Lab 1: Install DC/OS

In this lab we will walk you through how to install DC/OS **1.12.1** using the advanced install method.

Your instructor gave you a sheet with all the private and public IP addresses of your nodes. You will need to use the public IP addresses to SSH into your nodes remotely.

Use the following credentials to access your cluster nodes over SSH:

Property	Value
Username	**centos**
Password	**me$o$ph3r3_**

Lab Completion Time: 25 - 35 minutes.

Prepare the Bootstrap Node

Step 1

ssh into your bootstrap node with the **centos**/**me$o$ph3r3_** credentials and **cd** into the `1.12.1` directory:

```
$ ssh centos@<bootstrap_node_public_IP>
$ cd 1.12.1
```

Step 2

After connecting to your bootstrap node over SSH, you will find a template named `config.yaml.example` in the `~/1.12.1/genconf` directory. Make a copy of that template:

```
$ cp genconf/config.yaml.example genconf/config.yaml
```

Now open up the `config.yaml` file with your favorite text editor. There are a few things that you will need to change in that file before generating the install artifacts. Replace the following placeholders with the values that correspond to the cluster information your instructor provided:

- `<bootstrap_url>` should be set to the private IP address (e.g. 10.0.0.X) of your bootstrap node.
- `<cluster_name>` a name you would like to assign to the cluster. You can use your name for this value.
- `<master_list>` is a YAML list that should be set to the private IP address (e.g. 10.0.0.X) of your master node.

Warning

Make sure you use the private IP addresses (`10.0.0.X`) in your `config.yaml` file!

Use the snippet below as an example:

config.yaml

```
bootstrap_url: http://10.0.0.100
cluster_name: Mesosphere-Training
exhibitor_storage_backend: static
master_discovery: static
master_list:
- 10.0.0.50
resolvers:
- 8.8.8.8
- 8.8.4.4
ip_detect_public_filename: /genconf/ip-detect-public
```

Step 3

cd in to the `~/1.12.1` directory if you are not already there and run the DC/OS installer to generate customized install bits. The installer will extract a Docker container image that will use generic DC/OS installation files to create customized build files for your cluster. The build files generated will be written to disk in the `./genconf/serve/` directory:

```
$ cd ~/1.12.1
$ sudo bash dcos_generate_config.ee.sh
```

Note

It may take several minutes for the above command to complete execution.

Step 4

The build files will need to be distributed to our master and agent nodes in the next steps. To facilitate this, we will set up an **nginx** web server that will share out those files. Execute the following command exactly as it is written to start a web server on your bootstrap node:

```
$ sudo docker run -d -p 80:80 -v $PWD/genconf/serve:/usr/share/nginx/html:ro nginx:alpin
```

Install the Master Node

Step 1

SSH in to your master node with the **centos**/**me$o$ph3r3_** credentials:

```
$ ssh centos@<master_node_public_IP>
```

Step 2

Make a new directory and navigate to it:

```
$ mkdir /tmp/dcos && cd /tmp/dcos
```

Step 3

Download the DC/OS installer from the bootstrap node over HTTP:

```
$ curl -O http://<bootstrap_node_private_IP>/dcos_install.sh
```

Warning

Make sure you use the private IP of your bootstrap node in the command above!

Step 4

Run the following command to install and configure the DC/OS master components on this node:

```
$ sudo bash dcos_install.sh master
```

The output should resemble the following:

```
Starting DC/OS Install Process
Running preflight checks
Checking if DC/OS is already installed: PASS (Not installed)
PASS Is SELinux disabled?
Checking if docker is installed and in PATH: PASS
Checking docker version requirement (>= 1.6): PASS (1.13.1)
Checking if curl is installed and in PATH: PASS
Checking if bash is installed and in PATH: PASS
Checking if ping is installed and in PATH: PASS
Checking if tar is installed and in PATH: PASS
Checking if xz is installed and in PATH: PASS
Checking if unzip is installed and in PATH: PASS
Checking if ipset is installed and in PATH: PASS
Checking if systemd-notify is installed and in PATH: PASS
Checking if systemd is installed and in PATH: PASS
Checking systemd version requirement (>= 200): PASS (219)
Checking if group 'nogroup' exists: PASS
Checking if port 53 (required by spartan) is in use: PASS
Checking if port 80 (required by adminrouter) is in use: PASS
Checking if port 443 (required by adminrouter) is in use: PASS
Checking if port 1050 (required by dcos-diagnostics) is in use: PASS
Checking if port 2181 (required by zookeeper) is in use: PASS
Checking if port 5050 (required by mesos-master) is in use: PASS
Checking if port 7070 (required by cosmos) is in use: PASS
Checking if port 8080 (required by marathon) is in use: PASS
Checking if port 8101 (required by dcos-oauth) is in use: PASS
Checking if port 8123 (required by mesos-dns) is in use: PASS
Checking if port 8181 (required by exhibitor) is in use: PASS
Checking if port 9000 (required by metronome) is in use: PASS
Checking if port 9942 (required by metronome) is in use: PASS
Checking if port 9990 (required by cosmos) is in use: PASS
Checking if port 15055 (required by dcos-history) is in use: PASS
Checking if port 33107 (required by navstar) is in use: PASS
Checking if port 36771 (required by marathon) is in use: PASS
Checking if port 41281 (required by zookeeper) is in use: PASS
Checking if port 42819 (required by spartan) is in use: PASS
Checking if port 43911 (required by minuteman) is in use: PASS
Checking if port 46839 (required by metronome) is in use: PASS
Checking if port 61053 (required by mesos-dns) is in use: PASS
Checking if port 61420 (required by epmd) is in use: PASS
Checking if port 61421 (required by minuteman) is in use: PASS
Checking if port 62053 (required by spartan) is in use: PASS
Checking if port 62080 (required by navstar) is in use: PASS
Checking Docker is configured with a production storage driver: PASS (overlay)
Creating directories under /etc/mesosphere
Creating role file for master
Configuring DC/OS
Setting and starting DC/OS
Created symlink from /etc/systemd/system/multi-user.target.wants/dcos-setup.service to /
```

The installation is done on this system and services will begin to start up once you have returned back to your shell prompt.

Install the Public Agent Node

Step 1

SSH in to your public agent node with the **centos**/**me$o$ph3r3_** credentials:

```
$ ssh centos@<public_agent_public_IP>
```

Step 2

Make a new directory and navigate to it:

```
$ mkdir /tmp/dcos && cd /tmp/dcos
```

Step 3

Download the DC/OS installer from the bootstrap node over HTTP:

```
$ curl -O http://<bootstrap_node_private_IP>/dcos_install.sh
```

Warning

Make sure you use the private IP of your bootstrap node in the command above!

Step 4

Run the following command to install and configure the DC/OS public agent components on this node:

```
$ sudo bash dcos_install.sh slave_public
```

The output should resemble the following:

```
Starting DC/OS Install Process
Running preflight checks
Checking if DC/OS is already installed: PASS (Not installed)
PASS Is SELinux disabled?
Checking if docker is installed and in PATH: PASS
Checking docker version requirement (>= 1.6): PASS (1.13.1)
Checking if curl is installed and in PATH: PASS
Checking if bash is installed and in PATH: PASS
Checking if ping is installed and in PATH: PASS
Checking if tar is installed and in PATH: PASS
Checking if xz is installed and in PATH: PASS
Checking if unzip is installed and in PATH: PASS
Checking if ipset is installed and in PATH: PASS
Checking if systemd-notify is installed and in PATH: PASS
Checking if systemd is installed and in PATH: PASS
Checking systemd version requirement (>= 200): PASS (219)
Checking if group 'nogroup' exists: PASS
Checking if port 53 (required by spartan) is in use: PASS
Checking if port 5051 (required by mesos-agent) is in use: PASS
Checking if port 34451 (required by navstar) is in use: PASS
Checking if port 39851 (required by spartan) is in use: PASS
Checking if port 43995 (required by minuteman) is in use: PASS
Checking if port 61001 (required by agent-adminrouter) is in use: PASS
Checking if port 61420 (required by epmd) is in use: PASS
Checking if port 61421 (required by minuteman) is in use: PASS
Checking if port 62053 (required by spartan) is in use: PASS
Checking if port 62080 (required by navstar) is in use: PASS
Checking Docker is configured with a production storage driver: PASS (overlay)
Creating directories under /etc/mesosphere
Creating role file for slave_public
Configuring DC/OS
Setting and starting DC/OS
Created symlink from /etc/systemd/system/multi-user.target.wants/dcos-setup.service to /
```

The installation is done on this system and services will begin to start up once you have returned back to your shell prompt.

Install the Private Agent Nodes

Step 1

SSH in to one of your private agent nodes with the **centos**/**me$o$ph3r3_** credentials:

```
$ ssh centos@<private_agent_public_IP>
```

Step 2

Make a new directory and navigate to it:

```
$ mkdir /tmp/dcos && cd /tmp/dcos
```

Step 3

Download the DC/OS install from the bootstrap node over HTTP:

```
$ curl -O http://<bootstrap_node_private_IP>/dcos_install.sh
```

Warning

Make sure you use the private IP of your bootstrap node in the command above!

Step 4

Run the following command to install and configure the DC/OS private agent components on this node:

```
$ sudo bash dcos_install.sh slave
```

The output should resemble the following:

```
Starting DC/OS Install Process
Running preflight checks
Checking if DC/OS is already installed: PASS (Not installed)
PASS Is SELinux disabled?
Checking if docker is installed and in PATH: PASS
Checking docker version requirement (>= 1.6): PASS (1.13.1)
Checking if curl is installed and in PATH: PASS
Checking if bash is installed and in PATH: PASS
Checking if ping is installed and in PATH: PASS
Checking if tar is installed and in PATH: PASS
Checking if xz is installed and in PATH: PASS
Checking if unzip is installed and in PATH: PASS
Checking if ipset is installed and in PATH: PASS
Checking if systemd-notify is installed and in PATH: PASS
Checking if systemd is installed and in PATH: PASS
Checking systemd version requirement (>= 200): PASS (219)
Checking if group 'nogroup' exists: PASS
Checking if port 53 (required by spartan) is in use: PASS
Checking if port 5051 (required by mesos-agent) is in use: PASS
Checking if port 34451 (required by navstar) is in use: PASS
Checking if port 39851 (required by spartan) is in use: PASS
Checking if port 43995 (required by minuteman) is in use: PASS
Checking if port 61001 (required by agent-adminrouter) is in use: PASS
Checking if port 61420 (required by epmd) is in use: PASS
Checking if port 61421 (required by minuteman) is in use: PASS
Checking if port 62053 (required by spartan) is in use: PASS
Checking if port 62080 (required by navstar) is in use: PASS
Checking Docker is configured with a production storage driver: PASS (overlay)
Creating directories under /etc/mesosphere
Creating role file for slave
Configuring DC/OS
Setting and starting DC/OS
Created symlink from /etc/systemd/system/multi-user.target.wants/dcos-setup.service to /
```

The installation is done on this system and services will begin to start up once you have returned back to your shell prompt.

Step 5

Perform steps 1-4 on your other private agent node.

Once each node has completed executing the installation script, your DC/OS environment should be up and running! We will verify this in the next lab. Real nice work in this exercise!

© Copyright 2018, Mesosphere. Back to top