EC2

Navigate to Services => Compute => EC2 => INSTANCES => Instances. Enter your username in the search field (i.e. user99). The web application is hosted on webaz1.0 in one availability zone and webaz2.0

in another availability zone. Highlight web-az1.0.

The screenshot shows the AWS Management Console in the 'Instances' tab. A search filter 'user55' is applied. The table lists several EC2 instances. The instance 'web-az1.0' is highlighted. It is an 'm4.large' instance in the 'us-east-1a' availability zone, in a 'running' state. The instance has a public IP address of '174.129.63.235' and a private IP address of '18.208.30.193'. The instance is associated with the 'MyKeyPair-us...' key pair and was launched on July 9, 2018.

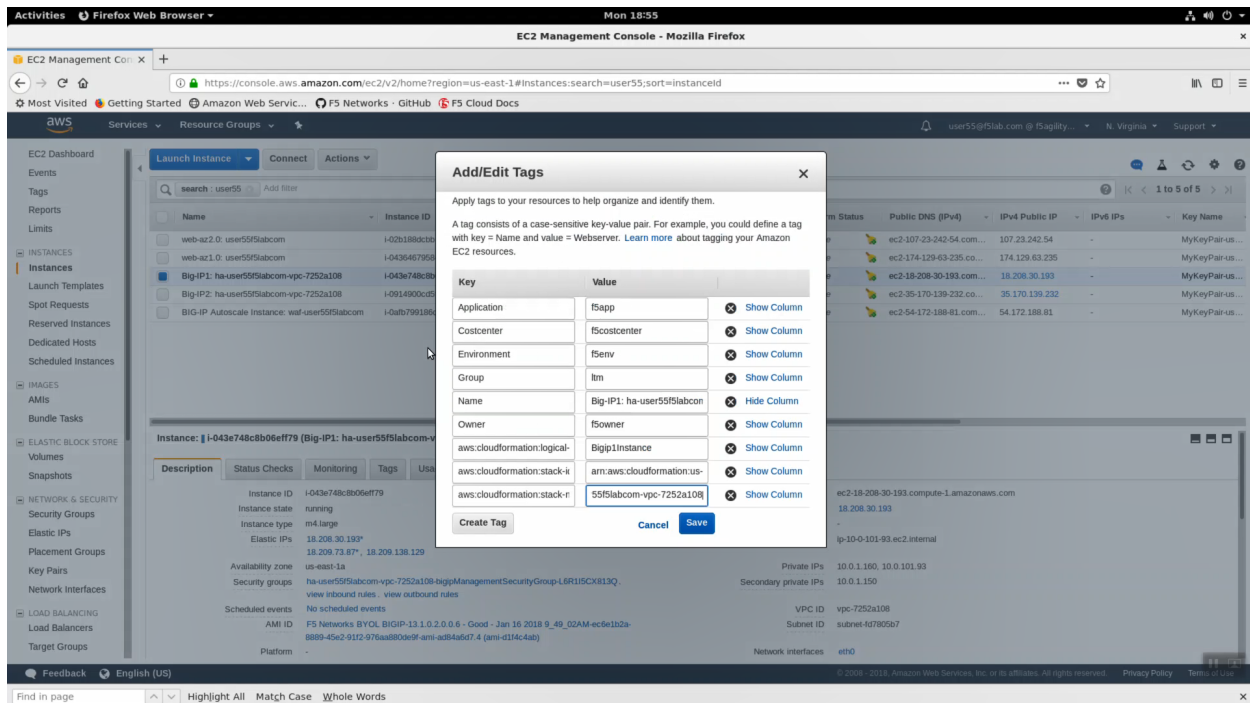
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name	Monitoring	Launch Time
web-az2.0	i-02b188dc6b1d283d0	t2.micro	us-east-1b	running	2/2 checks ...	None	ec2-107-23-242-54.com...	107.23.242.54	-	MyKeyPair-us...	disabled	July 9, 2018
web-az1.0	i-0436467956c62c0e3	t2.micro	us-east-1a	running	2/2 checks ...	None	ec2-174-129-63-235.co...	174.129.63.235	-	MyKeyPair-us...	disabled	July 9, 2018
Big-IP1: ha-u...	i-043e748c8b06eff79	m4.large	us-east-1a	running	2/2 checks ...	None	ec2-18-208-30-193.com...	18.208.30.193	-	MyKeyPair-us...	disabled	July 9, 2018
Big-IP2: ha-u...	i-091490cc5f33d51e	m4.large	us-east-1b	running	2/2 checks ...	None	ec2-35-170-139-232.co...	35.170.139.232	-	MyKeyPair-us...	disabled	July 9, 2018
BIG-IP Auto...	i-0a0b799186d2a7ee6	m4.xlarge	us-east-1b	running	2/2 checks ...	None	ec2-54-172-188-81.com...	54.172.188.81	-	MyKeyPair-us...	disabled	July 9, 2018

- In the “Description” tab below, note the availability zone. Highlight web-az2.0 and do the same.

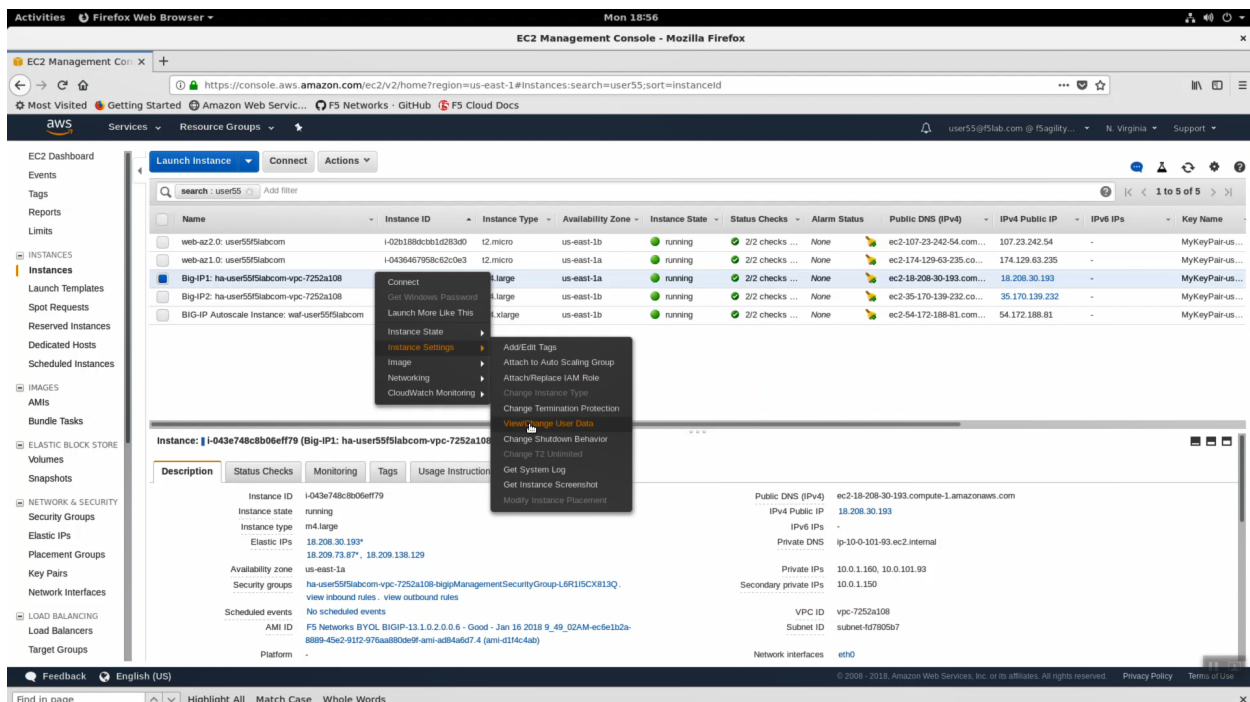
The screenshot shows the 'Description' tab for the instance 'web-az2.0'. The instance is in the 'us-east-1b' availability zone. The 'Description' tab shows details about the instance, including its state, type, availability zone, security groups, and network configuration.

Field	Value
Instance ID	i-02b188dc6b1d283d0
Instance state	running
Instance type	t2.micro
Elastic IPs	18.208.30.193*
Availability zone	us-east-1b
Security groups	ha-user55labcom-vpc-7252a108-bigipManagementSecurityGroup-L6R15CX813Q
Scheduled events	No scheduled events
AMI ID	F5 Networks BYOL BIGIP-13.1.0.2.0.0.6 - Good - Jan 16 2018 9_49_02AM-ec2e1b2a-8889-45e2-9152-975aa8f00e9f-ami-ad84ad07.4 (ami-d14c4ab)
Platform	-
IAM role	-
Key pair name	MyKeyPair-user55@f5lab.com
EBS-optimized	False
Root device type	ebs
Root device	/dev/xvda
Block devices	/dev/xvda
Elastic GPU	-
Elastic GPU type	-
Elastic GPU status	-
Public DNS (IPv4)	ec2-18-208-30-193.compute-1.amazonaws.com
IPv4 Public IP	18.208.30.193
IPv6 IPs	-
Private DNS	ip-10-0-101-93.ec2.internal
Private IPs	10.0.1.160, 10.0.101.93
Secondary private IPs	10.0.1.150
VPC ID	vpc-7252a108
Subnet ID	subnet-f0780967
Network interfaces	eth0, eth1
Source/dest. check	True
T2 Unlimited	-
Owner	457112751961
Launch time	July 9, 2018 at 6:47:14 PM UTC-7 (less than one hour)
Termination protection	False
Lifecycle	normal
Monitoring	basic
Alarm status	None
Kernel ID	-
RAM disk ID	-

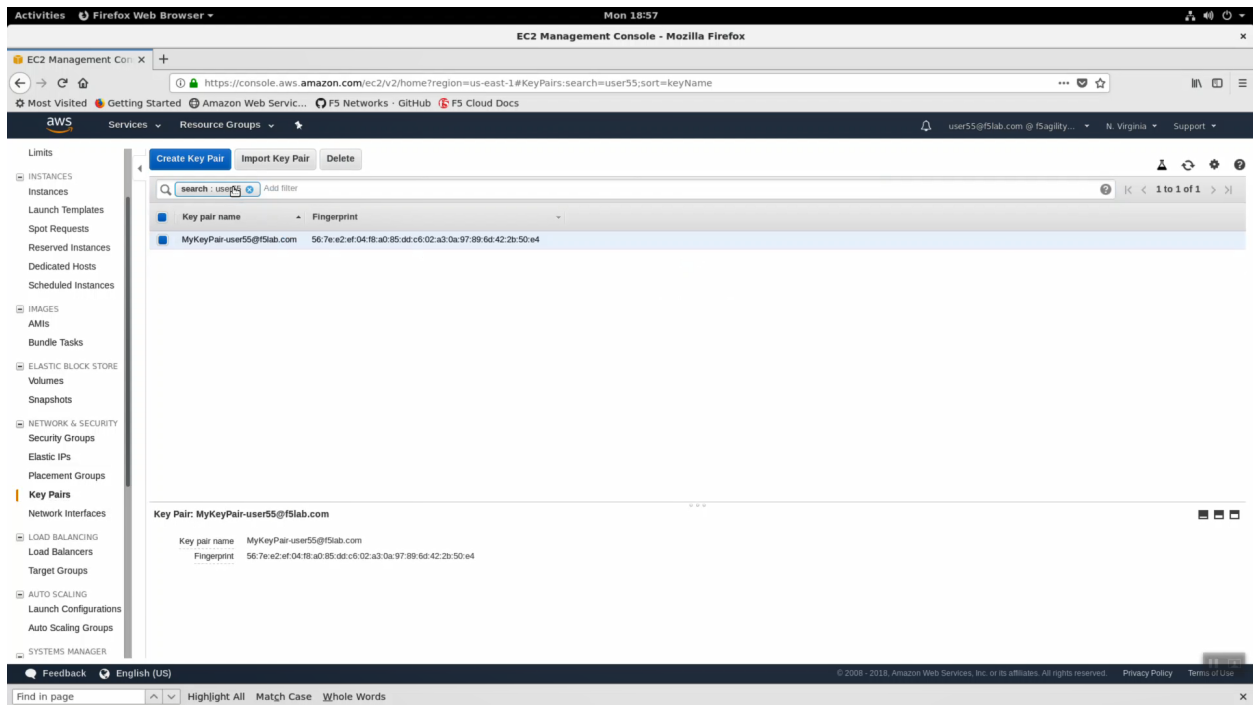
- Take a look at the tags big-IP1-ha... has been assigned. In public cloud deployments you can use tags (key-value pairs) to group your devices.



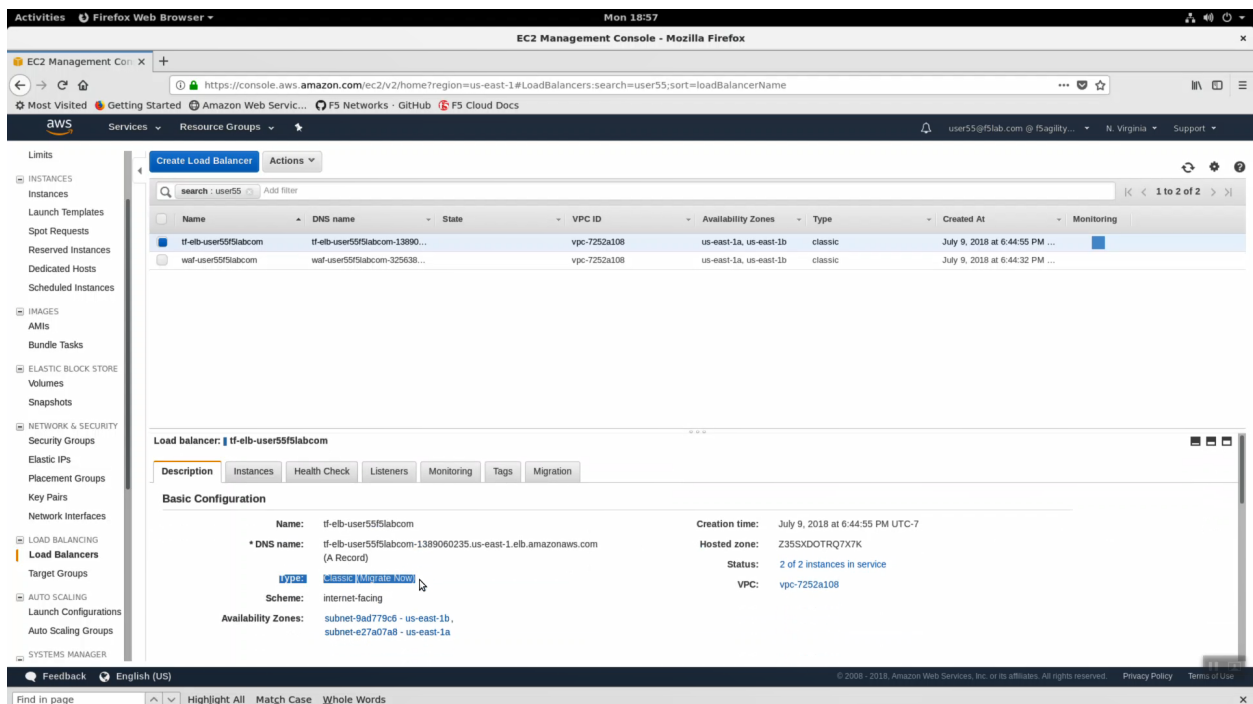
- Cloud-init. Version 13 of Big-IP supports cloud-init. Right click on BIGIP1 => Instance Settings => View/Change User Data. Cloud-init is the industry standard way to inject commands into an F5 cloud image to automate all aspects of the on-boarding process: <https://cloud-init.io/>.



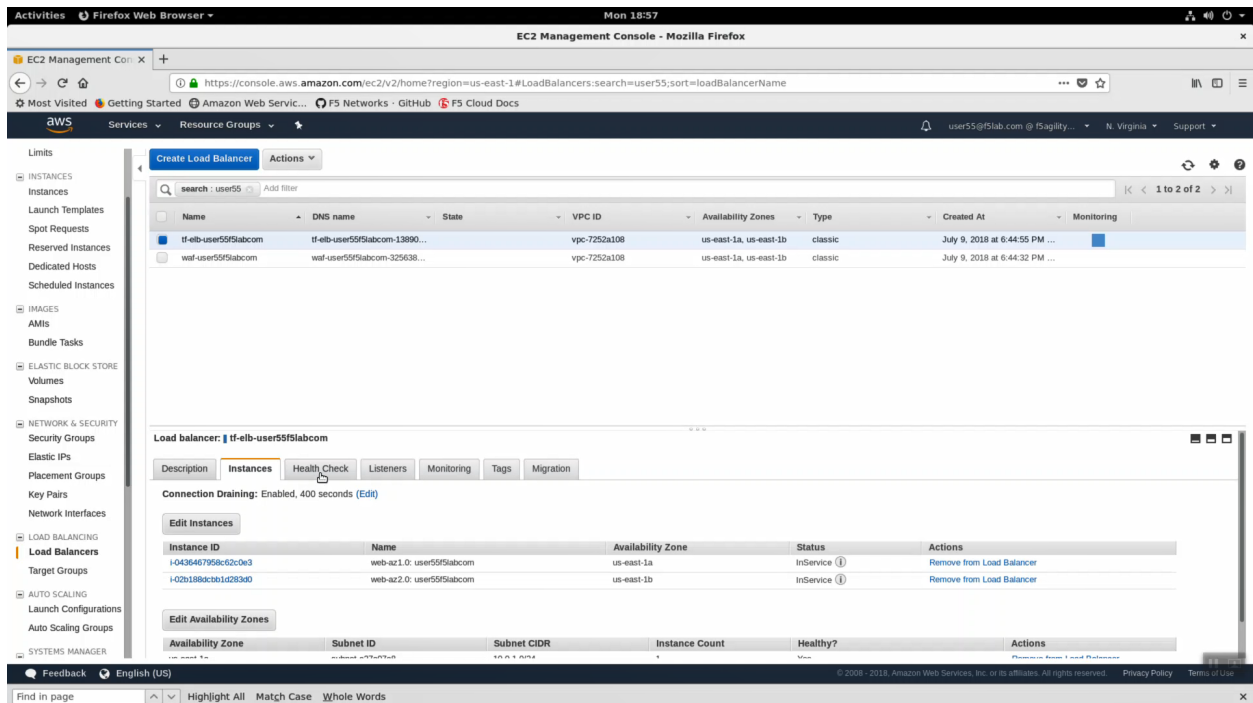
Navigate to Services => Compute => EC2 => # Key Pairs. Type your username in the search field (i.e. user99). You will see the ssh key that was created for you and upload to AWS. By default, F5 Big-IP VE appliances deployed to AWS do not have any default root or admin account access. You have to enable or create these accounts. Initially, you can only connect via ssh using your private key. From the Super-NetOps terminal, see if you can find the private key in your home directory.



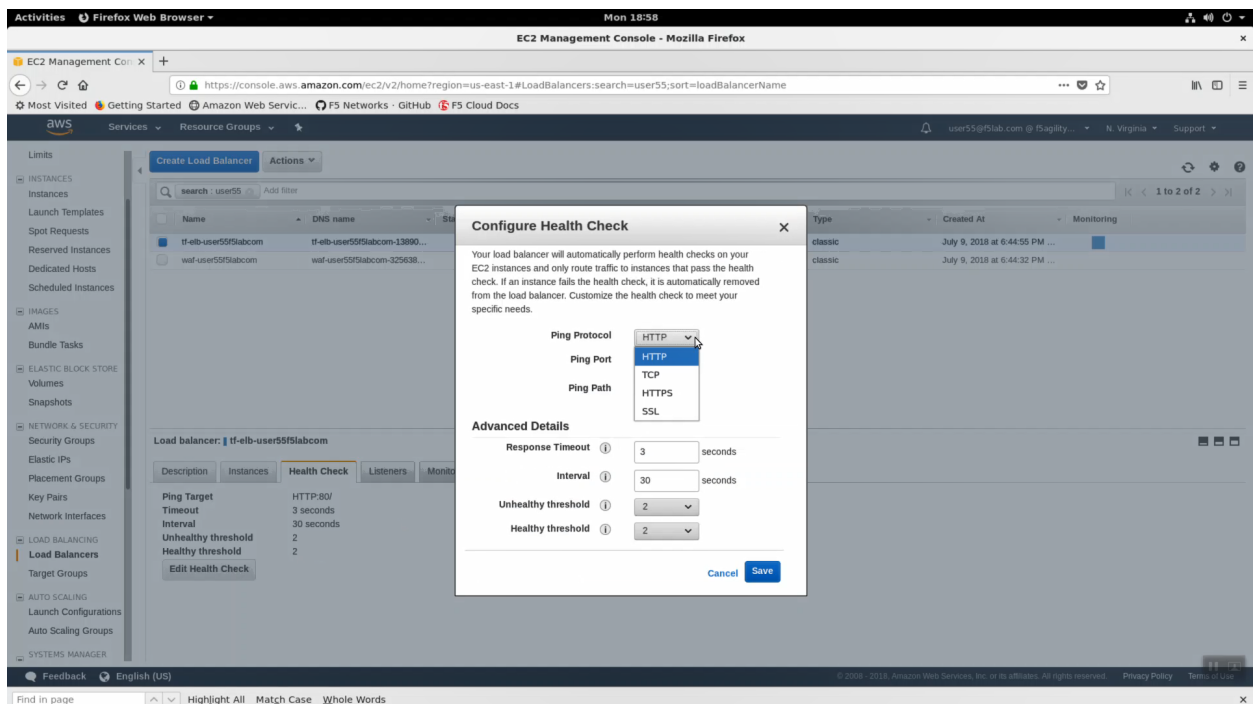
Navigate to Services => Compute => EC2 => LOAD BALANCING => Load Balancers. In the search filter enter your username. You should see two load balancers. One named tf-elb-* is your newly created AWS load balancer.



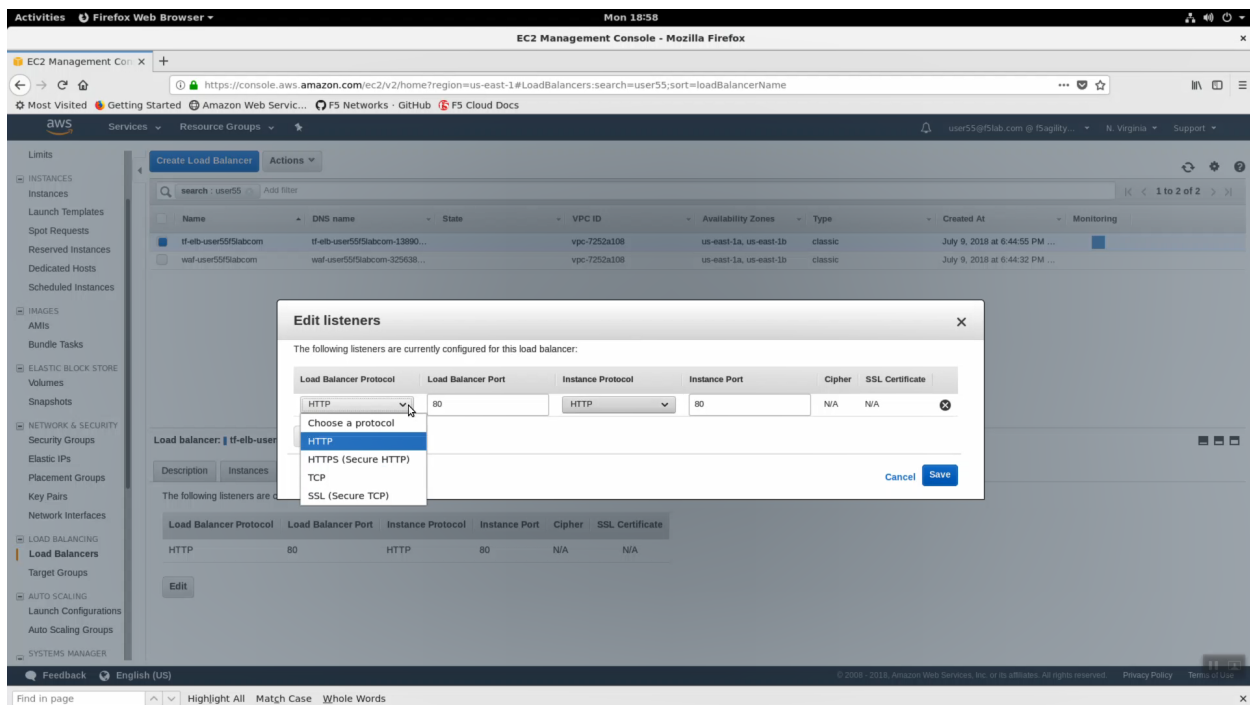
- Highlight the 'Description' tab. Note:
 - Scheme: internet-facing
 - Type: Classic



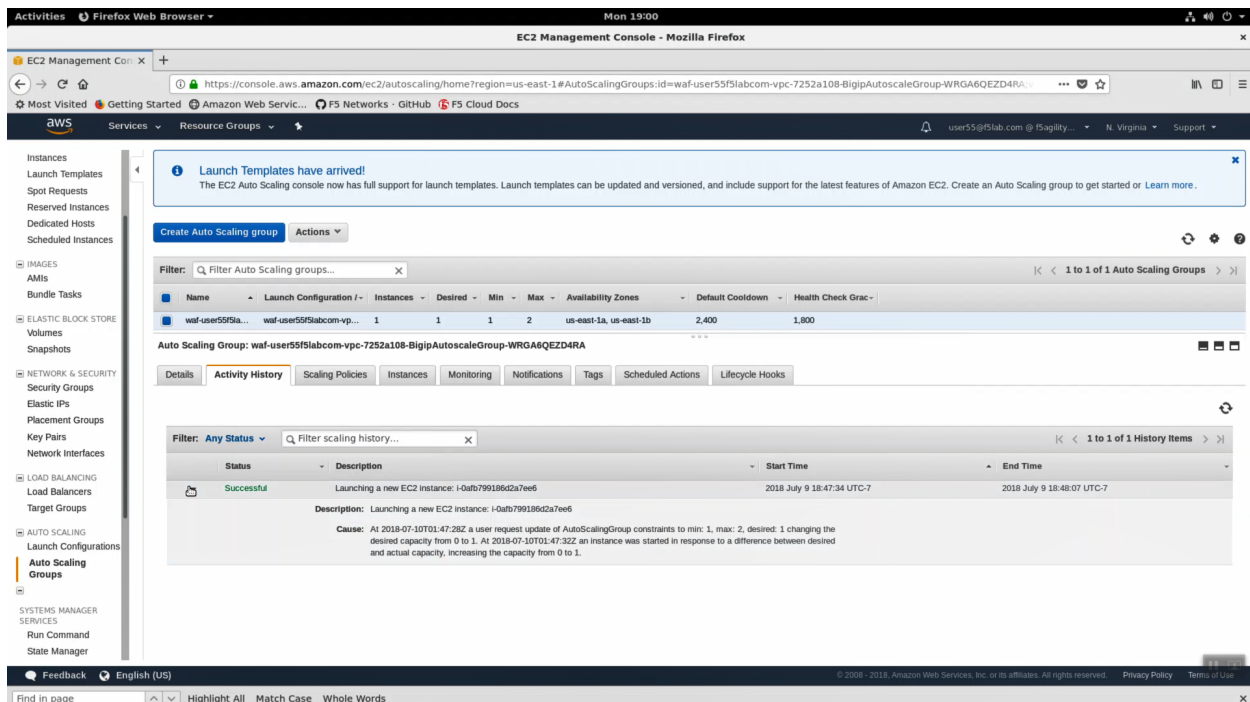
- Click the “Health Check” tab => [Edit health Check]. The classic load-balancer is limited to basic health checks.



- Click the “Listeners” tab => [Edit]. The classic load-balancer is limited to HTTP, HTTPS, TCP and SSL (no UDP).



Navigate to Services => Compute => EC2 => AUTO SCALING => Auto Scaling Group. Highlight the “Activity History” tab. You can the autoscale WAF CloudFormation template created an auto scaling group. Read the Description and Cause.



- Click the “Scaling Policies” tab. Read through the polices to understand how the autoscale WAF deployment is programmed to both scale out during a surge and scale in when the surge subsides.

Activities Firefox Web Browser EC2 Management Console - Mozilla Firefox

EC2 Management Console

Most Visited Getting Started Amazon Web Services F5 Networks GitHub F5 Cloud Docs

user55@f5lab.com @ f5agility... N. Virginia Support

Instances

Launch Templates

Spot Requests

Reserved Instances

Dedicated Hosts

Scheduled Instances

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

NETWORK & SECURITY

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

LOAD BALANCING

Load Balancers

Target Groups

AUTO SCALING

Launch Configurations

Auto Scaling Groups

SYSTEMS MANAGER

SERVICES

Run Command

State Manager

Launch Templates have arrived!

The EC2 Auto Scaling console now has full support for launch templates. Launch templates can be updated and versioned, and include support for the latest features of Amazon EC2. Create an Auto Scaling group to get started or [Learn more](#).

Create Auto Scaling group Actions

Filter: Filter Auto Scaling groups...

Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grac
waf-user55f5labcom-vp...	waf-user55f5labcom-vp...	1	1	1	2	us-east-1a, us-east-1b	2,400	1,800

Auto Scaling Group: waf-user55f5labcom-vpc-7252a108-BigipAutoscaleGroup-WRGA6QEZD4RA

Details Activity History Scaling Policies Instances Monitoring Notifications Tags Scheduled Actions Lifecycle Hooks

Add policy

waf-user55f5labcom-vpc-7252a108-BigipScaleDownPolicy-1RRYQBLEAOAD

Policy type: Simple scaling

Execute policy when: waf-user55f5labcom-vpc-7252a108-BigipLowCpuAlarm-UX6XJKQNL62Q breaches the alarm threshold: tmm-stat < 0 for 10 consecutive periods of 300 seconds for the metric dimensions

Take the action: Remove 1 instances

And then wait: 2400 seconds before allowing another scaling activity

waf-user55f5labcom-vpc-7252a108-BigipScaleUpPolicy-1JPD8YR0YBZVR

Policy type: Simple scaling

Execute policy when: waf-user55f5labcom-vpc-7252a108-BigipHighCpuAlarm-31S1WER46NFD breaches the alarm threshold: tmm-stat > 80 for 60 seconds

Feedback English (US)

© 2008 - 2018 Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Find in page Highlight All Match Case Whole Words

- Click the “Instances” tab. The single instance running the F5 WAF. Notice the instance is “Protected from: Scale in”. This means that AWS will guarantee a minimum of one F5 WAF instance is running at all times. If someone were to accidentally stop or terminate an instance, this policy would automatically trigger the creation of a new one.

Activities Firefox Web Browser EC2 Management Console - Mozilla Firefox

EC2 Management Console

Most Visited Getting Started Amazon Web Services F5 Networks GitHub F5 Cloud Docs

user55@f5lab.com @ f5agility... N. Virginia Support

Instances

Launch Templates

Spot Requests

Reserved Instances

Dedicated Hosts

Scheduled Instances

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

NETWORK & SECURITY

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

LOAD BALANCING

Load Balancers

Target Groups

AUTO SCALING

Launch Configurations

Auto Scaling Groups

SYSTEMS MANAGER

SERVICES

Run Command

State Manager

Launch Templates have arrived!

The EC2 Auto Scaling console now has full support for launch templates. Launch templates can be updated and versioned, and include support for the latest features of Amazon EC2. Create an Auto Scaling group to get started or [Learn more](#).

Create Auto Scaling group Actions

Filter: Filter Auto Scaling groups...

Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grac
waf-user55f5labcom-vp...	waf-user55f5labcom-vp...	1	1	1	2	us-east-1a, us-east-1b	2,400	1,800

Auto Scaling Group: waf-user55f5labcom-vpc-7252a108-BigipAutoscaleGroup-WRGA6QEZD4RA

Details Activity History Scaling Policies **Instances** Monitoring Notifications Tags Scheduled Actions Lifecycle Hooks

Actions

Filter: Any Health Status Any Lifecycle State Filter instances...

Instance ID	Lifecycle	Launch Configuration Name	Availability Zone	Health Status	Protected from
i-04b799189d2a7ee6	InService	waf-user55f5labcom-vpc-7252a108-BigipLaunchConfig-NFZCOEF8ZV16	us-east-1b	Healthy	Scale In

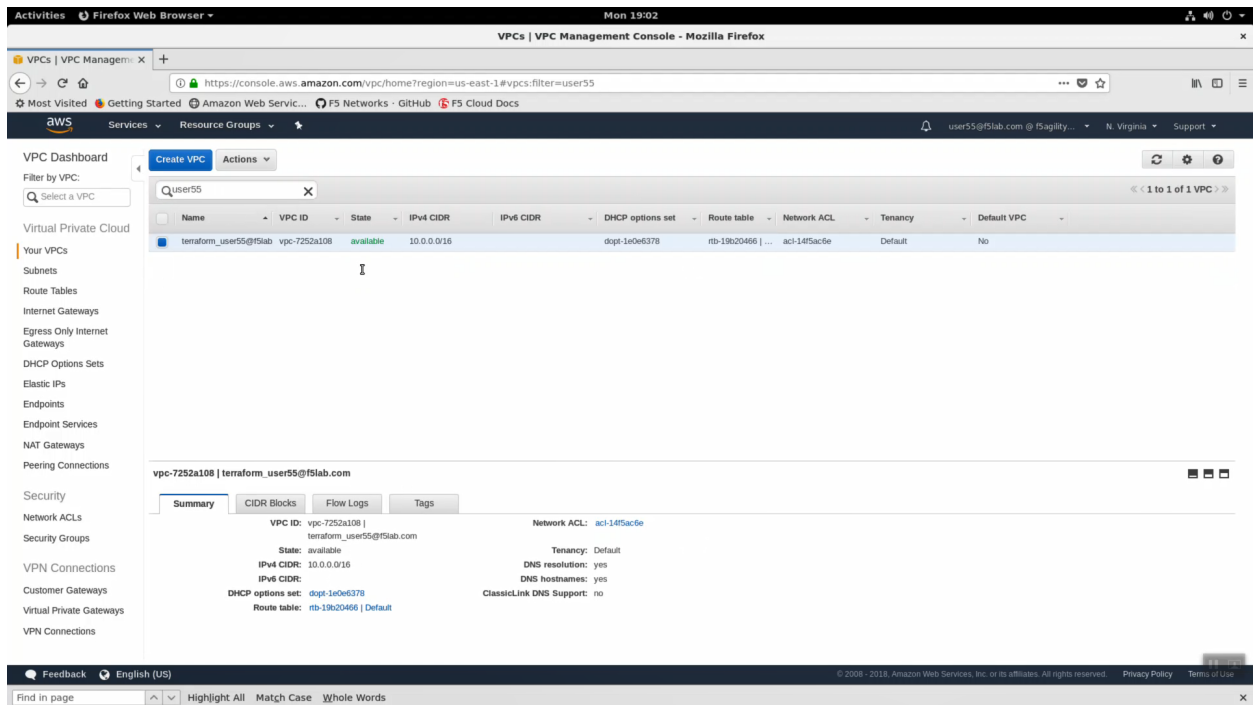
Feedback English (US)

© 2008 - 2018 Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Find in page Highlight All Match Case Whole Words

2.2.4 VPC

Navigate to Services => Networking & Content Deliver => VPC. click on VPCs. Enter your username in the search filter (i.e. user99). This is the Virtual Private Cloud (VPC) that has been dedicated to your lab environment. Select the Summary tab. You can see the IPv4 CIDR assigned is 10.0.0.0/16. Your on-premises datacenter has been assigned 10.1.0.0/16 to not conflict.



2.2.5 Github

- Fully supported F5 Networks Solutions are hosted in the official F5 Networks GitHub repository: <https://github.com/f5networks>
- We are running the lab from the F5 Super-NetOps container: <https://github.com/f5devcentral/f5-super-netops-container>
- AWS CloudFormation templates: <https://github.com/F5Networks/f5-aws-cloudformation>
- Native template formats are also available for Microsoft Azure (arm templates): <https://github.com/F5Networks/f5-azure-arm-templates>
- Native template formats are also available for Google Cloud Platform (gdm templates): <https://github.com/F5Networks/f5-google-gdm-templates>

← → ↻ GitHub, Inc. [US] | https://github.com/f5networks

This organization Search Pull requests Issues Marketplace Gist

f5 F5 Networks ¹

F5 Networks Solutions - Please also see https://www.github.com/f5devcentral for Community Solutions

📍 Seattle 🌐 https://f5.com

Repositories People 0

Pinned repositories

f5-common-python

Python SDK for configuration and monitoring of F5® BIG-IP® devices via the iControl® REST API.

Python ★ 93 🍏 40

f5-ansible

Ansible modules that can manipulate F5 products

Python ★ 89 🍏 53

f5-openstack-lbaasv2-driver

F5 LBaaSv2 service provider driver for OpenStack Liberty and beyond

Python ★ 9 🍏 27

k8s-bigip-ctrl

Repository for F5 BIG-IP Controller for Kubernetes.

Go ★ 16 🍏 13

f5-aws-cloudformation

CloudFormation Templates for quickly deploying BIG-IP services in Amazon Web Services EC2

Python ★ 25 🍏 9

f5-azure-arm-templates

Azure Resource Manager Templates for quickly deploying BIG-IP services in Azure

Shell ★ 13 🍏 12

Search repositories... Type: All Language: All

f5-openstack-agent

The F5 Agent for OpenStack Neutron allows you to deploy BIG-IP services in an OpenStack environment.

openstack bigip

Python ★ 5 🍏 24 Updated 20 hours ago

f5-openstack-lbaasv2-driver

F5 LBaaSv2 service provider driver for OpenStack Liberty and beyond

openstack load-balancer openstack-neutron f5-ltm

Python ★ 9 🍏 27 Updated 20 hours ago

f5-openstack-docs

Technical documentation related to the use of F5 products and tools in OpenStack

openstack bigip f5-ltm

★ 2 🍏 10 Updated 21 hours ago

Top languages

Python JavaScript Shell Tcl Go

Most used topics

bigip openstack openstack-neutron orchestration automation

People 0 >

This organization has no public members. You must be a member to see who's a part of this organization.

2.3 Explore the F5 Big-IP Virtual Editions Deployed

In this lab we'll take a close look at the Big-IP Virtual Editions deployed.

2.3.1 Explore the F5 Big-IP Virtual Editions Deployed

From the Super-NetOps terminal, run the handy lab-info utility. Confirm that "MCPD is up, System Ready" for all three of your instances.

```
lab-info
```

```
Activities Terminal Mon 19:04 student@docker: -
File Edit View Search Terminal Help
managementSubnetA22 = subnet-45db7519
restrictedSrcAddress = 0.0.0.0/0
sshKey = MyKeyPair-user55@f5lab.com
ssl_certificate_id = arn:aws:iam::457112751901:server-certificate/elb_cert_user55@f5lab.com
vpc-id = vpc-7252a108
vpc-private-a = 10.0.100.0/24
vpc-private-a-id = subnet-37017c7d
vpc-private-b = 10.0.200.0/24
vpc-private-b-id = subnet-61c46a3d
vpc-public-a = 10.0.1.0/24
vpc-public-a-id = subnet-e27a07a8
vpc-public-b = 10.0.2.0/24
vpc-public-b-id = subnet-9ad779c6
[root@f5-super-netops] [~/marfil-f5-terraform] $ lab-info
AWS Console
URL: https://f5agility2018.signin.aws.amazon.com/console?us-east-1
Username: user55@f5lab.com / Password: cloudy
WAF ELB
URL: https://waf-user55f5labcom-3256385.us-east-1.elb.amazonaws.com

web-az1.0: user55f5labcom
PRIVATE IP: 10.0.1.190

Big-IP1: ha-user55f5labcom-vpc-7252a108
MGMT IP: 10.208.30.193
STATUS: MCPD is up, System Ready
MGMT URL: https://10.208.30.193
SSH: ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no -o ConnectTimeout=3 -i MyKeyPair-user55@f5lab.com.pem admin@10.208.30.193
VIP IP: 10.0.1.190
Elastic IP: 10.209.130.129

web-az2.0: user55f5labcom
PRIVATE IP: 10.0.2.118

BIG-IP Autoscale Instance: waf-user55f5labcom
MGMT IP: 54.172.188.81
STATUS: MCPD is up, System Ready
MGMT URL: https://54.172.188.81:8443
SSH: ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no -o ConnectTimeout=3 -i MyKeyPair-user55@f5lab.com.pem admin@54.172.188.81

Big-IP2: ha-user55f5labcom-vpc-7252a108
MGMT IP: 35.170.139.232
STATUS: MCPD is up, System Ready
MGMT URL: https://35.170.139.232
SSH: ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no -o ConnectTimeout=3 -i MyKeyPair-user55@f5lab.com.pem admin@35.170.139.232
VIP IP: 10.0.2.58

[root@f5-super-netops] [~/marfil-f5-terraform] $
```

Attention: Do not attempt to reset the Big-IP password until **MCPD is up, System Ready**.

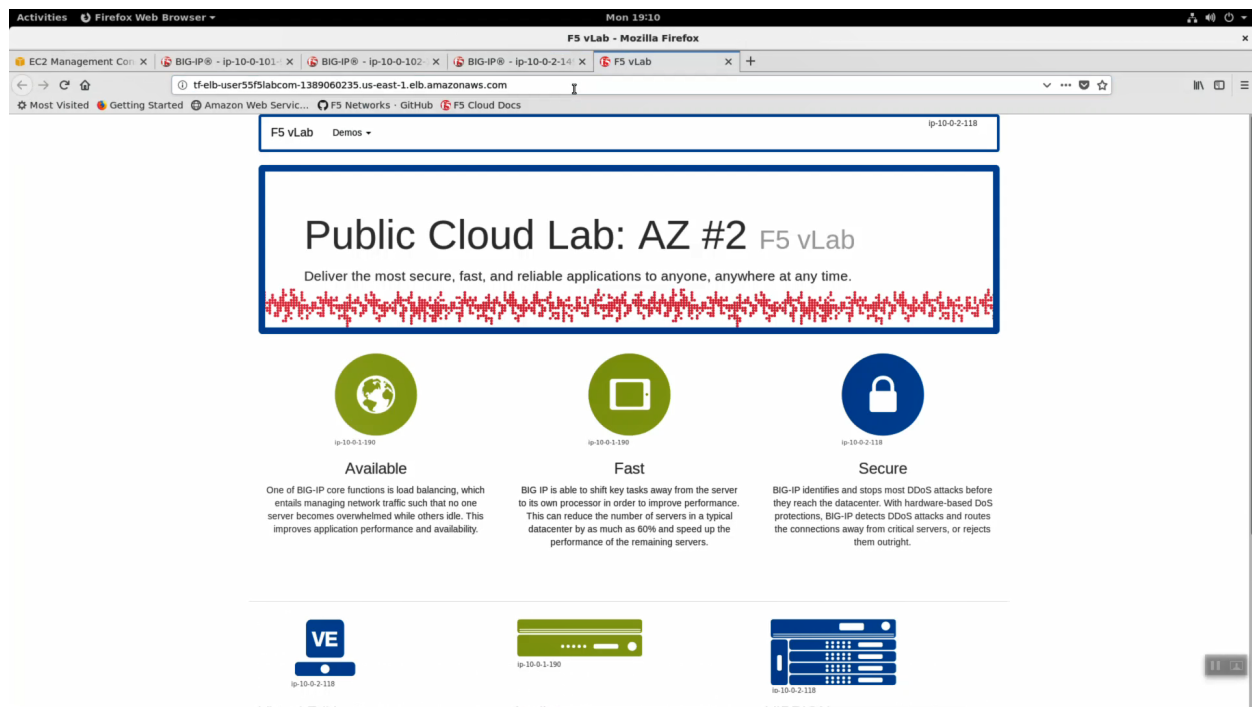
Initially, you can only login to an F5 Big-IP VE in AWS via SSH using an SSH key. You will have to enable admin and root password access. Invoke the reset-password utility with the IP address of each of your Big-IP VE's as the argument. **REPLACE THE x.x.x.x PLACEHOLDER WITH THE MANAGEMENT IP ADDRESSES OF YOUR THREE F5 BIG-IP VE'S.** This will enable the admin account on all three of your Big-IP's and change the password to the value of the shortUrl.

```
reset-password x.x.x.x
reset-password y.y.y.y
reset-password z.z.z.z
```

Run terraform output and note the value of elb_dns_name.

```
terraform output
```

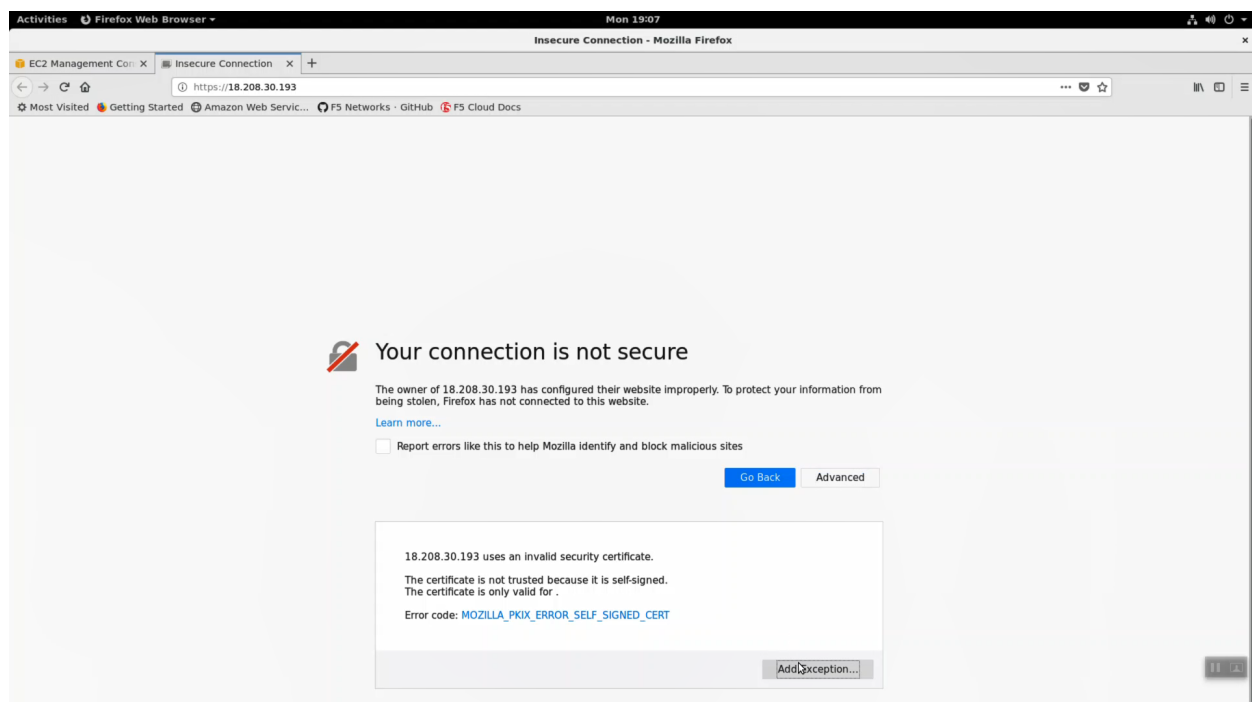
Open a new tab in the Firefox browser. HTTP to elb_dns_name. Confirm the sample application is up.



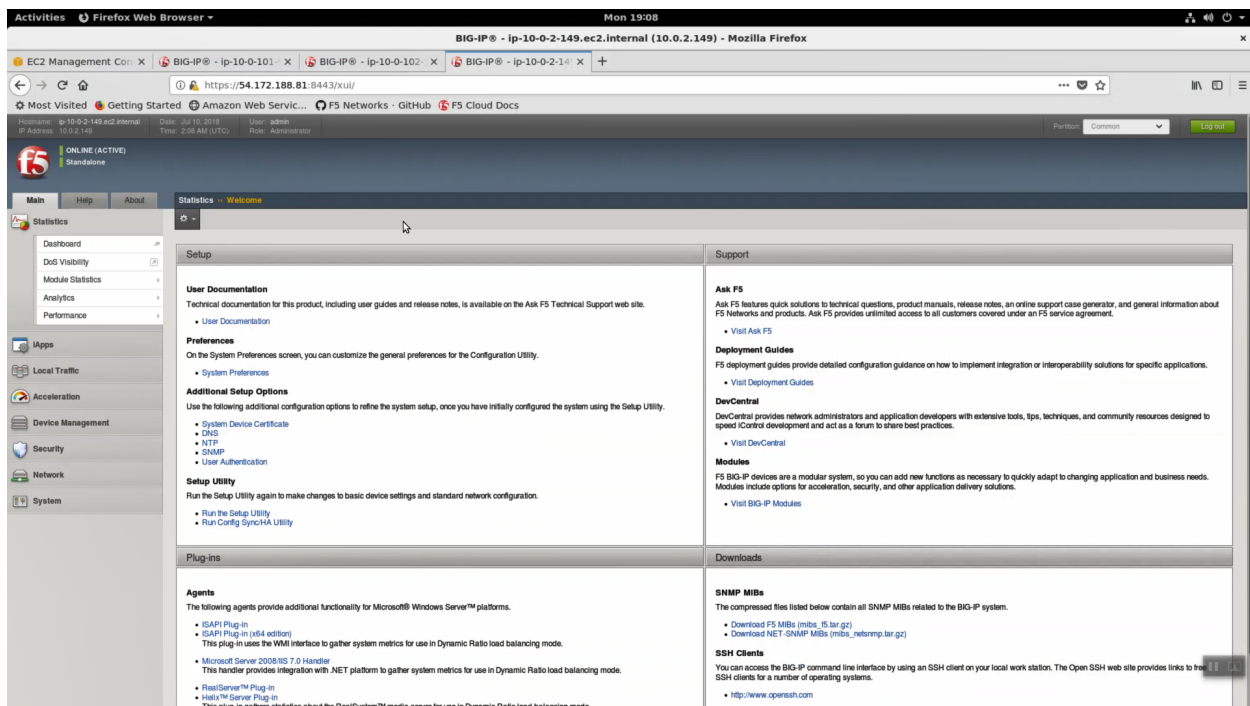
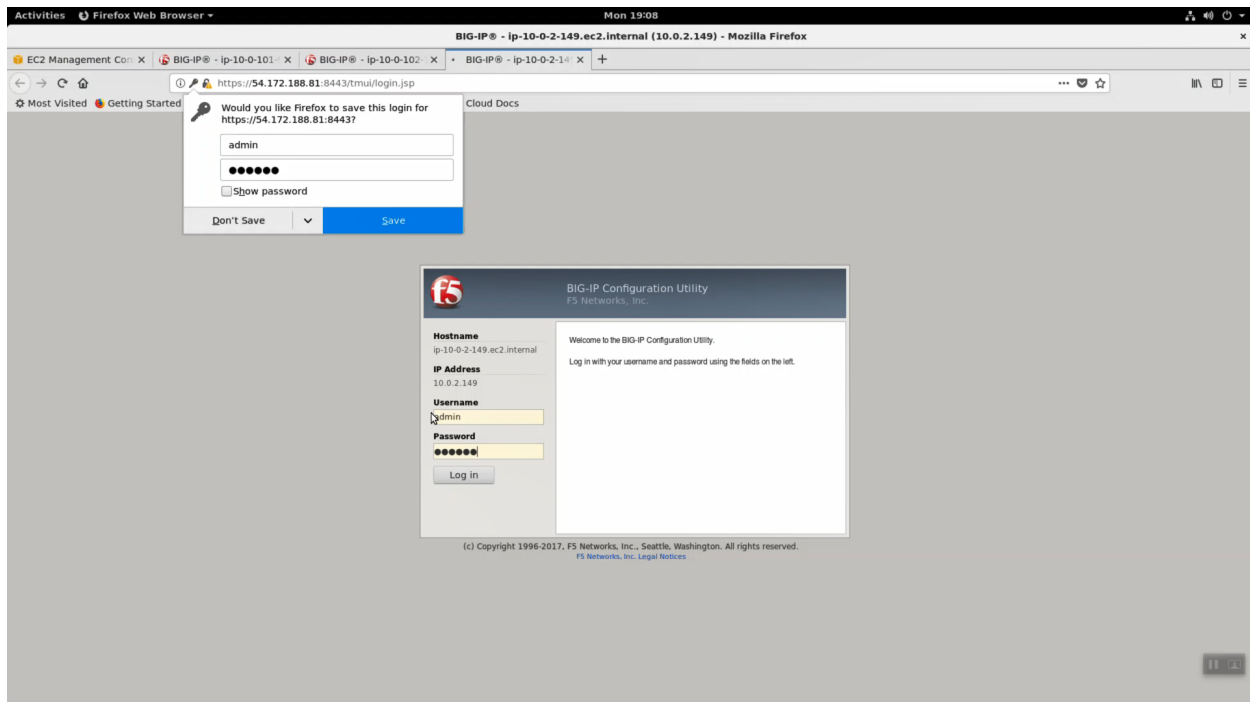
Open a new tab in the Firefox browser. HTTPS to the MGMT URL of BIG-IP Autoscale Instance. Don't miss management port is :8443!

lab-info

Attention: This lab makes use of insecure self-signed certificates. Bypass the warnings by clicking on "Confirm Security Exception".



Login with Username: admin Password: value of shortUrl.



Main => System => Resource Provision. Note an F5 WAF is provisioned for both LTM and ASM.

Current Resource Allocation

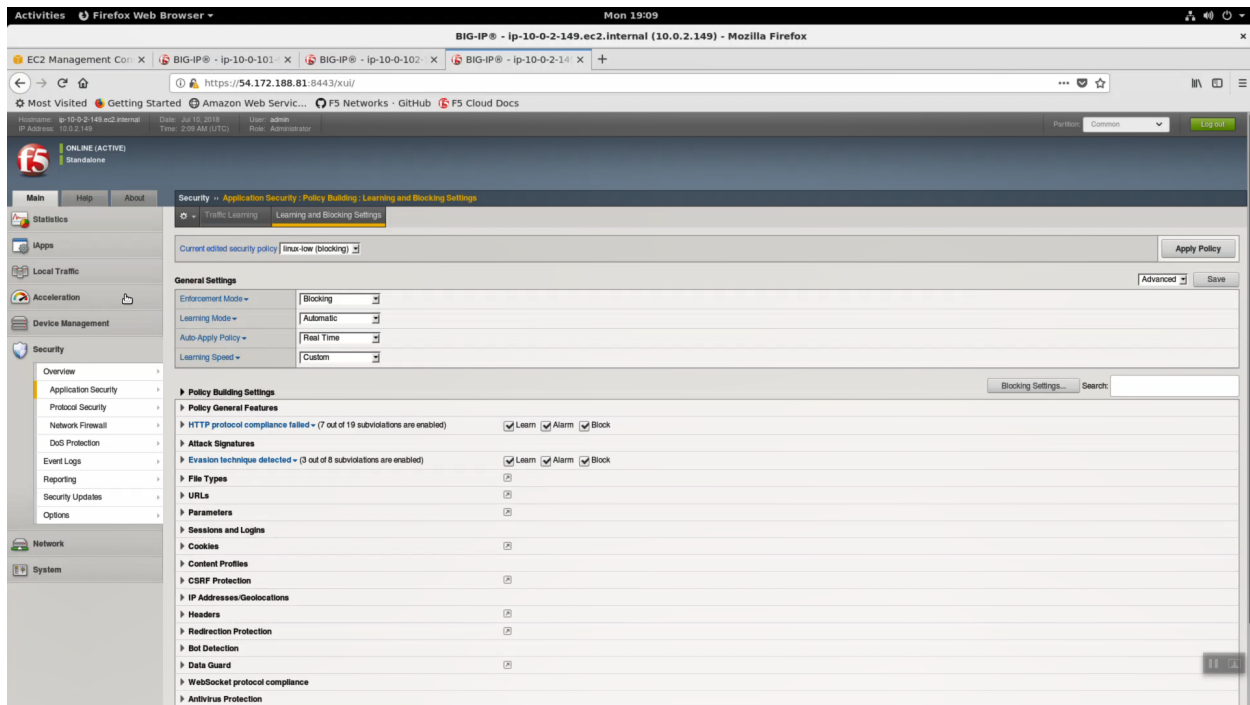
Module	Provisioning	License Status	Required Disk (GB)	Required Memory (MB)
Management (MGMT)	Small	N/A	0	1564
Carrier Grade NAT (CGNAT)	Disabled	Unlicensed	0	0
Local Traffic (LTM)	Normal	Licensed	0	1856
Application Security (ASM)	Normal	Licensed	20	1492
Fraud Protection Service (FPS)	None	N/A	12	544
Global Traffic (DNS)	None	Unlicensed	0	148
Link Controller (LC)	None	Unlicensed	0	148
Access Policy (APM)	None	Licensed	12	494
Application Visibility and Reporting (AVR)	None	Licensed	16	576
Policy Enforcement (PEM)	None	Unlicensed	16	1223
Advanced Firewall (AFM)	None	Unlicensed	16	1058
Application Acceleration Manager (AAM)	None	Unlicensed	32	2050
Secure Web Gateway (SWG)	None	Unlicensed	24	4096
Rules Language Extensions (RulesLX)	None	Licensed	0	748
URLDB Minimal (URLDB)	None	Unlicensed	36	2048

Main => Security => Application Security => Policies List. A starter “linux-low” policy has been deployed.

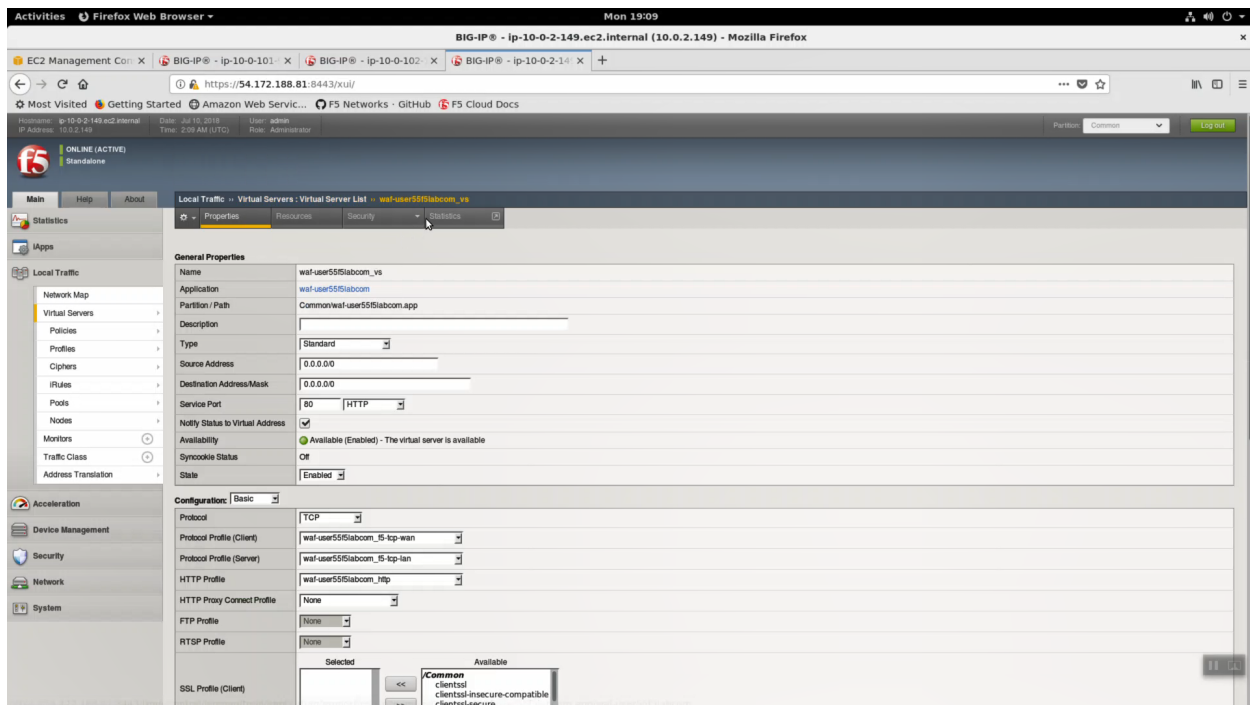
Policy Summary

Field	Value	Description
Policy Name	linux-low	Specifies the name of the policy.
Description	N/A	Specifies an optional description of the policy.
Policy Type	Security	Specifies the type of the policy.
Parent Policy	None	Specifies the parent of the policy.
Version	2018-07-10 01:57:37	Displays additional information about policy version.
Application Language	Unicode (utf-8)	Specifies the language encoding for the web application, which determines how the security policy processes the character sets.
Virtual Server	waf-user55labcom_vs	Displays the name of the protected virtual server, or virtual servers, which have assigned to them a security policy with Application Security enabled.
Enforcement Mode	Blocking	Specifies how the system processes a request that triggers a security policy violation.

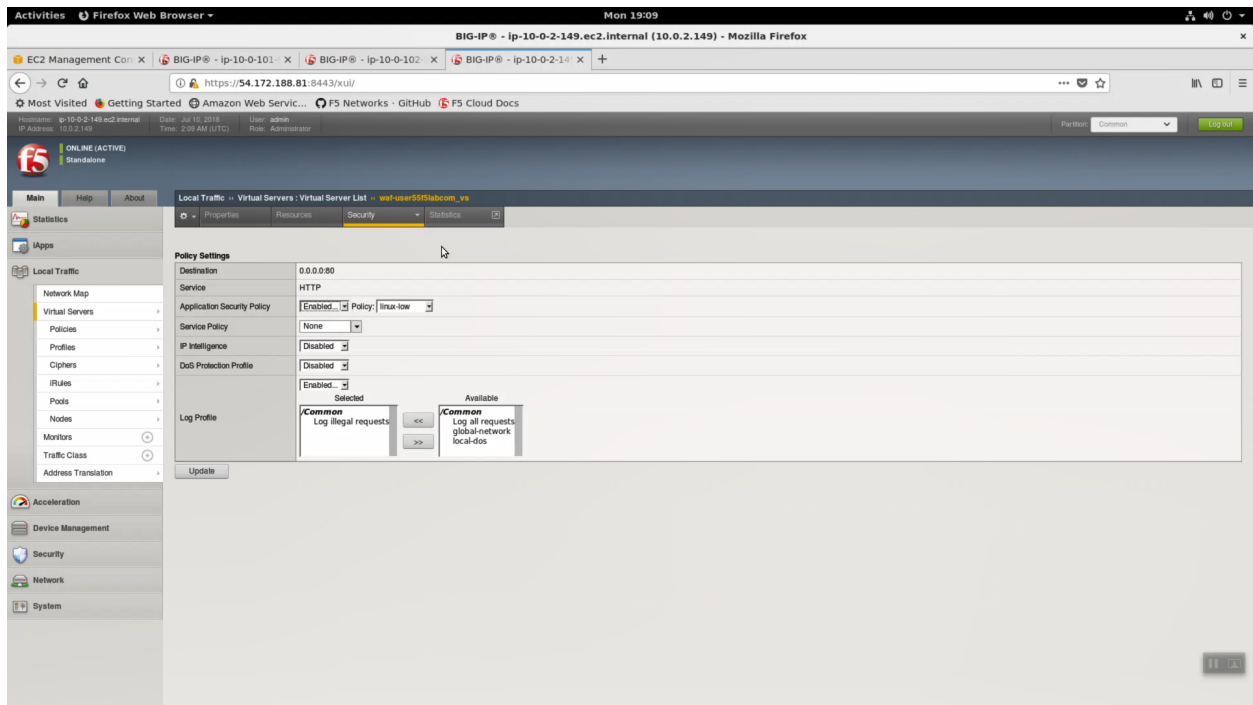
Click on “Learning and Blocking” settings to see exactly what a “linux-low” policy consists of. This starter policy is often times imported in to Big-IQ for central management.



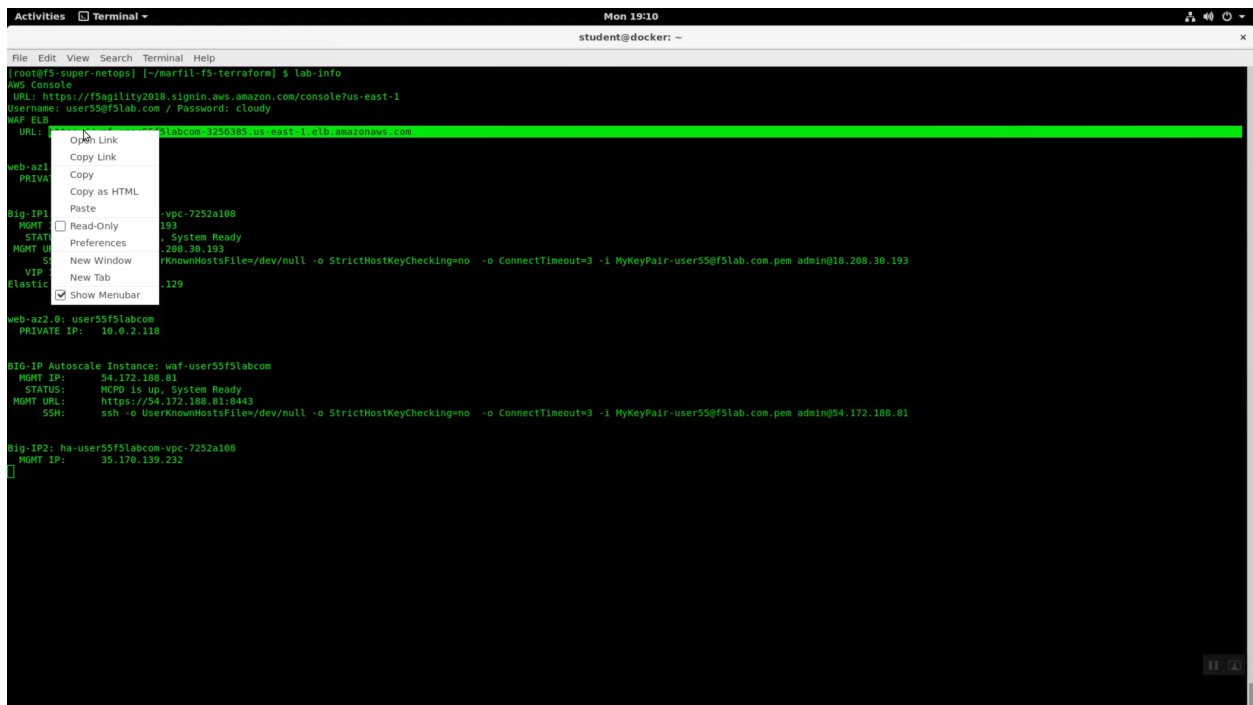
Local Traffic => Virtual Server => Properties. A virtual server with a “catch-all” listener of 0.0.0.0/0 has been deployed.

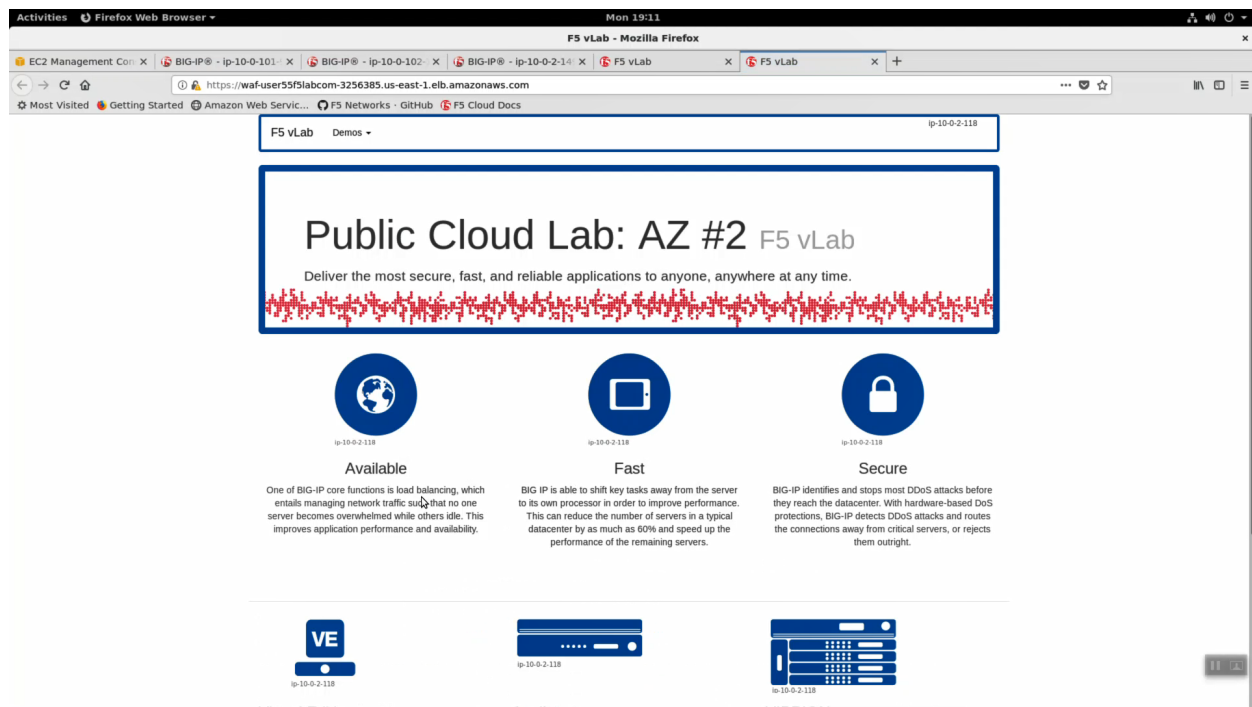


The “linux-low” security policy is attached to this virtual server.

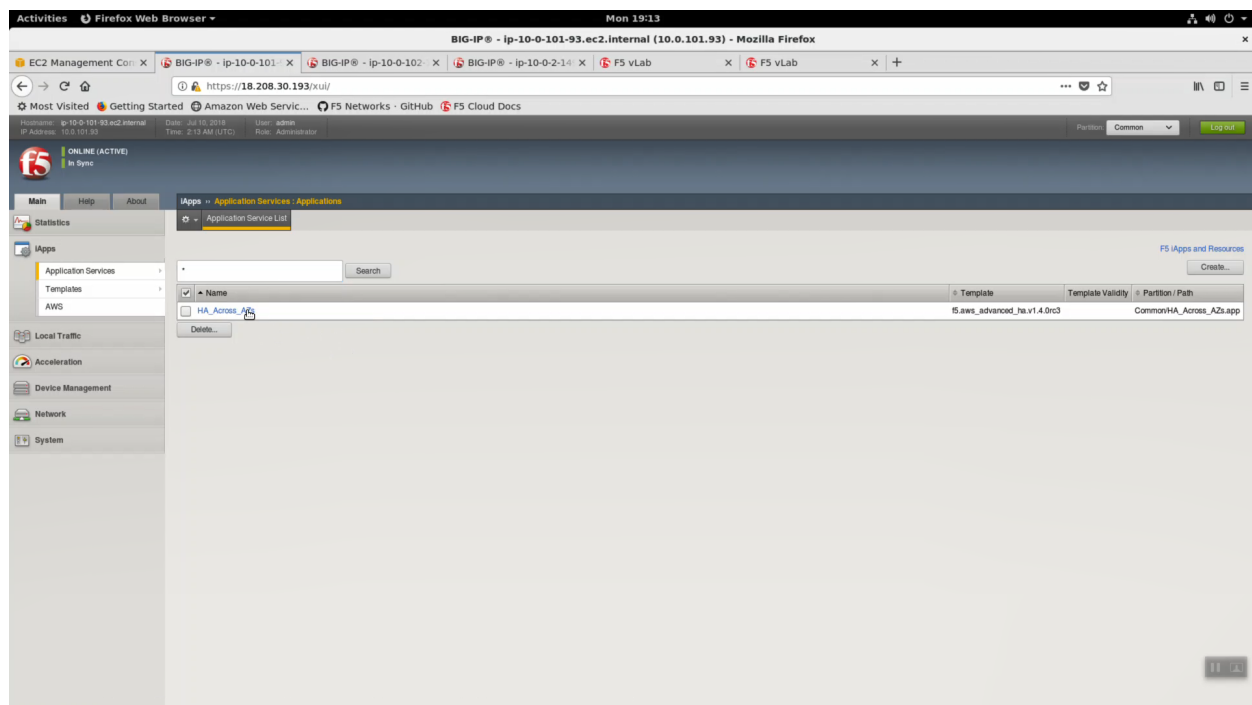


From the Super-NetOps terminal run “lab-info” and copy the value for WAF ELB -> URL. Open a new browser tab and HTTPS to the WAF ELB URL. Your sample application is protected behind an F5 WAF.





Login to either Big-IP1 or Big-IP2. Main => iApps => Application Services. The Cross-AZ HA Big-IP has been deployed with the F5 AWS HA iApp.



2.4 Extending and Securing your Cloud

This lab will use the Lab Environment created previously to explore other capabilities including

- Service Discovery
- Failover Across Availability Zones

We can now start configuring the Big-IPs to responsibly fulfill our part of the shared responsibility security model: <https://aws.amazon.com/compliance/shared-responsibility-model/>

2.4.1 Deploy the Service Discovery iApp on a BigIP Cluster across two Availability Zones

From the Super-NetOps terminal, run the handy lab-info utility. Copy the Big-IP1 MGMT IP.

```
lab-info
```

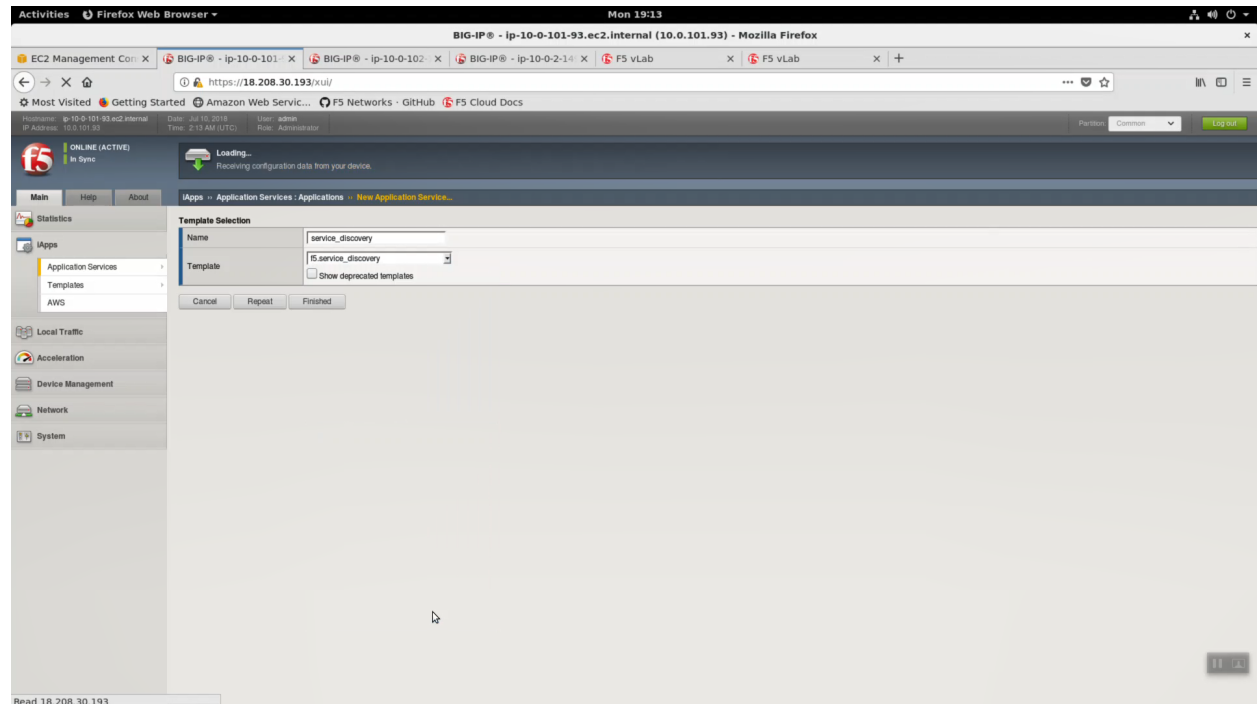
The Service Discovery iApp will automatically discover and populate nodes in the cloud based on tags. Open a new browser tab and HTTPS to the MGMT IP. Login to the Big-IP Configuration utility (Web UI).

- Username: admin
- Password: value for <shortURI> will be unique to your lab.

Navigate to iApps => Application Services. Create a new iApp deployment:

- Name: service_discovery
- Template: choose f5.service_discovery from the dropdown.

Click [Finished]



Question	value
Name	service_discovery
Template	f5.service.discovery
Pool	
What is the tag key on your cloud provider for the members of this pool?	findme
What is the tag value on your cloud provider for the members of this pool?	web
Do you want to create a new pool or use an existing one?	Create new pool. . .
Application Health	
Create a new health monitor or use an existing one?	http

Finished

Activities Firefox Web Browser Mon 19:14

BIG-IP® - ip-10-0-101-93.ec2.internal (10.0.101.93) - Mozilla Firefox

EC2 Management Console x BIG-IP® - ip-10-0-101- x BIG-IP® - ip-10-0-102- x BIG-IP® - ip-10-0-2-1- x F5 vLab x F5 vLab x +

https://10.208.30.193/xui/

Most Visited Getting Started Amazon Web Servic... F5 Networks - GitHub F5 Cloud Docs

Network System

IMPORTANT

This iApp uses the tag key and tag values you configure on your cloud provider to discover pool members. Make sure the tags and IP addresses you use are unique. You should not tag multiple nodes with the same key/tag combination if those nodes use the same IP address. See your cloud provider documentation for specific information on tagging.

Tagging your cloud resources

You have the following options for tagging nodes on your cloud provider.

- * Tag a VM resource: If you tag a VM resource, the BIG-IP VE will discover the primary public or private IP address for the primary NIC configured for the tagged VM.
- * Tag a NIC resource: If you tag a NIC resource, the BIG-IP VE will discover the primary public or private IP addresses for the tagged NIC. Use this option if you want to use the secondary NIC of a VM in the pool.
- * (Azure only) Tag a Virtual Machine Scale Set resource: If you tag a Scale Set resource, the BIG-IP VE will discover the primary private IP address for the primary NIC configured for each Scale Set instance. Note you must select Private IP addresses if you are tagging a Scale Set.

The iApp first looks for NIC resources with the tags you specify. If it finds NICs with the proper tags, it does not look for VM resources. If it does not find NIC resources, it looks for VM resources with the proper tags. If you are using Microsoft Azure only: in either case, it then looks for Scale Set resources with the proper tags.

Template Options

Which configuration mode do you want to use? Basic - Use F5's recommended settings

Do you want to see inline help? No, do not show inline help

Cloud Provider

In which cloud provider is this BIG-IP deployed? AWS EC2

Which region do you want to search? Default

Do you want to assume a role? No

Pool

What is the tag key on your cloud provider for the members of this pool? findme

What is the tag value on your cloud provider for the members of this pool? web

Do you want to search for public or private IP addresses? Private IP addresses

How often do you want to query the cloud provider for updates (in seconds)? 60

Do you want to create a new pool or use an existing one? Create a new pool...

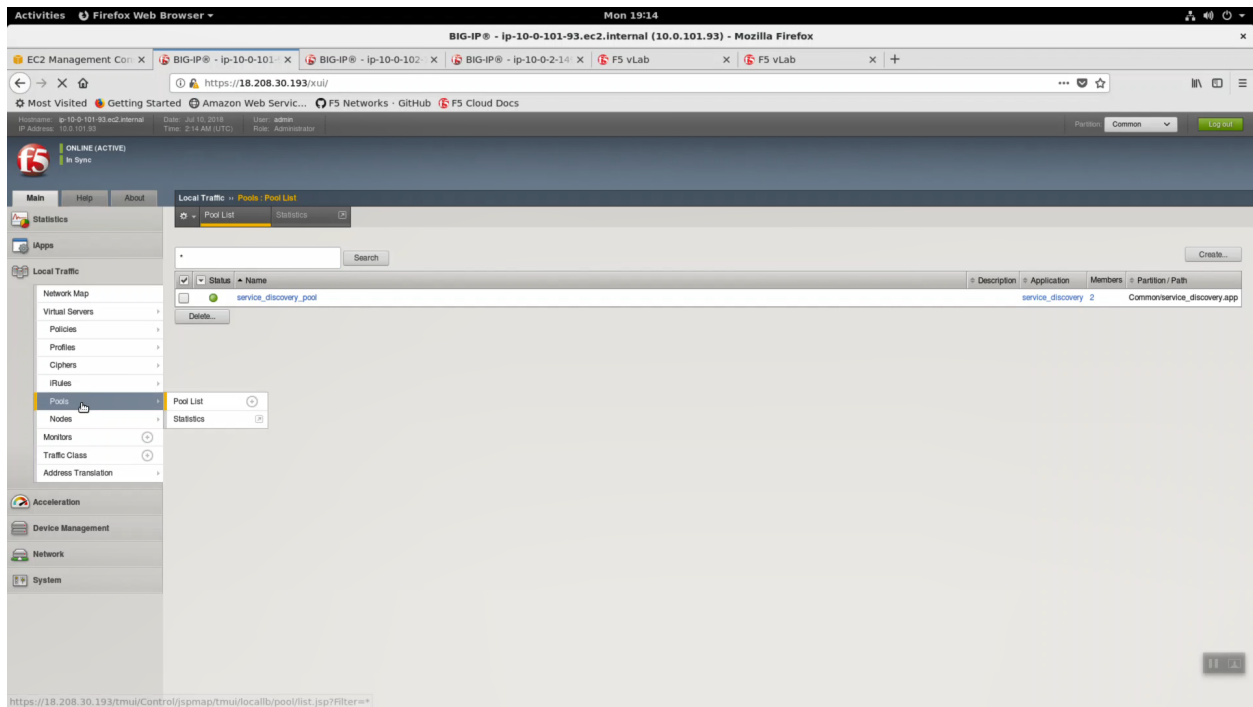
Which port should be assigned to the pool members? 80

Connection limit (optional) 0

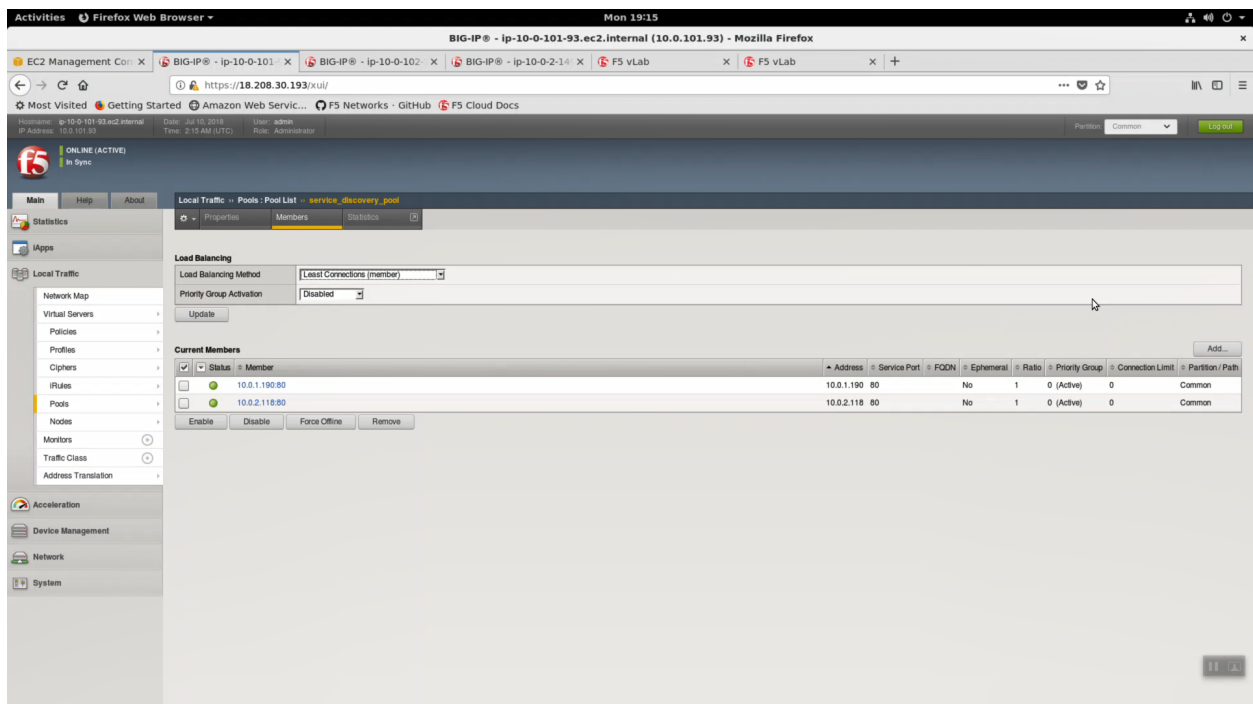
Application Health

Create a new health monitor or use an existing one? http

Cancel Repeat Finished



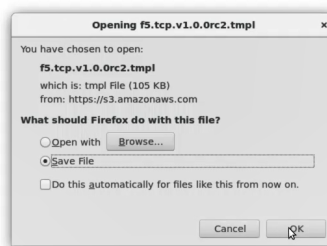
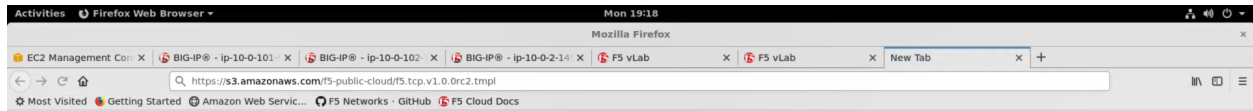
Local Traffic => Pools => track the newly created service_discovery_pool. Within 60 seconds it should light up green. The service_discovery iApp has discovered and auto-populated the service_discovery_pool with two web servers.



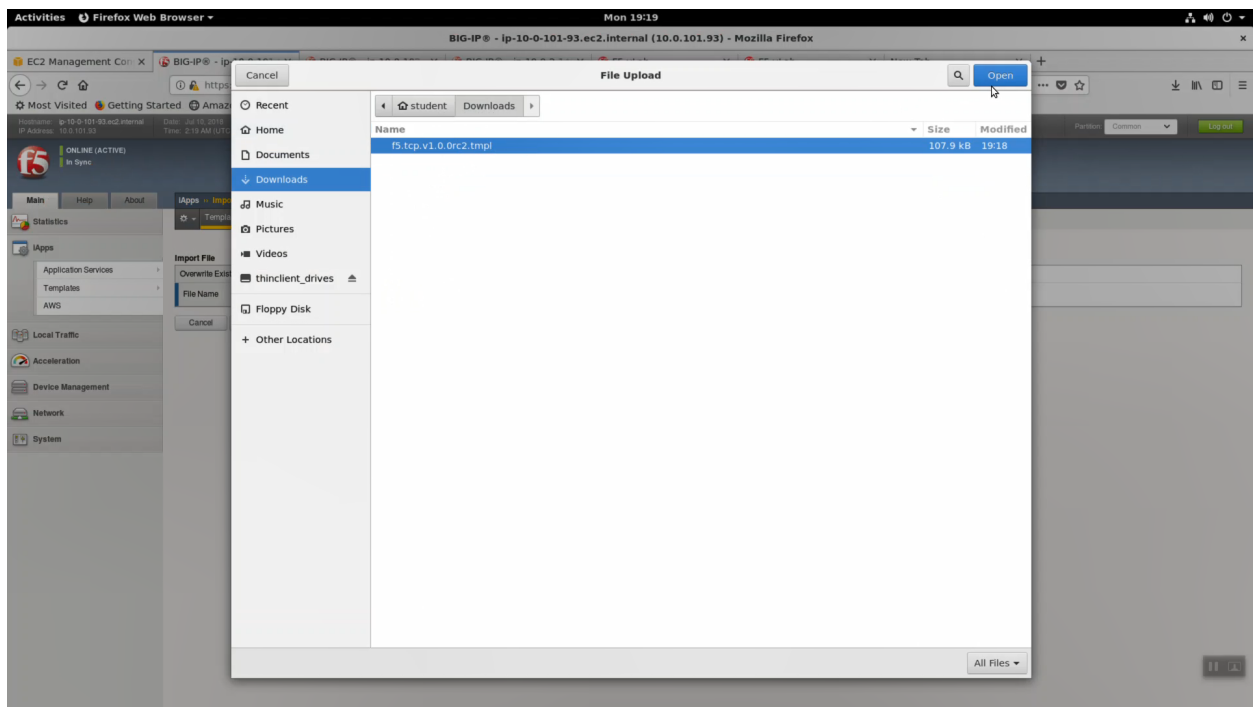
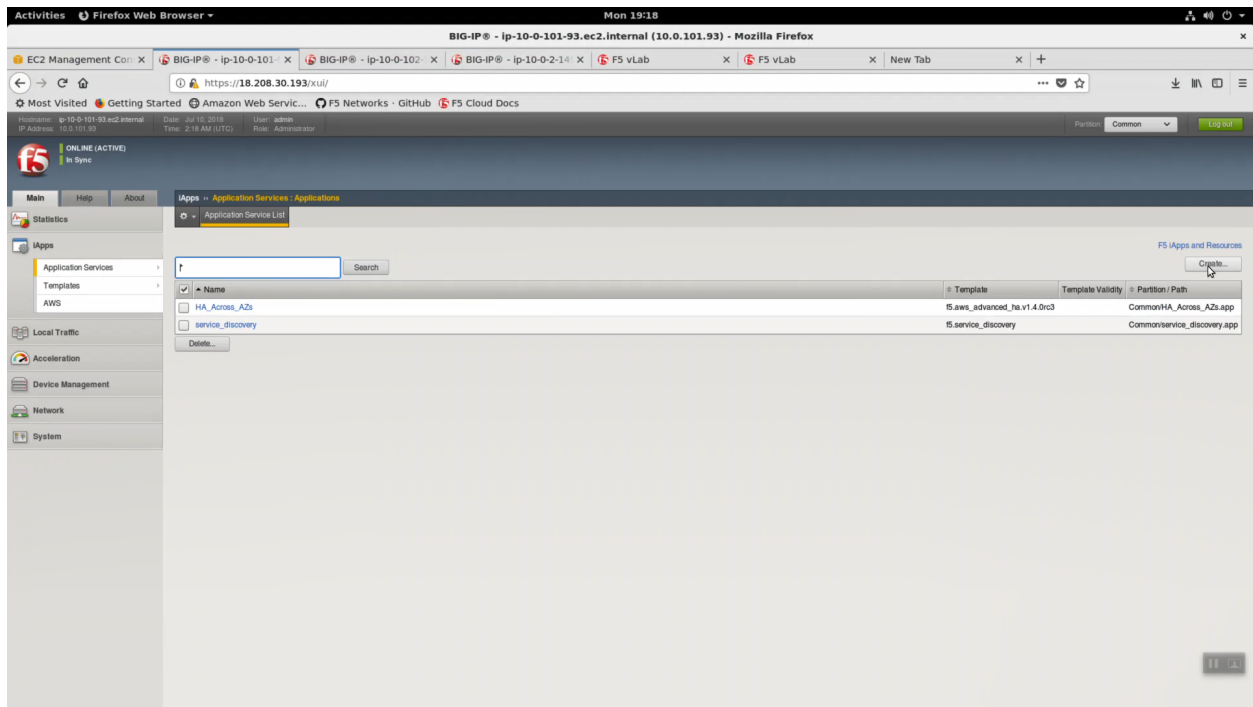
2.4.2 Deploy an AWS High-Availability-aware virtual server across two Availability Zones

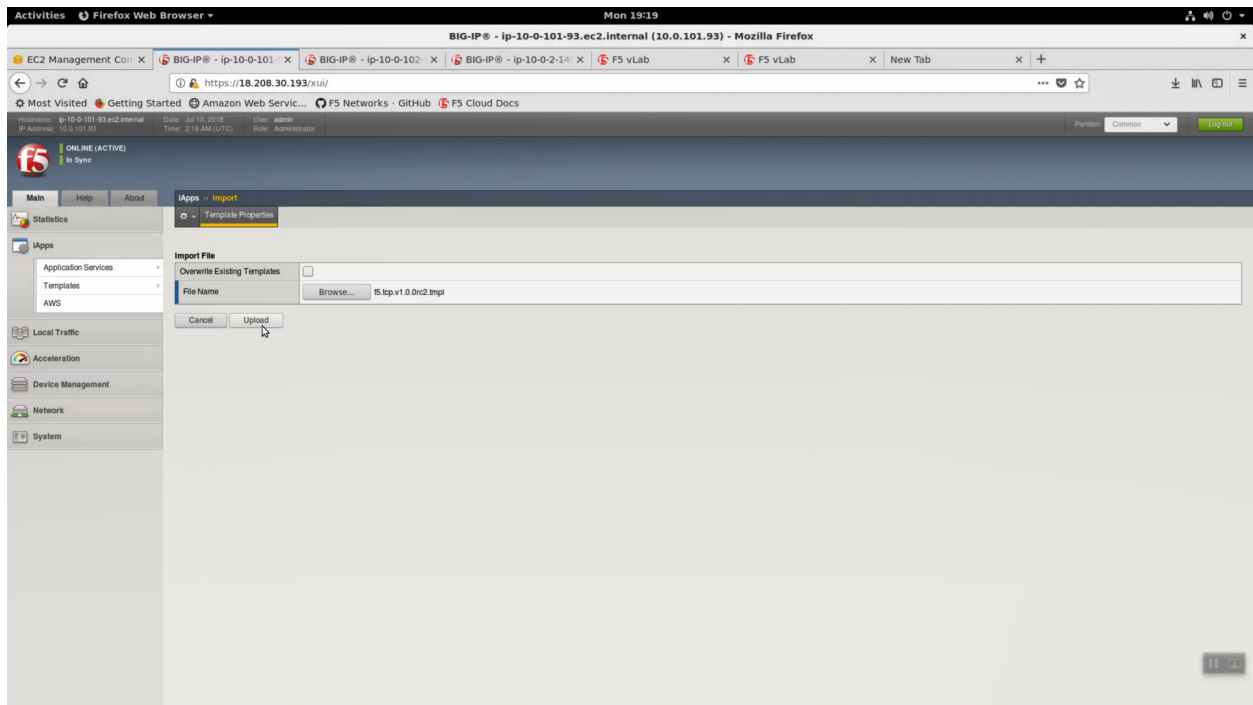
Login to the active Big-IP1 Configuration utility (Web UI). The “HA_Across_AZs” iApp will already be deployed in the Common partition.

Download the latest tcp iApp template from <https://s3.amazonaws.com/f5-public-cloud/f5.tcp.v1.0.0rc2.tmpl>.



iApps -> Templates -> import. Import f5.tcp.v1.0.0rc2.tmpl to the primary BigIP. The secondary BigIP should pick up the configuration change automatically.

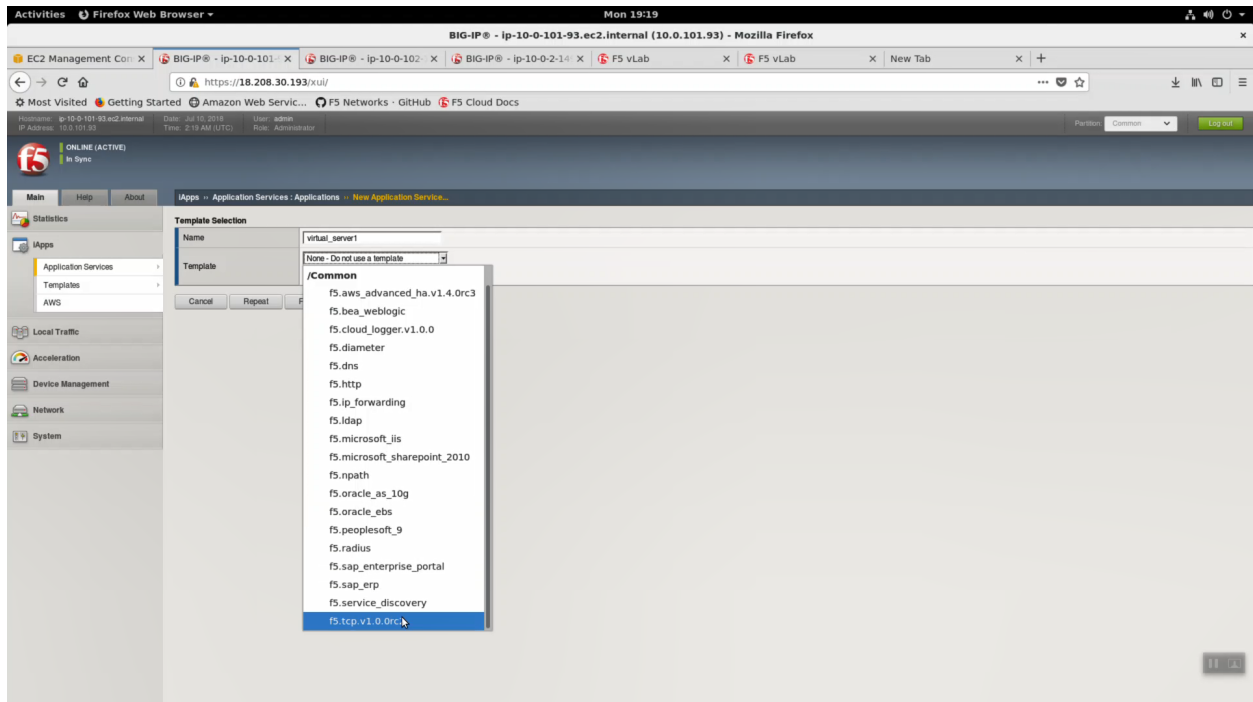




Deploy an iApp using the f5.tcp.v1.0.0rc2.tmpl template.

iApps => Application Services => Select f5.tcp.v1.0.0rc2 template from the dropdown. Name: virtual_server_1.

Configure iApp: Select "Advanced" from "Template Selection".



Traffic Group: UNCHECK "Inherit traffic group from current partition / path"

Question	value
Name:	virtual_server_1
Inherit traffic group from current partition / path	uncheck
High Availability. What IP address do you want to use for the virtual server?	VIP IP of Big-IP1
What is the associated service port?	HTTP (80)
What IP address do you wish to use for the TCP virtual server in the other data center or availability zone?	VIP IP of Big-IP2
Do you want to create a new pool or use an existing one?	ser-vice_discovery_pool

From the Super-NetOps terminal. Invoke `terraform output` and copy the value for Big-IP1 => VIP IP. Use this value in the iApp as explained in the chart above.

```

[student@fs-super-netops] [~/marfil-f5-terraform] $ lab-info
AWS Console
URL: https://f5agility2018.signin.aws.amazon.com/console?us-east-1
Username: user55@f5lab.com / Password: cloudy
WAF ELB
URL: https://waf-user55f5labcom-3256385.us-east-1.elb.amazonaws.com

web-az1.0: user55f5labcom
PRIVATE IP: 10.0.1.190

Big-IP1: ha-user55f5labcom-vpc-7252a108
MGMT IP: 18.208.30.193
STATUS: MCPD is up, System Ready
MGMT URL: https://18.208.30.193
SSH: ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no -o ConnectTimeout=3 -i MyKeyPair-user55@f5lab.com.pem admin@18.208.30.193
VIP IP: 10.0.2.193
Elastic IP: 18.209.139.150 = 0x96

web-az2.0: user55f5labcom
PRIVATE IP: 10.0.2.110

BIG-IP Autoscale Instance
MGMT IP: 54.172.188.81
STATUS: MCPD is up, System Ready
MGMT URL: https://54.172.188.81
SSH: ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no -o ConnectTimeout=3 -i MyKeyPair-user55@f5lab.com.pem admin@54.172.188.81

Big-IP2: ha-user55f5labcom-vpc-7252a108
MGMT IP: 35.170.139.232
STATUS: MCPD is up, System Ready
MGMT URL: https://35.170.139.232
SSH: ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no -o ConnectTimeout=3 -i MyKeyPair-user55@f5lab.com.pem admin@35.170.139.232
VIP IP: 10.0.2.58

[student@fs-super-netops] [~/marfil-f5-terraform] $

```

From the Super-NetOps terminal. Invoke `terraform output` and copy the value for Big-IP2 => VIP IP. Use this value in the iApp as explained in the chart above.

Activities Terminal Mon 19:20 student@docker: ~

```
File Edit View Search Terminal Help
[root@fs-super-netops] [~/mar11-fs-terraform] $ lab-info
AWS Console
URL: https://fsagility2018.signin.aws.amazon.com/console?us-east-1
Username: user55@fs5lab.com / Password: clouidy
WAF ELB
URL: https://waf-user55@fs5labcom-3256385.us-east-1.elb.amazonaws.com

web-az1.0: user55@fs5labcom
PRIVATE IP: 10.0.1.190

Big-IP1: ha-user55@fs5labcom-vpc-7252a108
MGMT IP: 10.208.30.193
STATUS: MCPD is up, System Ready
MGMT URL: https://10.208.30.193
SSH: ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no -o ConnectTimeout=3 -i MyKeyPair-user55@fs5lab.com.pem admin@10.208.30.193
VIP IP: 10.0.1.150
Elastic IP: 10.209.138.129

web-az2.0: user55@fs5labcom
PRIVATE IP: 10.0.2.118

BIG-IP Autoscale Instance: waf-user55@fs5labcom
MGMT IP: 54.172.188.81
STATUS: MCPD is up, System Ready
MGMT URL: https://54.172.188.81:8443
SSH: ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no -o ConnectTimeout=3 -i MyKeyPair-user55@fs5lab.com.pem admin@54.172.188.81

Big-IP2: ha-user55@fs5labcom-vpc-7252a108
MGMT IP: 35.170.139.232
STATUS: MCPD is up, System Ready
MGMT URL: https://35.170.139.232
SSH: ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no -o ConnectTimeout=3 -i MyKeyPair-user55@fs5lab.com.pem admin@35.170.139.232
VIP IP: 10.0.2.150

[Copy]
[Copy as HTML]
[Paste]
[Read-Only]
[Preferences]
[New Window]
[New Tab]
[Show Menubar]
```

Activities Firefox Web Browser Mon 19:21 BIG-IP® - ip-10-0-101-93.ec2.internal (10.0.101.93) - Mozilla Firefox

EC2 Management Console x BIG-IP® - ip-10-0-101-93.ec2.internal x BIG-IP® - ip-10-0-102-14.ec2.internal x BIG-IP® - ip-10-0-2-14.ec2.internal x FS vLab x FS vLab x New Tab x

Most Visited Getting Started Amazon Web Services FS Networks GitHub FS Cloud Docs

https://10.208.30.193/xul/

How have you configured routing on your servers?

It is important to ensure that responses to client requests made using the BIG-IP virtual server address are returned through the BIG-IP system. If the client receives a response directly from the TCP server, the connection is dropped. The way the BIG-IP system handles this depends on your network topology.

For environments in which the virtual server IP address is on a subnet different from the TCP servers, select BIG-IP virtual server IP and the TCP servers are on different subnets.

For environments in which the virtual server IP address provided is on the same subnet as the TCP servers in the associated pool, select BIG-IP virtual server IP and the TCP servers are on the same subnet. This enables Secure Network Address Translation (SNAT Auto Map). This configuration results in the BIG-IP system replacing the client IP address of an incoming connection with its local floating self IP address, ensuring the server response returns through the BIG-IP system.

Servers do not have a route to clients through the BIG-IP

For environments in which the virtual server IP is on a subnet different from the TCP servers, information regarding the IP setting of the TCP servers is required to ensure the correct BIG-IP system configuration.

If the TCP servers use the BIG-IP system as their default gateway, select TCP servers have a route for clients through the BIG-IP. In this scenario, no configuration is needed to support your environment to ensure correct server response handling.

If the TCP servers do not have a route through the BIG-IP system, select TCP servers do not have a route for clients through the BIG-IP. This enables Secure Network Address Translation (SNAT Auto Map). This configuration results in the BIG-IP system replacing the client IP address of an incoming connection with its local floating self IP address ensuring the server response returns through the BIG-IP system.

How many connections per server do you expect?

Fewer than 64,000 concurrent connections per server

For environments with fewer than 64,000 concurrent connections per server, the BIG-IP system enables SNAT Auto Map, which uses a unique IP:port combination for each client request it sends to the TCP server. For environments with more than 64,000 concurrent connections per TCP server, the BIG-IP system enables a SNAT pool, and additional IP addresses are reserved to ensure the system has enough unique combinations. If the system exhausts all combinations, new client connections are refused until one is available.

High Availability

What IP address do you want to use for the virtual server?

10.0.1.150

This IP address, along with the port you specify, becomes the BIG-IP virtual server address and port, which clients use to access the application. The system intercepts requests to this IP:Port and distributes them to the TCP servers. To specify a Route Domain, use the format a.b.c.d.N, where N is the route domain. For information on route domains, see the BIG-IP documentation.

What is the associated service port?

HTTP (80)

Select a service port, or you can enter a custom port if you choose 1 want to specify a custom port

If you are using a network virtual address, what is the IP mask?

If you specified a network address for the virtual server (allowing the virtual server to handle multiple IP addresses), you must enter the full network mask that represents the address range. If you specified a single address for the virtual server, you may leave this field blank.

What IP address do you wish to use for the TCP virtual server in the other data center or availability zone?

10.0.2.58

If using a network virtual address, what is the IP mask in the other data center or availability zone?

Do you want to create a new pool or use an existing one?

service_discovery_pool

A load balancing pool is a logical set of devices, such as TCP servers, grouped together to receive and process traffic. When clients attempt to access the application via the BIG-IP virtual server, the BIG-IP system distributes requests to any of the servers that are members of that pool.

Delivery Optimization and Server Offload

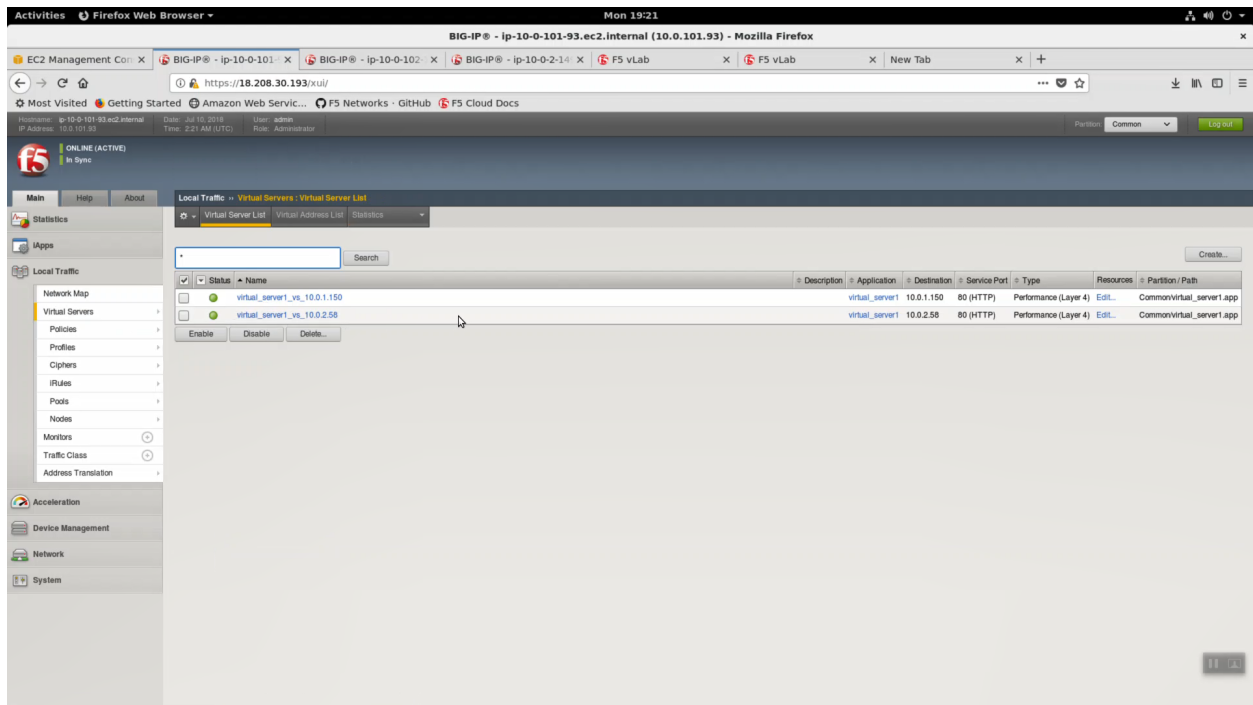
Which FastL4 profile do you want to apply to your virtual server(s)?

Create a new Fast L4 profile (recommended)

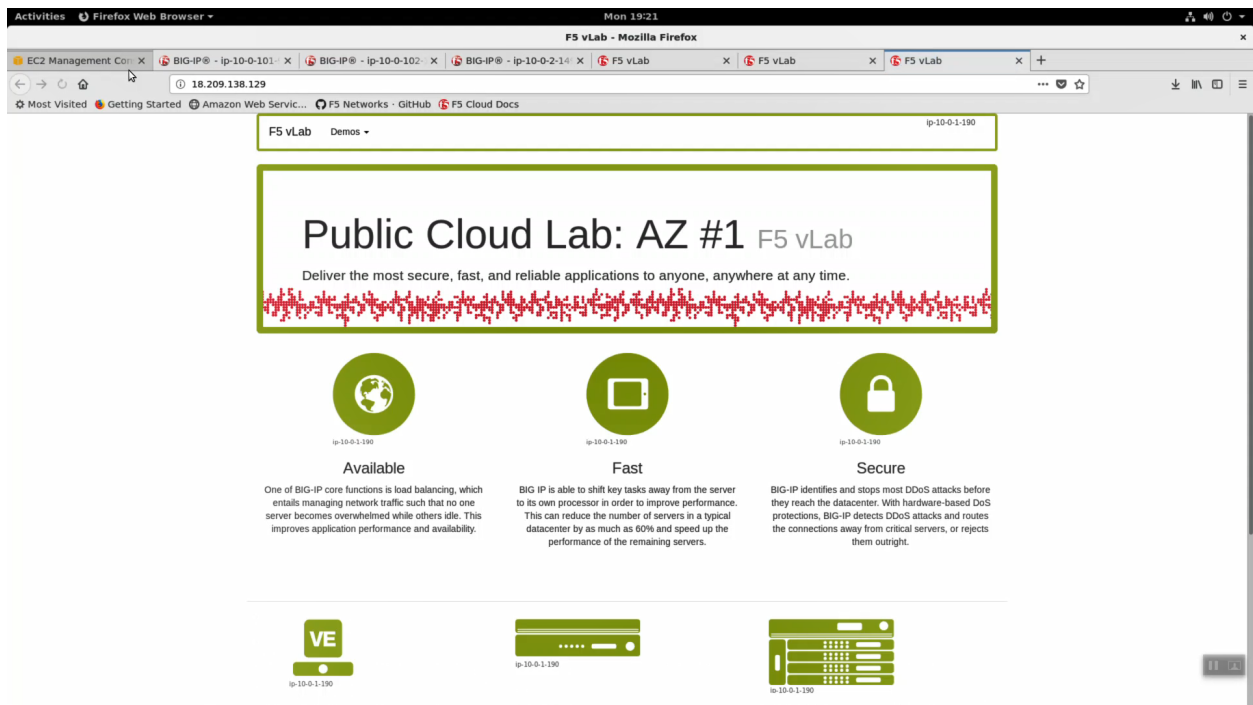
The FastL4 profile helps manage Layer 4 traffic more efficiently. Unless you have manually created a custom FastL4 profile for this configuration, we recommend allowing the iApp to create a new one. Only existing Fast L4 profiles appear in this list.

Cancel Repeat Finished

The iApp will create two virtual servers on *both* Big-IP's. The iApp deployment on Big-IP1 will automatically and immediately sync to Big-IP2.

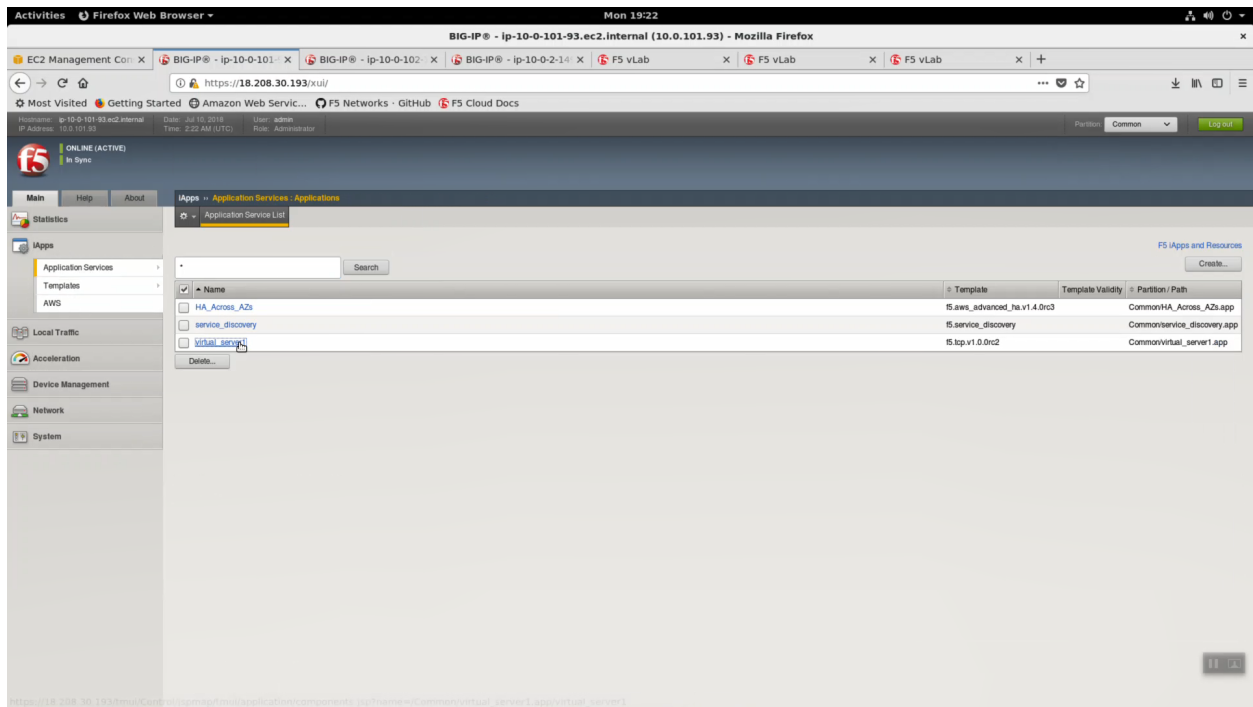


From the Super-NetOps terminal. Invoke `terraform output` and copy the value for the primary Big-IP's Elastic IP. Open a browser tab and HTTP to this Elastic IP.

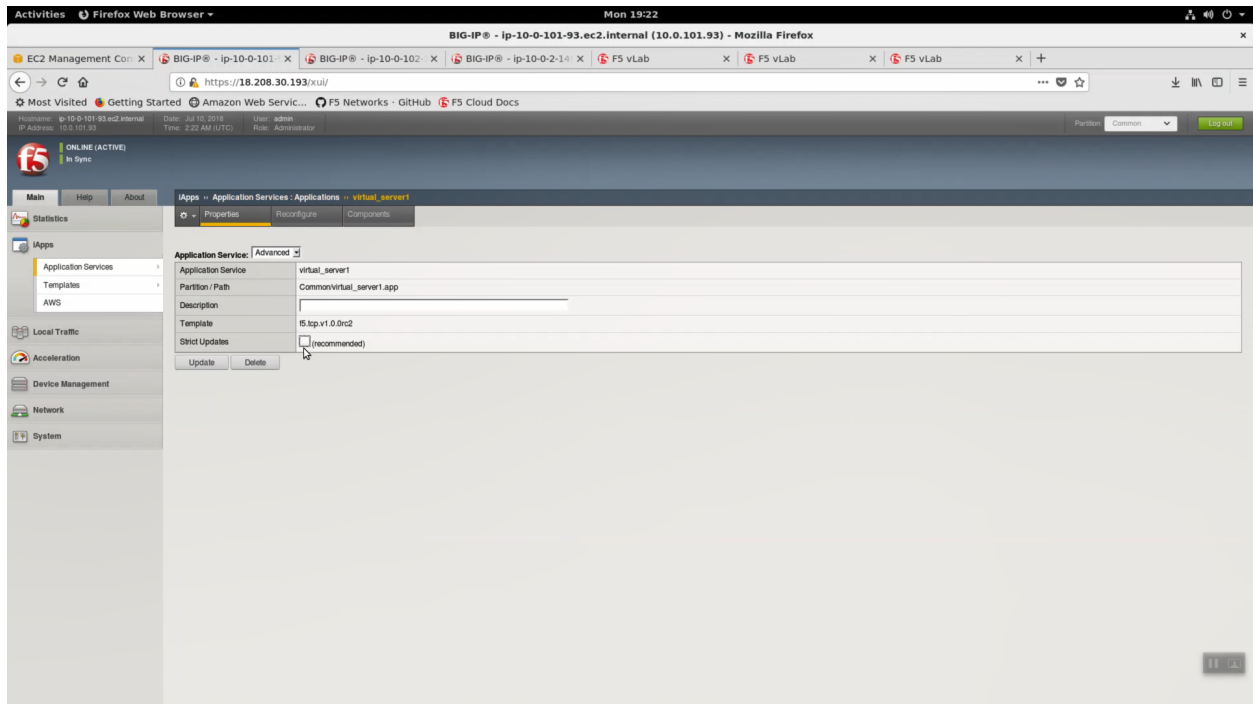


In order to enable request logging and apply a client SSL profile, let's re-configure our TCP / Fast L4 virtual server to a Standard virtual server with an http profile applied.

iApps => Application Services => select the "virtual_server_1" iApp we just deployed.



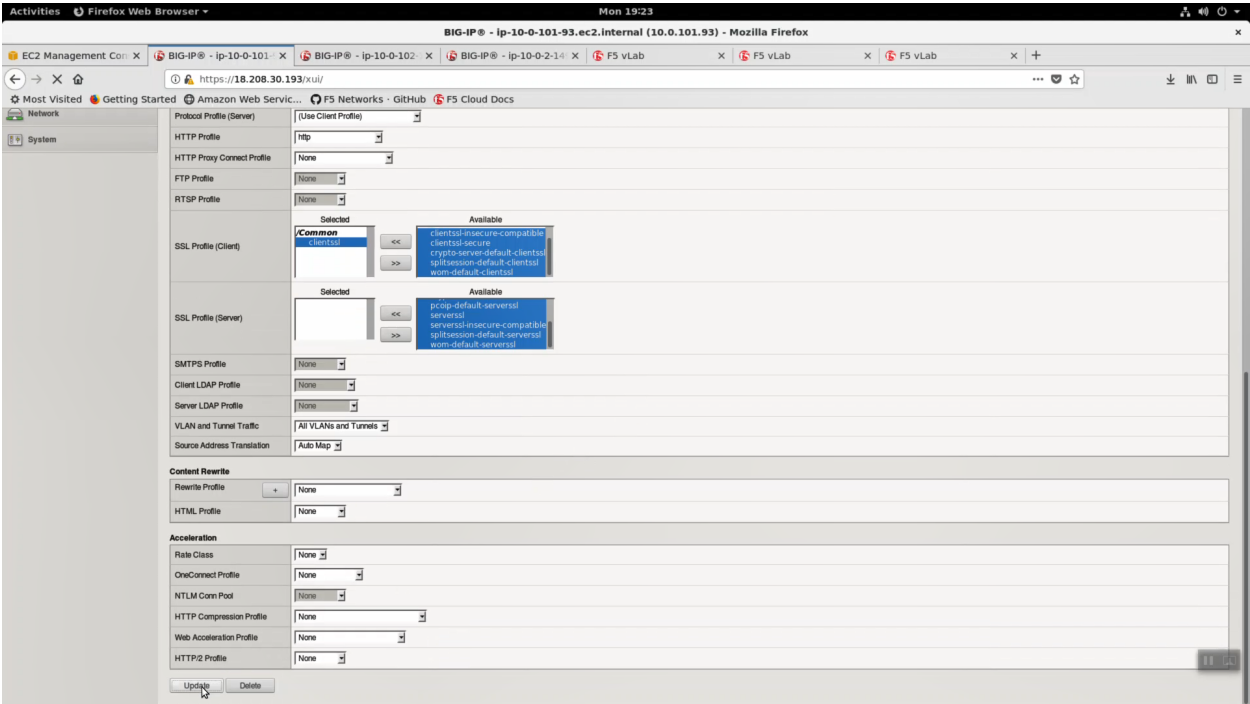
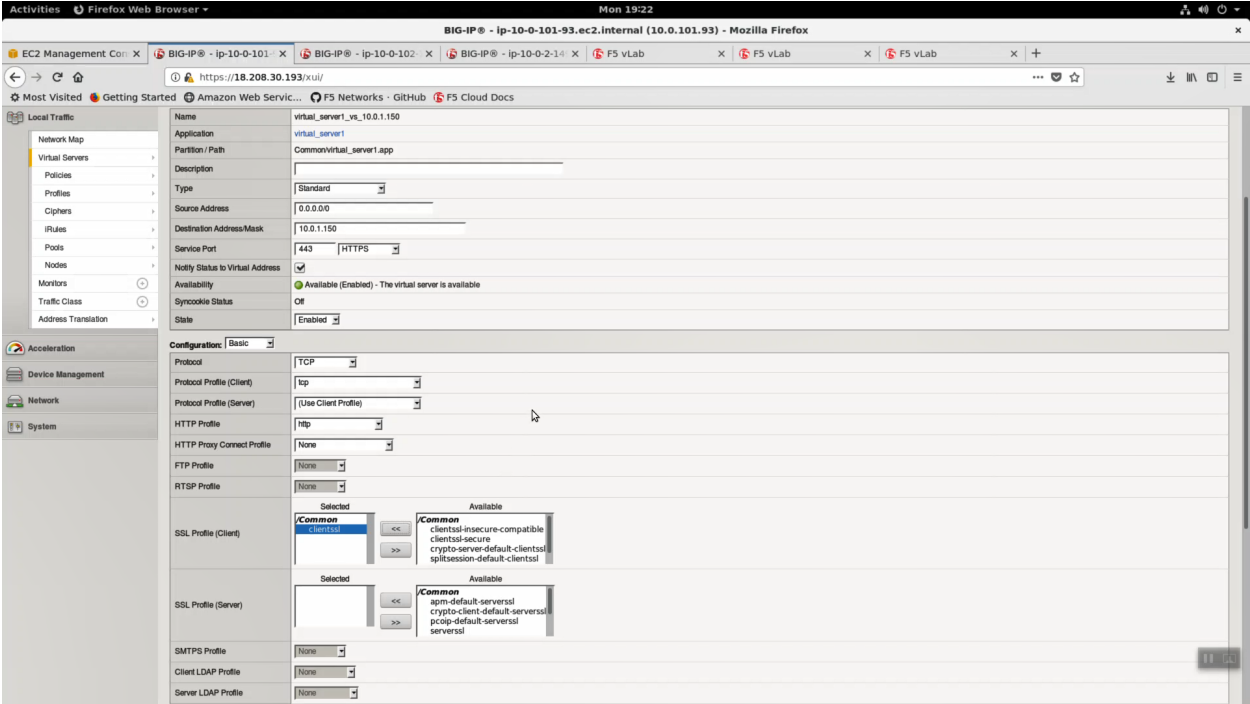
Properties => uncheck/disable "Strict Updates"



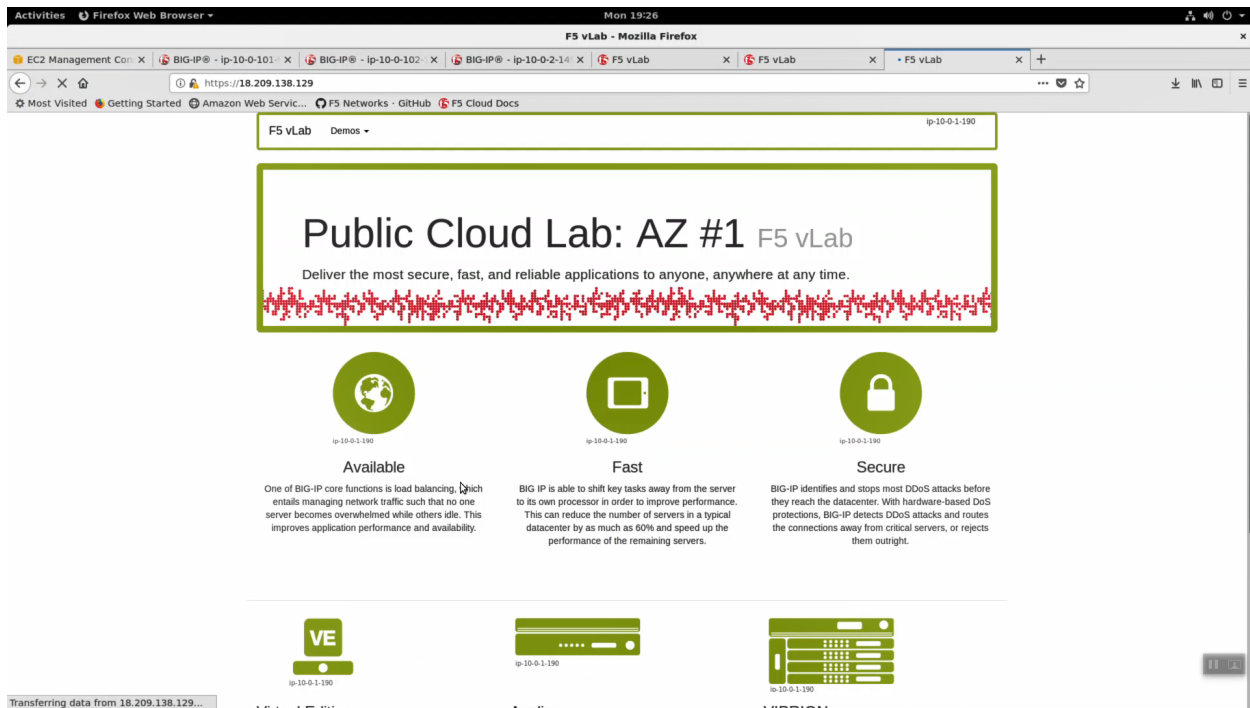
Local Traffic => Virtual Servers => virtual_server1. Change only the values below and leave the rest as they are.

Question	value
Type	Standard
Service Port	443 / HTTPS
HTTP Profile	http
SSL Profile (Client)	clientssl

[Update]



From the Super-NetOps terminal. Invoke `terraform output` and copy the value for the primary Big-IP's Elastic IP. Let's test the http profile and clientssl profile are working. Open a browser tab and HTTPS (different than before, when we accessed our example application via HTTP) to this Elastic IP.

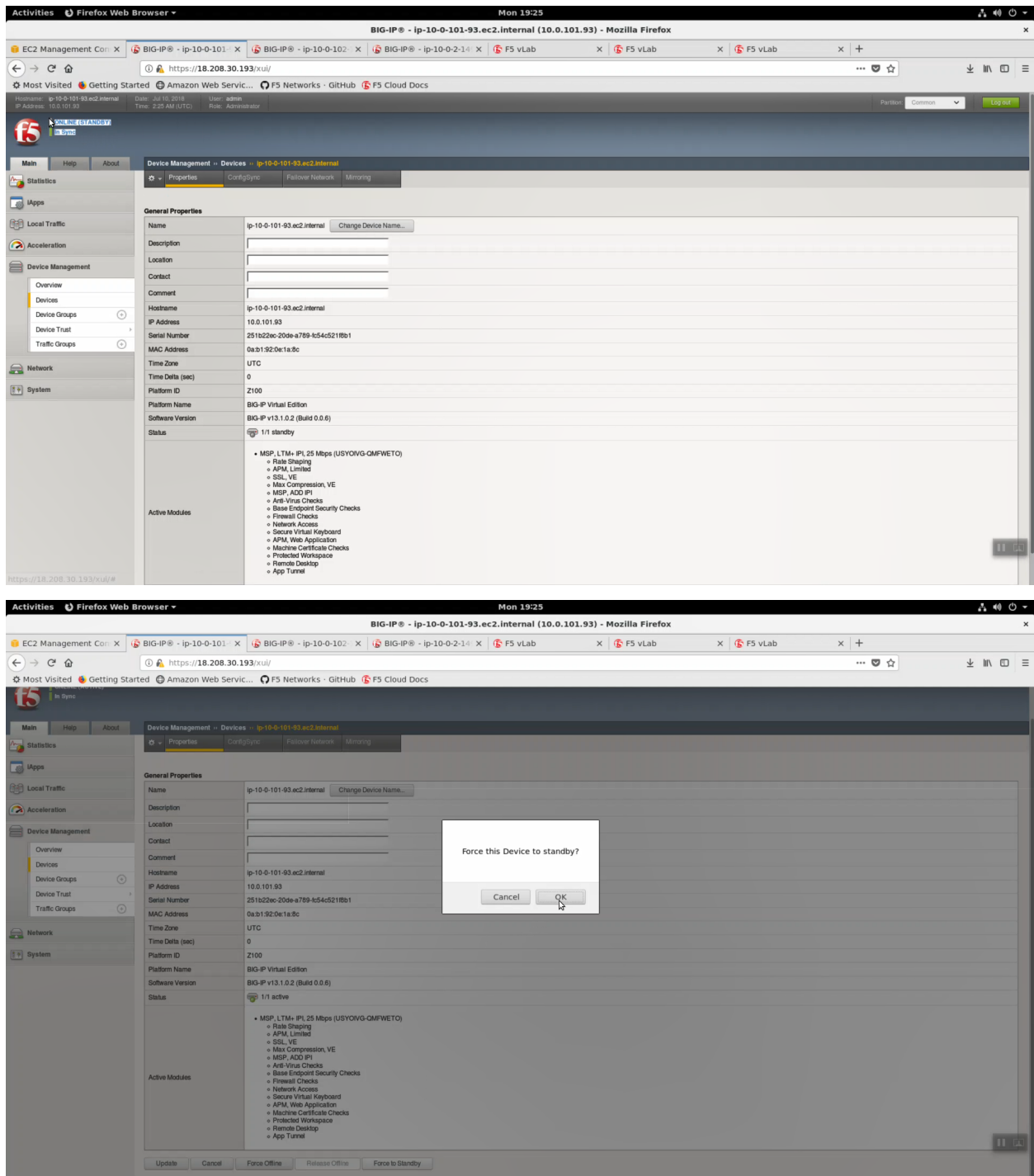


2.4.3 Test Failover

From the Super-NetOps terminal, run the handy lab-info utility. Confirm that “MCPD is up, System Ready” for all three of your instances.

```
lab-info
```

From the HTTPS Configuration Utility (Web UI) of the active Big-IPX device: Device Management => Devices. [Force Offline]. Click [OK] to confirm.



From the Super-NetOps terminal, run the lab-info utility. Notice how the Elastic IP previously associated with Big-IP1 has now “floated over” and is associated with Big-IP2.

lab-info


```
Activities Terminal Mon 19:26 student@docker: -
File Edit View Search Terminal Help
[root@fs-super-netops] [~/marfil-f5-terraform] $ lab-info
AWS Console
URL: https://f5agility2018.signin.aws.amazon.com/console?us-east-1
Username: user55@f5lab.com / Password: cloudy
WAF ELB
URL: https://waf-user55f5labcom-3256385.us-east-1.elb.amazonaws.com

web-az1.0: user55f5labcom
PRIVATE IP: 10.0.1.190

Big-IP1: ha-user55f5labcom-vpc-7252a108
MGMT IP: 10.200.30.193
STATUS: MCPD is up, System Ready
MGMT URL: https://10.200.30.193
SSH: ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no -o ConnectTimeout=3 -i MyKeyPair-user55@f5lab.com.pem admin@10.200.30.193
VIP IP: 10.0.1.190

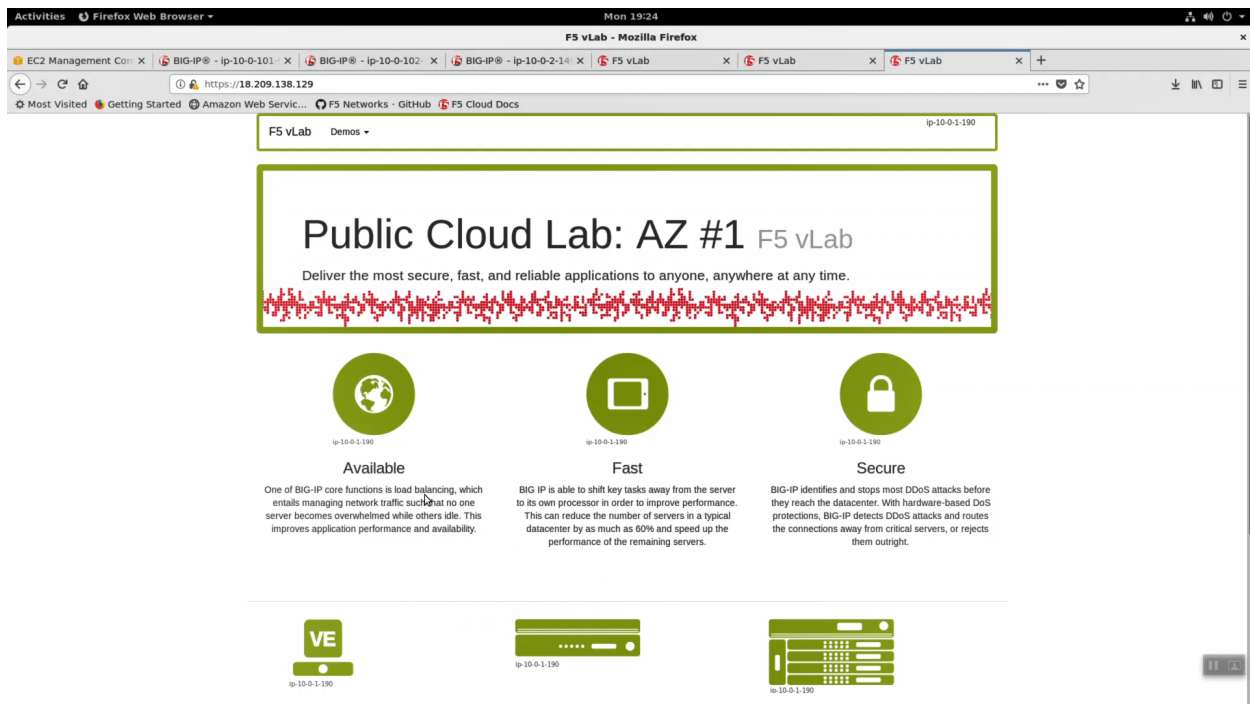
web-az2.0: user55f5labcom
PRIVATE IP: 10.0.2.118

BIG-IP Autoscale Instance: waf-user55f5labcom
MGMT IP: 54.172.188.81
STATUS: MCPD is up, System Ready
MGMT URL: https://54.172.188.81:8443
SSH: ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no -o ConnectTimeout=3 -i MyKeyPair-user55@f5lab.com.pem admin@54.172.188.81

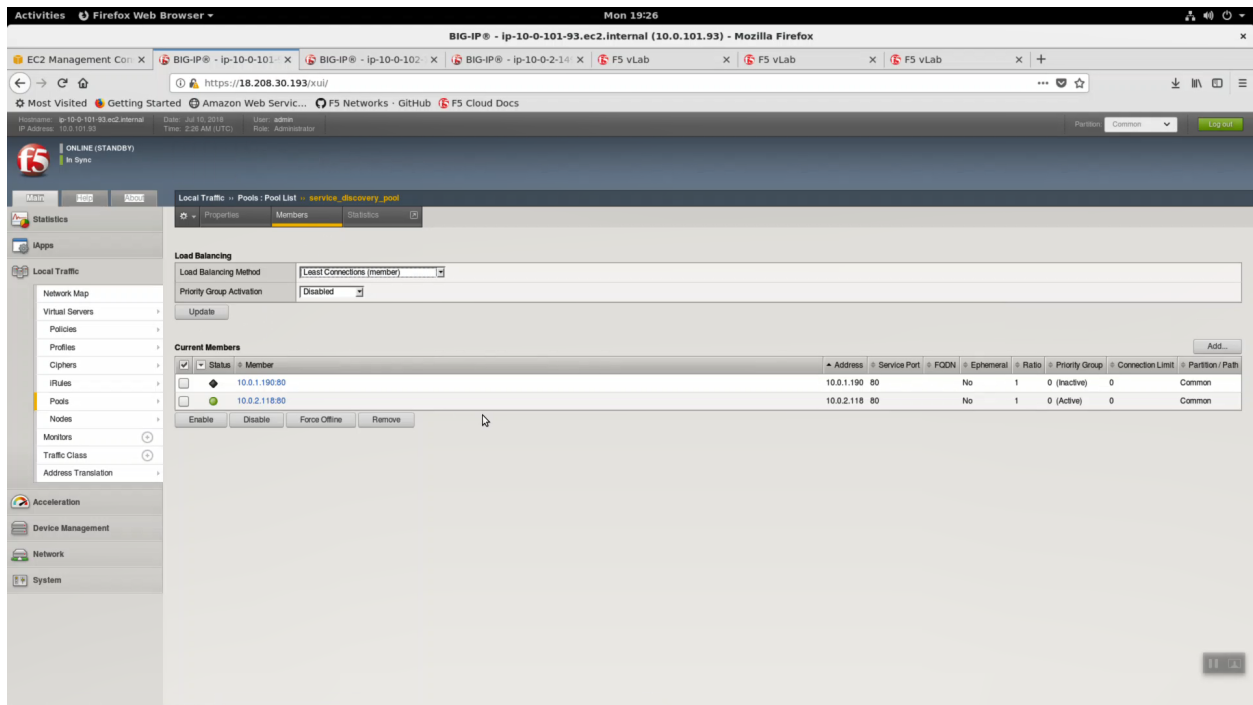
Big-IP2: ha-user55f5labcom-vpc-7252a108
MGMT IP: 35.170.139.232
STATUS: MCPD is up, System Ready
MGMT URL: https://35.170.139.232
SSH: ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no -o ConnectTimeout=3 -i MyKeyPair-user55@f5lab.com.pem admin@35.170.139.232
VIP IP: 10.0.2.50
Elastic IP: 18.209.130.129

[root@fs-super-netops] [~/marfil-f5-terraform] $
```

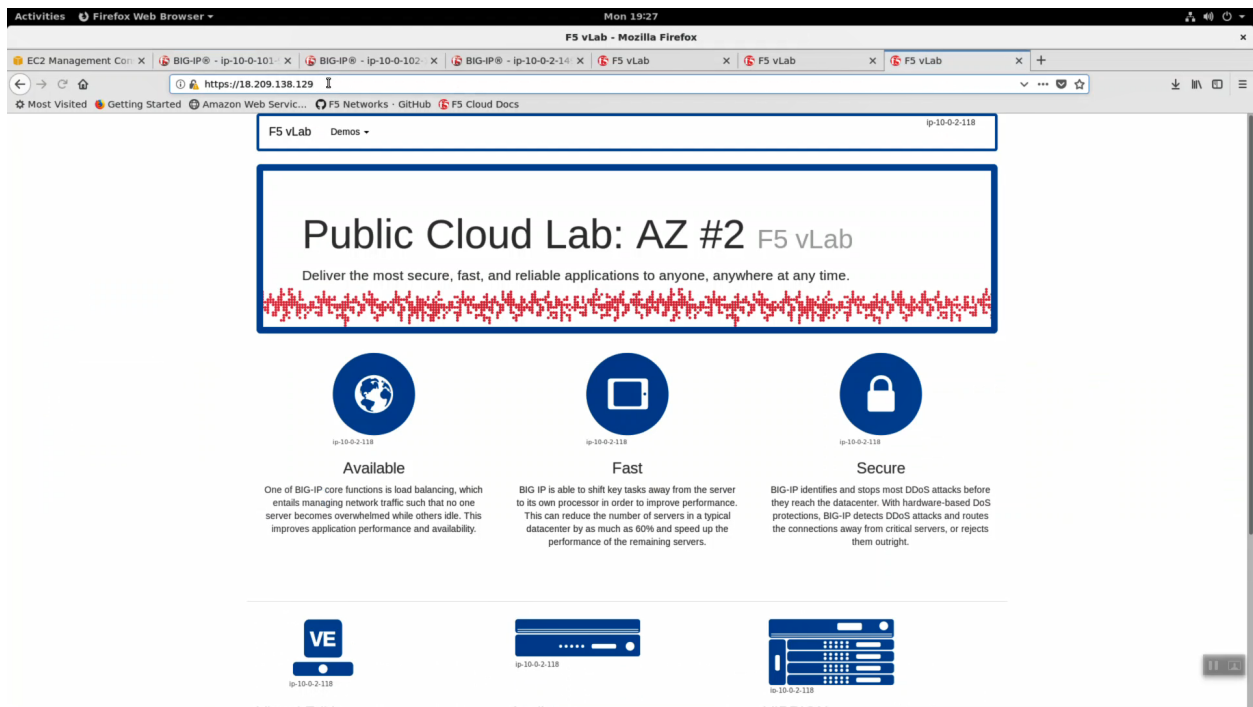
HTTPS to the Elastic IP. We simulated a failover event and our sample application is still up. Because only the Big-IP has failed, not the whole Availability Zone, and the client is configured for persistence, the application is still served up from the same Availability Zone.



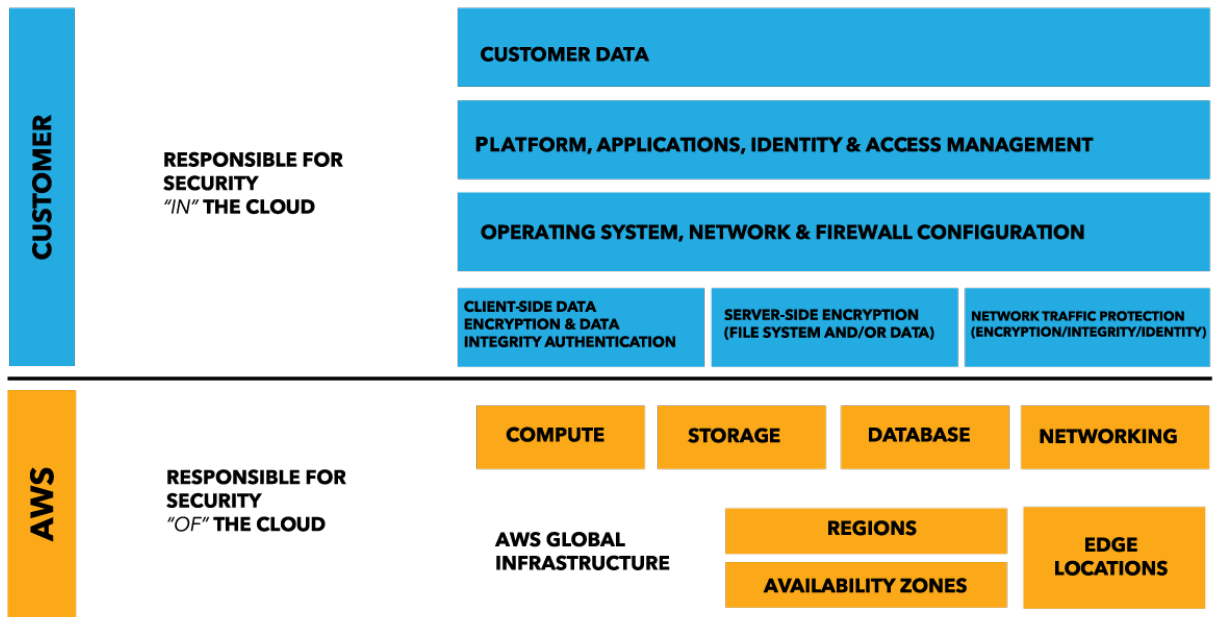
Now we'll simulate an Availability Zone outage. From the https Configuration Utility (Web UI) of the active Big-IPX device: Local Traffic => Pools => Members => Select the pool member in Availability Zone #1 (almost always the first pool member) and [Force Offline].



HTTPS to the Elastic IP. Hit refresh [F5] a few times to refresh the cache. Notice we are not connecting to the application on AZ#2.



Note: Traditional HA failover relies on Layer 2 connectivity and a heartbeat to trigger a fail-over event and move a 'floating IP' to a new active unit. There is no Layer 2 connectivity in the cloud across availability zones. The Big-IP will detect an availability zone outage or trouble with a Big-IP VE and the elastic IP will 'float' over to the new active device as you just saw.



2.5 Logging to CloudWatch

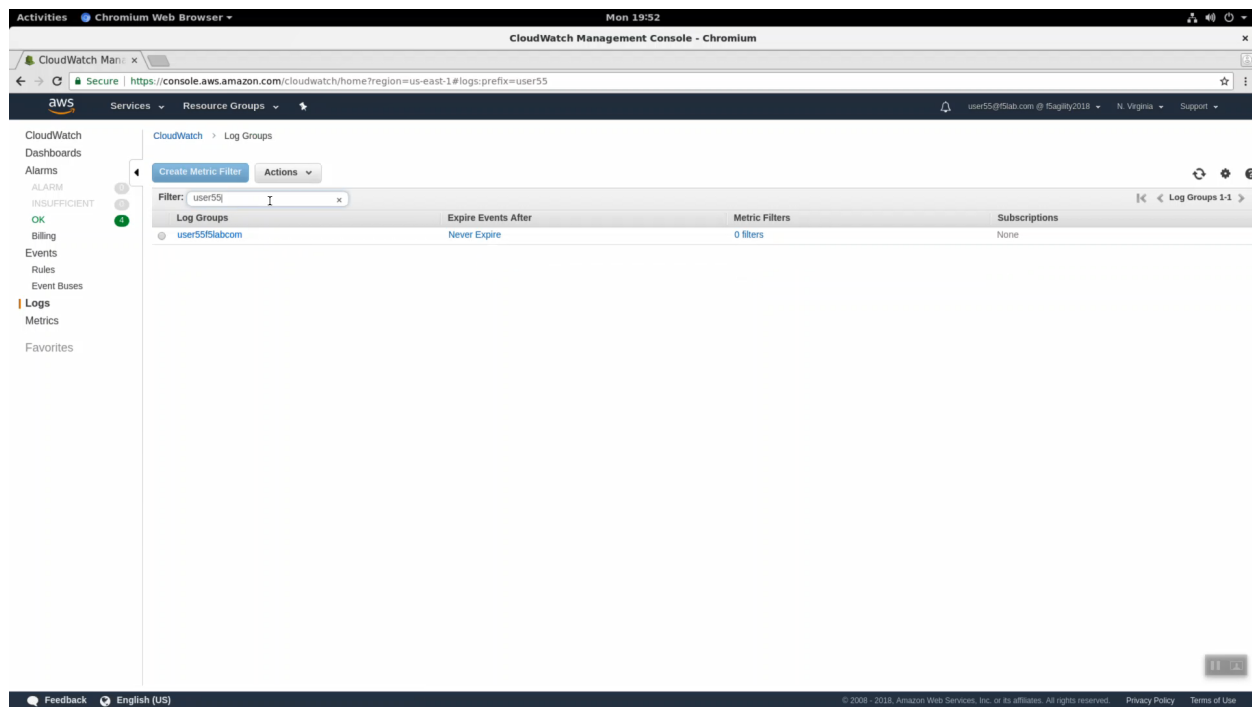
F5 Virtual Editions support comprehensive request and security logging for compliance and troubleshooting using two AWS native features: S3 Buckets and CloudWatch. In this lab we'll configure logging to CloudWatch.

2.5.1 LTM Request Logging to CloudWatch

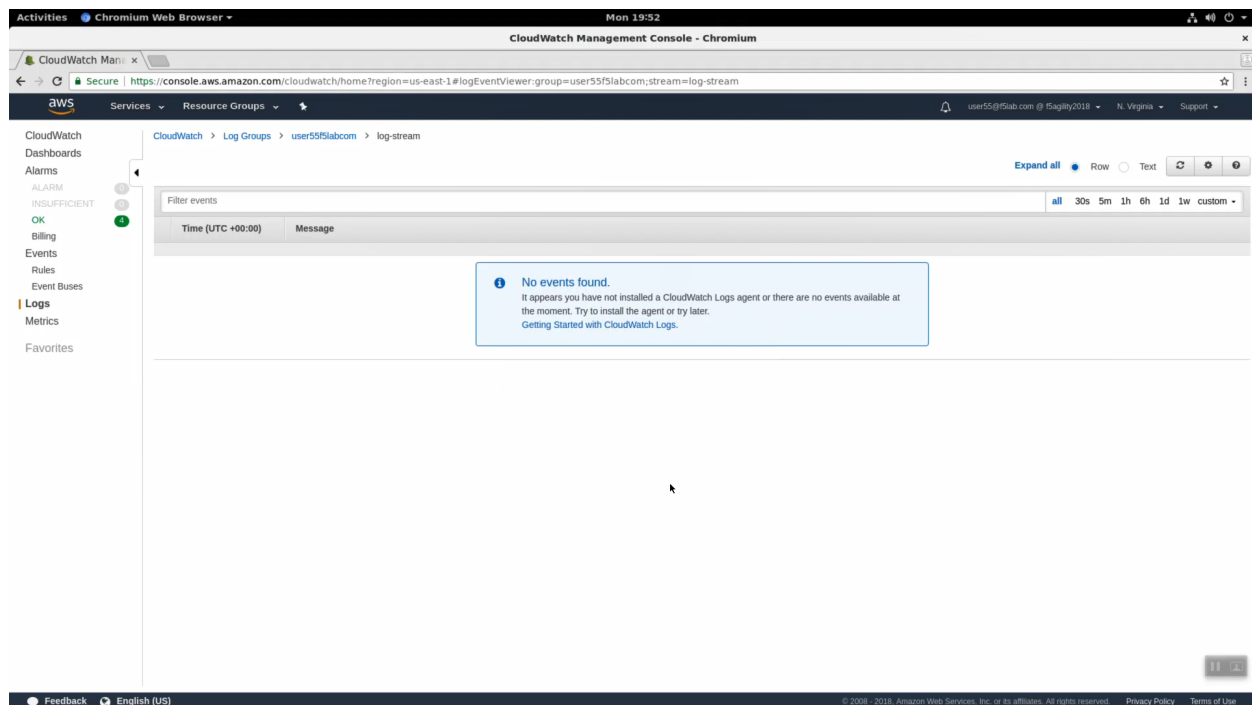
From the Super-NetOps terminal, run the handy lab-info utility. Confirm that "MCPD is up, System Ready" for all three of your instances.

```
lab-info
```

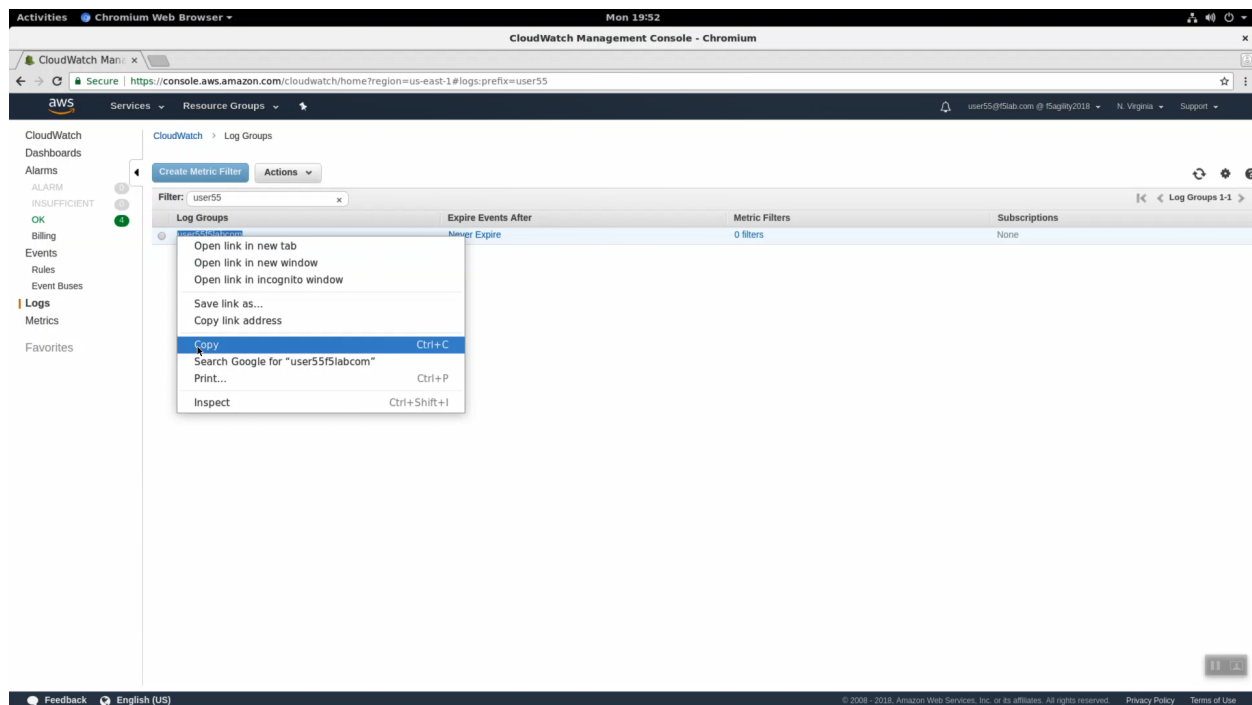
From the AWS management console, navigate to Services => Management Tools => CloudWatch => Log Groups. In the search filter enter your username (i.e. user55). Terraform created a Log Group for you.



Click on your log group. Click on your log stream named “log-stream”. Notice the Message column has no messages.



Right-click and copy your log group name (i.e. user55labcom). **Save in notepad or your preferred text editor / note taking method for later use.**



For convenience working through the next few steps, split your screen into two halves: Super-NetOps terminal on the left and the Firefox or Chrome browser on the right. On a standard Windows US/English Windows keyboard you can split the screen with <Windows Key + left arrow> and <Windows Key + right arrow>.

From your Super-NetOps terminal, there are multiple ways to see your AWS access keys. You can echo the environment variables:

```
echo $AWS_ACCESS_KEY_ID
echo $AWS_SECRET_ACCESS_KEY
```

... or you can cat the hidden ~/.aws/config file:

```
cat ~/.aws/config
```

Copy your AWS_ACCESS_KEY_ID and AWS_SECRET_ACCESS_KEY values. **Save in notepad or your preferred text editor / note taking method for later use.**

Create a new cloud_logger iApp. HTTPS to the Configuration Utility (Web UI) of Big-IP1 (assuming that is the ACTIVE device and not STANDBY).

iApps => Application Services => Name: *cloudwatch*. Template: *f5.cloud_logger.v1.0.0*. Click [Finished].

Activities Terminal

student@docker: ~

```
[root@fs-super-netops] [~/marfil-f5-terraform] $ cat ~/.aws/config
[default]
aws_secret_access_key = rnfXAdYyWb4MCNoofJk2HL+T3+ZBRBkpRq21PTPx
region = us-east-1
aws_access_key_id = AKIA3BBLT3DJETPMYI3Q
[root@fs-super-netops] [~/marfil-f5-terraform] $ echo $AWS_ACCESS_KEY_ID
AKIA3BBLT3DJETPMYI3Q
[root@fs-super-netops] [~/marfil-f5-terraform] $ echo $AWS_SECRET_ACCESS_KEY
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
[root@fs-super-netops] [~/marfil-f5-terraform] $
```

BIG-IP® - ip-10-0-102-111.ec2.internal (10.0.102.111) - Mozilla Firefox

EC2 Man... CloudWai... BIG-IP® X BIG-IP® F5 vLab F5 vLab F5 vLab

https://35.170.139.232/xul/

Most Visited Getting Started Amazon Web Servic... F5 Networks - GitHub F5 Cloud Docs

IP Address: 10.0.102.111.ec2.internal Date: Jul 16, 2018 User: admin Role: Administrator Partition: Common Log out

ONLINE (ACTIVE) In Sync

Main Help About

Statistics

Apps

Application Services

Templates

AWS

Local Traffic

Acceleration

Device Management

Network

System

Apps > Application Services > Applications

Application Service List

Search

Create

Name	Template	Template Validity	Partition / Path
HA_Across_AZs	f5.aws_advanced_ha.v1.4.0rc3		CommonHA_Across_AZs.app
service_discovery	f5.service_discovery		Commonservice_discovery.app
Virtual_server1	f5.tsp.v1.0.0rc2		Commonvirtual_server1.app

Delete...

Activities Firefox Web Browser

student@docker: ~

```
[root@fs-super-netops] [~/marfil-f5-terraform] $ cat ~/.aws/config
[default]
aws_secret_access_key = rnfXAdYyWb4MCNoofJk2HL+T3+ZBRBkpRq21PTPx
region = us-east-1
aws_access_key_id = AKIA3BBLT3DJETPMYI3Q
[root@fs-super-netops] [~/marfil-f5-terraform] $ echo $AWS_ACCESS_KEY_ID
AKIA3BBLT3DJETPMYI3Q
[root@fs-super-netops] [~/marfil-f5-terraform] $ echo $AWS_SECRET_ACCESS_KEY
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
[root@fs-super-netops] [~/marfil-f5-terraform] $
```

BIG-IP® - ip-10-0-102-111.ec2.internal (10.0.102.111) - Mozilla Firefox

EC2 Man... CloudWai... BIG-IP® X BIG-IP® F5 vLab F5 vLab F5 vLab

https://35.170.139.232/xul/

Most Visited Getting Started Amazon Web Servic... F5 Networks - GitHub F5 Cloud Docs

IP Address: 10.0.102.111.ec2.internal Date: Jul 16, 2018 User: admin Role: Administrator Partition: Common Log out

ONLINE (ACTIVE) In Sync

Main Help About

Statistics

Apps

Application Services

Templates

AWS

Local Traffic

Acceleration

Device Management

Network

System

Apps > Application Services > Applications > New Application Service...

Template Selection

Name

cloudwatch

Template

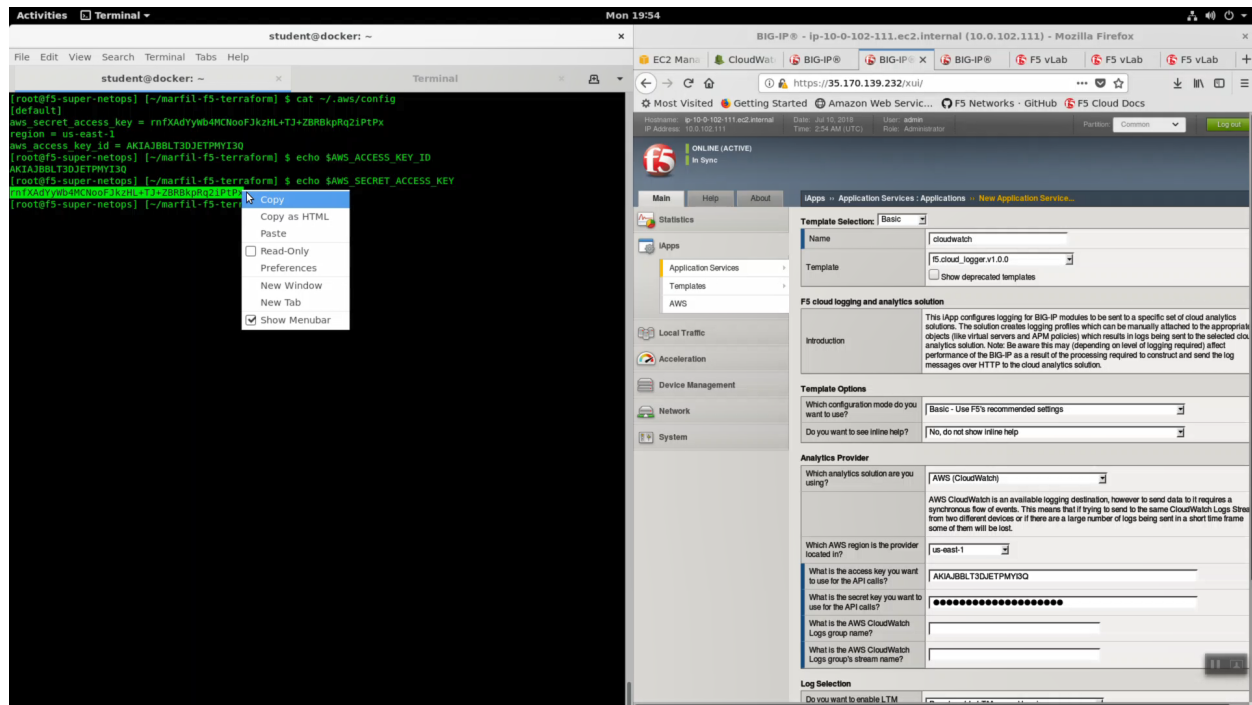
None - Do not use a template

None - Do not use a template

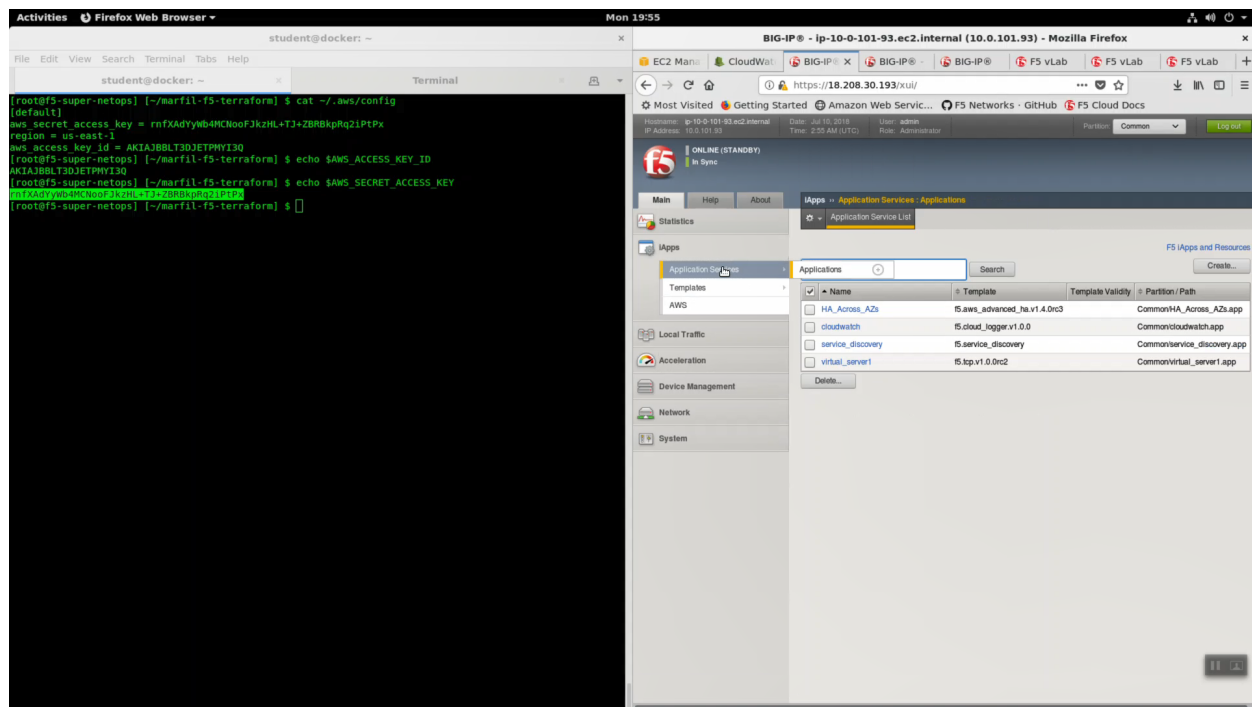
Cancel Repeat

- f5.aws_advanced_ha.v1.4.0rc3
- f5.bea_weblogic
- f5.cloud_logger.v0.0
- f5.diameter
- f5.dns
- f5.http
- f5.ip_forwarding
- f5.idap
- f5.microsoft_jis
- f5.microsoft_sharepoint_2010
- f5.npath
- f5.oracle_as_10g
- f5.oracle_ebs
- f5.peoplesoft_9
- f5.radius
- f5.sap_enterprise_portal
- f5.sap_erp
- f5.service_discovery

Question	value
Name	cloudwatch
Template	f5.cloud_logger.v1.0.0
Which AWS region is the provider located in?	us-east-1
What is the access key you want to use for the API calls?	value of \$AWS_ACCESS_KEY_ID
What is the secret key you want to use for the API calls?	value of \$AWS_SECRET_ACCESS_KEY
What is the AWS CloudWatch Logs group name?	log group name i.e. user55labcom
What is the AWS CloudWatch Logs group's stream name?	log-stream
Do you want to enable LTM Request logging?	Enable LTM request logging



Click [Finished].



HTTPS to the Configuration Utility (Web UI) of Big-IP1 (assuming that is the ACTIVE device and not STANDBY).

iApps => Application Services => virtual_server1.

Attention: Before completing the next few steps, DISABLE STRICT UPDATES for the f5.tcp.v1.0.0rc2 iApp named virtual_server1 in our example.

Local Traffic => Virtual Servers => virtual_server1_vs.10.0.1.x.

- Choose “Advanced” from the dropdown.
- Select SSL Profile(Client): clientssl
- Change HTTP Profile to “http”
- Request Logging Profile: cloudwatch_remote_logging

Click [Update].

Activities Firefox Web Browser student@docker: - Mon 19:56

student@docker: - Terminal

```
[root@fs-super-netops] [~/marfil-f5-terraform] $ cat ~/.aws/config
[default]
aws_secret_access_key = rnfXADyYnB4MCNoofJk2HL+T3+ZBRBkpRq21PtPx
region = us-east-1
aws_access_key_id = AKIA3BBLT3DJETPMYI3Q
[root@fs-super-netops] [~/marfil-f5-terraform] $ echo $AWS_ACCESS_KEY_ID
AKIA3BBLT3DJETPMYI3Q
[root@fs-super-netops] [~/marfil-f5-terraform] $ echo $AWS_SECRET_ACCESS_KEY
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
[root@fs-super-netops] [~/marfil-f5-terraform] $
```

BIG-IP® - ip-10-0-102-111.ec2.internal (10.0.102.111) - Mozilla Firefox

EC2 Man... CloudWat... BIG-IP® - BIG-IP® - BIG-IP® - F5 vLab - F5 vLab - F5 vLab

Most Visited Getting Started Amazon Web Servic... F5 Networks - GitHub F5 Cloud Docs

IP Address: 10.0.102.111 Date: Jul 16, 2018 User: admin Role: Administrator Partition: Common Log out

ONLINE (ACTIVE) In Sync

Main Help About Local Traffic Virtual Servers Virtual Server List

Virtual Server List

Virtual Server List	Virtual Address List	Statistics
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Status	Name	Description
<input checked="" type="checkbox"/>	cloudwatch_cred_vs	cloudwatch
<input checked="" type="checkbox"/>	cloudwatch_formal_vs	cloudwatch
<input checked="" type="checkbox"/>	cloudwatch_send_vs	cloudwatch
<input checked="" type="checkbox"/>	virtual_server1_vs_10.0.1.150	virtual_server1 10.0.1.150 443 (HTTPS) Standard Edit
<input checked="" type="checkbox"/>	virtual_server1_vs_10.0.2.58	virtual_server1 10.0.2.58 443 (HTTPS) Standard Edit

Enable Disable Delete

Network Map Virtual Servers Policies Profiles Ciphers iRules Pools Nodes Monitors Traffic Class Address Translation Acceleration Device Management Network System

https://35.170.139.232/tmui/Control/jspmap/tmui/locallb/virt...e=/Common/virtual_server1.app/virtual_server1_vs_10.0.1.150

Activities Firefox Web Browser student@docker: - Mon 19:56

student@docker: - Terminal

```
[root@fs-super-netops] [~/marfil-f5-terraform] $ cat ~/.aws/config
[default]
aws_secret_access_key = rnfXADyYnB4MCNoofJk2HL+T3+ZBRBkpRq21PtPx
region = us-east-1
aws_access_key_id = AKIA3BBLT3DJETPMYI3Q
[root@fs-super-netops] [~/marfil-f5-terraform] $ echo $AWS_ACCESS_KEY_ID
AKIA3BBLT3DJETPMYI3Q
[root@fs-super-netops] [~/marfil-f5-terraform] $ echo $AWS_SECRET_ACCESS_KEY
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
[root@fs-super-netops] [~/marfil-f5-terraform] $
```

BIG-IP® - ip-10-0-102-111.ec2.internal (10.0.102.111) - Mozilla Firefox

EC2 Man... CloudWat... BIG-IP® - BIG-IP® - BIG-IP® - F5 vLab - F5 vLab - F5 vLab

Most Visited Getting Started Amazon Web Servic... F5 Networks - GitHub F5 Cloud Docs

IP Address: 10.0.102.111 Date: Jul 16, 2018 User: admin Role: Administrator Partition: Common Log out

ONLINE (ACTIVE) In Sync

Main Help About Local Traffic Virtual Servers Virtual Server List

Virtual Server List

Virtual Server List	Virtual Address List	Statistics
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Status	Name	Description
<input checked="" type="checkbox"/>	cloudwatch_cred_vs	cloudwatch
<input checked="" type="checkbox"/>	cloudwatch_formal_vs	cloudwatch
<input checked="" type="checkbox"/>	cloudwatch_send_vs	cloudwatch
<input checked="" type="checkbox"/>	virtual_server1_vs_10.0.1.150	virtual_server1 10.0.1.150 443 (HTTPS) Standard Edit
<input checked="" type="checkbox"/>	virtual_server1_vs_10.0.2.58	virtual_server1 10.0.2.58 443 (HTTPS) Standard Edit

Enable Disable Delete

Network Map Virtual Servers Policies Profiles Ciphers iRules Pools Nodes Monitors Traffic Class Address Translation Acceleration Device Management Network System

https://35.170.139.232/tmui/Control/jspmap/tmui/locallb/virt...e=/Common/virtual_server1.app/virtual_server1_vs_10.0.1.150

Availability: Available (Enabled) - The virtual server is available

Sync cookie Status: Off

State: Enabled

Configuration: Basic

Protocol: Basic TCP

Protocol Profile: Advanced tcp

Protocol Profile (Server): (Use Client Profile)

HTTP Profile: None

HTTP Proxy Connect Profile: None

FTP Profile: None

RTSP Profile: None

SSL Profile (Client): Selected: Common clientsssl Available: Common clientsssl-insecure-compatible, clientsssl-secure, crypto-server-default-clientsssl, split-session-default-clientsssl

SSL Profile (Server): Selected: Common sslm-default-serverssl Available: Common sslm-default-serverssl, crypto-client-default-serverssl, pcsp-default-serverssl, serverssl

SMTPS Profile: None

Client LDAP Profile: None

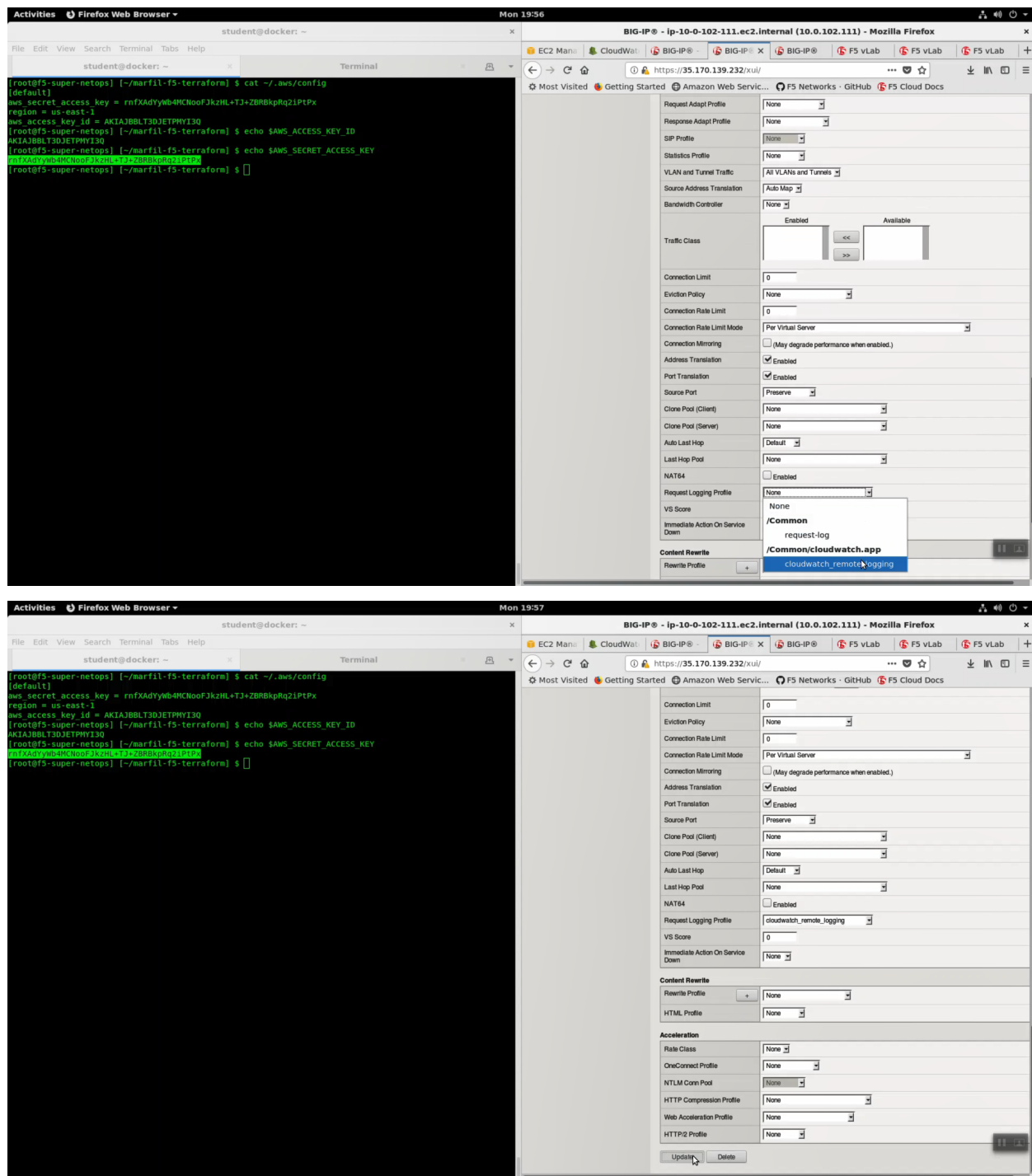
Server LDAP Profile: None

VLAN and Tunnel Traffic: All VLANs and Tunnels

Source Address Translation: Auto Map

Content Rewrite: Rewrite Profile: None HTML Profile: None

Acceleration: Rate Class: None OneConnect Profile: None



Do the same for the second virtual server. Local Traffic => Virtual Servers => virtual_server1_vs.10.0.1.x.

- Choose "Advanced" from the dropdown.
- Select SSL Profile(Client): clientssl
- Change HTTP Profile to "http"
- Request Logging Profile: cloudwatch_remote_logging

Click [Update].

Activities Firefox Web Browser Mon 19:57

student@docker: ~

Terminal

```
[root@f5-super-netops] [~/marfil-f5-terraform] $ cat ~/.aws/config
[default]
aws_secret_access_key = rnfXADyYnB4MCNooF3k2HL+T3+ZBRBkpRq21PtPx
region = us-east-1
aws_access_key_id = AKIA3BBLT3DJETPMYI3Q
[root@f5-super-netops] [~/marfil-f5-terraform] $ echo $AWS_ACCESS_KEY_ID
AKIA3BBLT3DJETPMYI3Q
[root@f5-super-netops] [~/marfil-f5-terraform] $ echo $AWS_SECRET_ACCESS_KEY
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
[root@f5-super-netops] [~/marfil-f5-terraform] $
```

Firefox Web Browser

BIG-IP® - ip-10-0-102-111.ec2.internal (10.0.102.111) - Mozilla Firefox

EC2 Man... CloudWai... BIG-IP® - BIG-IP® - BIG-IP® F5 vLab F5 vLab F5 vLab

Most Visited Getting Started Amazon Web Servic... F5 Networks - GitHub F5 Cloud Docs

Address: ip-10-0-102-111.ec2.internal Date: Jul 16, 2018 User: gdmr Role: Administrator Partition: Common Log out

IP Address: 10.0.102.111

ONLINE (ACTIVE) In Sync

Loading... Retrieving configuration data from your device.

Local Traffic Virtual Servers: Virtual Server List

Virtual Server List

Virtual Server List	Virtual Address List	Statistics
cloudwatch_cred_vs	cloudwatch	255.255.255.254 41003 Standard Edit
cloudwatch_formal_vs	cloudwatch	255.255.255.254 41001 Standard Edit
cloudwatch_send_vs	cloudwatch	255.255.255.254 41002 Standard Edit
virtual_server1_vs_10.0.1.150	virtual_server1	10.0.1.150 443 (HTTPS) Standard Edit
virtual_server1_vs_10.0.2.58	virtual_server1	10.0.2.58 443 (HTTPS) Standard Edit

Enable Disable Delete

Waiting for 35.170.139.232...

Activities Firefox Web Browser Mon 19:57

student@docker: ~

Terminal

```
[root@f5-super-netops] [~/marfil-f5-terraform] $ cat ~/.aws/config
[default]
aws_secret_access_key = rnfXADyYnB4MCNooF3k2HL+T3+ZBRBkpRq21PtPx
region = us-east-1
aws_access_key_id = AKIA3BBLT3DJETPMYI3Q
[root@f5-super-netops] [~/marfil-f5-terraform] $ echo $AWS_ACCESS_KEY_ID
AKIA3BBLT3DJETPMYI3Q
[root@f5-super-netops] [~/marfil-f5-terraform] $ echo $AWS_SECRET_ACCESS_KEY
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
[root@f5-super-netops] [~/marfil-f5-terraform] $
```

Firefox Web Browser

BIG-IP® - ip-10-0-102-111.ec2.internal (10.0.102.111) - Mozilla Firefox

EC2 Man... CloudWai... BIG-IP® - BIG-IP® - BIG-IP® F5 vLab F5 vLab F5 vLab

Most Visited Getting Started Amazon Web Servic... F5 Networks - GitHub F5 Cloud Docs

Bandwidth Controller: None

Traffic Class: Enabled Available

Connection Limit: 0

Evasion Policy: None

Connection Rate Limit: 0

Connection Rate Limit Mode: Per Virtual Server

Connection Mirroring: (May degrade performance when enabled.)

Address Translation: Enabled

Port Translation: Enabled

Source Port: Preserve

Clone Pool (Client): None

Clone Pool (Server): None

Auto Last Hop: Default

Last Hop Pool: None

NAT64: Enabled

Request Logging Profile: None

VS Score: None

Immediate Action On Service Down: /Common request-log

Content Rewrite: /Common/cloudwatch.app

Rewrite Profile: cloudwatch_remake_logging

HTML Profile: None

Acceleration: Rate Class: None OneConnect Profile: None NTLM Conn Pool: None HTTP Compression Profile: None

Public Cloud Lab: AZ #2 F5 vLab

Deliver the most secure, fast, and reliable applications to anyone, anywhere at any time.

Available

One of BIG-IP core functions is load balancing, which entails managing network traffic such that no one server becomes overwhelmed while others idle. This improves application performance and availability.

Fast

```
[default]
aws_secret_access_key = rnfXAdYyNb4HCNoofJkzHL+Tj+ZBRBkpRq21PtPx
region = us-east-1
aws_access_key_id = AKIAJBBLT3DJETPMYI3Q
[root@f5-super-netops] [~/marfil-f5-terraform] $ echo $AWS_ACCESS_KEY_ID
AKIAJBBLT3DJETPMYI3Q
[root@f5-super-netops] [~/marfil-f5-terraform] $ echo $AWS_SECRET_ACCESS_KEY
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
[root@f5-super-netops] [~/marfil-f5-terraform] $ lab-info
AWS Console
URL: https://f5agility2018.signin.aws.amazon.com/console?us-east-1
Username: user55@f5lab.com / Password: cloudy
WAF ELB
URL: https://waf-user55f5labcom-3256385.us-east-1.elb.amazonaws.com

web-az1.0: user55f5labcom
PRIVATE IP: 10.0.1.190

Big-IP1: ha-user55f5labcom-vpc-7252a108
MGMT IP: 10.208.30.193
STATUS: MCPD is up, System Ready
MGMT URL: https://10.208.30.193
SSH: ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no -o ConnectTimeout=3 -i M
yKeyPair-user55f5lab.com.pem admin@10.208.30.193
VIP IP: 10.0.1.190

web-az2.0: user55f5labcom
PRIVATE IP: 10.0.2.118

BIG-IP Autoscale Instance: waf-user55f5labcom
MGMT IP: 54.172.188.81
STATUS: MCPD is up, System Ready
MGMT URL: https://54.172.188.81:8443
SSH: ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no -o ConnectTimeout=3 -i M
yKeyPair-user55f5lab.com.pem admin@54.172.188.81

Big-IP2: ha-user55f5labcom-vpc-7252a108
MGMT IP: 35.170.139.232
STATUS: MCPD is up, System Ready
MGMT URL: https://35.170.139.232
SSH: ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no -o ConnectTimeout=3 -i M
yKeyPair-user55f5lab.com.pem admin@35.170.139.232
VIP IP: 10.0.2.58
Elastic IP: 10.209.130.129

[root@f5-super-netops] [~/marfil-f5-terraform] $
```

Attention: Some lab testers reported an incompatibility issue with Mozilla Firefox on Linux and the AWS CloudWatch console. If Firefox doesn't render the CloudWatch console, switch to Google Chrome for this part of the lab.

From the AWS Console, Services => Management Tools => CloudWatch => Log Groups. Select your log group and log-stream.

CloudWatch Management Console - Chromium

Services | Resource Groups | **Log Groups** | Streams for user55f5labcom

Search Log Group | Create Log Stream | Delete Log Stream

Filter: Log Stream Name Prefix x

Log Streams

log-stream

Last Event Time

Log Streams 1-1

https://console.aws.amazon.com/cloudwatch/home?region=us-east-1#logEventV...

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

You will see the http request logs.

The screenshot shows the AWS CloudWatch Management Console in a Chromium browser. The page title is "CloudWatch Management Console - Chromium". The address bar shows the URL: <https://console.aws.amazon.com/cloudwatch/home?region=us-east-1#logEventViewer:group=user55f5labcom;stream=log-stream>. The left sidebar contains navigation links: CloudWatch, Dashboards, Alarms, Rules, Event Buses, Logs (selected), Metrics, and Favorites. The main content area shows a log stream for the group "user55f5labcom" and stream "log-stream". The "Filter events" bar is empty. The log entries are displayed in a table with columns "Time (UTC +00:00)" and "Message". The messages are JSON objects representing log data. The first entry is at 02:59:53 and contains a large JSON object. The subsequent entries are at 02:59:54, 03:00:01, 03:00:02, 03:00:02, 03:00:02, and 03:00:02. The console footer shows the copyright notice: "© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved." and links to "Privacy Policy" and "Terms of Use".

Expand a log entry to see more detail.

The screenshot shows the AWS CloudWatch Management Console with the same log stream as the previous image. The log entry at 02:59:53 is expanded, showing its full details in a JSON format. The details include the time, host, log source, log version, event source, log group, client IP, server IP, HTTP method, HTTP URI, and virtual name. A tooltip is visible over the "Access statistics, performance graphs, and links to helpful tools." link. The console footer shows the copyright notice: "© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved." and links to "Privacy Policy" and "Terms of Use".

Copy the CLIENT_IP of a request and use this CLIENT_IP in the “Filter events” search filter. In production you would filter search results by attributes such as CLIENT-IP to home in on relevant logs.

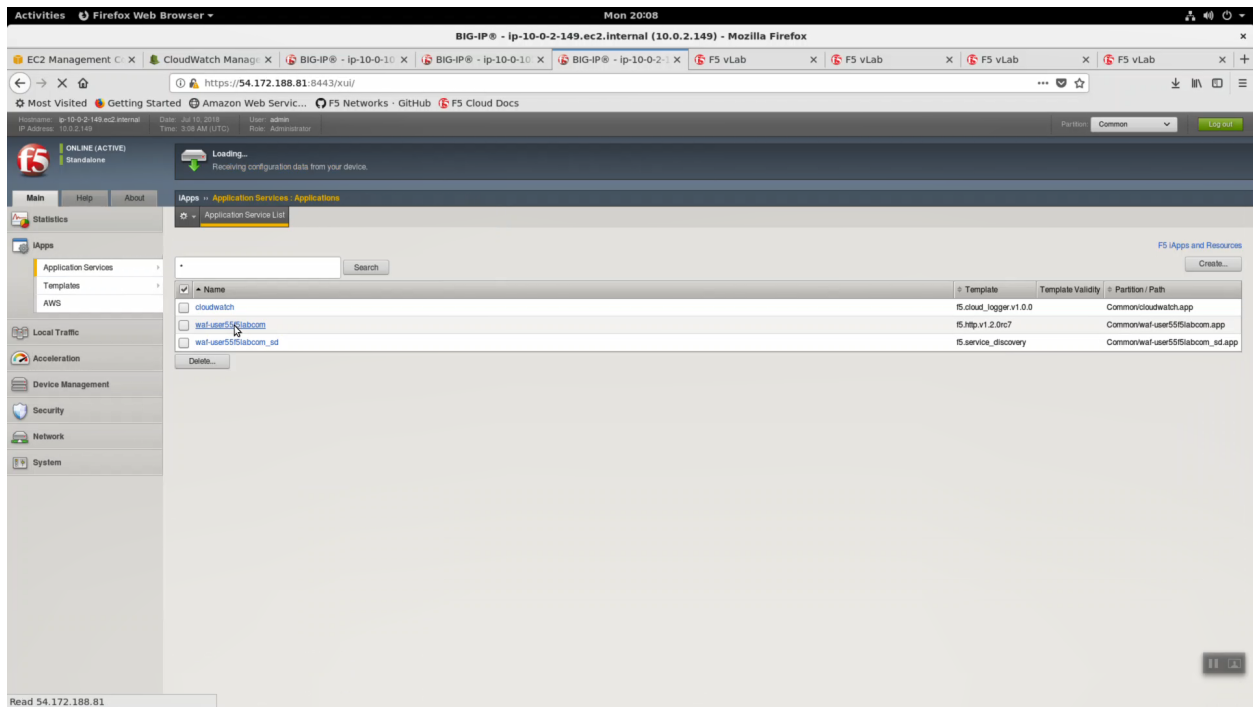
The first screenshot shows the CloudWatch Management Console with a log stream named 'log-stream' under the group 'user55f5labcom'. A context menu is open over a log event, showing options like 'Copy', 'Go to 129.213.189.8', 'Print...', and 'Inspect'. The log event contains a JSON object with details about a request, including the client IP '129.213.189.8'.

The second screenshot shows the same console with the log stream filtered by the client IP '129.213.189.8'. The filter is applied to the 'CLIENT_IP' field, and the log stream shows only events from this specific client IP.

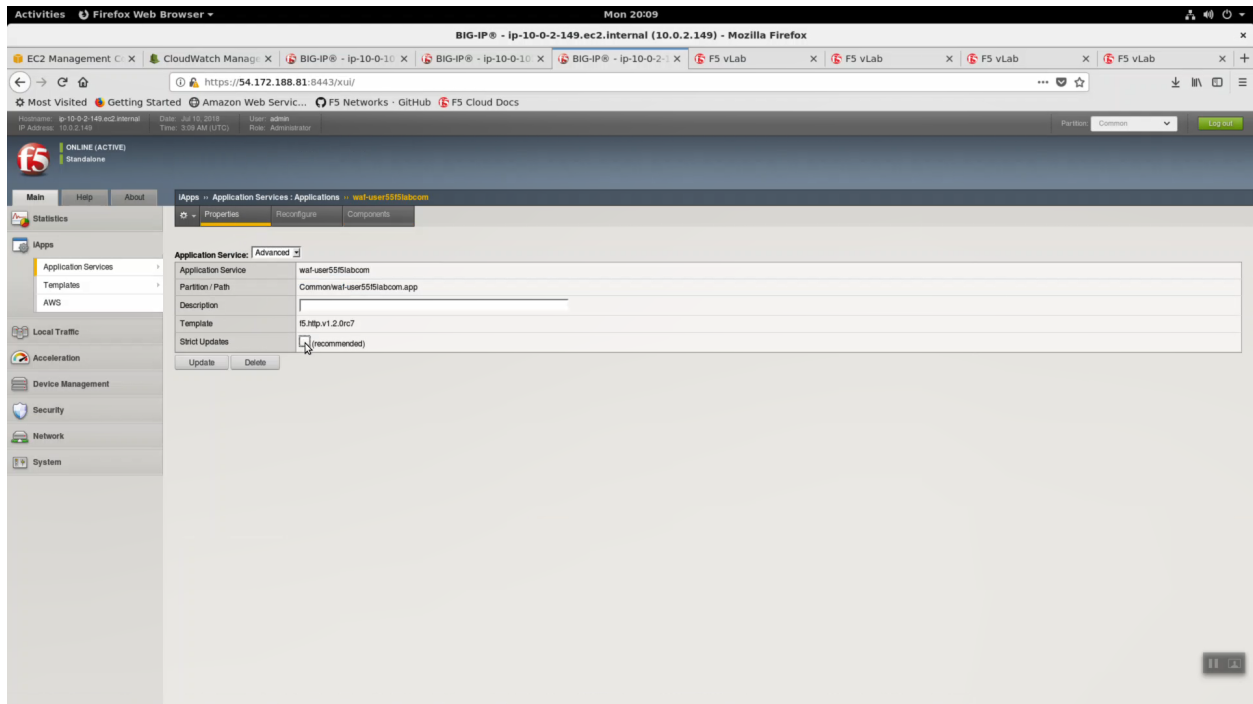
2.5.2 WAF HTTP Request and Security Logging to CloudWatch

HTTPS to the Configuration Utility (Web UI) of the BIG-IP Autoscale Instance: waf...

iApps => Application Services => waf=userxxf5labcom.



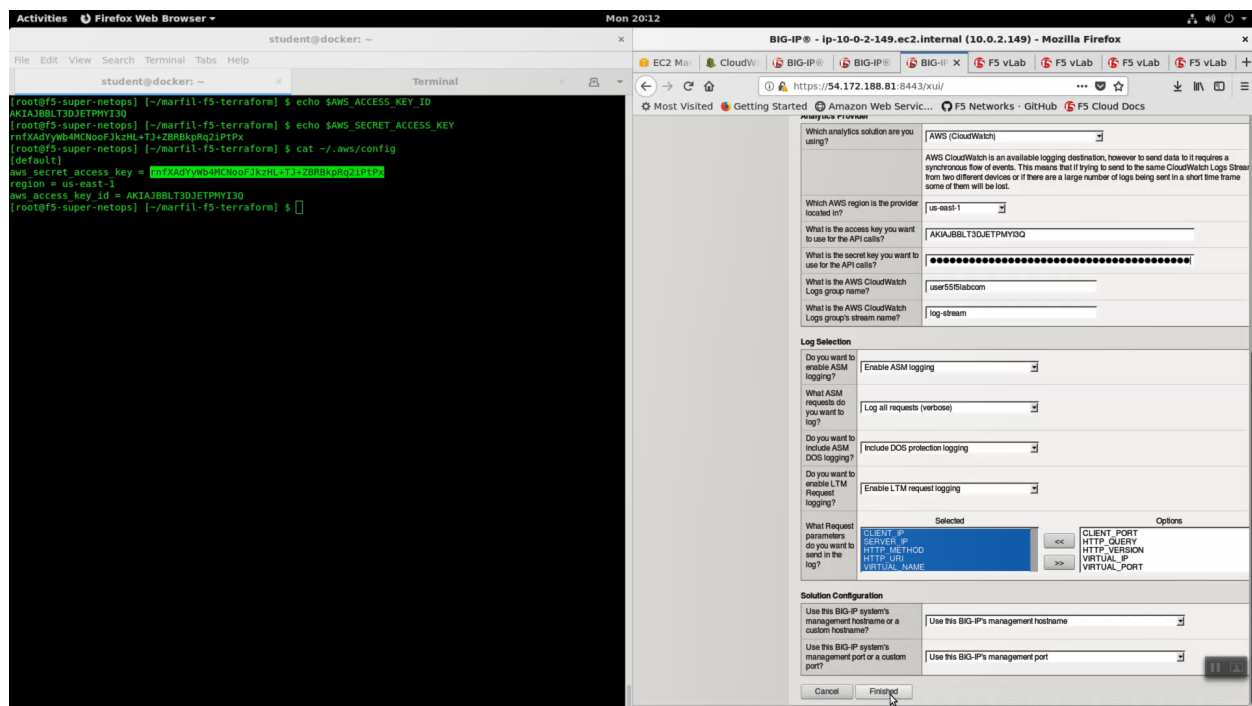
Properties => UNCHECK "Strict Updates". [Update].



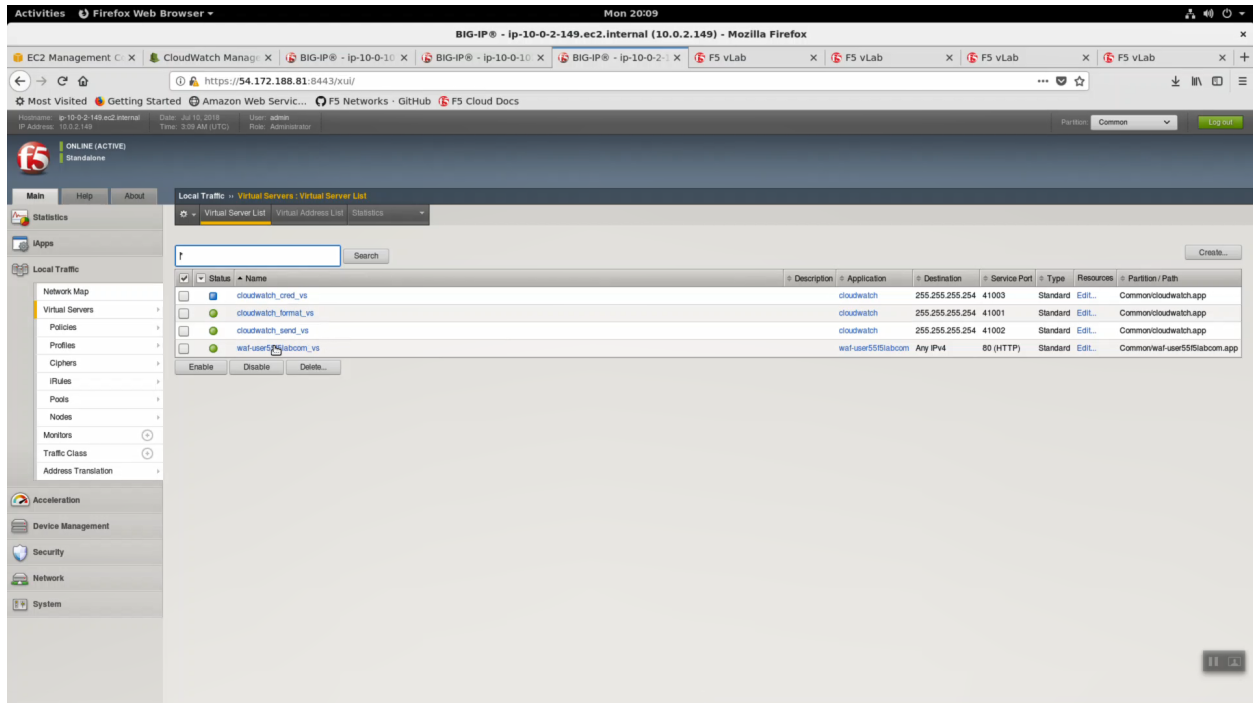
Create a new cloud_logger iApp. iApps => Application Services => Name: *cloudwatch*. Template: *f5.cloud_logger.v1.0.0*. Click [Finished].

Question	value
Name	cloudwatch
Template	f5.cloud_logger.v1.0.0
Which AWS region is the provider located in?	us-east-1
What is the access key you want to use for the API calls?	value of \$AWS_ACCESS_KEY_ID
What is the secret key you want to use for the API calls?	value of \$AWS_SECRET_ACCESS_KEY
What is the AWS CloudWatch Logs group name?	log group name i.e. user55labcom
What is the AWS CloudWatch Logs group's stream name?	log-stream
Do you want to enable ASM logging?	Enable ASM logging
What ASM requests do you want to log?	Log all requests (verbose)
Do you want to include ASM DOS logging?	Include DOS protection logging
Do you want to enable LTM Request logging?	Enable LTM request logging
What Request parameters do you want to send in the log?	leave defaults

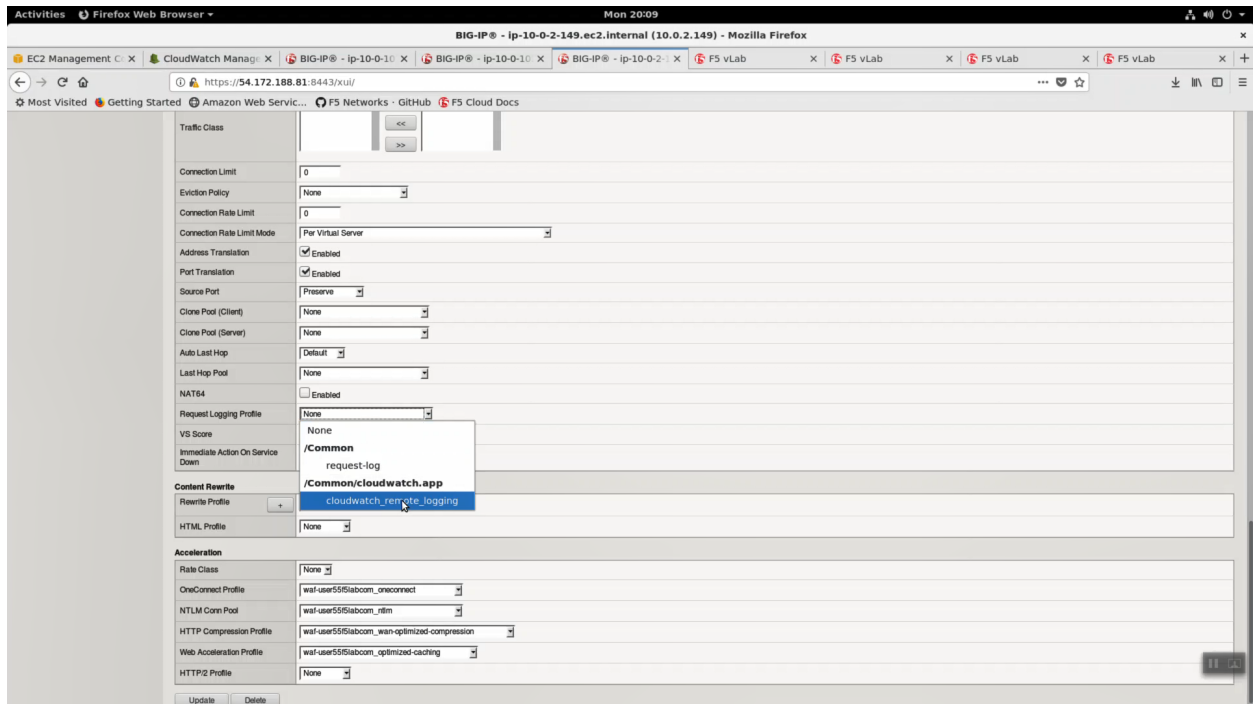
Click [Finished].



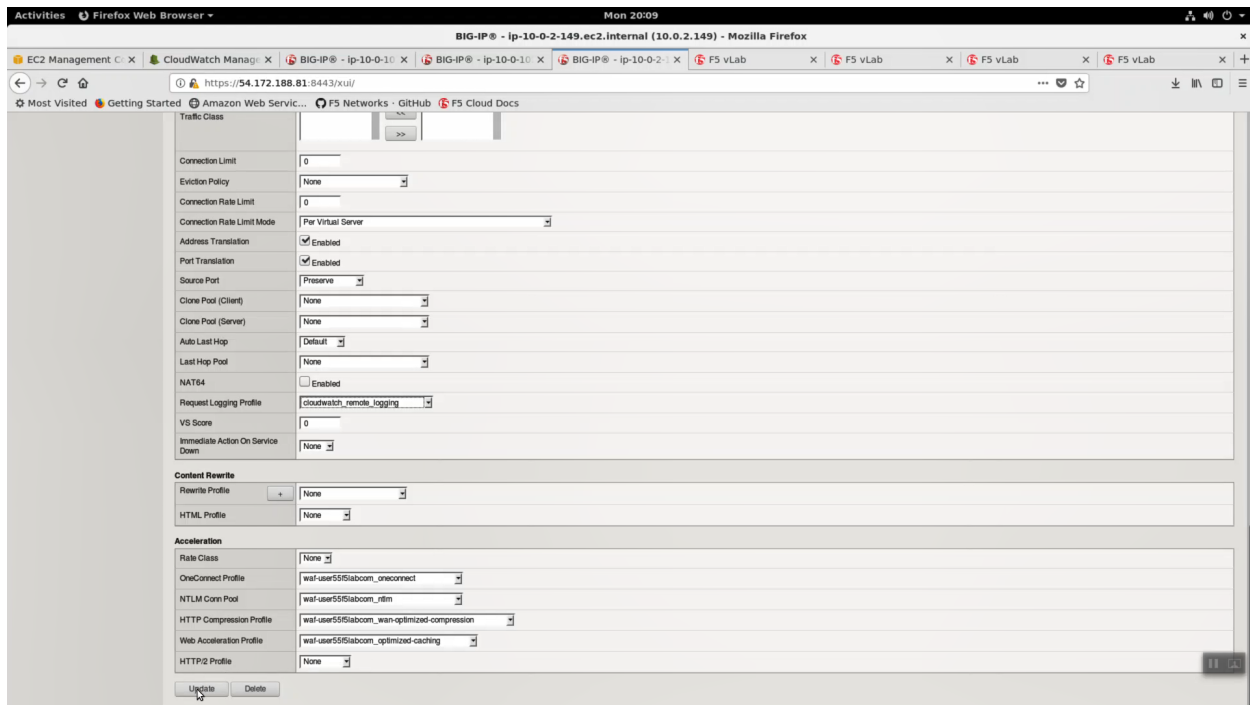
Local Traffic => Virtual Server => waf-userXXf5labcom_vs.



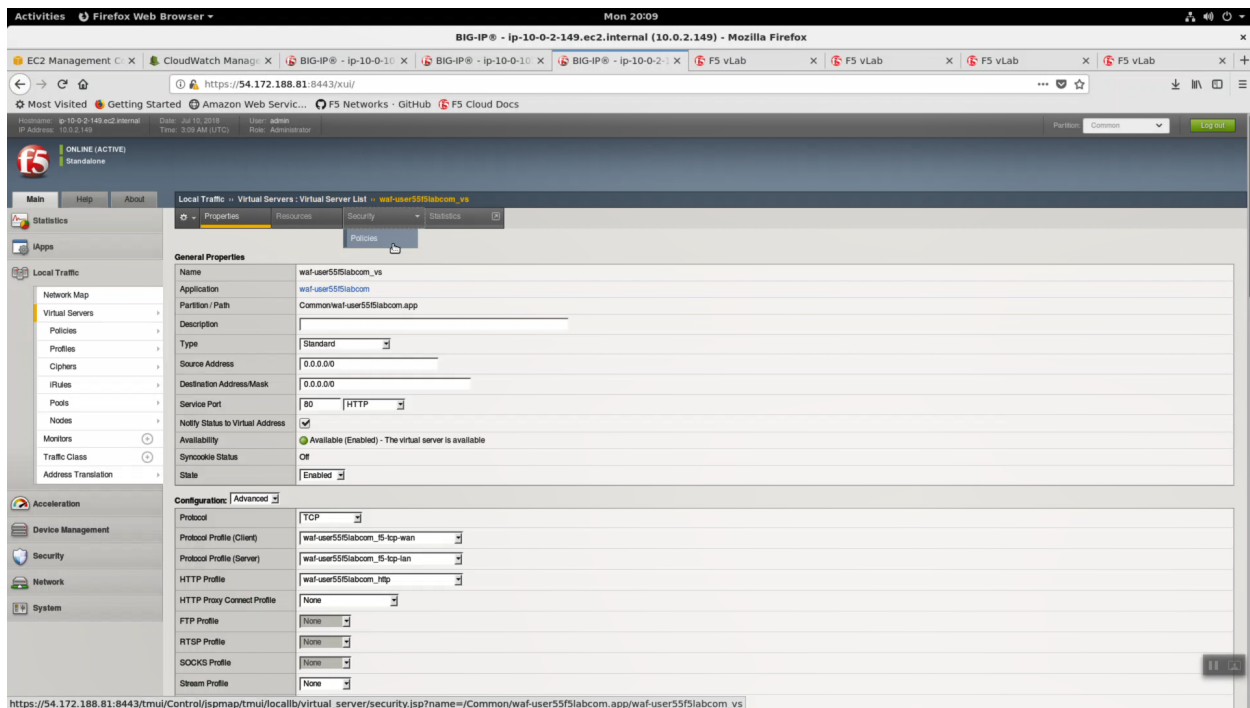
Change Request Logging Profile to *cloudwatch_remote_logging*.



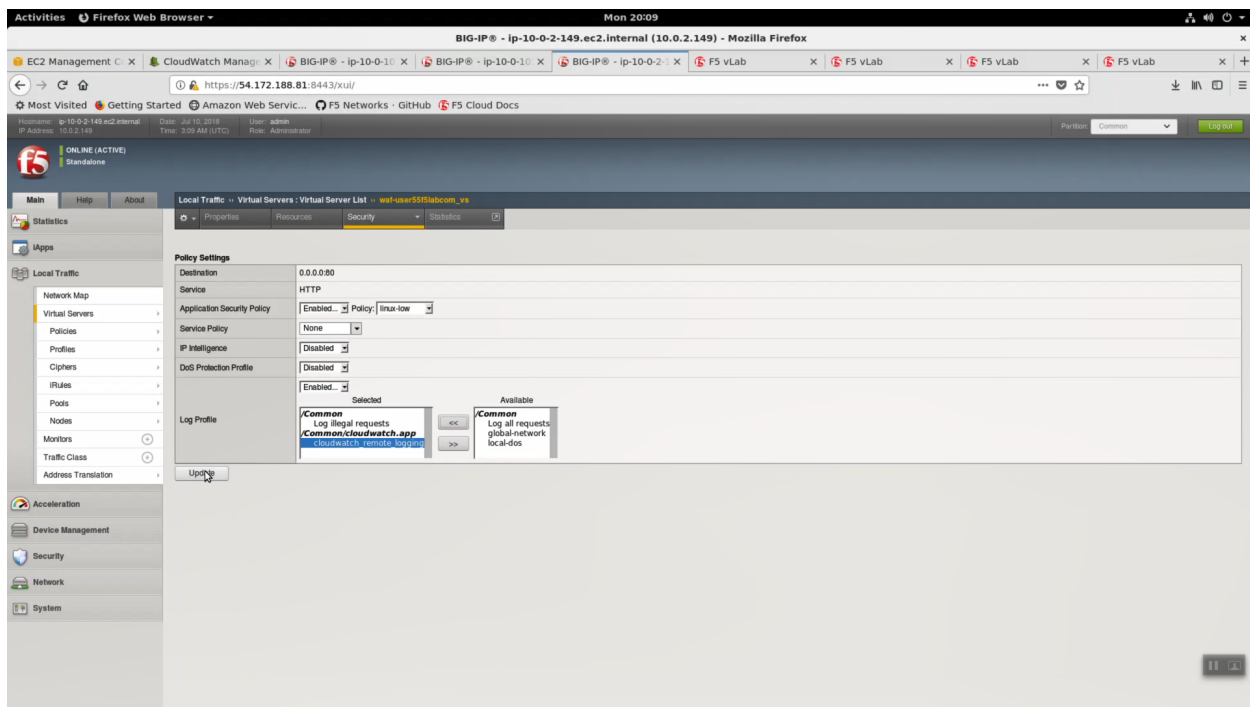
Click [Update].



Local Traffic => Virtual Server => waf-userXXf5labcom_vs => Security => Policies.



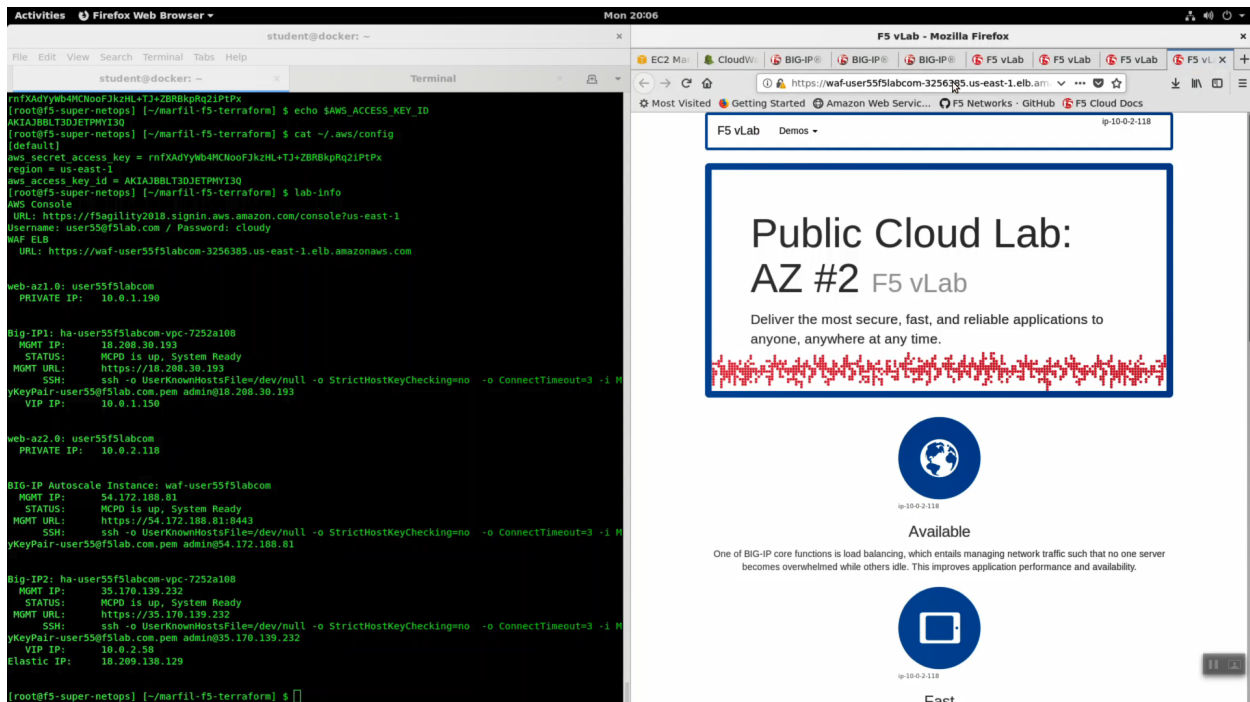
Log Profile. Select *cloudwatch_remote_logging*. Click [Update].



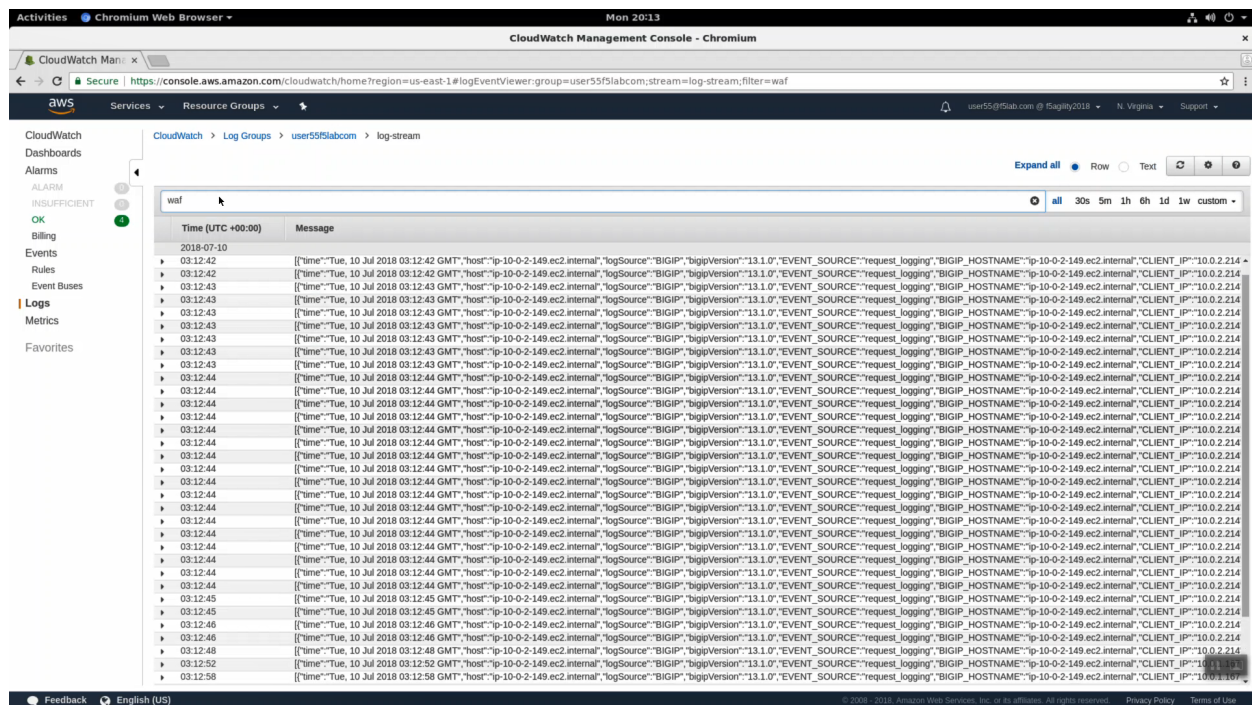
From the Super-NetOps terminal, run the lab-info utility.

```
lab-info
```

HTTPS to the WAF ELB URL. Refresh the browser with <CTRL+F5> for 15 seconds to generate a modest amount of traffic.



Back in the CloudWatch console. Use the search term *waf* to see logs coming from your F5 WAF.

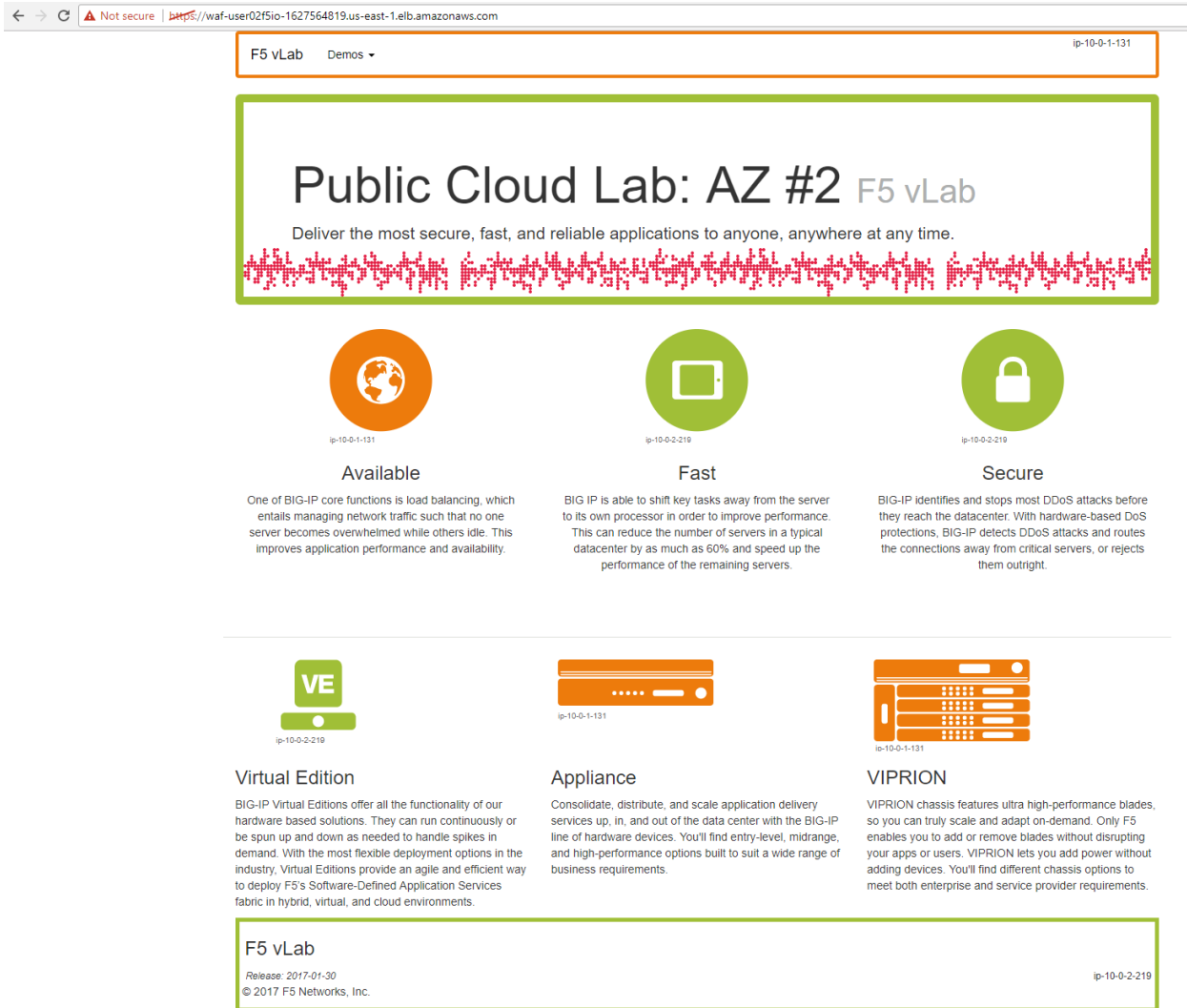


2.6 Autoscale WAF

Automatically scale out your Web Application Firewall to service a surge and scale in when surge subsides.

2.6.1 Autoscale WAF

HTTPS to the WAF ELB URL.



From the AWS console, navigate to Services => AUTO SCALING => Auto Scaling Groups. Filter on your username and select your waf-userxx... auto scaling group.

Select the 'Instances' tab below, and select your Instance ID (there should be only one). If your instance is "Protected from... Scale in" then it will always stay up regardless of scale up/down thresholds configured. It's common to keep a single minimum WAF instance running at all times and scale the 2nd, 3rd, Nth WAF during surges.

Filter:

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace
waf-user02f5io-vc-b7b1c7ce-BigipAutoscaleGroup-8GKCW91ZHJ4M	waf-user02f5io-vc-b7b...	2	1	2	us-east-1a, us-east-1b	1,500	1,500	

Auto Scaling Group: waf-user02f5io-vc-b7b1c7ce-BigipAutoscaleGroup-8GKCW91ZHJ4M

Details | Activity History | Scaling Policies | **Instances** | Monitoring | Notifications | Tags | Scheduled Actions

Filter:

Instance ID	Lifecycle	Launch Configuration Name	Availability Zone	Health Status	Protected from
i-015f46cebf0749b1	Pending	waf-user02f5io-vc-b7b1c7ce-BigipLaunchConfig-STCZT92TTBHC	us-east-1a	Healthy	
i-096736e827c0f02cd	InService	waf-user02f5io-vc-b7b1c7ce-BigipLaunchConfig-STCZT92TTBHC	us-east-1b	Healthy	Scale In

Select the Scaling Policies tab. These policies were deployed via the CloudFormation template and can be changed via the CloudFormation template.

Filter:

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace
waf-user02f5io-vc-b7b1c7ce-BigipAutoscaleGroup-8GKCW91ZHJ4M	waf-user02f5io-vc-b7b...	2	2	1	2	us-east-1a, us-east-1b	1,500	1,500

Auto Scaling Group: waf-user02f5io-vc-b7b1c7ce-BigipAutoscaleGroup-8GKCW91ZHJ4M

Details | Activity History | **Scaling Policies** | Instances | Monitoring | Notifications | Tags | Scheduled Actions

Add policy

waf-user02f5io-vc-b7b1c7ce-BigipScaleDownPolicy-G5JDZJU41J1V

Policy type: Simple scaling

Execute policy when: waf-user02f5io-vc-b7b1c7ce-BigipLowbytesAlarm-3DLRJVDNCXV breaches the alarm threshold: throughput-per-sec < 1000 for 10 consecutive periods of 300 seconds for the metric dimensions

Take the action: Remove 1 instances

And then wait: 1500 seconds before allowing another scaling activity

waf-user02f5io-vc-b7b1c7ce-BigipScaleUpPolicy-42WVPMQZH1G

Policy type: Simple scaling

Execute policy when: waf-user02f5io-vc-b7b1c7ce-BigipHighbytesAlarm-V8V51C7AR4M breaches the alarm threshold: throughput-per-sec > 3500 for 60 seconds for the metric dimensions

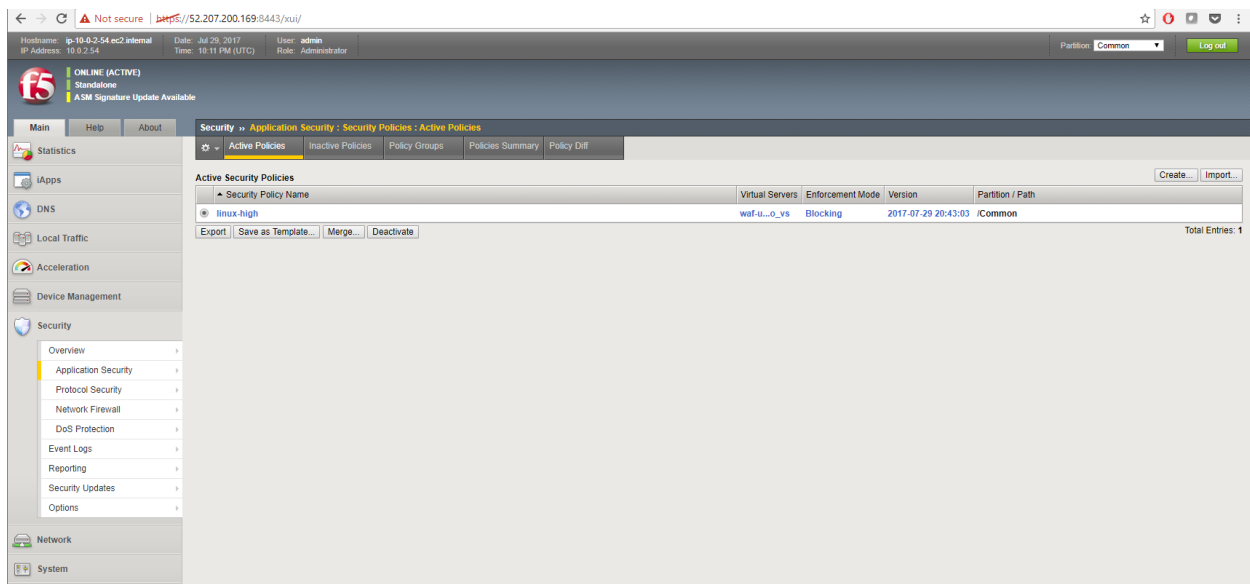
Take the action: Add 1 instances

And then wait: 1500 seconds before allowing another scaling activity

Login to the active BIG-IP Autoscale Instance MGMT IP **on port 8443** configuration utility (web ui).

lab-info

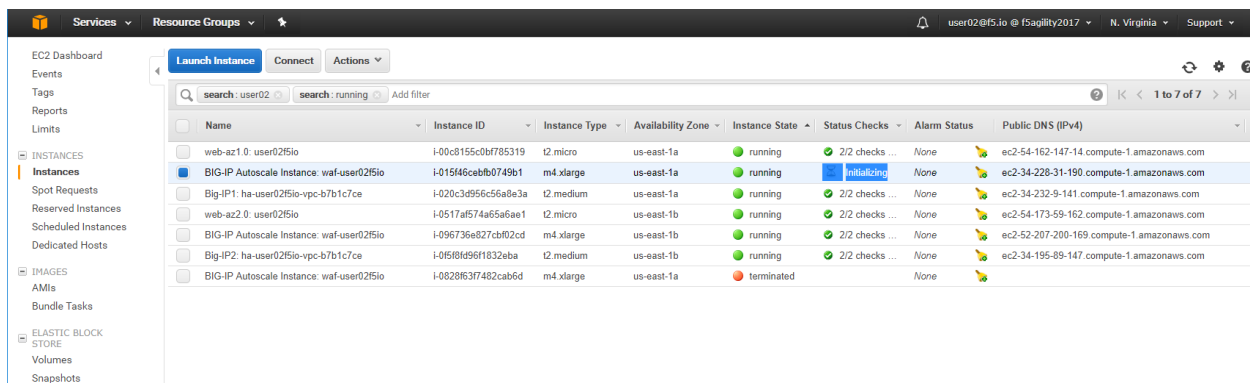
In the Big-IP Configuration utility (Web UI) navigate to Security -> Application Security -> Security Policies -> Active Polices. A "linux-low" policy was deployed via CloudFormation template and is in Enforcement Mode: Blocking.



From the f5-super-netops container, let's launch some traffic against the application behind our WAF and watch it autoscale to service the surge! Replace the <https://waf-userxx...> in the command below with the one in the output of lab-info and don't miss that critical forward slash / at the end!

```
base64 /dev/urandom | head -c 3000 > payload
ab -t 120 -c 200 -c 5 -T 'multipart/form-data; boundary=1234567890' -p payload https://
↪ /waf-user11f5democom-xxxxxxxxx.us-east-1.elb.amazonaws.com/
```

Services => Compute => EC2 => INSTANCES => Instances. Filter on your username and after 60 seconds (the lowest configurable time threshold) hit refresh to see your 2nd autoscale WAF instance starting.



2.7 Clean Up Environment

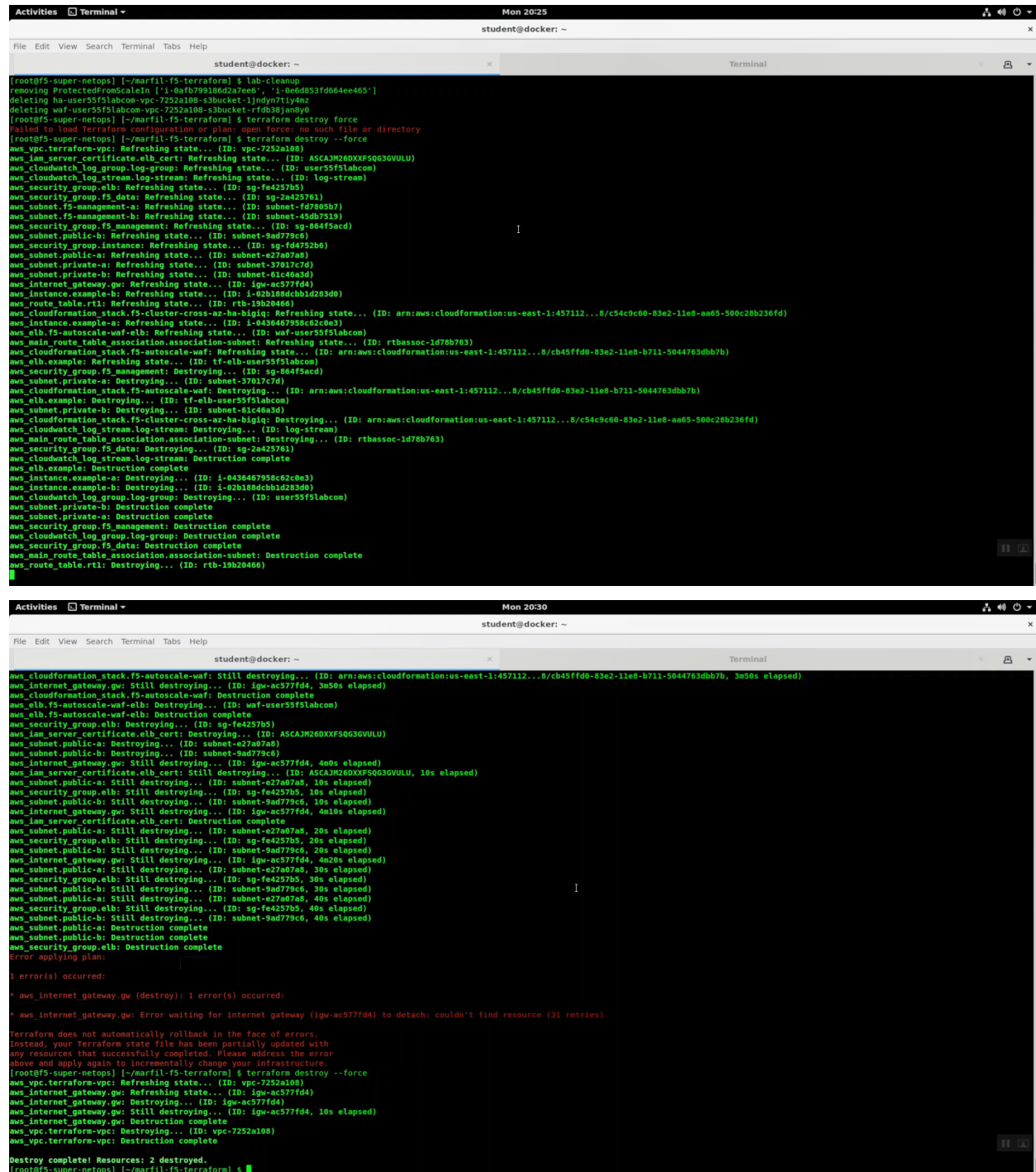
The exciting promise of public cloud is not only to stand up application environments quickly, consistently and with minimum capex, but also the inverse: to tear down application environments quickly, cleanly and completely.

2.7.1 Clean up the lab environment

From the Super-NetOps terminal, clean up, then destroy the environment.


```
lab-cleanup
terraform destroy --force
```

Attention: You might need to run *terraform destroy --force* a second time. Watch the console output. Nothing serious: sometimes the Internet gateways take longer to delete than the time we have configured for terraform to timeout.



```
Mon 20:25
student@docker: ~

[root@fs-super-netops] [/~/marfil-f5-terraform] $ lab-cleanup
removing ProtectedFromScaleIn ['1-0afb799186d2a7ee8', '1-0edd853fd664ee405']
deleting ha-users55flabcom-vpc-7252a108-s3bucket-1jndyn71iyamz
deleting waf-user55flabcom-vpc-7252a108-s3bucket-rf0b3bjmny0
[root@fs-super-netops] [/~/marfil-f5-terraform] $ terraform destroy --force
Failed to load Terraform configuration or plan: open force: no such file or directory
[root@fs-super-netops] [/~/marfil-f5-terraform] $ terraform destroy --force
aws_vpc.terraform-vpc: Refreshing state... (ID: vpc-7252a108)
aws_iam_server_certificate.elb_cert: Refreshing state... (ID: ASCAJM26DXXFSQ63GVULU)
aws_cloudwatch_log_group.log-group: Refreshing state... (ID: users55flabcom)
aws_cloudwatch_log_stream.log-stream: Refreshing state... (ID: log-stream)
aws_security_group.elb: Refreshing state... (ID: sg-fe4257b5)
aws_security_group.fs_data: Refreshing state... (ID: sg-2a425761)
aws_subnet.f5-management-a: Refreshing state... (ID: subnet-fd7805b7)
aws_subnet.f5-management-b: Refreshing state... (ID: subnet-45d07819)
aws_security_group.f5_management: Refreshing state... (ID: sg-864f5acd)
aws_subnet.public-b: Refreshing state... (ID: subnet-9ad779c8)
aws_security_group.instance: Refreshing state... (ID: sg-fd4752b6)
aws_subnet.public-a: Refreshing state... (ID: subnet-e27a07a8)
aws_subnet.private-a: Refreshing state... (ID: subnet-37017c7d)
aws_subnet.private-b: Refreshing state... (ID: subnet-61c4a3d)
aws_internet_gateway.gw: Refreshing state... (ID: igw-ac577fd4)
aws_instance.example-b: Refreshing state... (ID: i-02b188dcbbid283d0)
aws_route_table.rtl: Refreshing state... (ID: rtb-19b28466)
aws_cloudformation_stack.f5-cluster-cross-az-ha-high: Refreshing state... (ID: arn:aws:cloudformation:us-east-1:457112.../c/c54c9c60-83e2-11e8-a665-500c28b236fd)
aws_instance.example-a: Refreshing state... (ID: i-0436467958c52ce83)
aws_elb.f5-autoscale-waf-elb: Refreshing state... (ID: waf-user55flabcom)
aws_main_route_table_association.association-subnet: Refreshing state... (ID: rtbassoc-1d78b763)
aws_cloudformation_stack.f5-autoscale-waf: Refreshing state... (ID: arn:aws:cloudformation:us-east-1:457112.../c/cb45ff00-83e2-11e8-b711-5044763dbb7b)
aws_elb.example: Refreshing state... (ID: elb-user55flabcom)
aws_security_group.f5_management: Destroying... (ID: sg-864f5acd)
aws_subnet.private-a: Destroying... (ID: subnet-37017c7d)
aws_cloudformation_stack.f5-autoscale-waf: Destroying... (ID: arn:aws:cloudformation:us-east-1:457112.../c/cb45ff00-83e2-11e8-b711-5044763dbb7b)
aws_elb.example: Destroying... (ID: elb-user55flabcom)
aws_subnet.private-b: Destroying... (ID: subnet-61c4a3d)
aws_cloudformation_stack.f5-cluster-cross-az-ha-high: Destroying... (ID: arn:aws:cloudformation:us-east-1:457112.../c/c54c9c60-83e2-11e8-a665-500c28b236fd)
aws_cloudwatch_log_stream.log-stream: Destroying... (ID: log-stream)
aws_main_route_table_association.association-subnet: Destroying... (ID: rtbassoc-1d78b763)
aws_security_group.fs_data: Destroying... (ID: sg-2a425761)
aws_cloudwatch_log_stream.log-stream: Destruction complete
aws_elb.example: Destruction complete
aws_instance.example-a: Destroying... (ID: i-0436467958c52ce83)
aws_instance.example-b: Destroying... (ID: i-02b188dcbbid283d0)
aws_cloudwatch_log_group.log-group: Destroying... (ID: users55flabcom)
aws_subnet.private-b: Destruction complete
aws_subnet.private-a: Destruction complete
aws_security_group.f5_management: Destruction complete
aws_cloudwatch_log_group.log-group: Destruction complete
aws_security_group.fs_data: Destruction complete
aws_main_route_table_association.association-subnet: Destruction complete
aws_route_table.rtl: Destroying... (ID: rtb-19b28466)

aws_cloudformation_stack.f5-autoscale-waf: Still destroying... (ID: arn:aws:cloudformation:us-east-1:457112.../c/cb45ff00-83e2-11e8-b711-5044763dbb7b, 3m50s elapsed)
aws_internet_gateway.gw: Still destroying... (ID: igw-ac577fd4, 3m50s elapsed)
aws_cloudformation_stack.f5-autoscale-waf: Destruction complete
aws_elb.f5-autoscale-waf-elb: Destroying... (ID: waf-user55flabcom)
aws_elb.f5-autoscale-waf-elb: Destruction complete
aws_security_group.elb: Destroying... (ID: sg-fe4257b5)
aws_iam_server_certificate.elb_cert: Destroying... (ID: ASCAJM26DXXFSQ63GVULU)
aws_subnet.public-a: Destroying... (ID: subnet-e27a07a8)
aws_subnet.public-b: Destroying... (ID: subnet-9ad779c8)
aws_internet_gateway.gw: Still destroying... (ID: igw-ac577fd4, 4m0s elapsed)
aws_iam_server_certificate.elb_cert: Still destroying... (ID: ASCAJM26DXXFSQ63GVULU, 10s elapsed)
aws_subnet.public-a: Still destroying... (ID: subnet-e27a07a8, 10s elapsed)
aws_subnet.public-b: Still destroying... (ID: subnet-9ad779c8, 10s elapsed)
aws_security_group.elb: Still destroying... (ID: sg-fe4257b5, 10s elapsed)
aws_subnet.private-a: Still destroying... (ID: subnet-37017c7d, 4m10s elapsed)
aws_internet_gateway.gw: Still destroying... (ID: igw-ac577fd4, 4m10s elapsed)
aws_iam_server_certificate.elb_cert: Destruction complete
aws_subnet.public-a: Still destroying... (ID: subnet-e27a07a8, 20s elapsed)
aws_security_group.elb: Still destroying... (ID: sg-fe4257b5, 20s elapsed)
aws_subnet.public-b: Still destroying... (ID: subnet-9ad779c8, 20s elapsed)
aws_internet_gateway.gw: Still destroying... (ID: igw-ac577fd4, 4m20s elapsed)
aws_subnet.private-b: Still destroying... (ID: subnet-61c4a3d, 4m20s elapsed)
aws_security_group.elb: Still destroying... (ID: sg-fe4257b5, 30s elapsed)
aws_subnet.public-b: Still destroying... (ID: subnet-9ad779c8, 30s elapsed)
aws_subnet.private-a: Still destroying... (ID: subnet-e27a07a8, 40s elapsed)
aws_security_group.elb: Still destroying... (ID: sg-fe4257b5, 40s elapsed)
aws_subnet.private-b: Still destroying... (ID: subnet-9ad779c8, 40s elapsed)
aws_subnet.public-a: Destruction complete
aws_subnet.public-b: Destruction complete
aws_security_group.elb: Destruction complete
Error applying plan:

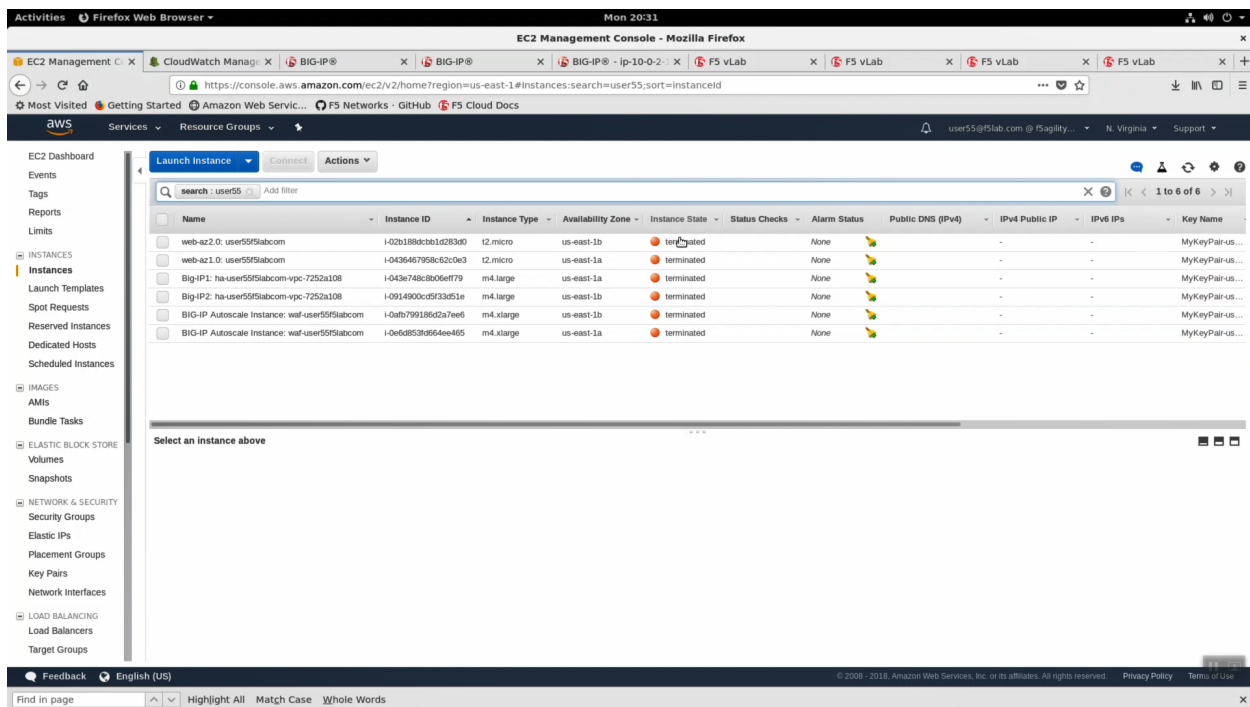
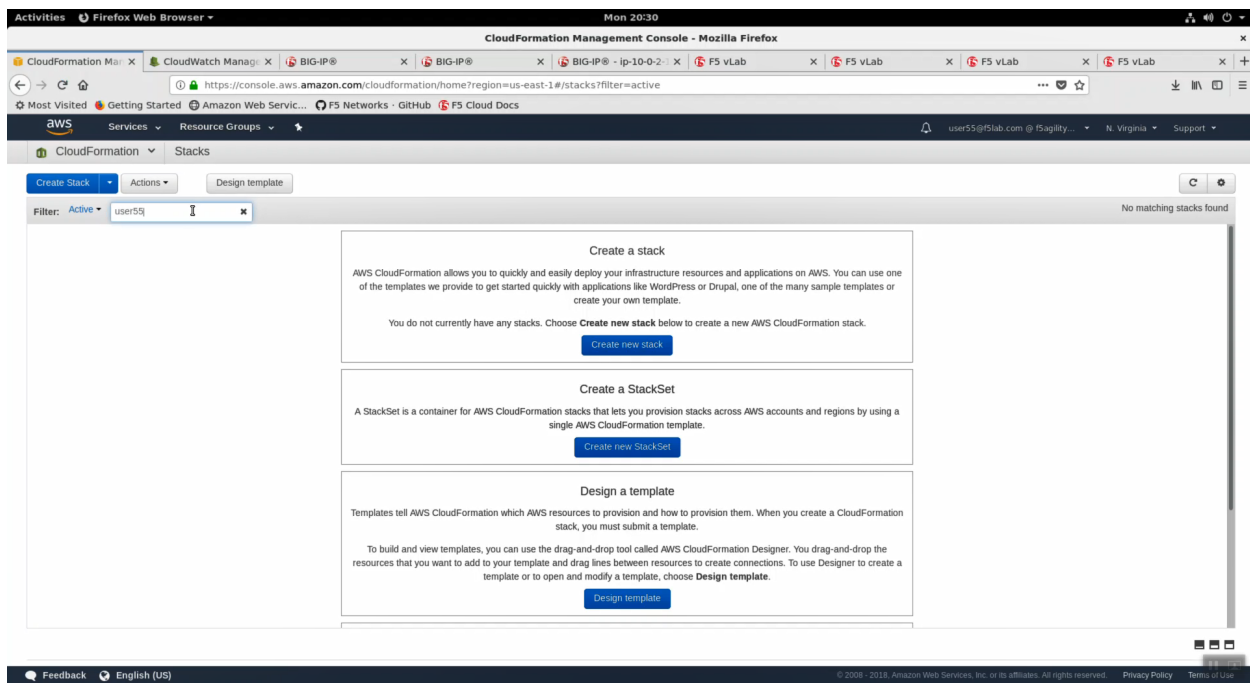
1 error(s) occurred:

* aws_internet_gateway.gw (destroy): 1 error(s) occurred:

* aws_internet_gateway.gw: Error waiting for internet gateway (igw-ac577fd4) to detach: couldn't find resource (31 retries)

Terraform does not automatically rollback in the face of errors.
Instead, your Terraform state file has been partially updated with
any resources that successfully completed. Please address the error
above and apply again to incrementally change your infrastructure.
[root@fs-super-netops] [/~/marfil-f5-terraform] $ terraform destroy --force
aws_vpc.terraform-vpc: Refreshing state... (ID: vpc-7252a108)
aws_internet_gateway.gw: Refreshing state... (ID: igw-ac577fd4)
aws_internet_gateway.gw: Destroying... (ID: igw-ac577fd4)
aws_internet_gateway.gw: Still destroying... (ID: igw-ac577fd4, 10s elapsed)
aws_internet_gateway.gw: Destruction complete
aws_vpc.terraform-vpc: Destroying... (ID: vpc-7252a108)
aws_vpc.terraform-vpc: Destruction complete

Destroy complete! Resources: 2 destroyed.
[root@fs-super-netops] [/~/marfil-f5-terraform] $
```



WE MAKE APPS  FASTER.
SMARTER.
SAFER.

F5 Networks, Inc. | f5.com



US Headquarters: 401 Elliott Ave W, Seattle, WA 98119 | 888-882-4447 // Americas: info@f5.com // Asia-Pacific: apacinfo@f5.com // Europe/Middle East/Africa: emeainfo@f5.com // Japan: f5j-info@f5.com
©2017 F5 Networks, Inc. All rights reserved. F5, F5 Networks, and the F5 logo are trademarks of F5 Networks, Inc. in the U.S. and in certain other countries. Other F5 trademarks are identified at f5.com. Any other products, services, or company names referenced herein may be trademarks of their respective owners with no endorsement or affiliation, express or implied, claimed by F5. These training materials and documentation are F5 Confidential Information and are subject to the F5 Networks Reseller Agreement. You may not share these training materials and documentation with any third party without the express written permission of F5.