Dens - Docs

Release alpha

Denis Grimard

Jun 12, 2017

Contents:

CHAPTER 1

Getting Started

This is a repository of my documentation for handy commonly used items:

Note: Please use at your own risk. These are from my own experience and may not work in your own lab. These scripts may break your system or lab or the whole time continuum !!!

CHAPTER 2

Sphinx Documentation

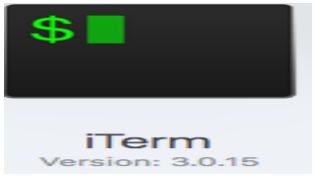
Sphinx is an application helper to build documentation.

Note: Sphinx is fun.

Starting the Sphinx server

To start the sphinx server do the following

1. Open a shell - in my case I use iterm.



2. Change to the correct directory (in my case - /Users/grimard/Documents/Sphinx)
3. **Start the server with the following command:** sphinx-autobuild DensDocs/ DensDocs/_build

How to build documents

To properly build Sphinx documentation you need to know a few things:

How to do formatting

1. To create a master heading use a bunch of —— on a separate line

2. To create a subheading use a bunch of ~~~~ on a separate line
3. To create an ordered list (numbered) use the # symbol then a period.
4. To add an image via a variable use:

```
|image12|
```

5. In this document I am using:

```
- for top level
~ for sub headings
```

Variables for images

This is another subheading and ordered list for variables

To create a variable for an image file to be used in a page do the following:

1. Define the variable as such:

```
.. |image8| image:: /_static/image008.png
:width: 0.46171in
:height: 0.43269in
```

Making Links

It is easy to make a link to yahoo or to some section inside this document (see *How to do formatting*) or another document.

To do this call something like this:

```
Link to Yahoo = `yahoo <http://yahoo.com>`_

Reference to a Page = see :ref:`task-1-formatting`
  You will also need to use .. _task-1-formatting: before your heading
```

You can also reference classes, modules, functions, etc that are documented using the sphinx autodoc facilites. For example, see the module `matplotlib.backend_bases` documentation, or the class `LocationEvent`, or the method `mpl_connect()`.

CHAPTER 3

Cloud'ish Things

I like clouds, here is stuff about clouds

Ansible

This is a collection of Ansible things

Ansible container

I have grabbed this one from Tom McGonagle

SuperNetOps

This will be for Cloud Team's SuperNetOps information.

SuperNetOps GitHub Repository

Super-NetOps is the new NetOps. This repo supports and auto-build integration with Docker Hub.

To learn more about Super-NetOps, read Does DevOps need a Super-NetOps'

To use the Super-NetOps container, visit

Azure

This is a placeholder for Azure

AWS

Placeholder for AWS

Cloud 5

Placeholder for Cloud 5

CHAPTER 4

Containers

This is a collection of the container things i have gathered - think hoarders - :)

Docker Basics

Install Docker for your OS - I am using a Mac so I downloaded and ran the Mac installer

List images

docker images

List containers

docker ps -a

Run shell within a container

docker exec -it -u root jenkins-2 /bin/bash

Container 2

In this lab we will review, line-by-line an example script that has been created to view the attributes of a BIG-IP Pool directly from the command line.

Task 1 – Review read_pool.py

1. Open `read_pool.py` in Notepad++
2. We will review the code. For brevity we have removed lines that are common with previous examples:

```
if not mgmt.tm.ltm.pools.pool.exists(partition=args.partition, name=args.pool_name):
raise Exception("Pool '%s' does not exist" % args.pool_name)
```

This if statement checks to see if a pool with the same name exists in the specified partition on the device. The key difference between this and the example in the previous lab is the inclusion of the 'not' keyword. This inverses the logic of the statement so that the Exception is raised when the pool DOES NOT exist

```
pool = mgmt.tm.ltm.pools.pool.load(partition=args.partition, name=args.pool_name)
```

This line loads the configuration of the pool into a variable

```
print "Pool %s:" % pool_path
pp.pprint(pool.raw)
```

These lines print the human-readable pool path and then uses the PrettyPrint library to dump all the attributes associated with the pool

Task 2 – Run read_pool.py

1. In the command prompt type `python read_pool.py 10.1.1.4 test_pool` and examine the output:

2. Notice the various attributes that are associated with the pool. Take note of the value of the `loadBalancingMode` attribute for the next lab

CHAPTER 5

Putty Things

I like Play Dough - also called Creative Putty

Putty

Used to do ssh type stuff

Putty Commands

1. Here is the first putty command

Plink

Plink is for automating putty commands

Plink Commands

1. This command will license the bigip: *"C:Program FilesPuTTYplink.exe" -l admin -pw admin 10.1.1.7 run /util bash -c "/usr/local/bin/SOAPLicenseClient –basekey J1701-07024-25118-65122-2084674"*
2. This command will login to the bigip and change the password to what is in the master-key.txt file: *"C:Program FilesPuTTYplink.exe" -l admin -pw admin 10.1.1.7 run /util bash -c "sh load-ucs.sh"*

   load-ucs.sh

```
cat master-key.txt |tmsh modify /sys crypto master-key prompt-for-password
tmsh save /sys config
tmsh load /sys ucs initial-config.ucs no-license
```

master-key.txt

```
default
default
```

3. This command will run “C:Program FilesPuTTYplink.exe” -load automation ”./run-demo.sh”. This is a linux box that has the GitHub repository on it and it runs the file - run-demo.sh
4. This command will run “C:Program FilesPuTTYplink.exe” -load automation ./test.sh. This is a linux box that has the GitHub repository on it and it runs the file - test.sh

CHAPTER 6

Rest Things

I like to REST, sometimes I even fall asleep.

REST

Useful rest information

REST Commands

1. Here is the first rest command

```
curl -sk -u admin:password https://bigip.domain.com/mgmt/tm/ltm/pool -H "Content-
↪Type: application/json" | sed 's/,/\'$'\n/g'
```