**Last Revised on June 9th, 2017**

**Version 1.0**



Ansible Orchestration for

F5 LTM and GSLB

**Document Owners:**

**Mark Lowcher**

**AT&T Labs**

**200 Laurel Avenue South**

**Middletown, NJ 07748**

Contents

[1 Revision History/Document Control 2-2](#_Toc484782660)

[1.1 Review / Retention Cycle: 2-2](#_Toc484782661)

[1.2 Updates 2-3](#_Toc484782662)

[1.3 Change History Detail: 2-3](#_Toc484782663)

[1.4 Purpose: 2-3](#_Toc484782664)

[1.5 Using Departments 2-3](#_Toc484782665)

[1.6 Distribution 2-3](#_Toc484782666)

[2 Executive Summary 2-3](#_Toc484782667)

[2.1 Introduction 2-3](#_Toc484782668)

[2.2 Test Overview 2-3](#_Toc484782669)

[2.3 Scope 2-3](#_Toc484782670)

[3 Ansible Installation 2-4](#_Toc484782671)

[3.1 Install Openssl Server 2-4](#_Toc484782672)

[3.2 Update APT 2-4](#_Toc484782673)

[3.3 Install Ansible and Dependencies 2-4](#_Toc484782674)

[4 Ansible Orchestration 2-5](#_Toc484782675)

[4.1 Setting up the Connectivity between Ansible and the F5s. 2-5](#_Toc484782676)

[4.2 Ansible Setup Playbook 2-6](#_Toc484782677)

[4.3 Ansible Methods 2-6](#_Toc484782678)

[4.4 The Ansible Playbooks 2-6](#_Toc484782679)

[4.5 Licensing the F5 devices 2-7](#_Toc484782680)

[4.6 Disable the Setup Wizard 2-7](#_Toc484782681)

[4.7 Provision the Desired F5 Modules 2-8](#_Toc484782682)

[4.8 Setting Up the HA Configuration 2-8](#_Toc484782683)

[5 System Configuration 2-10](#_Toc484782684)

[5.1 Changing System Preferences 2-10](#_Toc484782685)

[5.2 Adding an Advisory Banner 2-11](#_Toc484782686)

[5.3 Updating the Geolocation Database 2-11](#_Toc484782687)

[6 Creating Profiles for Applications 2-12](#_Toc484782688)

[7 Creating Monitors for Applications 2-13](#_Toc484782689)

[8 Adding Applications to the Active/Standby Pair 2-13](#_Toc484782690)

[8.1 Creating non SSL/TLS Applications 2-13](#_Toc484782691)

[8.2 Creating a Wildcard IP-Forwarding Virtual 2-14](#_Toc484782692)

[9 Configuring F5 DNS Module for GSLB 2-14](#_Toc484782693)

Revision History/Document Control

Review / Retention Cycle:

This document will be reviewed and updated as needed. The record retention period for this document is “until superseded”. This document will be located at: <https://github.com/mlowcher61> in the documentation folder.

Updates

Updates to this document must be approved by the following: Mark Lowcher

Change History Detail:

|  |  |  |  |
| --- | --- | --- | --- |
| Revision Number | Date | Approver |  |
| Version 1.0 | 06/7/2017 | Mark Lowcher |  |

Note: This document was printed from an online system and must be used for reference purposes only. The online document is the current valid document. It is the user’s responsibility to ensure that they are using the latest copy, as found in https://github.com/mlowcher61

Purpose:

This document was created to show how to get Ansible on Ubuntu working with F5 modules and uri ReST commands.

Distribution

The following are on distribution for this document, and will be copied as part of the document review prior to publication.

|  |  |
| --- | --- |
| Name | E-Mail ID |
| Mark Lowcher | m.lowcher@f5.com |
|  |  |

# *Executive Summary*

Introduction

This document describes how to install Ansible 2.3 on Ubuntu 16.04 .

Test Overview

All Ansible playbooks located at <https://github.com/mlowcher61> were tested on F5 version 12.1.2. They should also work on 11.6 but that was not tested.

Scope

The scope for this test is limited to specified features and functionality of F5 devices as outlined in this document.

Ansible Installation

Ansible must be installed on a Linux platform. We will spin up a new Ubuntu 16.04 within a hypervisor and run the following commands to get Ansible installed with all its dependencies. During initial setup of the Linux, create user ansible with your desired password.

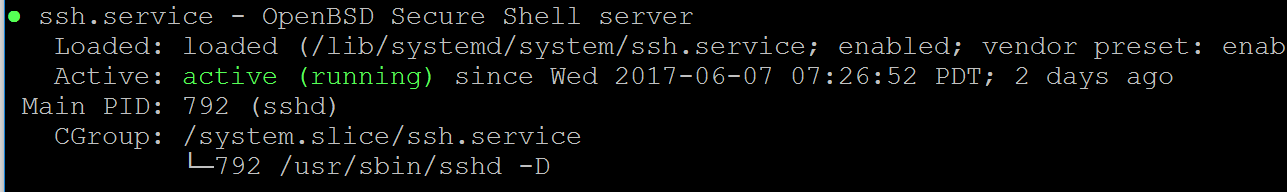
Once the Ubuntu instance has booted we will need to install openssl server from the hypervisor console so that we can access the Linux instance from an SSH session.

Install Openssl Server

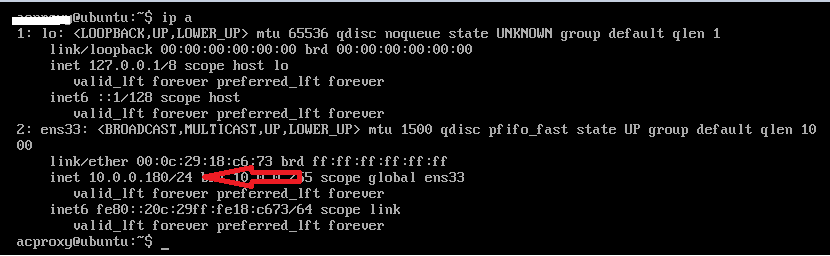
Run “sudo apt-get install openssh-server”

After the openssl server installation is complete, check the SSH status.

Run “sudo service ssh status” It should show as “active”



From the Linux console run “ip a” and determine what ip address was given through DHCP. You may want to shut down the Linux instance and put the network adapter into “bridged” mode.



Update APT

Run “sudo apt-get update”

Install Ansible and Dependencies

Perform the following from the home directory:

Run “cd ~”

Run “sudo apt-get install software-properties-common”

Run “sudo apt-add-repository ppa:ansible/ansible”

Run “sudo apt-get install python-pip”

Run “sudo pip install ansible”

Run “sudo pip install bigsuds

Run “sudo pip install f5-sdk

Run “sudo mkdir /usr/share/ansible”

Run “sudo chown ansible /usr/share/ansible”

Run “sudo chgrp ansible /usr/share/ansible”

Run “sudo mkdir ansible”

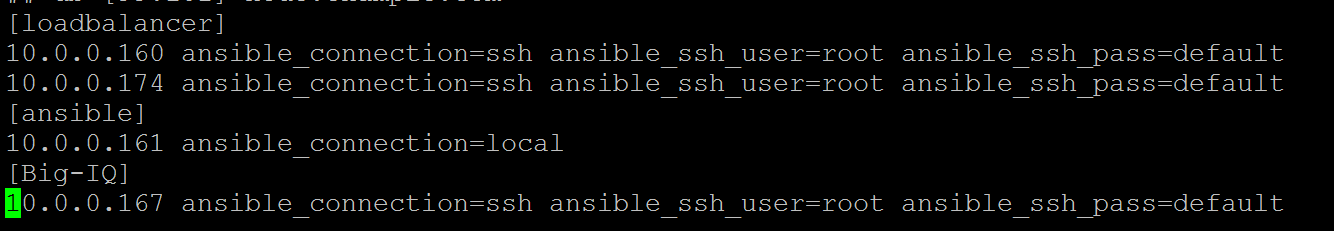
Run “sudo mkdir ./ansible/playbooks

Ansible Orchestration

Setting up the Connectivity between Ansible and the F5s.

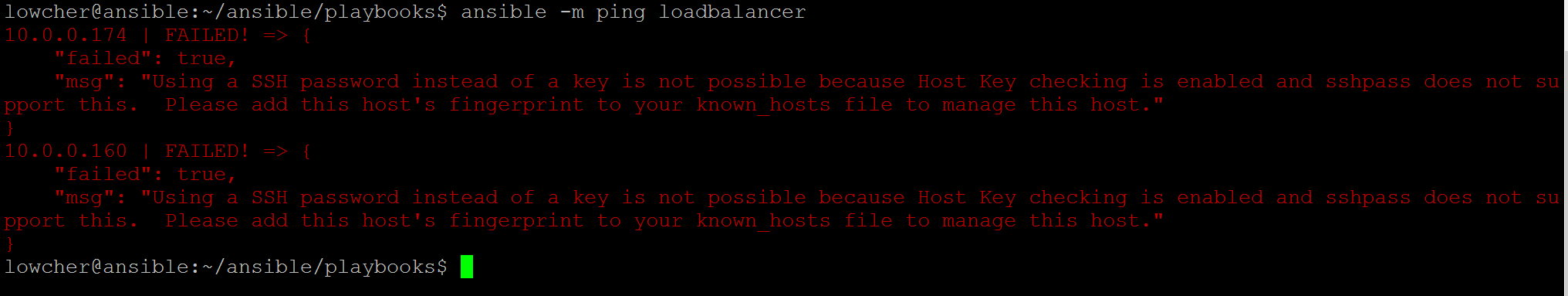
Now that Ansible is installed and we know the management ip address. We will configure Ansible for use.

First edit the /etc/ansible/hosts file and add your loadbalancer and ansible ip addresses and connection method. The example below shows setting up the environment with the default root user and default password of default.



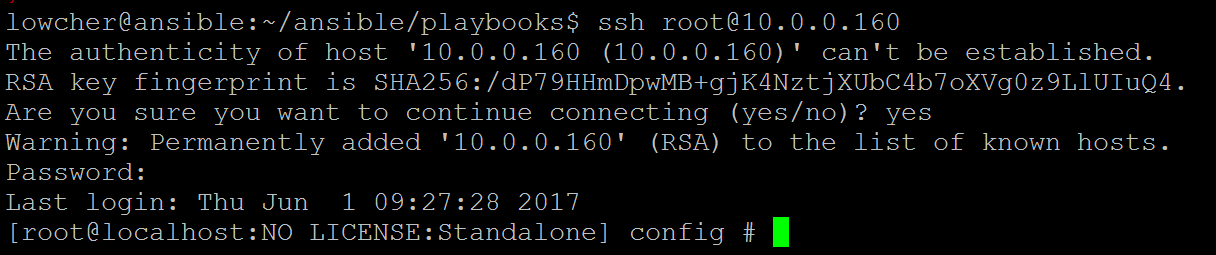
In our Ansible playbooks we will be able to target the individual loadbalancers by using “Loadbalancer[0]” for the .160 and loadbalancer[1] for the .174.

Test connectivity to the F5s using the ansible -m ping command

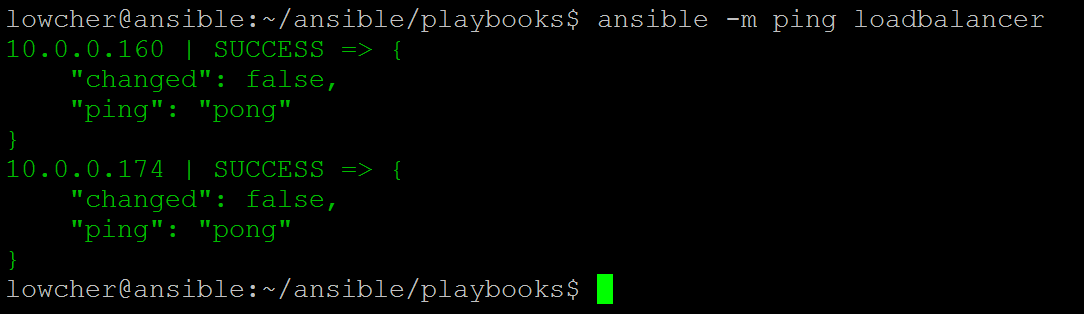


Since the Ansible workstation has never connected to the F5s, this will fail.

From Ansible, ssh as root to each of your loadbalancers with password “default” and type yes to continue.



Now retry the ansible -m ping loadbalancer command.



Now Ansible can communicate with the F5s.

Ansible Setup Playbook

From the <https://github.com/mlowcher61> site download all the Ansible playbooks located under f5-Ansible.

Upload all the playbooks to the ~/ansible/playbooks directory.

Now run the f5\_ansible\_setup.yml playbook by running the following command.

Run “ansible-playbook f5\_ansible\_setup.yml.

Ansible is now ready to be used.

Ansible Methods

I will be showing three different ways to use Ansible playbooks.

* Using F5 supported Ansible modules which are available at <https://github.com/f5Networks/f5-ansible>
* Using uri ReST commands
* Using native tmsh commands from within Ansible

Best practice is to use the F5 supported modules. If you run into tasks that don’t currently have F5 modules, you can get the job done using one of the other methods. I will be using a mix of the three so that you can get an understanding of all of them.

The Ansible Playbooks

Ansible playbooks can be individual tasks or all task can be combined into a single playbook. I have broken the tasks into Individual playbooks. This was done so that progress of the configuration can be verified before moving on to next steps. The following playbooks should now be in your ~/ansible/playbooks directory.

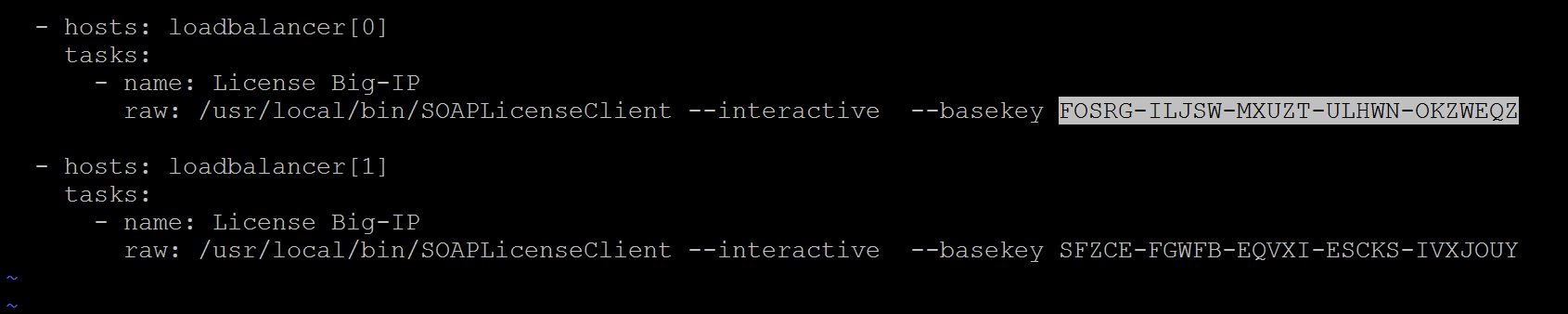
The playbooks we will be using are:

1. License\_f5\_pair.yml – this will remotely license both F5s in the active/standby cluster.
2. setup\_disable.yml – Since we won’t be using the GUI for configuration we want to disable the setup wizard. This playbook will disable the setup wizard on both devices
3. f5\_provision.yml – this playbook will provision the desired modules for your deployment. You will need to edit this playbook as needed.
4. f5\_HA.yml – this playbook will perform the network and HA configuration for the two devices. If you will be aggregating interfaces (Etherchannel) for your deployment, use the f5\_HA\_trunks.yml instead of the f5-HA.yml
5. f5\_preferences – this will setup global preferences for the F5 device. Open and read the playbook to understand the changes that it will make.
6. f5\_advisory\_banner.yml – This will add a system message to the GUI. This is good for alerting other admins that the F5 device is going into a maintenance window. It can be turned off by running the f5\_disable\_advisory.yml playbook.
7. Update\_geo\_db.yml – will update the geolocation database. Open the playbook to learn how to download and stage the appropriate .RPM files.

Licensing the F5 devices

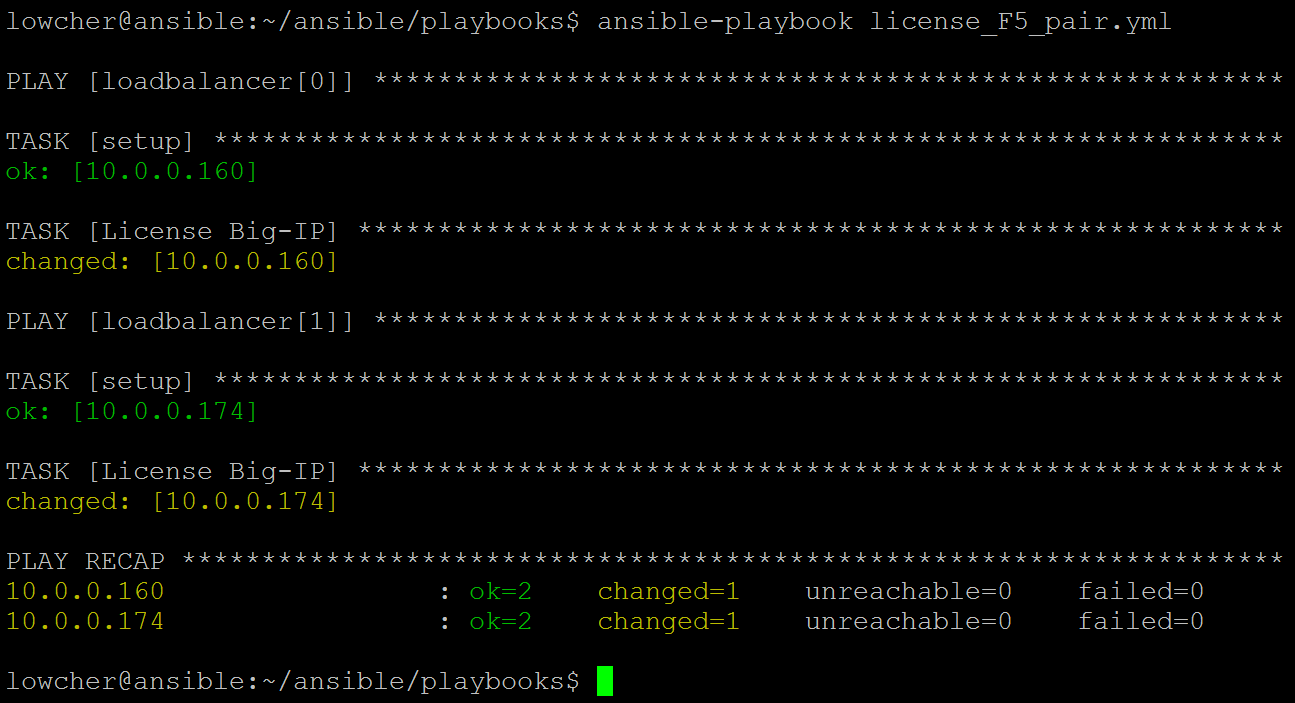
Edit the “license\_f5\_pair.yml playbook and change the existing registration keys to your registration keys. (WARNING: This playbook will not work unless the F5s have access to the internet)

From your F5s, ping an external ip such as 8.8.8.8 and verify the F5s can get to the internet.



Now run the license\_f5\_pair.yml playbook by using the following syntax.

Ansible-playbook license\_f5\_pair.yml



The two F5s at .160 and .174 are now licensed. It will take a minute for the processes to restart.

Disable the Setup Wizard

Now edit the host ip addresses in the setup\_disable.yml playbook. Change them to the management ip addresses for the two F5 devices.

Now run the ansible playbook with the “ansible-playbook setup\_disable.yml” command.



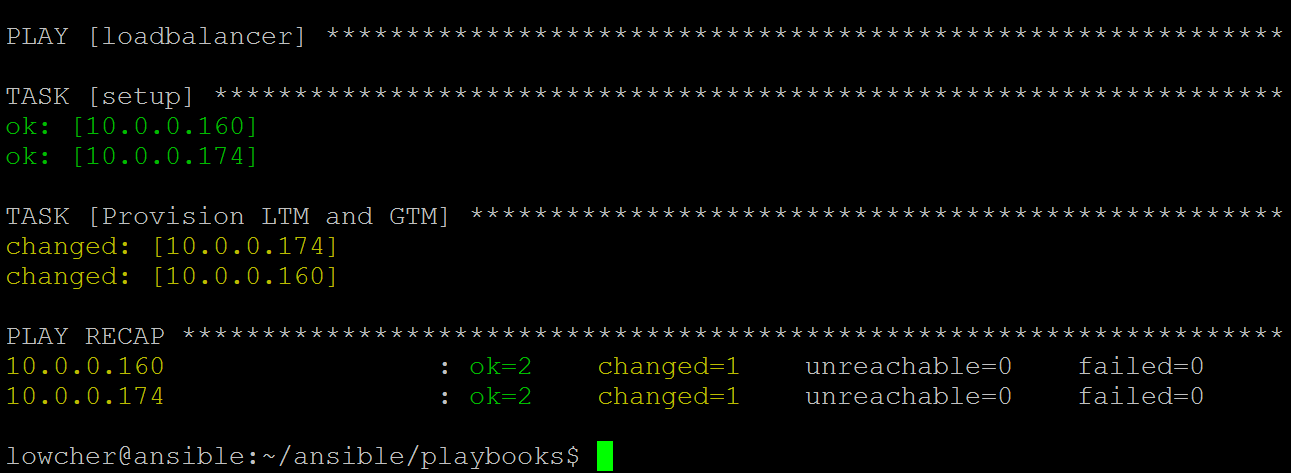
Provision the Desired F5 Modules

We will now provision the appropriate modules.

Open and look at the f5\_provision\_LTM\_GTM\_tmsh.yml playbook. You will see that I am using native tmsh commands in this instance. This is done using the Ansible “raw” command. This method calls the “Loadbalancer” from the /etc/ansible/hosts file. If you only needed to provision the first device you would call it using “Loadbalancer[0]”. In our case we will be provisioning both devices at the same time so we are calling out “Loadbalancer”.

You could also do provisioning using the bigip\_command module to issue tmsh commands or there is a bigip\_provision module from F5 which is currently in “preview” and requires Ansible version 2.3 or greater. That’s why it’s good to have more than one way of doing things. If an F5 Ansible module hasn’t been developed yet, you can still get the task done using either the uri ReST method or the “raw” tmsh commands.

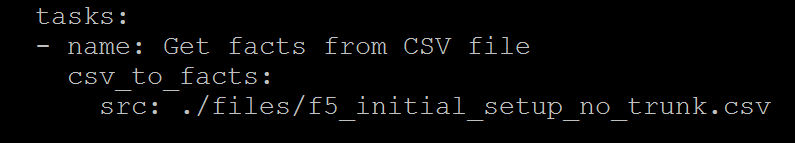
Now run the f5\_provision\_LTM\_GTM\_tmsh.yml playbook.



Setting Up the HA Configuration

Now that we have the device licensed and provisioned we will get the systems to a full network configuration with HA set up as active/standby. The pair will then be ready to have applications configured on one unit and pushed to the other using configsync.

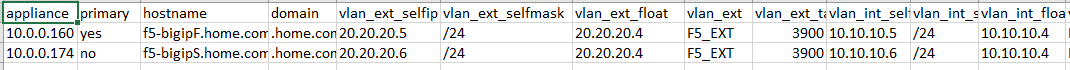
The next playbook we will look at is the f5\_HA.yml. On line 22 of the playbook we see the following



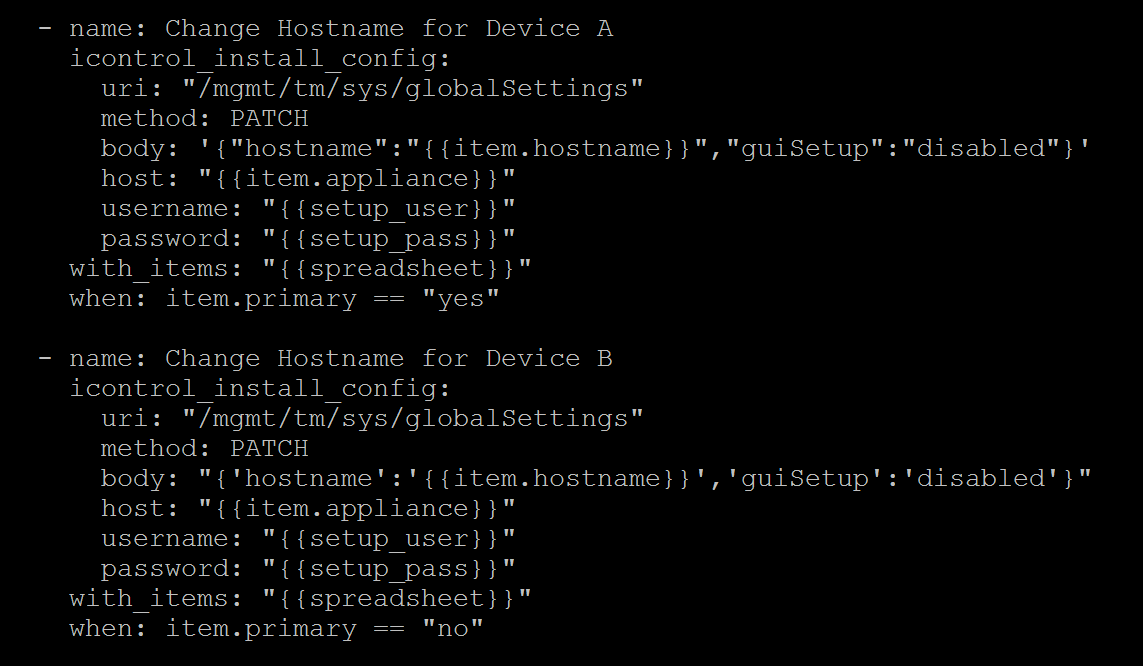
Within this playbook we will be using variables that are in a .csv file (spreadsheet). The spreadsheet should be in a folder named “files” under the “playbook” directory. If you are not using trunks in your configuration, only one interface per vlan, use the f5\_HA.yml playbook with the f5\_initial\_setup\_no\_trunk.csv file.

If you will be using multiple interfaces per vlan use the f5\_HA\_trunk.yml with the f5\_initial\_setup.csv file.

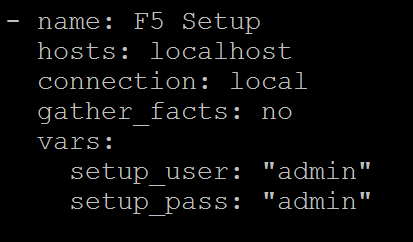
Variables called within the playbook are called in this format “{{items.appliance}}” from “with\_items: “{{spreadsheet}}”.



Because we are dealing with two devices, I am using the item “primary” to differentiate between the two devices. If you look at the playbook you can see I use “when: item.primary == "yes"” to call our the device that has “yes” in the “primary” column and I use “no” to call out the other. The snippet above does not show all the variables that are in the .csv file. Open the file and edit accordingly.



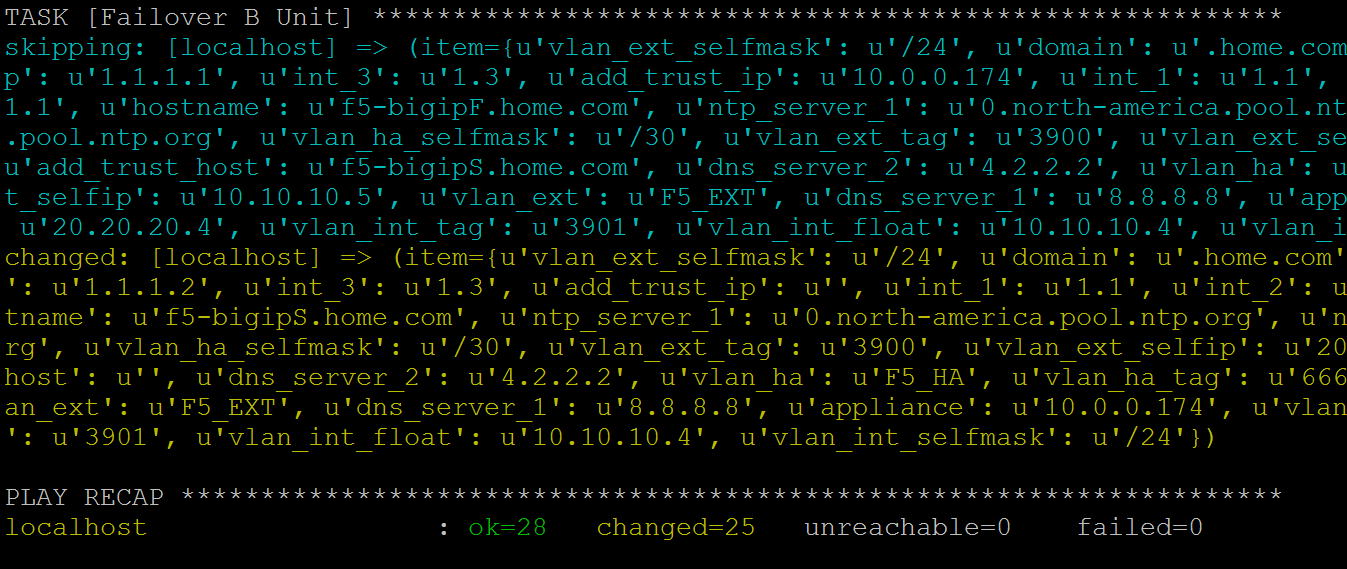
You will also need to change the Admin password at line 20 from the default of admin to the management password of the two devices.



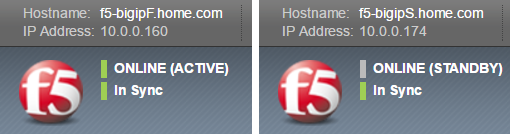
After you have gone through the f5\_initial\_setup\_no\_trunk.csv file and populated with your variables, run the f5\_HA.yml playbook and it will do the following.

* Change the hostname of both f5s to the appropriate FQDN as listed in the .csv file.
* Create 3 vlans named External, Internal and HA
* Assign the vlans to physical interfaces. (1.1, 1.2 etc…)
* Create self and floating self ips for the External and Internal vlans and only non-floating for the HA.
* Define the DNS, NTP and Syslog servers to be used by the F5s
* Setup the ip addresses to use for ConfigSync, HA and connection mirroring.
* Set up the device service cluster between the two F5s.
* Perform the initial ConfigSync between the devices.

The playbook should complete with 25 changes and no errors displayed at the bottom of the output.



Now we have an active/standby cluster that is in sync and ready for application configuration on one unit, which is then config-synced to the other unit.



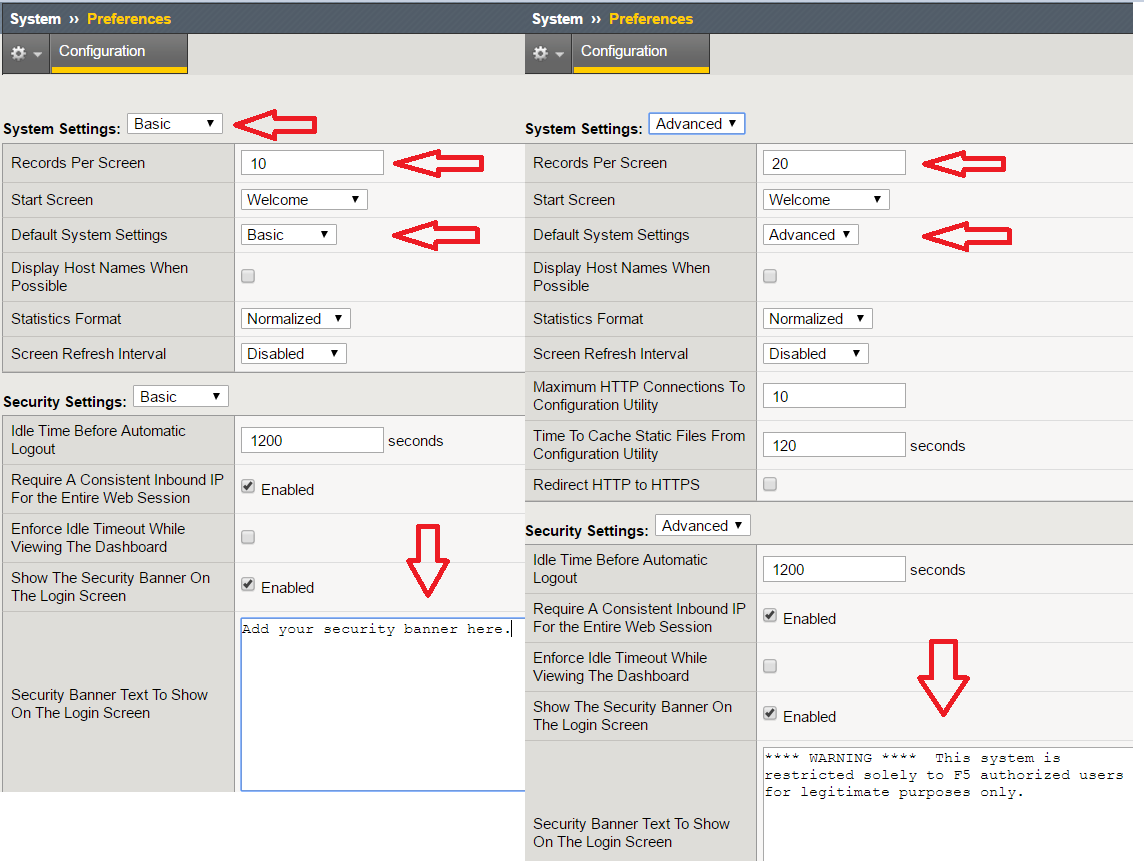
System Configuration

Changing System Preferences

The f5\_HA playbook setup DNS, NTP and Syslog. We will now run the f5\_preferences playbook to change the number of items shown in the GUI from the default of 10 to 20. It will also change default view from “Basic” to “Advanced” for all pages navigated to in the GUI. Lastly it will insert a security banner at the GUI logon screen.

Edit the security banner to display your desired message. Also, make sure the “appliance” line has the correct management ip address.

After running this playbook, you can click on the red F5 ball in the GUI which will take you to the welcome screen. From there click on “System preferences” and notice the changes.



Adding an Advisory Banner

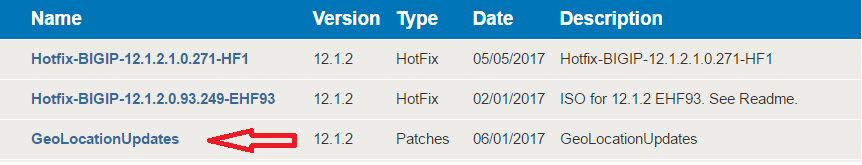
To alert Administrators to maintenance windows you can enable a banner that will alert users in the GUI that the system will be in a maintenance window. All configuration changes can be made while the banner is present. After the maintenance window, you remove the advisory banner.

The playbook named “f5\_advisory\_banner.yml” will turn the banner on and the f5\_disable\_advisory.yml will remove the banner. The f5\_advisory\_banner will need to be edited to display your preference of message and message color. Both the advisory and disable playbooks need to have the management ip address of the f5 device set to the correct ip address.

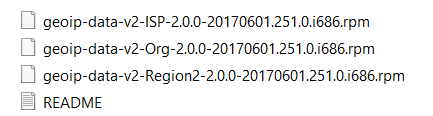
Updating the Geolocation Database

All F5 products ship with a Geolocation database that allow the policies and irules to determine the location of a source ip address. This makes it possible to for example to block clients from a particular country when using the firewall features or to create topologies for GSLB (Global Server Load Balancing using the F5 DNS module formerly known as GTM. For example, we could create a GSLB topology policy that would send clients from west of the Mississippi to a California datacenter while sending clients coming in from east of the Mississippi to Secaucus NJ.

To update this database, you need to download the current Geolocation database from the f5 download site. <https://downloads.f5.com>. Navigate the desired code level and download the geo update files.

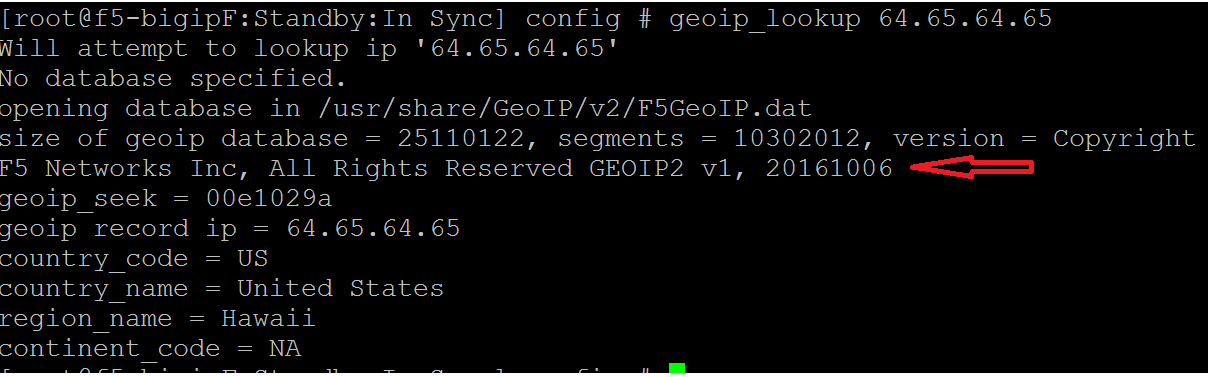


The geo download will contain 3 .rpm files. Similar to the following



You will need to edit the “update\_geo\_db.yml” playbook to reflect the name changes of the new files. This playbook will push the files from Ansible to the F5 devices and then install them. You will also need to edit the file paths in the playbook.

After running the playbook, you can verify that the geolocation database has been updated by running the following command at the bash CLI. The date by the red arrow should match the date within the rpm name.



Creating Profiles for Applications

Virtual Servers use profiles to perform various functions. Profiles are the building blocks for defining how your application should be processed and what services to provide. There are:

* Protocol profiles to describe how to process protocols
* Persistence profiles to describe how to persist to a specific server
* SSL/TLS profiles to configure how to terminate and/or encrypt traffic

And many more.

F5 devices come with default profiles. It is a best practice to not directly use the default profiles but to rather create child profiles based on the default profiles. A good example of why you would want to do this would be, if you use the same default http profile for 1000 vips and then it becomes necessary to make a change in that profile to accommodate a particular application. You have now made that change to all 1000 applications. By creating a new child profile for each application, you can make a change to just one application. Or if there is a need to make that change to all applications, you can make the change on the parent profile.

In the playbook bundle are playbooks for creating various child profiles. Be sure to change the management ip to reflect your environment and change the name of the child profile that you will be creating to the desired name.

Here is the current list of profile playbooks.

* Child\_tcp\_profile
* Child\_udp\_profile
* child\_http\_profile
* push\_cert\_key\_to\_F5
* create\_clientssl\_profile
* child\_cookie\_persist
* child\_ssl\_persist

Creating Monitors for Applications

Monitors are used to perform health checks on pool members. You should create child monitors for every application for the same reason you do so for profiles. The list of current child monitor playbooks are as follows.

* child\_tcp\_monitor
* child\_http\_monitor

Adding Applications to the Active/Standby Pair

Now we can create an application on one device and push the configuration to the other device. It doesn’t matter which is which. You can create on the active device and push to the standby device or the other way around.

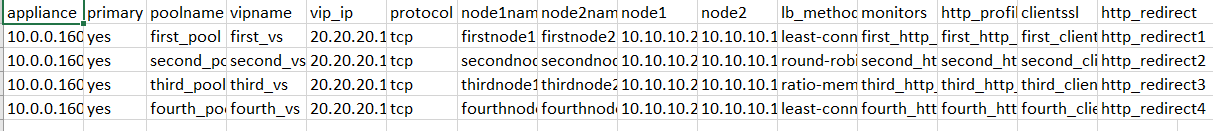
The add\_multiple\_ssl\_and\_redirect.yml will create two vips, an SSL/TLS vip that will terminate SSL/TLS traffic, and a port 80 redirect vip. This is to ensure clients get to the https vip even if they put http in their browser rather than https.

Prior to running this playbook you will need to run the push\_cert\_key\_to\_F5.yml. Edit the path to the certificate and key. This will stage the certificate and key on the F5 device.

Now run the create\_clientssl\_profile.yml playbook to create a child clientssl profile with the certificates and keys loaded using the push\_cert\_key\_to\_F5 playbook.

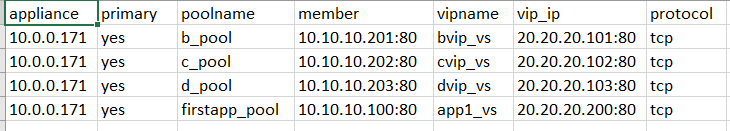
Lastly, edit the add\_multiple\_ssl\_and\_redirect.yml playbook to reflect the correct names for all the profiles.

By filling out the multipl\_ssl\_redirect csv file you will create one application per row. So with one execution of the add\_multiple\_ssl\_and\_redirect.yml playbook you can configure all your ssl applications with one run of the playbook



Creating non SSL/TLS Applications

Like the f5\_HA.yml playbook, you will need to change the admin password accordingly in the add\_app.yml playbook.



You will also need to change the management ip address in the add\_app.csv file under appliance. Notice the primary is yes for all rows. We will be adding this configuration to just one device and do a configsync to the partner device. The member column must be an ip:port pairing. Select the desired protocol, TCP or UDP.

Creating a Wildcard IP-Forwarding Virtual

The wildcard\_ip-forwarder playbook will create a wildcard ip-forwarding virtual

Configuring F5 DNS Module for GSLB

At this point we can add a GSLB configuration. Verify that the F5 device is licensed for DNS (GTM) and that it is provisioned. If you are installing F5 DNS on just one device in an active/standby cluster, you only need the DNS license on the device that will perform GSLB. The other unit does not need a DNS license but it MUST be provisioned for DNS. An active/standby pair always needs the provisioning to match its peers.

The gslb playbook uses the gslb.csv file for its variable inputs. The gslb playbook will do the following.

* Create two listeners. A TCP and a UDP listener.
* Create two datacenters.
* Add a server object to both datacenters
* Add one virtual server to each server object
* Create the GSLB pool with the two virtual servers
* Create the wideip with the GSLB pool attached.

Now you have one F5 DNS configured for GSLB in the first datacenter. Run the gslb\_for\_secondary playbook to configure your second F5 DNS in the second datacenter. The gslb\_for\_secondary will do the following.

* Create two listeners. A TCP and a UDP listener.
* Create two datacenters.
* Add the local server object

You will now need to logon to the CLI of the first f5 DNS and do the following.

* Run bigip\_add script to exchange certificate and key to setup iQuery. The syntax is as follows
  + bigip\_add <self ip of other F5 devices separated by a space>
  + Example bigip\_add 10.1.1.1 172.16.1.1 192.168.1.1
  + Verify that you have connectivity from the first F5 to all others. Any firewalls in the way will need to allow TCP ports 22, 443 and 4353
* Run big3d\_install to ensure big3d compatibility
  + Same syntax as before and same ports open on the firewalls
  + Example big3d\_install 10.1.1.1 172.16.1.1 192.168.1.1

Now run the iqdump command to every other f5 one at a time. The syntax is iqdump 10.1.1.1.

There will be two possible outcomes. You will see XML data scroll across the screen every few seconds or there will be an ssl error. If you see XML then iQuery is working. If not you will need to troubleshoot. Chances are either there is a firewall blocking communication between the f5 devices. The devices will require TCP ports 22, 443 and 4353. Or the system certificate is expired.

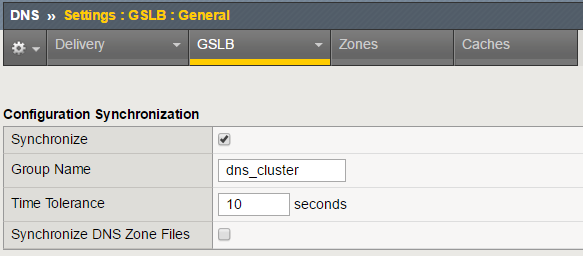
After you verify that iQuery communications is working properly, run the gtm\_add script from the secondary F5 DNS and point it to a self ip on the original F5 DNS.

Example: gtm\_add 10.1.1.1.

This will pull the remaining configuration from the original F5 DNS to the secondary.

At this point log into the GUI of both F5 DNS and navigate to DNS -> Settings -> GSLB -> General

Change the “Group Name” from “default” to anything else and check the “Synchronize” box.



The GSLB configuration is now complete.