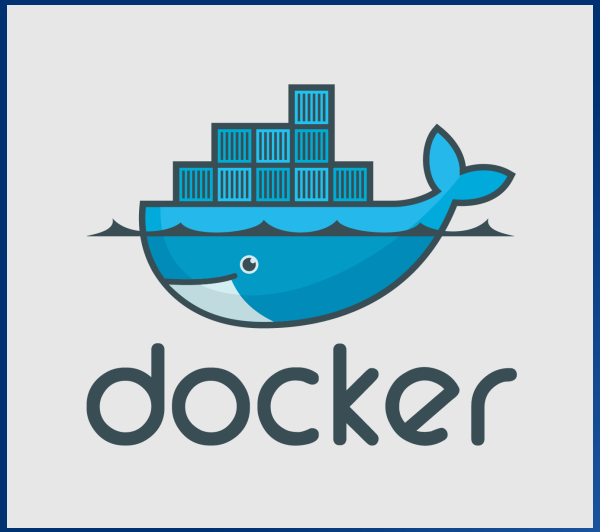


Apache jclouds and Docker

Andrea Turli
@turlinux

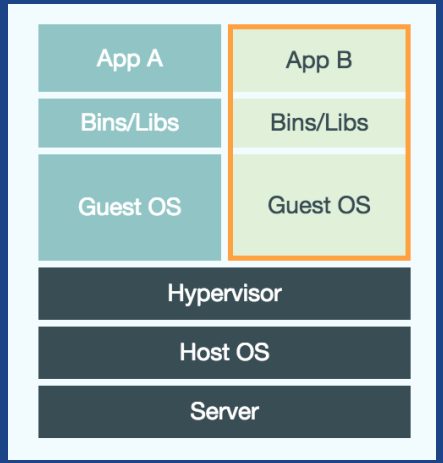
What is jclouds?

An open source multi-cloud toolkit for the Java platform

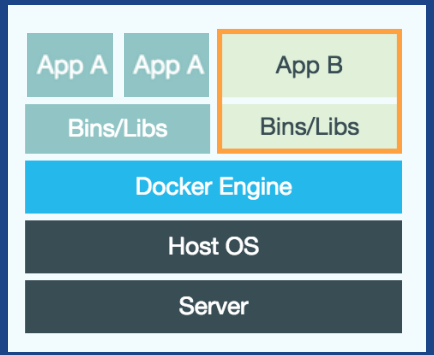
BlobStore and ComputeService API abstraction for 20+ IaaS providers

What is Docker?

Docker Engine is an open source container virtualization technology.



Virtual Machine



Docker

From jclouds perspective, Docker is a cloud provider which spins containers up instead of virtual machines

Docker REST API

Docker 1.3.x contains Docker REST API v1.15
Docker Remote API <- supported by *jclouds-docker*

- ContainerAPI
- ImageAPI
- MiscAPI

Docker Hub API
Docker Registry API

The API tends to be REST, but for some complex commands, like **attach** or **pull**, the HTTP connection is hijacked to transport STDOUT, STDIN and STDERR.

Docker Engine daemon configuration

By default, the Docker Engine daemon listens on *unix:///var/run/docker.sock* and the client must have **root** access to interact with the daemon.

Docker Engine daemon can be set to use an (encrypted) TCP socket

http://host:2375
https://host:2376

If the Docker Engine daemon is set to use an encrypted TCP socket (*--tls*, or *--tlsverify*) then you need to add extra parameters to ***curl*** or ***wget*** when making test API requests:

```
curl --insecure --cert ~$DOCKER_CERT_PATH/cert.pem --key
~$DOCKER_CERT_PATH/key.pem https://boot2docker:2376/images/json
```

Docker Remote API

2.1 Containers

- List containers
- Create a container
- Inspect a container
- List processes running insi...
- Get container logs
- Inspect changes on a cont...
- Export a container
- Resize a container TTY
- Start a container
- Stop a container
- Restart a container
- Kill a container
- Pause a container
- Unpause a container
- Attach to a container
- Wait a container
- Remove a container
- Copy files or folders from ...

2.2 Images

- List Images
- Create an image
- Inspect an image
- Get the history of an image
- Push an image on the regi...
- Tag an image into a reposit...
- Remove an image
- Search images

2.3 Misc

- Build an image from Dock...
- Check auth configuration
- Display system-wide infor...
- Show the docker version i...
- Ping the docker server
- Create a new image from a...
- Monitor Docker's events
- Get a tarball containing all ...
- Get a tarball containing all ...
- Load a tarball with a set of...
- Image tarball format
- Exec Create
- Exec Start
- Exec Resize

: supported by jclouds

Example: List Images

List Images

`GET /images/json`

Example request:

```
GET /images/json?all=0 HTTP/1.1
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "RepoTags": [
      "ubuntu:12.04",
      "ubuntu:precise",
      "ubuntu:latest"
    ],
    "Id": "8dbd9e392a964056420e5d58ca5cc376ef18e2de93b5cc90e868a1bbc8318c1c",
    "Created": 1365714795,
    "Size": 131506275,
    "VirtualSize": 131506275
  },
```

```java
@Consumes(MediaType.APPLICATION_JSON)
public interface ImageApi {

   /**
    * @return the images available.
    */
   @Named("images:list")
   @GET
   @Path("/images/json")
   @Fallback(EmptyListOnNotFoundOr404.class)
   List<ImageSummary> listImages();

   /**
    * @param options the configuration to list images (@see ListImageOptions)
    * @return the images available.
    */
   @Named("images:list")
   @GET
   @Path("/images/json")
   @Fallback(EmptyListOnNotFoundOr404.class)
   List<ImageSummary> listImages(ListImageOptions options);
```

```java
@AutoValue
public abstract class ImageSummary {

   public abstract String id();

   public abstract long created();

   public abstract String parentId();

   public abstract int size();

   public abstract int virtualSize();

   public abstract List<String> repoTags();

   ImageSummary() {
   }

   @SerializedNames({"Id", "Created", "ParentId", "Size", "VirtualSize", "RepoTags"})
   public static ImageSummary create(String id, long created, String parentId, int size, int virtualSize,
                                     List<String> repoTags) {
      return new AutoValue_ImageSummary(id, created, parentId, size, virtualSize, copyOf(repoTags));
   }
```

Auto Value: Immutable value-type code generation for Java 1.6+

Example: Create Container

Create a container

`POST /containers/create`

Create a container

Example request:

```
POST /containers/create HTTP/1.1
Content-Type: application/json

{
     "Hostname":"",
     "Domainname": "",
     "User":"",
     "Memory":0,
     "MemorySwap":0,
     "CpuShares": 512,
     "Cpuset": "0,1",
     "AttachStdin":false,
     "AttachStdout":true,
     "AttachStderr":true,
     "Tty":false,
     "OpenStdin":false,
     "StdinOnce":false,
     "Env":null,
     "Cmd":[
             "date"
     ],
     "Entrypoint": ""
     "Image":"base",
     "Volumes":{
             "/tmp": {}
     },
     "WorkingDir":"",
     "NetworkDisabled": false,
     "ExposedPorts":{
             "22/tcp": {}
     },
     "SecurityOpts": [""],
     "HostConfig": {
       "Binds":["/tmp:/tmp"],
       "Links":["redis3:redis"],
       "LxcConf":{"lxc.utsname":"docker"},
       "PortBindings":{ "22/tcp": [{ "HostPort": "11022" }] },
       "PublishAllPorts":false,
       "Privileged":false,
       "Dns": ["8.8.8.8"],
       "DnsSearch": [""],
       "VolumesFrom": ["parent", "other:ro"],
       "CapAdd": ["NET_ADMIN"],
       "CapDrop": ["MKNOD"],
       "RestartPolicy": { "Name": "", "MaximumRetryCount": 0 },
       "NetworkMode": "bridge",
       "Devices": []
     }
}
```

```
@Named("container:create")
@POST
@Path("/containers/create")
Container createContainer(@QueryParam("name") String name, @BinderParam(BindToJsonPayload.class) Config config);
```

Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json

{
     "Id":"f91ddc4b01e079c4481a8340bbbeca4dbd33d6e4a10662e499f8eacbb5bf252b"
     "Warnings":[]
}
```

Building jclouds-docker

official: https://github.com/jclouds/jclouds-labs/tree/master/docker
latest version (1.3.2): https://github.com/andreaturli/jclouds-labs/tree/1.3.2

TLS (optional; starting from 1.3.0)

If you are using boot2docker, set up the following VM options for jclouds-docker:

```
-Dtest.docker.identity=$DOCKER_CERT_PATH/cert.pem
-Dtest.docker.credential=$DOCKER_CERT_PATH/key.pem
-Dtest.docker.endpoint=https://192.168.59.103:2376 # this is the default, but check with boot2docker up
```

When using encrypted TCP socket, you should know that Docker removed sslv3 from server's TLS supported versions
https://github.com/docker/docker/pull/8588/files

JDK	sslContext.getDefaultSSLParameters().getProtocols()
6	SSLv3, TLSv1, SSLv2Hello
7	SSLv3, TLSv1

To have more fine-grained control, OkHttp driver will be the default choice for providers that need SSL/TLS
OkHttpCommandExecutorServiceModule vs JavaUrlHttpCommandExecutorServiceModule

References

Apache jclouds	https://jclouds.apache.org/
Apache jclouds Docker	https://jclouds.apache.org/guides/docker/
Docker	https://www.docker.com
Docker Rest API	http://goo.gl/Cuhpcx
Google AutoValue	https://github.com/google/auto/tree/master/value
How to disable SSL v3.0 in JDK and JRE	http://goo.gl/ZiK6Z7
OkHttp	https://github.com/square/okhttp/