



MobyCraft – Docker in 8-bit

Aditya Gupta

Student and Hacker, @aditya_g

dockercon 16

Before we start...



8



5



2

What does it do?

- **Client-side Docker mod inspired by server-side Dockercraft**
 - Thanks @gaetan_dv and @aduermael
 - Client-side mods have more capabilities
- **Manage Docker containers visually using Minecraft**
- **Run Docker commands from the game**
 - Common commands
 - Convenience commands
- **Add new Minecraft blocks and items**
- **Runs from a Java development environment**
 - Easy to debug, test new features, and contribute

Visualizing Containers

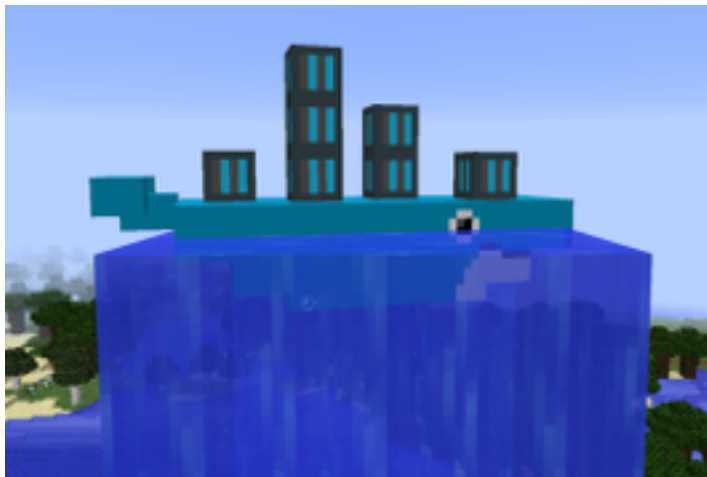
- Show containers in Minecraft
- Each container is a box with buttons and signs
 - Start/stop button
 - Detailed information button
 - Name sign
 - Image sign
- Container look changes based on status
 - Gray for running, red for stopped
- Container Wand - new Minecraft item
 - Right click on name sign to remove container
- Automatic container building at specified rate



Docker commands in Minecraft

- Common commands
 - `/docker ps`
 - `/docker run`
 - `/docker images`
- Convenience commands
 - `/docker kill_all`
 - `/docker rm_stopped`
 - `/docker heat_map`
- `/docker help` for in-game help about commands

Fun stuff



Moby the 3D Minecraft whale mascot – custom 3D model and texture



Netflix's Chaos Monkey

More fun stuff



AdityaGupta1 / mobycraft

Code Issues 24 Pull requests 0 Wiki Pulse Graphs

A Minecraft client-side mod to manage and visualize Docker containers in Minecraft

107 commits 2 branches 1 release

Branch: master New pull request Create new file Up

AdityaGupta1 committed on GitHub Merge pull request #63 from aspyker/master

images	Added Eclipse play button image for use in wiki
src	merge upstream
.gitignore	Fixed #63
LICENSE	adding license
readme.asciidoc	updated readme

bit.ly/mobycraft

What I learned from this project

- **Basic Docker terminology**
 - Container; client, engine, and hub; Docker Machine; `/docker ps`; `--help`, docs.docker.com, Docker Swarm
- **Some Java skills**
 - Access modifiers
 - `break`; and `continue`; statements
- **Design patterns**
 - DRY – Don't Repeat Yourself
- **Shading JARs in Gradle**
- **Asynchronous callbacks**
 - e.g. `StatisticsResultCallback` class

Upcoming features

- Docker Swarm cluster visualization and management ([#21](#))
 - Waiting for Docker-Java to add Swarm support
- Packaged JAR file for running in release Minecraft ([#38](#))
 - Only runs well from development environment
- Show containers from a multi-cloud Docker Swarm cluster ([#8](#))
- Add more modes to Container Wand ([#53](#))
- Add support for Docker for Mac ([#84](#))

Netflix Titus

- Container cloud at Netflix - bit.ly/QConNetflixTitus
- Video from Titus in production
 - Time-lapse - 4 hours
 - Thanks @aspyker
- Specs (at peak):
 - 130 r3.8xl's
 - 4000 vCPUs
 - 31 TB memory
 - 325 containers

Titus in Production

**Get by with a little
help from new kid's
originals this season...**

bit.ly/MobycraftTitus

Multiple Docker Providers

```
8 public class MobyCraftModule extends AbstractModule {
9     @Override
10    protected void configure() {
11        bind(MobycraftBasicCommands.class).to(org.redfrog404.mobycraft.commands.common.BasicDockerCommands.class);
12        bind(MobycraftBuildContainerCommands.class).to(org.redfrog404.mobycraft.commands.common.BuildContainerCommands.class);
13        bind(MobycraftConfigurationCommands.class).to(org.redfrog404.mobycraft.commands.dockerjava.ConfigurationCommands.class);
14
15        // for docker-machine local
16        bind(MobycraftContainerListCommands.class).to(org.redfrog404.mobycraft.commands.dockerjava.ContainerListCommands.class);
17        bind(MobycraftContainerLifecycleCommands.class).to(org.redfrog404.mobycraft.commands.dockerjava.ContainerLifecycleCommands.class);
18        bind(MobycraftImageCommands.class).to(org.redfrog404.mobycraft.commands.dockerjava.ImageCommands.class);
19        bind(MobycraftDockerClient.class).to(MobycraftDockerClientImpl.class);
20
21        // for mocking via JSON files under resources
22        // bind(MobycraftContainerListCommands.class).to(org.redfrog404.mobycraft.commands.mock.ContainerListCommands.class);
23        // bind(MobycraftContainerLifecycleCommands.class).to(org.redfrog404.mobycraft.commands.mock.ContainerLifecycleCommands.class);
24        // bind(MobycraftImageCommands.class).to(org.redfrog404.mobycraft.commands.mock.ImageCommands.class);
25
26        // for Titus
27        // bind(MobycraftContainerListCommands.class).to(org.redfrog404.mobycraft.commands.titus.ContainerListCommands.class);
28        // bind(MobycraftContainerLifecycleCommands.class).to(org.redfrog404.mobycraft.commands.titus.ContainerLifecycleCommands.class);
29        // bind(MobycraftImageCommands.class).to(org.redfrog404.mobycraft.commands.titus.ImageCommands.class);
30    }
31 }
```

How you can help - bit.ly/mobycraft

- Code review
- Pull requests
- Issues
- Testing in your environment
 - Find bugs and file issues
 - Contribute new features

If **NETFLIX** can use
Mobycraft
to manage their containers,
you can too!

Thank you!

@aditya_g

