CFEngine

The Future of In-Container Configuration Management

Diego Zamboni, Mark Burgess
CFEngine AS

diego.zamboni@cfengine.com

zzamboni

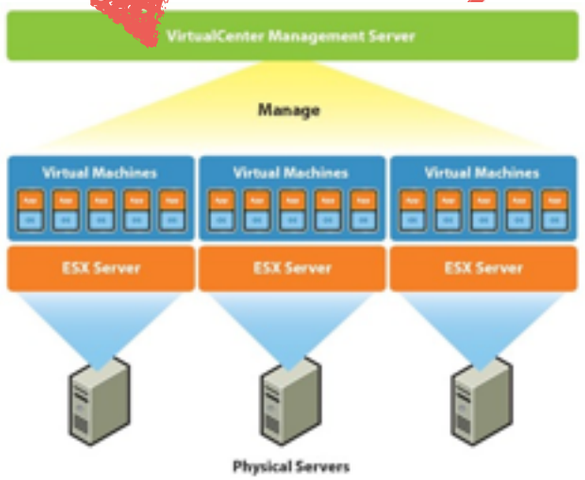
A grossly oversimplified history of computing











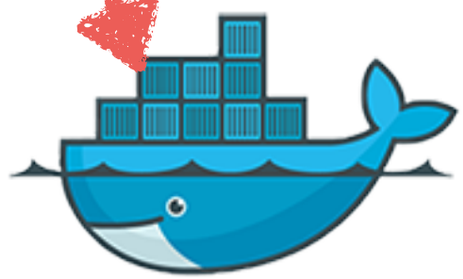


Image credits: http://blog.softlayer.com/tag/mainframe, http://www.bloomberg.com/photo/ibm-makes-1-billion-bet-on-new-mainframe-in-shrinking-business-/222911.html, http://blogs.vmware.com/virtualreality/2008/06/a-look-at-some.html, http://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-single-node-cluster/

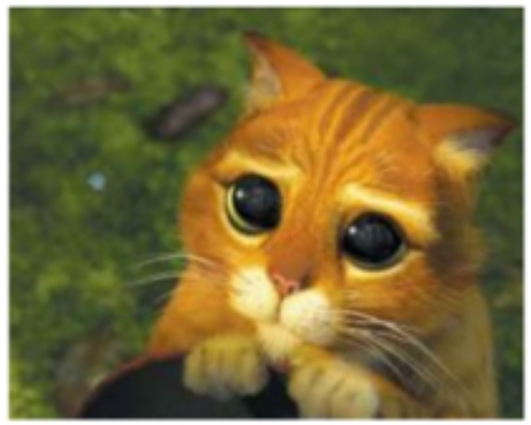
CFEngine

Pets vs cattle

Pets vs cattle

Service Model

cloudscaling



- Pets are given names like pussinboots.cern.ch
- They are unique, lovingly hand raised and cared for
- When they get ill, you nurse them back to health



- Cattle are given numbers like vm0042.cern.ch
- They are almost identical to other cattle
- When they get ill, you get another one

- Future application architectures should use Cattle but Pets with strong configuration management are viable and still needed

Gavin McCance, CERN 17

Engine

Credit: http://www.slideshare.net/gmccance/cern-data-centre-evolution

Pets vs cattle



- Custom-built server (physical or VM)
- Carefully configured (by hand or CM)
- Hard to replicate

amazon web services

- Automatically-built server (VM or container)
- Statically-configured (by hand or CM)
- Easy to replicate

CFEngine

Pets vs cattle

Traditional domain of configuration management



- Custom-built server (physical or VM)
- Carefully configured (by hand or CM)
- Hard to replicate

amazon web services

- Automatically-built server (VM or container)
- Statically-configured (by hand or CM)
- Easy to replicate

CFEngine

Configuration management in the age of containers?



CFEngine

Configuration management in the age of containers?

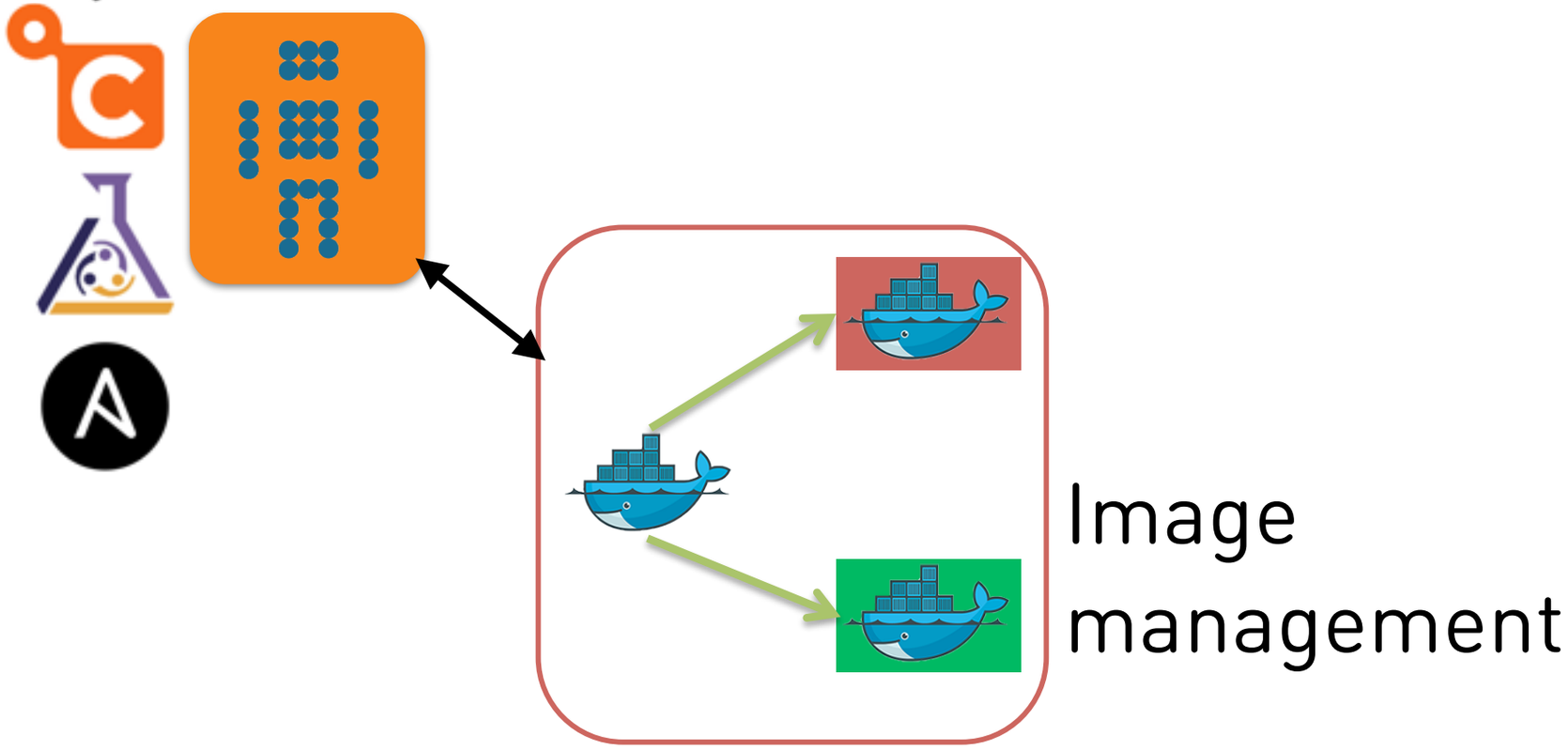
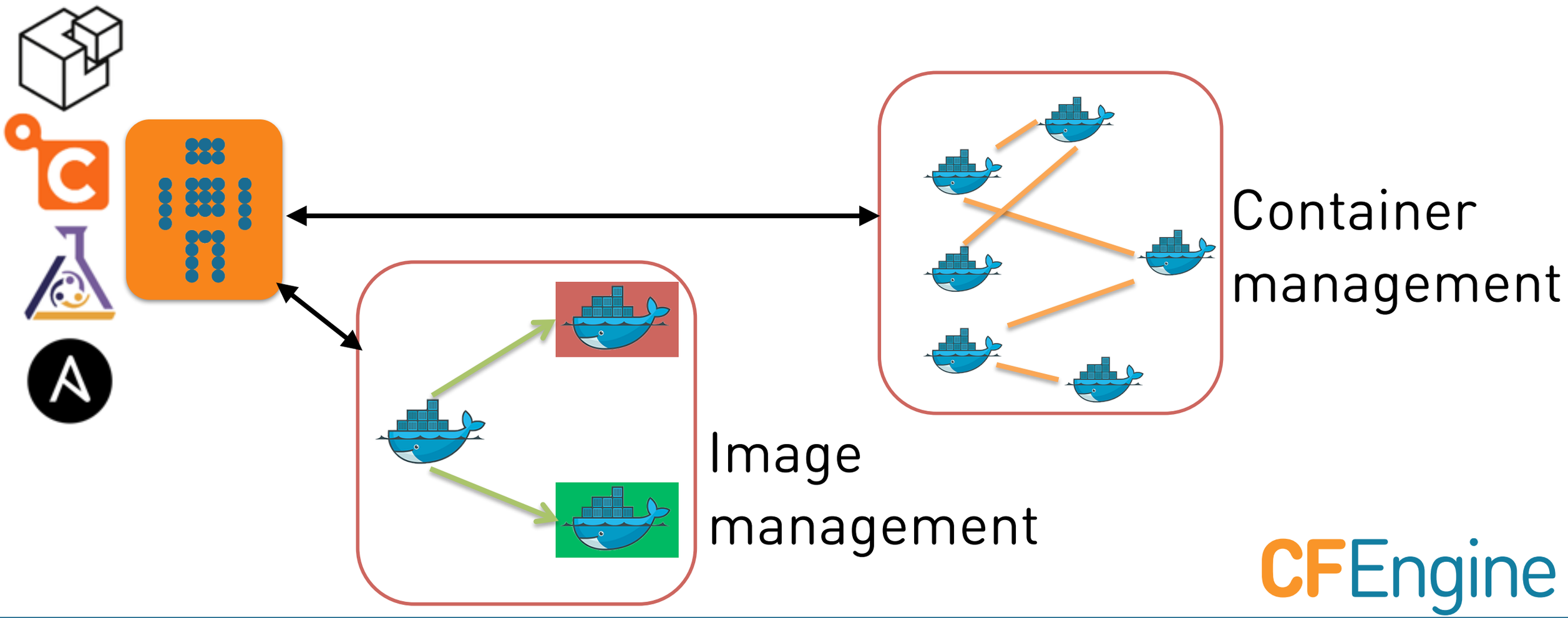


Image management

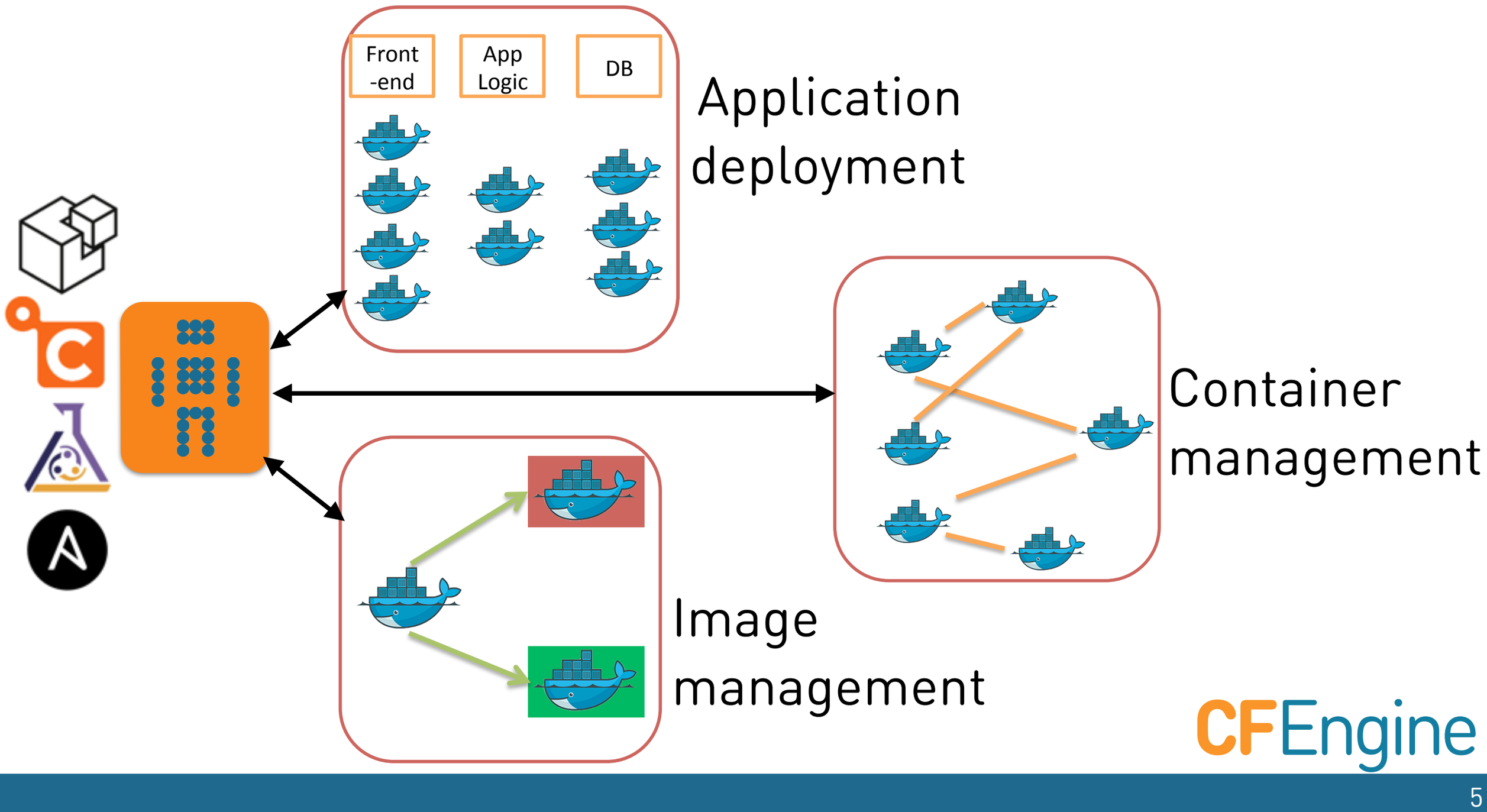
CFEngine

Configuration management in the age of containers?



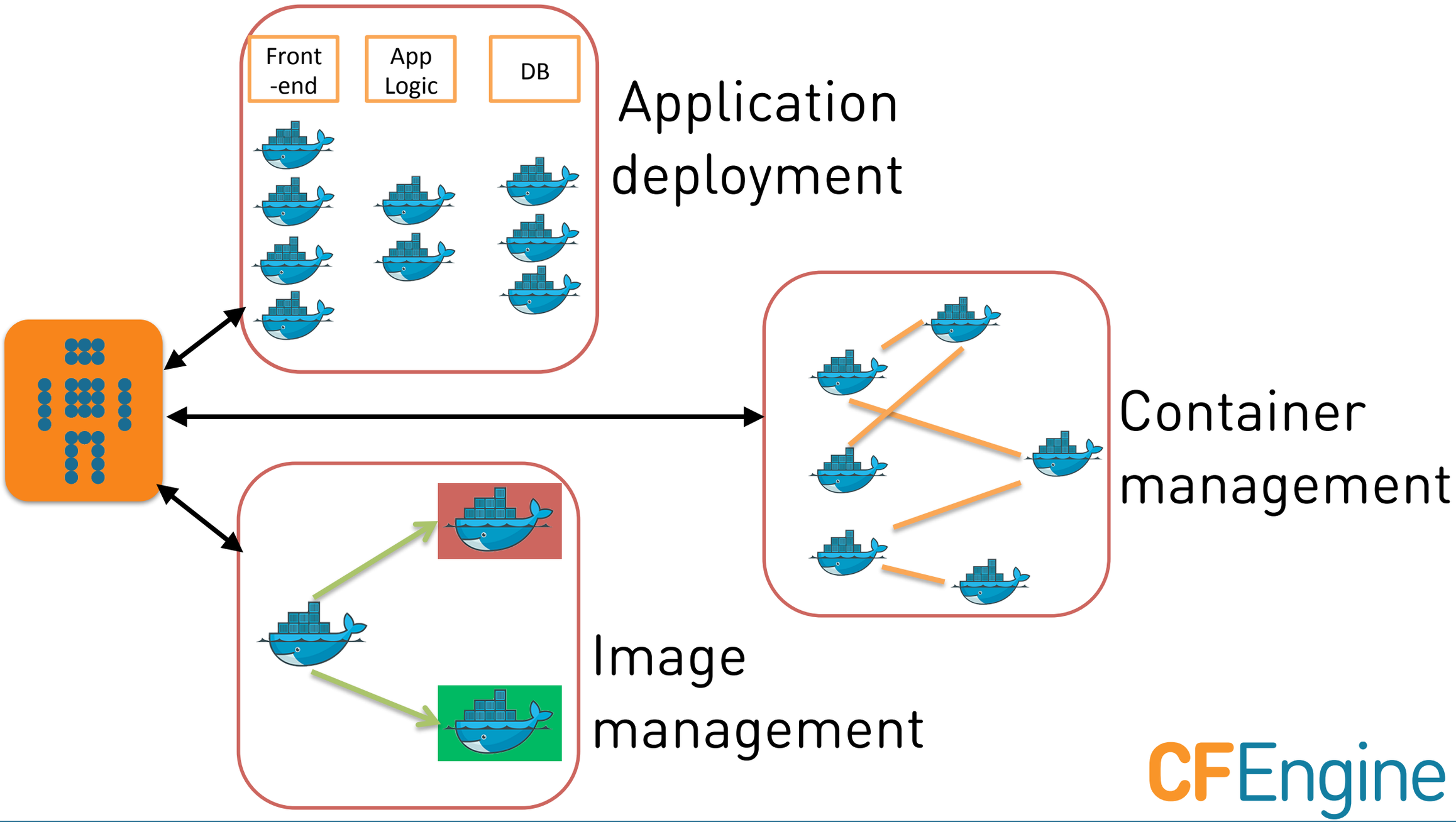
CFEngine

Configuration management in the age of containers?



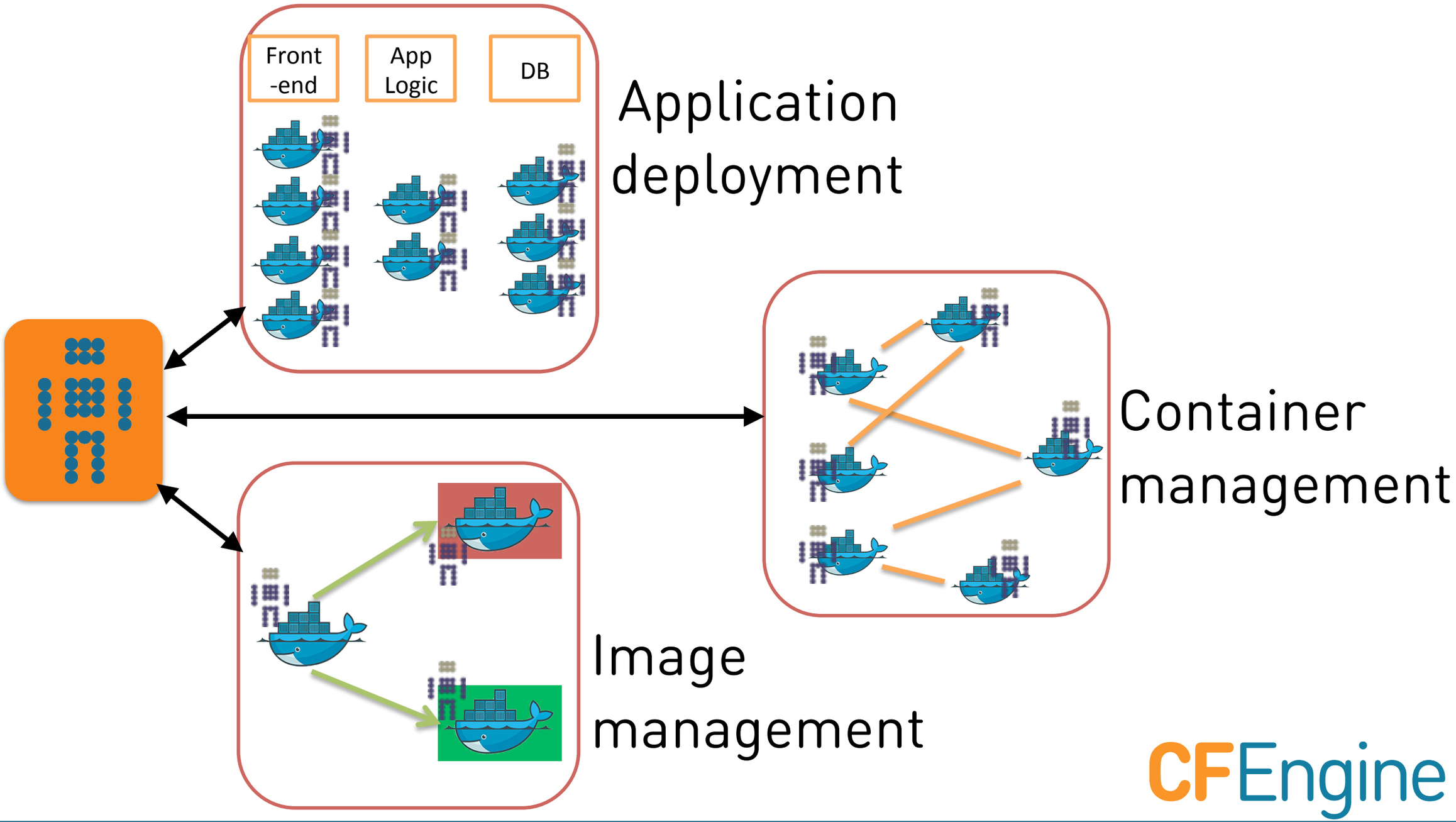
CFEngine

Configuration management in the age of containers!



CFEngine

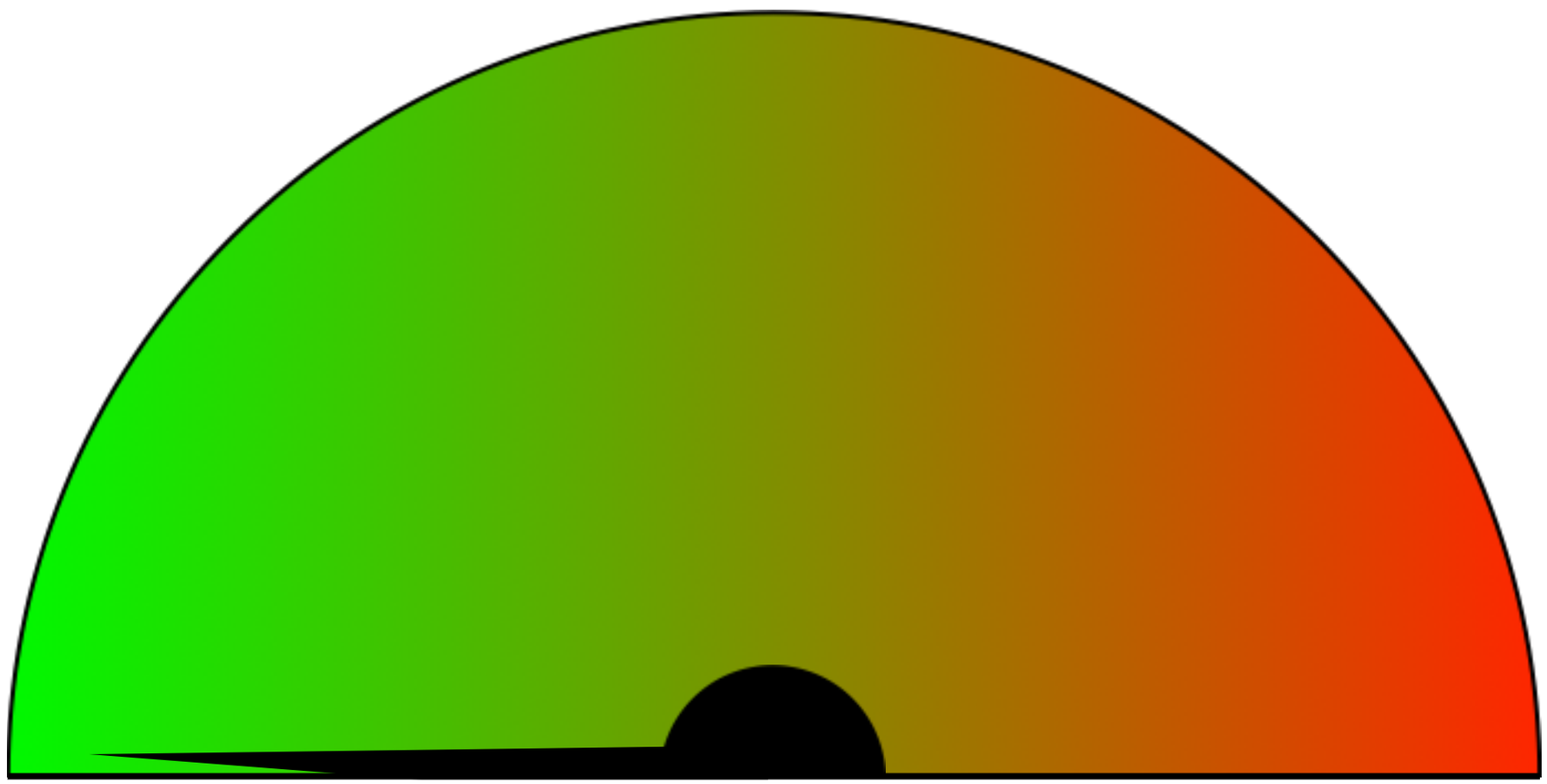
Configuration management in the age of containers!



CFEngine

The pet-cattle scale



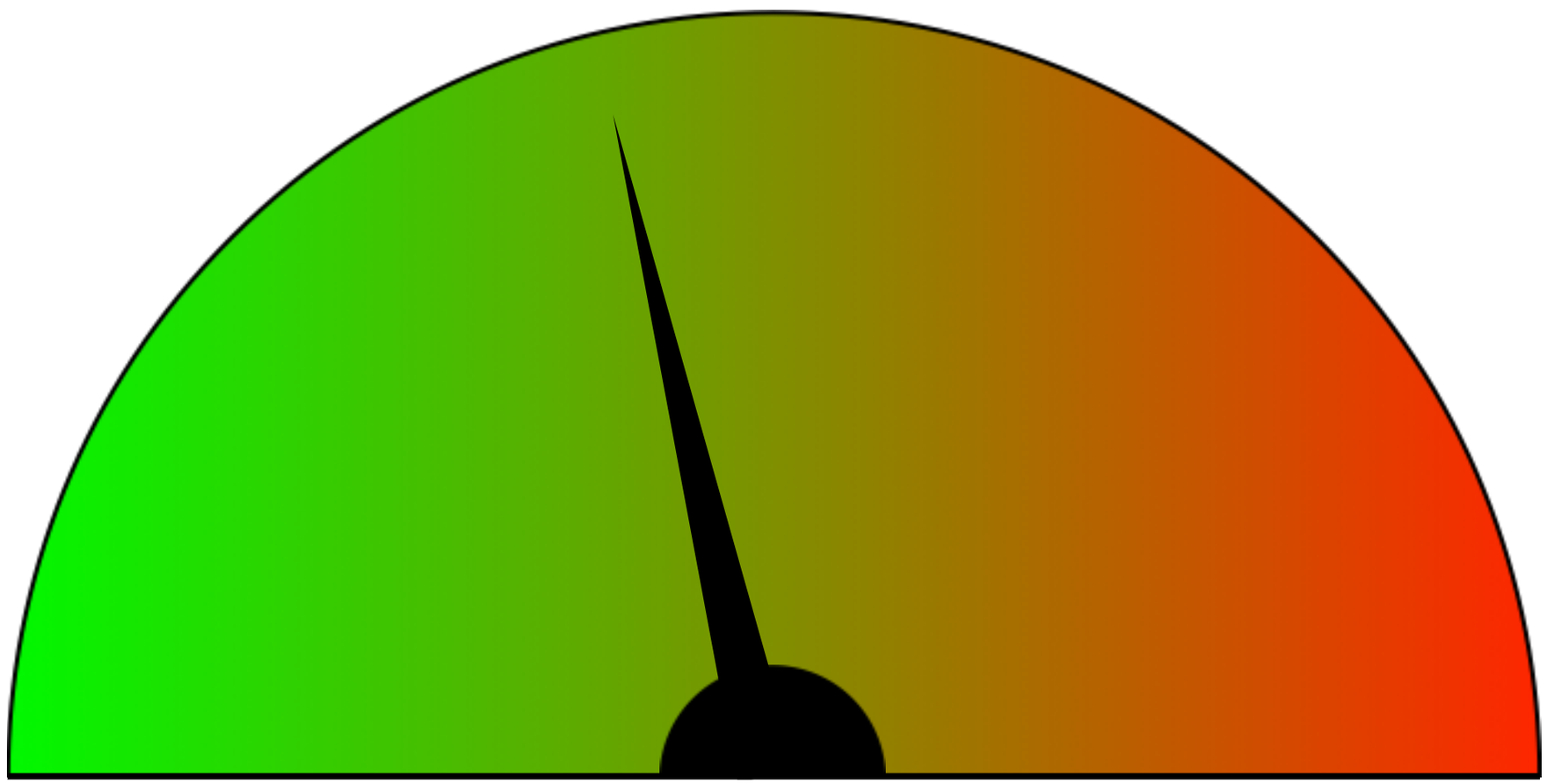




CFEngine

The pet-cattle scale







CFEngine

Balancing the scale

```
{
    "cfengine/cfubuntu": {
        "base_image": "ubuntu",
        "install_cfengine": true,
        "packages": [ "openssh-server" ],
        "ports": [ 22 ]
    },
    "cfengine/cfdbserver": {
        "base_image": "cfengine/cfubuntu",
        "packages": [ "mysql-server", "mysql-client" ],
        "ports": [ 3306 ]
    },
    "cfengine/cfwebserver": {
        "base_image": "cfengine/cfubuntu",
        "packages": [ "apache2", "libapache2-mod-php5",
                      "php5-mysql" ],
        "ports": [ 80, 443 ]
    },
    "cfengine/cfwordpress": {
        "base_image": "cfengine/cfwebserver",
        "extralines": [
            "RUN rm -rf /var/www/html",
            "ADD http://wordpress.org/latest.tar.gz /wordpress.tar.gz",
            "RUN tar xvzf /wordpress.tar.gz",
            "RUN mv /wordpress /var/www/html",
            "RUN chown -R www-data:www-data /var/www/"
        ],
    },
    "cfengine/cfhaserver": {
        "base_image": "cfengine/cfubuntu",
        "packages": [ "haproxy" ],
        "ports": [ 80, 443 ]
    },
}
```

CFEngine

Balancing the scale

```
{
    "cfengine/cfubuntu": {
        "base_image": "ubuntu",
        "install_cfengine": true,
        "packages": [ "openssh-server" ],
        "ports": [ 22 ]
    },
    "cfengine/cfdbserver": {
        "base_image": "cfengine/cfubuntu",
        "packages": [ "mysql-server", "mysql-client" ],
        "ports": [ 3306 ]
    },
    "cfengine/cfwebserver": {
        "base_image": "cfengine/cfubuntu",
        "packages": [ "apache2", "libapache2-mod-php5",
                      "php5-mysql" ],
        "ports": [ 80, 443 ]
    },
    "cfengine/cfwordpress": {
        "base_image": "cfengine/cfwebserver",
        "extralines": [
            "RUN rm -rf /var/www/html",
            "ADD http://wordpress.org/latest.tar.gz /wordpress.tar.gz",
            "RUN tar xvzf /wordpress.tar.gz",
            "RUN mv /wordpress /var/www/html",
            "RUN chown -R www-data:www-data /var/www/"
        ],
    },
    "cfengine/cfhaserver": {
        "base_image": "cfengine/cfubuntu",
        "packages": [ "haproxy" ],
        "ports": [ 80, 443 ]
    },
}
```

CFEngine

Balancing the scale

```
{
    "cfengine/cfubuntu": {
        "base_image": "ubuntu",
        "install_cfengine": true,
        "packages": [ "openssh-server" ],
        "ports": [ 22 ]
    },
    "cfengine/cfdbserver": {
        "base_image": "cfengine/cfubuntu",
        "packages": [ "mysql-server", "mysql-client" ],
        "ports": [ 3306 ]
    },
    "cfengine/cfwebserver": {
        "base_image": "cfengine/cfubuntu",
        "packages": [ "apache2", "libapache2-mod-php5",
                      "php5-mysql" ],
        "ports": [ 80, 443 ]
    },
    "cfengine/cfwordpress": {
        "base_image": "cfengine/cfwebserver",
        "extralines": [
            "RUN rm -rf /var/www/html",
            "ADD http://wordpress.org/latest.tar.gz /wordp
            "RUN tar xvzf /wordpress.tar.gz",
            "RUN mv /wordpress /var/www/html",
            "RUN chown -R www-data:www-data /var/www/"
        ],
    },
    "cfengine/cfhaserver": {
        "base_image": "cfengine/cfubuntu",
        "packages": [ "haproxy" ],
        "ports": [ 80, 443 ]
    },
}
```

```
{
    "dbserver": {
        "image": "cfengine/cfdbserver",
        "options": "-p 22"
    },

    "webserver": {
        "howmany": 5,
        "image": "cfengine/cfwordpress",
        "links": { "dbserver": "db" },
        "options": "-p 22"
    },

    "haserver": {
        "image": "cfengine/cfhaserver",
        "links": { "webserver": "web" },
        "options": "-p 80 -p 443 -p 22"
    },
}
```

CFEngine

Balancing the scale

```
{
    "cfengine/cfubuntu": {
        "base_image": "ubuntu",
        "install_cfengine": true,
        "packages": [ "openssh-server" ],
        "ports": [ 22 ]
    },
    "cfengine/cfdbserver": {
        "base_image": "cfengine/cfubuntu",
        "packages": [ "mysql-server", "mysql-clie
        "ports": [ 3306 ]
    },
    "cfengine/cfwebserver": {
        "base_image": "cfengine/cfubuntu",
        "packages": [ "apache2", "libapache2-mod-
                      "php5-mysql" ],
        "ports": [ 80, 443 ]
    },
    "cfengine/cfwordpress": {
        "base_image": "cfengine/cfwebserver",
        "extralines": [
            "RUN rm -rf /var/www/html",
            "ADD http://wordpress.org/latest.tar.
            "RUN tar xvzf /wordpress.tar.gz",
            "RUN mv /wordpress /var/www/html",
            "RUN chown -R www-data:www-data /var/
        ],
    },
    "cfengine/cfhaserver": {
        "base_image": "cfengine/cfubuntu",
        "packages": [ "haproxy" ],
        "ports": [ 80, 443 ]
    },
}
```

```
bundle agent service_catalogue
{
  methods:

    any::
      "docker" usebundle => docker_discovery,
                 comment => "Discover if we are in a docker container";

      "ssh"    usebundle => setup_ssh,
                 comment => "Start ssh and ensure correct keys are in";

    am_policy_hub::
      "setup_docker" usebundle => docker_infrastructure,
                       comment => "Set up docker containers";

    web_server::
      "wordpress"    usebundle => config_wordpress,
                       comment => "Configure wordpress on the local host";

    db_server::
      "mysql"        usebundle => config_mysql,
                       comment => "Configure MySQL for wordpress";

    load_balancer::
      "haproxy"      usebundle => config_haproxy,
                       comment => "Configure haproxy for load balancer";
}
```

CFEngine

In-container/off-container CM

Image size

Runtime size

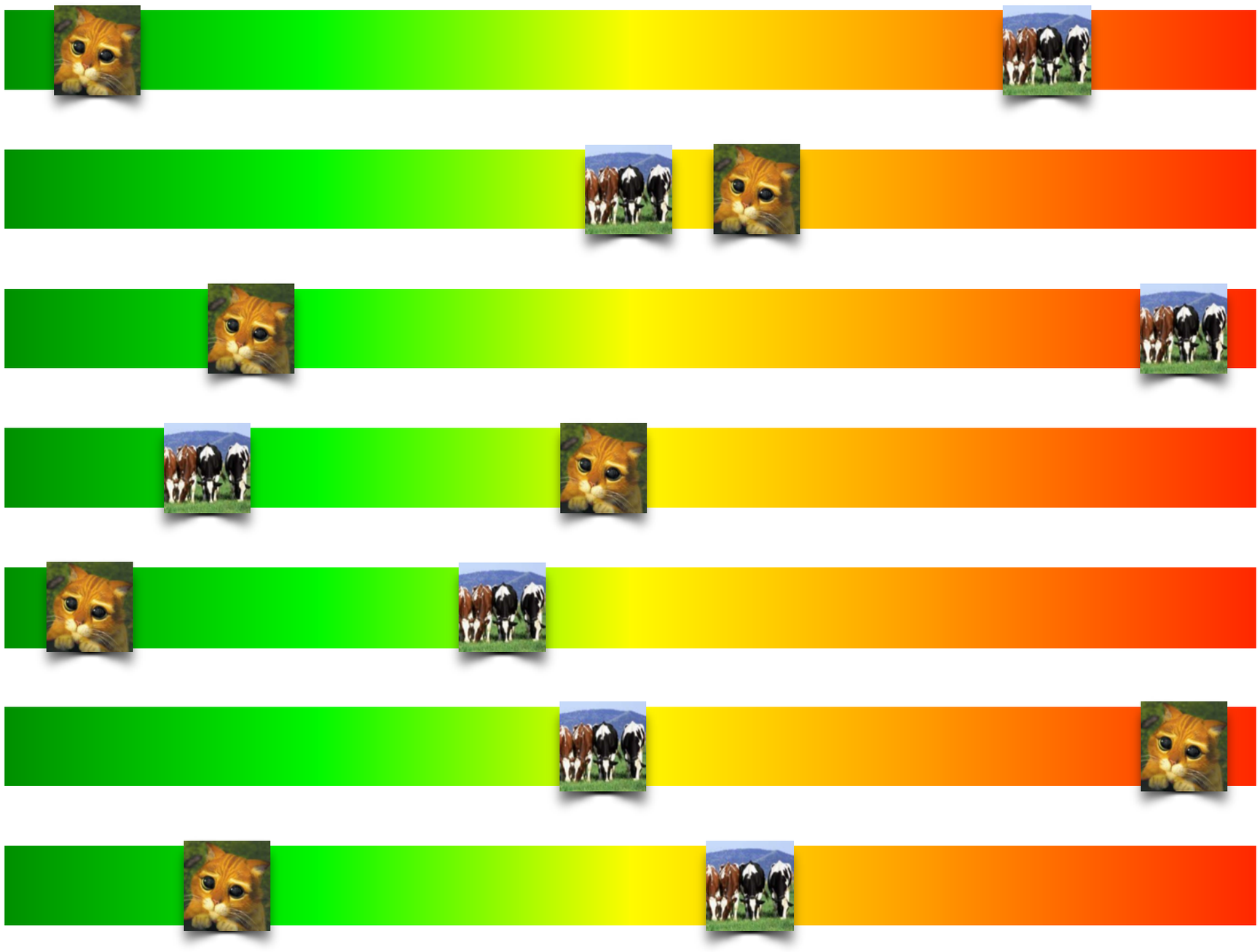
Image/Dockerfile proliferation

Startup time

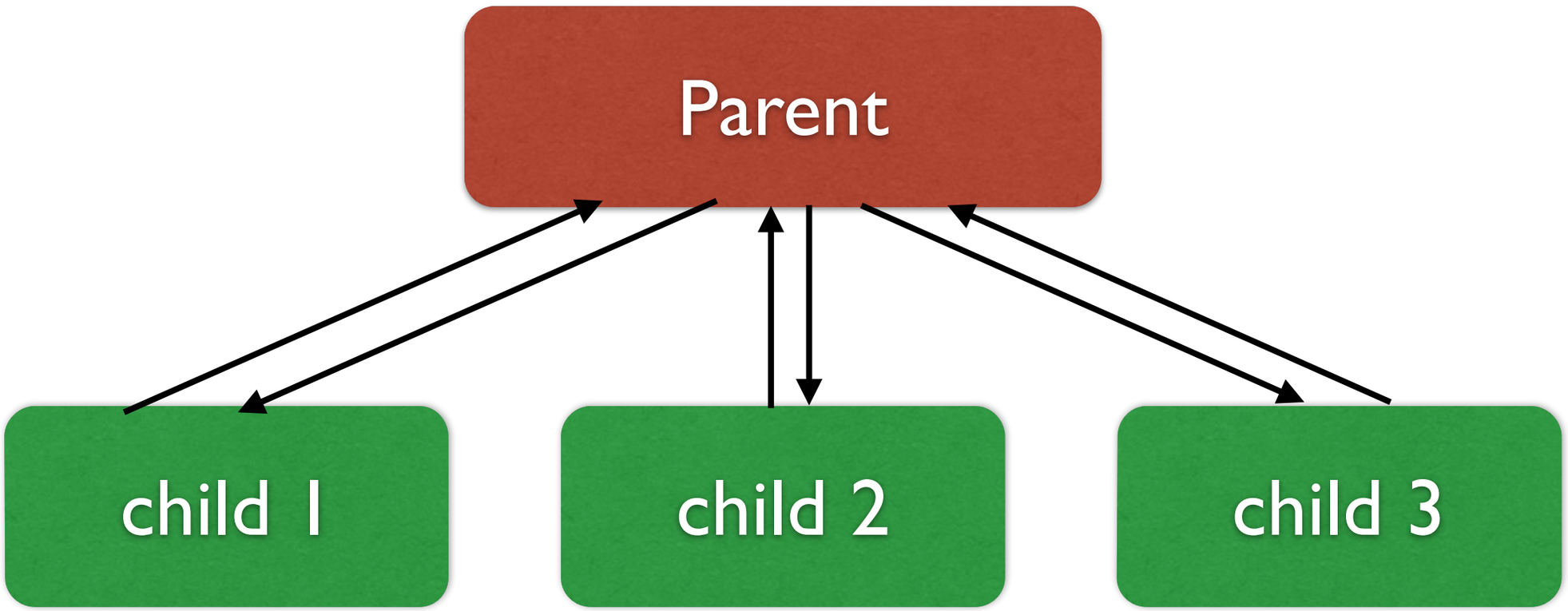
Runtime changes

Immutable infrastructure

Repeatability across infrastructures



Orchestration



CFEngine

Orchestration

```
bundle agent main
{
  methods:

    parent::

      "plant continuous delivery seeds" comment => "The files where people make changes";
      "pick the flowers"                comment => "Reap the harvest";
      "package for delivery"            comment => "Tie it with a bow";

    bud_containers::

      "feed from the parent" comment => "pick up files and build something";

  services:
    parent::
      "budding children" comment => "Spawn some fledgings";
}
```

CFEngine

Orchestration

```
bundle agent main
{
  methods:

    parent::

      "plant continuous delivery seeds"
      "pick the flowers"
      "package for delivery"

    bud_containers::

      "feed from the parent" comment =>

  services:
    parent::
      "budding children" comment => "Sp
}
```

```
bundle agent service_budding_children(name,state)
{
  guest_environments:

    parent.!cleanup::

      "$(my.buds)"
            guest_details => stem_cell,
            guest_state   => "create";

    cleanup::

      "$(my.buds)"
            guest_details => stem_cell,
            guest_state   => "delete";
}

##########################################################

body guest_details stem_cell
{
              guest_type => "docker";
        guest_image_name => "cf-stem-cell";
}
```

What do we need from in-container CM?

- Lightweight
- Distributed
- Resilient

CFEngine

CFEngine

Thank you

Diego Zamboni
diego.zamboni@cfengine.com