SoftLayer

Getting Started

Based on http://sftlyr.com/dockerlab

Today we will be exploring automation and devops tools that can help manage an automated Cloud deployment.

We will be using:
softlayer-python: A Python library that provides raw API access, a simplified manager layer for SoftLayer services, and a command line tool. – https://github.com/softlayer/softlayer-python

SaltStack: A server and infrastructure management utility. This tool allows for central management of multiple devices. In our case we will use it to issue commands to 'minion' servers rather than logging into them directly. - http://www.saltstack.com/

Docker: Docker is a utility to build, ship, and run, applications in Linux Containers(LXC) - https://www.docker.com/

Ghost: Ghost is a blogging application written in nodejs. Along with being awesome, it is simple to setup and setting up a Ghost blog will be the goal of this lab. - https://ghost.org/
When issuing commands look for words in: BLUE. These words will need to be changed to fit your lab environment.
Each section has a corresponding command. Section 1 has the command "1", section 2 has the command "2" and so on.

Prerequisites

An SSH Client
• Windows
Putty.exe (http://www.putty.org/)
• Linux/Mac
You should already have an SSH client which you can access through a **terminal** with the command **ssh**.

We will be deploying a salt-master and one salt minion.

Part 1: ensuring that we have the Python CLI deployed someplace

You should have the CLI configured as part of the previous homework. Recall that you simply need a system with the softlayer-python library installed. Before we can use it, we must
give it our account credentials.
1. Start the configuration process with **sl config setup**
2. When prompted enter your lab **username**

3. When prompted enter your lab **password**(not the password to your ***salt-master***)
4. When prompted for ***Endpoint URL***, enter **public**

We can test that the if the configuration was successful by running ***slcli vs list.*** You should see a listing of virtual machines on your account. If not, please ask for assistance.

Step 2: SSH Keys (if needed)

An SSH Key is a pair of very large numbers: A Public Key, a number which acts as a lock and is installed on the server you want to login to, and a Private Key, a number which acts as the actual key. When you login to a server that has a Public key that matches your Private key, no password is needed to login to the server. SoftLayer supports provisioning servers with your Public Key. When a server is created with a Public Key, it is not necessary to log into the portal to locate it's password.

1. Create your Public and Private keys with
   ssh-keygen
2. You will be prompted 3 times for input. Each time hit ***enter*** to accept the default configurations
3. Upload your Public Key to SoftLayer the SoftLayer API:
   slcli sshkey add --in-file ~/.ssh/id_rsa.pub test01
4. Keep track of the Public Key **Label** (your lab username), as you will need to specify it when creating new servers.
5. Treat your Private Key like a password; don't share it with anyone.

Step 3: Creating the salt-minion & salt-master

Salt is a server management system that takes the pain and suffering out of managing servers.

Use the –test flag if you want to test these commands:

Create the salt-master:

```
slcli vs create --datacenter=sjc01 --hostname=salt --domain=test01.labs.sftlyr.ws --billing=hourly
--cpu=1 --memory=1024 --image=ae67e7c9-cb9f-4615-8b42-ac5aa8aca687 --key=somekey
```

Create the salt-minion:

```
slcli vs create --datacenter=sjc01 --hostname=minion01 --domain=test01.labs.sftlyr.ws
--billing=hourly --key=somekey --cpu=1 --memory=1024
--image=242b3f9f-866d-4ca1-a732-230013de54bd
```

This will create a new virtual server. Make sure to specify your Public Key Label.
Keep an eye on the transaction process with the following command.

```
$> slcli vs detail <VS ID>
```

Once the "active_transaction" section is "-", we can continue.

Step 4: Logging in to the salt-master

We have already created the ***salt-master*** *for you, so all you need to do now is get the* ***login credentials*** *from the* ***portal****, and* ***login*** *with* ***ssh****.*

1. Navigate to - https://control.softlayer.com
2. Login using the credentials provided.
3. Go to Devices -> Device List and click the little triangle to expand the panel.
4. Click “Show Password” to display the password for your server (it should be called salt.username.labs.sftlyr.ws)
5. Locate the **Public IP** listed for your server
6. **SSH** the ***salt-master*** with the **Public IP** and **password** located in steps 4 & 5.
   $> ssh root@<Public IP>

Alternatively, connect to salt-master using your private key (ssh –I keyname ....)

Step 5: Configuring the Minion

To get the salt-minion working, we just have to tell it which IP address the salt-master server is
located at. This will be the Public IP you used in section 1.

1. Configure the ***salt-minion*** with the ***salt-master*** IP address with
   ssh <salt-minion IP> "echo 'master: <salt-master IP>' > /etc/salt/minion.d/master.conf"
2. Restart the ***salt-minion*** process with
   ssh <salt-minion IP> "service salt-minion restart"

Step 6: Working with Salt

Now that we have taught the ***salt-minion*** about the ***salt-master*** it is time to teach the ***saltmaster*** about the ***salt-minion***.

1. Find all the servers attempting to connect to the ***salt-master*** with
   salt-key –L

   NOTE: If you don't see the minion in the "Unaccepted Keys" section, you will need to run:
   ssh <salt-minion IP> "echo '<salt-master IP> salt.test01.labs.sftlyr.ws salt' > /etc/hosts"
   ssh <salt-minion IP> "service salt-minion restart"
   salt-key –L

2. Accept the ***salt-minion*** with
   salt-key –a minion01.test01.labs.sftlyr.ws
3. Test the salt configuration with
   salt '*' test.ping
4. See all supported salt commands with
   salt '*' sys.doc

This will give you a VERY long list of all the magic salt can do. But for now we are just going to focus on cmd.run, which will let you run commands remotely.

5. Test *cmd.run* with
   salt '*' cmd.run 'free -m'

expected output from free –m

```
minion01.test01.labs.sftlyr.ws:
                 total       used       free     shared    buffers     cached
    Mem:            989        179        810          0         14         84
    -/+ buffers/cache:          80        909
    Swap:          2047          0       2047
```

Step 7: Docker

Docker (https://www.docker.com/whatisdocker/) is a utility that helps you easily deploy applications. Combined with salt, we barely have to even think about our servers anymore.

So let's get started...
Docker has a wide variety of other containers which you can check out (https://registry.hub.docker.com/) or just build your own, but today we are going to install Ghost (https://ghost.org/) which is a neat little blogging platform.

The first command may take a few minutes to complete.

```
$> salt '*' cmd.run "docker pull ghost"
$> salt '*' cmd.run "docker run -d -p 80:2368 ghost"
```

The first command pulls down the Docker files for ghost to the local machine. The second command spins up the container, telling it to listen on port 80. And that's it!

Put minion01's Public IP address into your browser and see if you get a welcome page.

http://<salt-minion IP>

Feel free to mess around with the Ghost blog to see what it can do. Hopefully this presentation wasn't too overwhelming, if you have any questions please let us know.