

Lesson 04 Demo 07

Using Templates with Docker Create Service

Objective: To demonstrate Docker's template usage to dynamically configure container attributes like hostnames based on service name, node ID, and hostname, optimizing container management and deployment efficiency

Tools required: Docker and Ubuntu OS

Prerequisites: A Docker swarm cluster must be set up.

Steps to be followed:

1. Set a container template
2. Check the service status
3. Inspect the service

Step 1: Set a container template

- 1.1 Execute the following commands to set the template of the created containers based on service name, node ID, and hostname:

```
sudo docker service create \  
--name hosttempl \  
--hostname="{{.Node.Hostname}}-{{.Node.ID}}-{{.Service.Name}}" \  
busybox top
```

```
labsuser@ip-172-31-29-216:~$ sudo docker service create \  
> --name hosttempl \  
> --hostname="{{.Node.Hostname}}-{{.Node.ID}}-{{.Service.Name}}" \  
> busybox top  
lez36tii48ej76rdfqininiew  
overall progress: 1 out of 1 tasks  
1/1: running [=====>]  
verify: Service converged  
labsuser@ip-172-31-29-216:~$
```

Step 2: Check the service status

- 2.1 Run the following command to check the status and details of all tasks (containers) associated with the **hosttempl** service in a Docker swarm:

sudo docker service ps hosttempl

```
labsuser@ip-172-31-29-216:~$ sudo docker service ps hosttempl
ID                NAME          IMAGE          NODE                DESIRED STATE  CURRENT STATE
qih6bd4o6tou      hosttempl.1    busybox:latest ip-172-31-30-210    Running        Running 2 minutes ago
labsuser@ip-172-31-29-216:~$
```

Step 3: Inspect the service

- 3.1 Use the following command to inspect the **hosttempl** service:

sudo docker inspect SERVICE_ID --format "{{json .Status}}"

```
labsuser@ip-172-31-29-216:~$ sudo docker inspect qih6bd4o6tou --format "{{json .Status}}"
{"Timestamp":"2021-01-14T05:46:45.994532956Z","State":"running","Message":"started","ContainerStatus":{"ContainerID":"d46890ace8c186c925c3e494b7590142b45fa5d7594e0ab86f898bc1abadf4f9","PID":16592,"ExitCode":0,"PortStatus":{}}}
```

Note: Replace SERVICE_ID with the ID of the **hosttempl** service

By following these steps, you have successfully demonstrated Docker's template usage to dynamically configure container attributes such as hostnames based on service name, node ID, and hostname, thereby optimizing container management and deployment efficiency in your Docker environment.