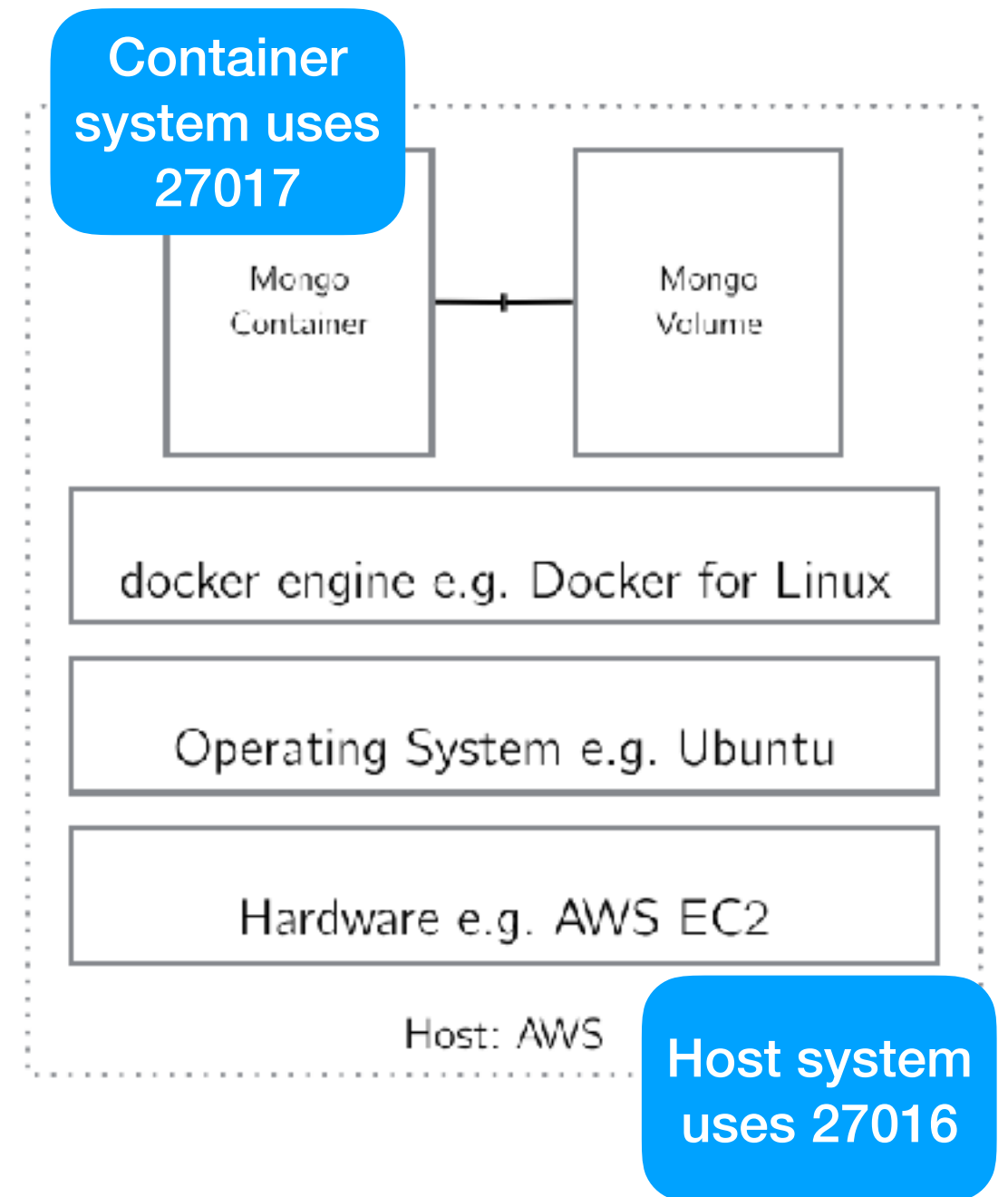


# Configuring Mongo on AWS

- Considerations:
  - Networking
    - Solve via AWS Security Groups
  - Data Persistence
    - Solve via Docker Volumes



# Configuring Mongo on AWS

- Note that this configuration will use Mongo on a separate instance from the instance on which you are running Jupyter.
- If you are running Jupyter on AWS, you will need a second `t2.micro`.
- Accessing a database managed by Docker from a different AWS instance is actually easier than accessing a database managed by Docker on the same system.

# Set up a new t2.micro

- From the AWS EC2 Dashboard, select “Launch Instance.”
- On the Choose AMI tab, choose Ubuntu Server 16.04.
- On the Choose Instance Type tab, choose t2.micro.
- On the Add Storage tab, use the default setting of 8GB.
- On the Configure Security Group tab, choose “Create a new security group.”
- a. Confirm that inbound SSH traffic can be accepted over port 22 from anywhere.
- b. Add a rule that accepts inbound traffic over port 2376. Make sure you choose “Anywhere” under the source. This port will allow you to pull images from Docker Hub.
- c. Add a rule that accepts inbound traffic over port 27016. Make sure you choose “Anywhere” under the source. This is one port less than the default port for accessing MongoDB.
- Review and launch an instance, taking care to confirm that you have access to the SSH keys stored with your AWS account.

# Configure the new t2.micro

- Take note of the IP address of the newly configured AWS instance.
- SSH into the instance using that IP address.
- Install Docker via a shell script.
- Add the ubuntu user to the docker group.
- Log out and back in.

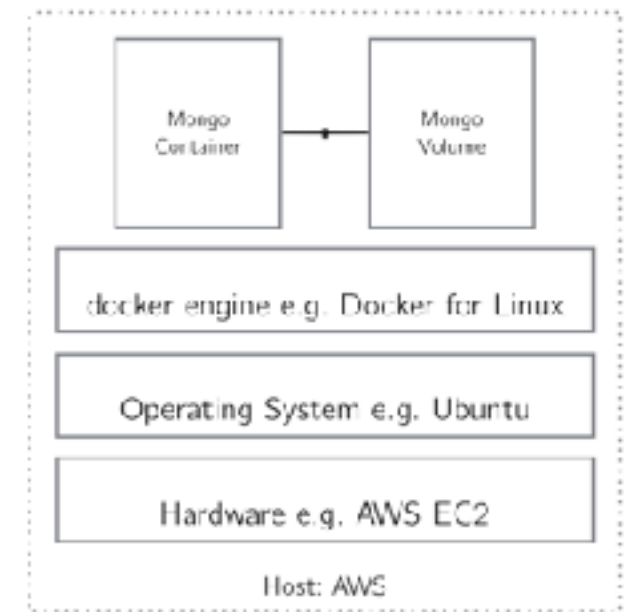
```
(local) $ ssh ubuntu@255.255.255.255  
(remote) $ curl -sSL https://get.docker.com/ | sh  
(remote) $ sudo usermod -aG docker ubuntu
```

# Run Mongo via Docker

- Pull the mongo image  
`$ docker pull mongo`

- Create a New Data Volume  
`$ docker volume create mongo-dbstore`

- Launch MongoDB as a Persistent Service  
`$ docker run -d --name this_mongo \`  
`-v mongo-dbstore:/data/db \`  
`-p 27016:27017 mongo`



# Verify MongoDB Installation

- You can verify that you are running the mongo service by connecting to the running MongoDB via the MongoDB client, `mongo`, issued via `docker exec`.
- To do this, connect and then insert a trivial document to a mongo collection. You are inserting the JSON object `{"foo": 1}` into the collection `test`. You then search for the document you inserted using the `.find()` command.

```
$ docker exec -it this_mongo mongo
> db.test.insert({"foo":1})
> db.test.find()
```

**What does the final ``mongo`` keyword do?**