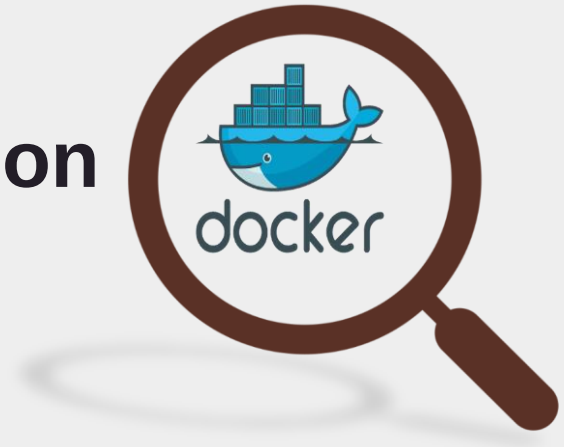
Persistent

Docker Orchestration - Docker machine



Docker Orchestration

- Docker Engine works well for packaging simple, single container apps.
- Production deployments are more complex.
- Typically apps composed of multiple containers, running across multiple hosts.
- To help the deployment of distributed applications, Docker has provided multiple Orchestration tools
  - Docker Machine
  - Docker Swarm
  - Docker Compose

Docker Machine

- Docker Machine takes the users from Zero to Docker in seconds in a single command.
- Before Docker Machine, developers need to run OS specific installation and configuration commands for Docker containers.
- Docker machine is a tool which automatically provisions docker hosts and installs docker machine on them.
- Docker Machine lets you install Docker Engine on virtual hosts and manage the hosts with docker-machine commands.
- Docker Machine can be used to create Docker hosts on Mac, Windows box, on Cloud like AWS, Azure.
- Why it should be used?
  - Provision Docker Hosts on remote systems.

Using machine on VMs

- Docker Machine can create hosts on most major virtualization hypervisors and in cloud service providers.
- Docker Machine has driver support for AWS, Google Cloud Platform, IBM Softlayer, Microsoft Azure and Hyper-V, OpenStack, Rackspace, VirtualBox, VMware Fusion®, vCloud® Air™ and vSphere®.

Docker Machine use cases

- **Running Docker on Mac or Windows**

- **Provisioning Docker hosts on remote systems**

Persistent

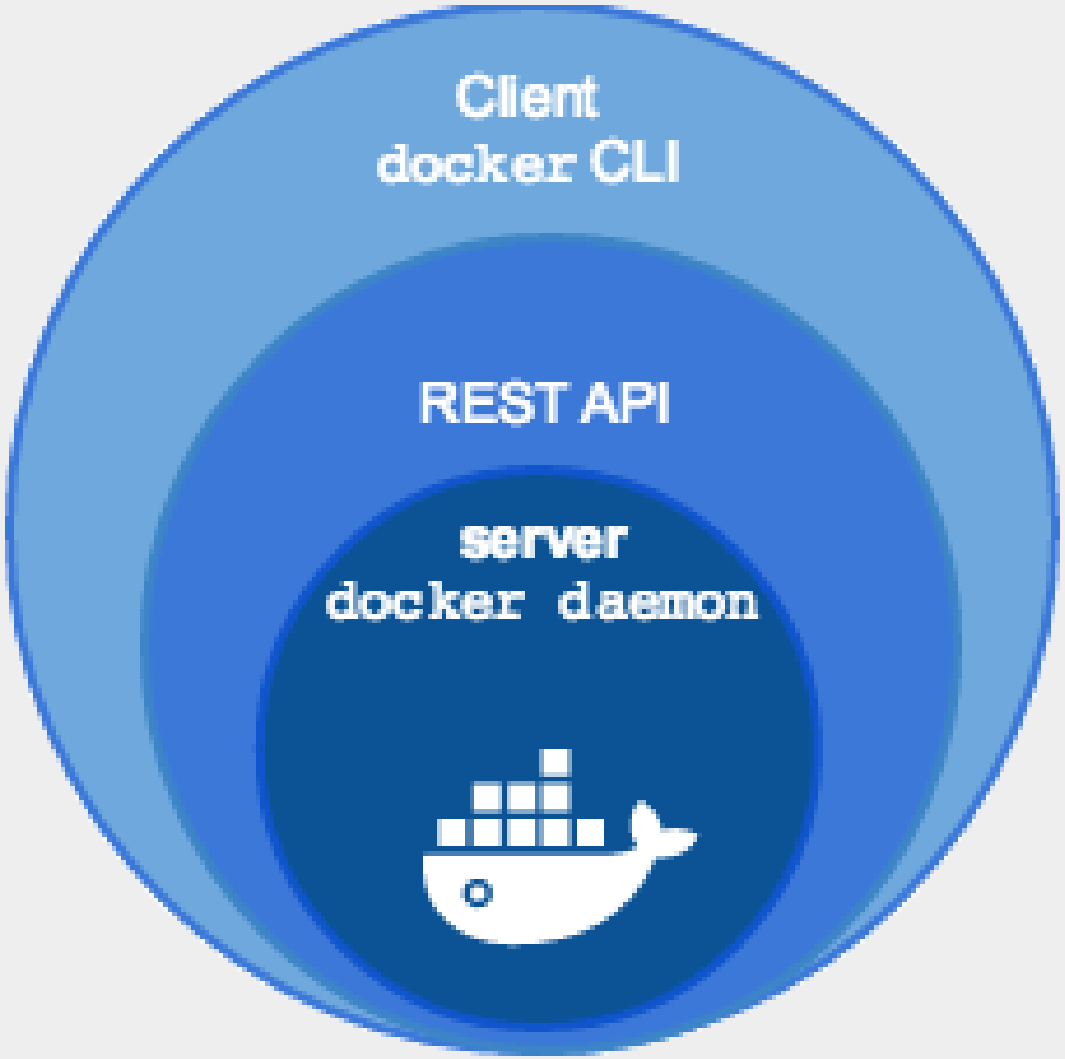
Docker Engine and Docker Machine differences

Docker Engine

- Client server application made up of

  Docker daemon, REST APIs and CLI client.
- Docker Engine accepts Docker

  commands from the CLI.



Persistent

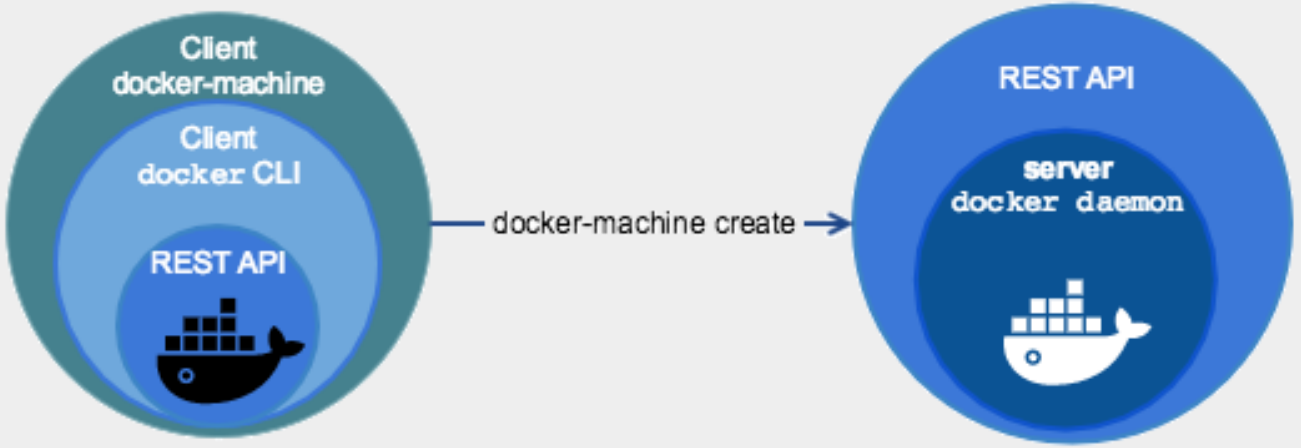
Docker Engine and Docker Machine differences…

Docker Machine

- A tool for provisioning and managing your Dockerized hosts (hosts with Docker Engine on them).
- Docker Machine has its own command line client docker-machine and the Docker Engine client, docker.
- You can use Machine to install Docker Engine on one or more virtual systems.
- Virtual systems can be local or remote (Cloud instances)
- The Dockerized hosts themselves can be thought of managed “machines”.

Docker Engine and Docker Machine differences…

Docker Machine



Persistent

Docker Machine installation

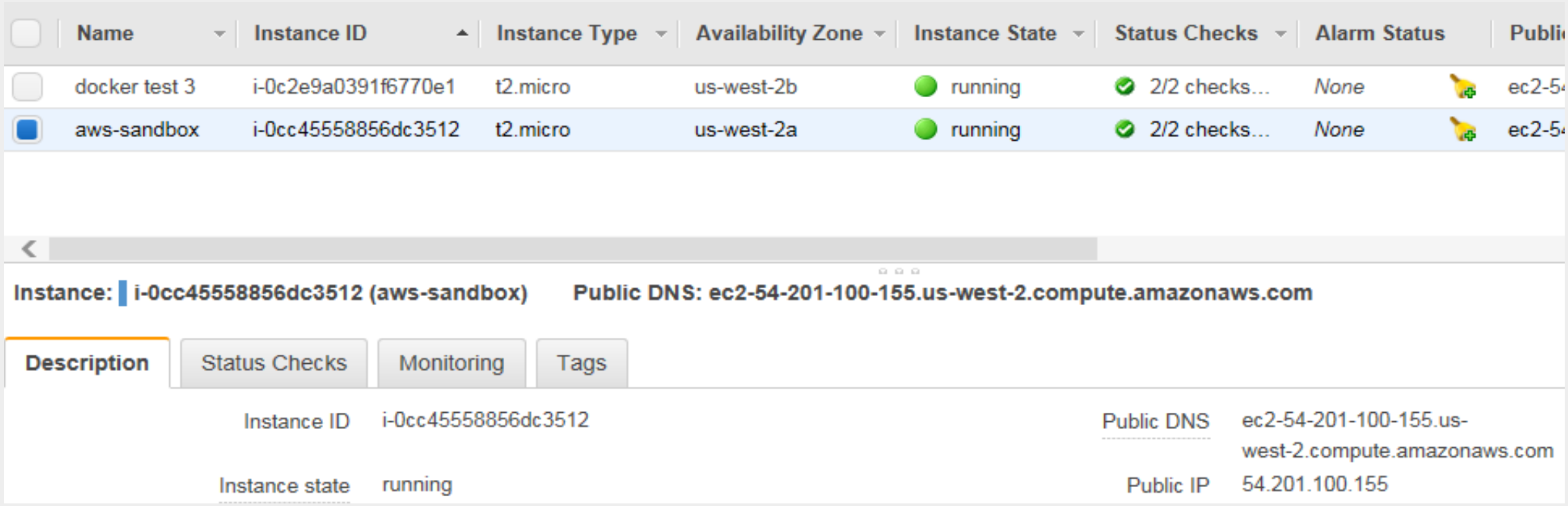
- Link for installing Docker Machine-
  https://docs.docker.com/machine/install-machine/
- Run the below command to install docker machine
  - $ sudo curl -L https://github.com/docker/machine/releases/download/v0.7.0/docker-machine-`uname -s`-`uname -m` > /usr/local/bin/docker-machine && chmod +x /usr/local/bin/docker-machine
- Make sure the user has access to directory - /usr/local/bin/
- Check if Docker machine is installed successfully or not
  - $ docker-machine version

Creating Docker Machine on AWS EC2

- Docker machine can be created on AWS Cloud, pls make sure you are using the correct region.
- Here is the link
  - https://docs.docker.com/machine/examples/aws/
- Run the command below to create a new Docker machine on AWS
  - docker-machine create --driver amazonec2 --amazonec2-region us-west-2 --amazonec2-access-key Your-Key --amazonec2-secret-key Your Secret Key aws-sandbox

Docker Machine on AWS EC2

- Docker machine command provisions a new EC2 instance on AWS, the Docker container is pre-installed. (check the ssh key pairs)



Persistent

Docker Machine on AWS EC2

- Let’s list down the list of Docker machines created on AWS

```
ubuntu@ip-172-31-16-130:~$ docker-machine ls
NAME           ACTIVE   DRIVER      STATE     URL                         SWARM   DOCKER
aws-sandbox    -        amazonec2   Error                                         Unknown
aws-sandbox1   -        amazonec2   Running   tcp://54.149.103.216:2376           v1.11.2
ubuntu@ip-172-31-16-130:~$
```

- Let’s inspect our new Docker container, using below command
- Docker-machine inspect aws-sandbox1

```
ubuntu@ip-172-31-16-130:~$ docker-machine inspect aws-sandbox1
{
    "ConfigVersion": 3,
    "Driver": {
        "IPAddress": "54.149.103.216",
        "MachineName": "aws-sandbox1",
        "SSHUser": "ubuntu",
        "SSHPort": 22,
        "SSHKeyPath": "/home/ubuntu/.docker/machine/machines/aws-sandbox1/id_rsa",
        "StorePath": "/home/ubuntu/.docker/machine",
        "SwarmMaster": false,
        "SwarmHost": "tcp://0.0.0.0:3376",
```

Running a Web server on Docker Machine

- Let’s run the below command to launch a web server on the Docker machine remotely.
  - **docker run -d -p 8000:80 --name webserver kitematic/hello-world-nginx**

```
ubuntu@ip-172-31-16-130:~$ sudo docker run -d -p 8000:80 --name webserver kitematic/hello-world-nginx
Unable to find image 'kitematic/hello-world-nginx:latest' locally
latest: Pulling from kitematic/hello-world-nginx

77c6c00e8b61: Pull complete
9b55a9cb10b3: Pull complete
```

- To connect to Docker machine using ssh, use the below command,
  - **docker-machine ssh <hostname>**

Using machine command to remove the instance.

- To remove an instance and all of its containers and images, first stop the machine, then use
  - **docker-machine rm**
  - **sudo docker-machine stop aws-sandbox1**
  - **sudo ocker-machine rm aws-sandbox1**

Reference Material : Websites & Blogs

- https://docs.docker.com/machine/examples/aws/
- The Docker Book by James Turnball
- https://www.youtube.com/watch?v=Q5POuMHxW-0

Docker up and Running by Karl Matthias and Sean kane

Docker Interactive

Dattatray Kulkarni

dattatray_kulkarni@persistent.co.in

Asif Immanad

asif_immanad@persistent.co.in

Vice President

Shubhangi Kelkar

shubhangi_kelkar@persistent.co.in

Persistent

Thank you!

Persistent University