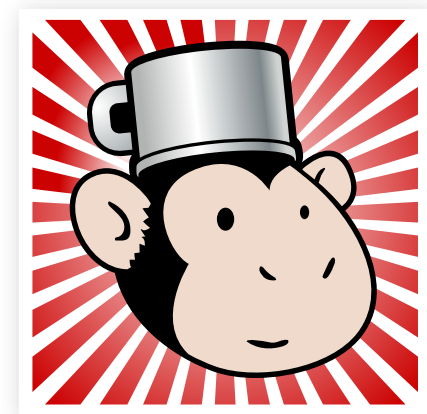


# JUDCon

JBoss Users & Developers Conference

London:2011



# BoxGrinder

Marc Savy



# Agenda

- Background, terminology and rationale
- BoxGrinder
  - Appliance definition files
  - Architecture overview
    - Build process
    - Writing a plugin
- Small demo

An **appliance** is a preconfigured disk image (virtual machine), with the operating system and requisite software to fulfil a specific task.



# Appliances, Appliances, Appliances

For any architecture, **those components we want to scale independently** are separate appliances.

# Consider a few examples...

- Classic n-tier layered architecture
  - Caching & load balancing
  - Presentation
  - Business logic
  - Data
- Master-Worker, MVC...



# Bake vs. Fry

**Bake:** Produce a complete virtual machine offline, before first use.

**Fry:** Produce a skeleton virtual machine, by booting a basic VM and then applying configuration.

# Out of the frying pan...

**Frying is usually slower**, waiting for essential components to be retrieved, installed and launched. **Repeatedly incurring time and performance costs** that slow deployment.



# Bake!

We think **baking** is The Right Way, especially for developers simply looking for reliable platforms.

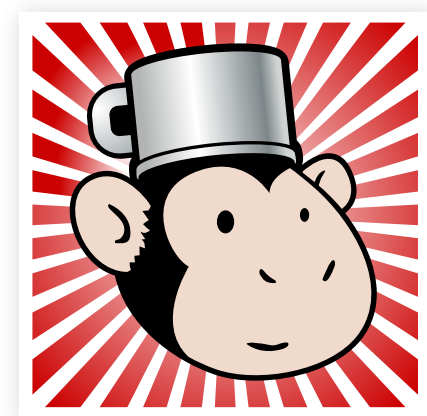
# Into the oven

**Bake once, use many times.**  
Bake again when stale.



# Bake, then fry

If you **bake** an image you can **fry** it later too! The baked image can be your **starting point**.



# BoxGrinder





BoxGrinder is a family of tools to grind out **appliances** for various **platforms**

# Why might I love BoxGrinder?

- Produce **lean and mean** appliances, with no bloat.
- **Fast and efficient.** Build quickly in the format you need, and deliver to the place you want, to exact specification.
- **Extensible.** Adding functionality is easy.
- Handles complexities of **dependency resolution** and **remote service interactions** on your behalf.
- **Repeatable process** that minimises arduous manual labour.





# BoxGrinder

**Build**



# BoxGrinder

**REST**



# BoxGrinder

**Studio**



## Current status

---



**BoxGrinder**  
Build



stable



**BoxGrinder**  
REST



development

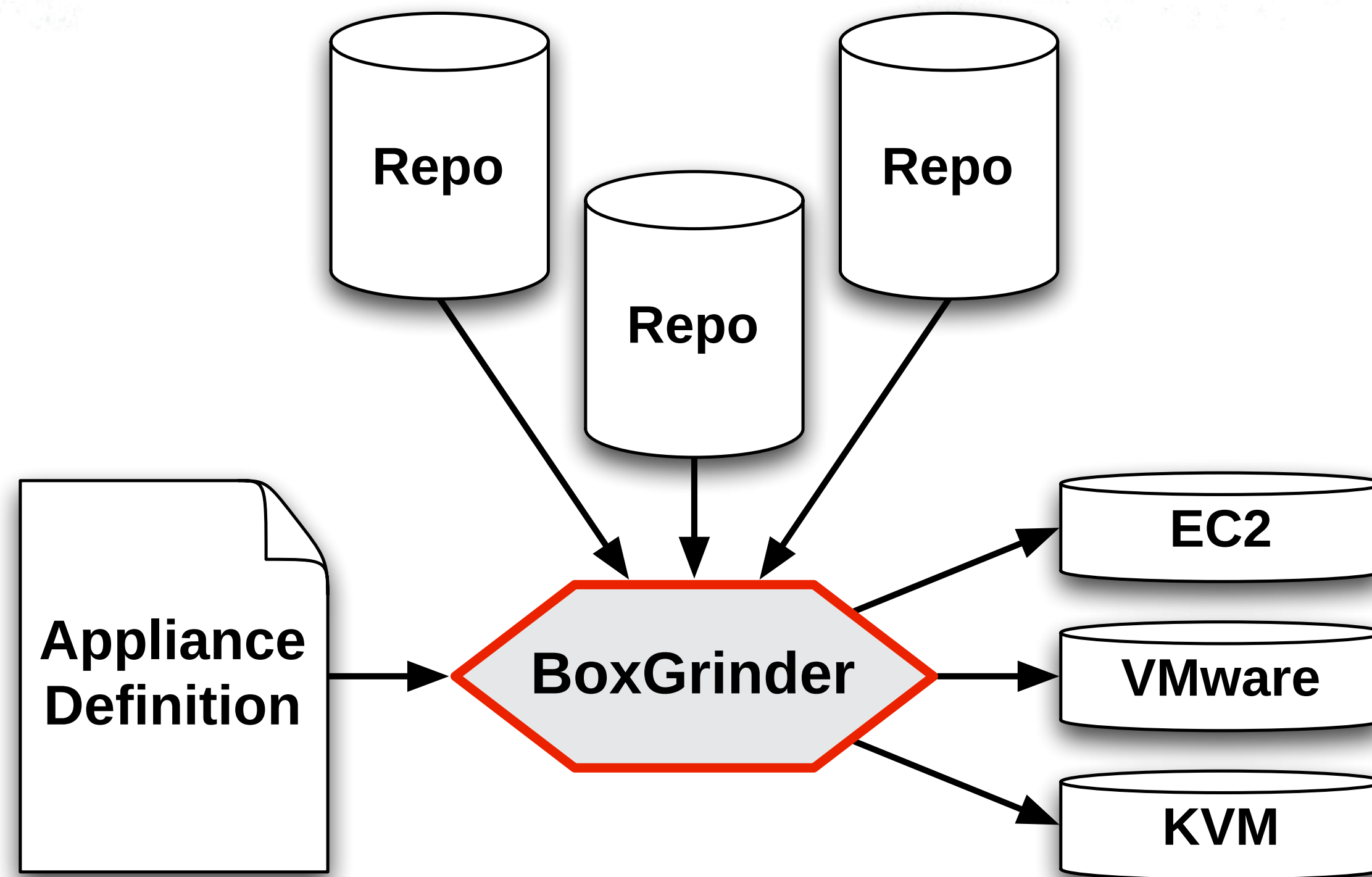


**BoxGrinder**  
Studio



planning





# A closer look at **appliance definition files**



# Appliance definition, huh?

- Plain text file – **YAML** format
- Very easy to understand, modify
- Inheritance (mixins)

# Appliance example

```
name: back-end
version: 1
release: 1
summary: back-end appliance with JBoss AS 6
hardware:
  memory: 512
  partitions:
    "/":
      size: 2
appliances:
  - fedora-base
packages:
  - jboss-as6
  - jboss-as6-cloud-profiles
  - java-1.6.0-openjdk
...
```



# General information

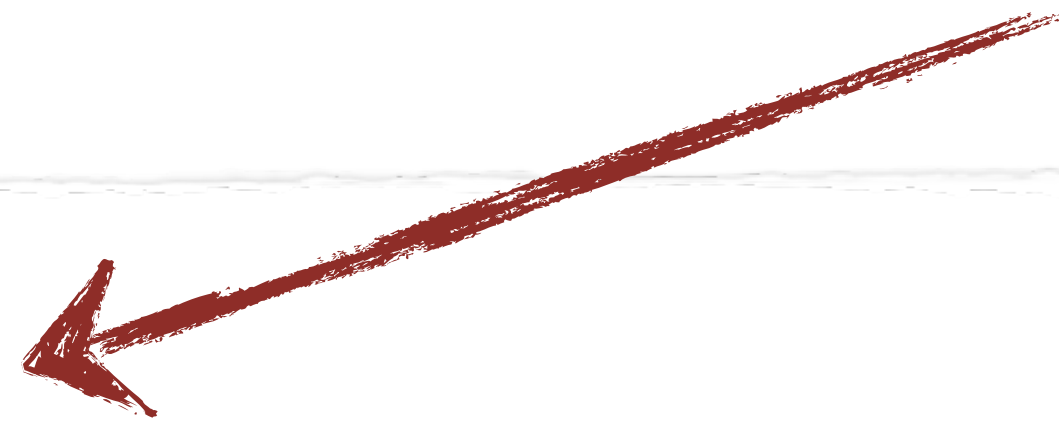
should match the filename: back-end.appl

name: back-end

version: 1

release: 1

summary: back-end appliance with JBoss AS 6



# Hardware

hardware:

memory: 512

partitions:

"/":

size: 2

**512MB**

**2GB**





# Appliance Mix-ins

Mixing in **fedora-base.appl**



```
appliances:  
- fedora-base
```

# back-end.appl

```
name: back-end
version: 1
release: 1
summary: back-end appliance with JBoss AS 6
hardware:
  memory: 512
  partitions:
    "/":
      size: 2
appliances:
  - fedora-base
packages:
  - jboss-as6
  - jboss-as6-cloud-profiles
  - java-1.6.0-openjdk
...
```



# fedora-base.appl

```
name: fedora-base
summary: Basic Fedora OS
os:
  name: fedora
  version: 16
hardware:
  memory: 256
  partitions:
    "/":
      size: 1
packages:
  - @core
  - openssh-server
  - openssh-clients
  - wget
```

# Appliance Mix-ins

**back-end.appl**

**fedora-base.appl**

*overrides*



```
hardware:  
  memory: 512  
  partitions:  
    "/":  
      size: 2
```

```
hardware:  
  memory: 256  
  partitions:  
    "/":  
      size: 1
```



# Appliance content

packages:

- jboss-as6
- jboss-as6-cloud-profiles
- java-1.6.0-openjdk

# Appliance content

packages:

- jboss-as6
- jboss-as6-cloud-profiles
- java-1.6.0-openjdk

Plus everything from  
**fedora-base.app1**

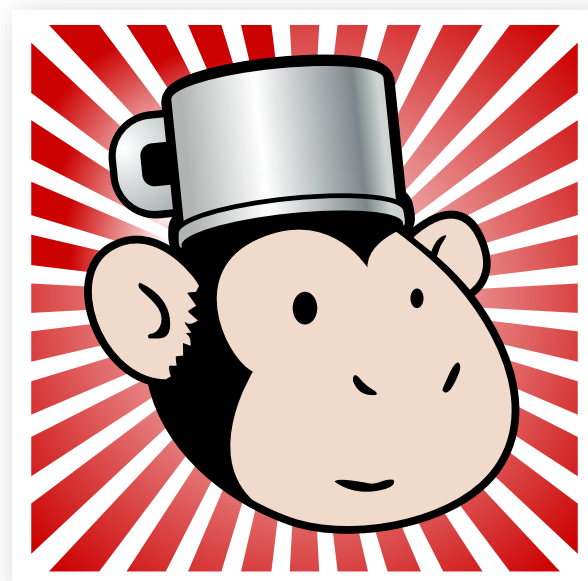


# There is a lot more!

- Additional sections
  - **files**
    - Add arbitrary files and directories to your appliance
    - Great when an RPM is not appropriate
  - **repos**
    - ephemeral repos
  - **post**
    - What should be done **after** you build your appliance
    - Different commands for different platforms
    - Using **libguestfs**



<http://boxgrinder.org/tutorials/appliance-definition/>

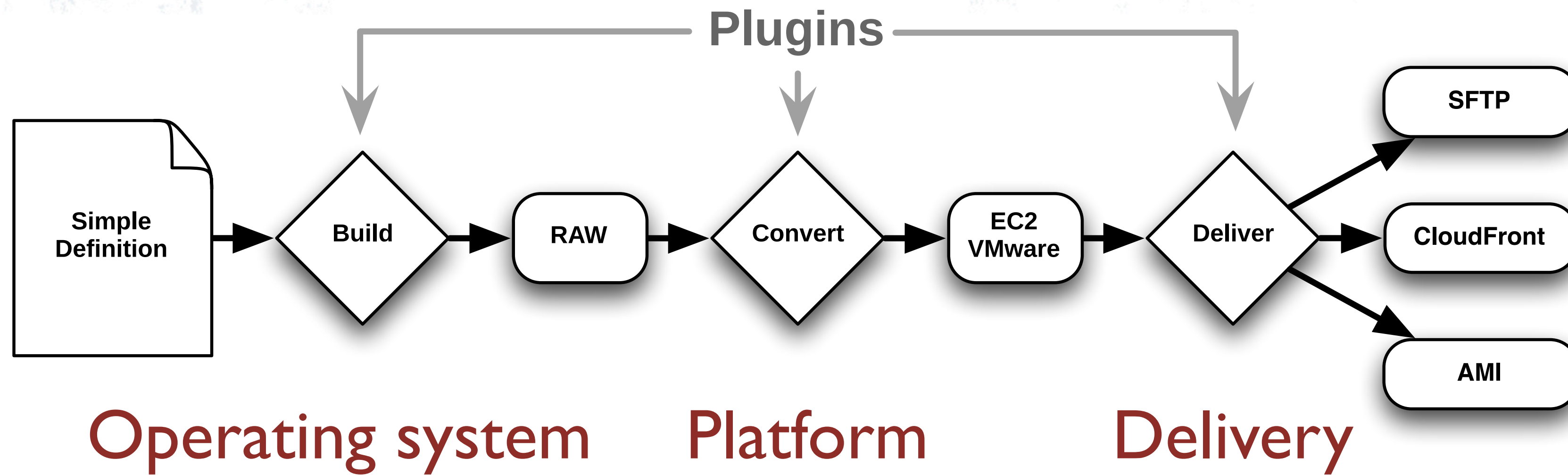


# BoxGrinder

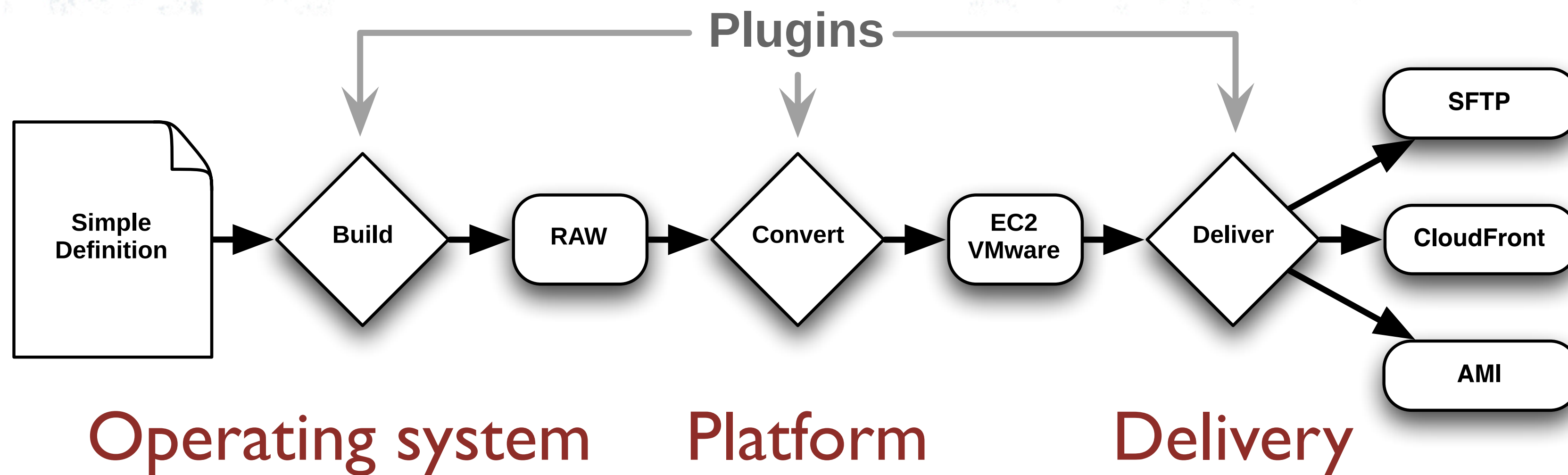
## Build



# BoxGrinder Build **architecture**







Write your **own** plugins, it's easy!

<http://boxgrinder.org/tutorials/how-to-write-a-plugin-for-boxgrinder-build/>

# Plugin skeleton

```
require 'boxgrinder-build/managers/plugin-manager'  
require 'boxgrinder-build/plugins/base-plugin'
```

```
class YourPlugin < BoxGrinder::BasePlugin  
  def execute  
    # PLACE YOUR CODE HERE  
  end  
end
```

```
plugin :class => YourPlugin,  
      :type => :platform,  
      :name => :mycloud,  
      :full_name => "MyCloud"
```



# Plugin skeleton

```
require 'boxgrinder-build/managers/plugin-manager'  
require 'boxgrinder-build/plugins/base-plugin'
```

```
class YourPlugin < BoxGrinder::BasePlugin  
  def execute  
    # PLACE YOUR CODE HERE  
  end  
end
```

```
plugin :class => YourPlugin,  
       :type => :platform,  
       :name => :mycloud,  
       :full_name => "MyCloud"
```

This will be removed  
in future releases



# Plugin registration

```
require 'boxgrinder-build/managers/plugin-manager'  
require 'xyz-your-plugin/your-plugin'
```

```
plugin :class => YourPlugin,  
      :type => :platform,  
      :name => :mycloud,  
      :full_name => "MyCloud"
```



# How to **install** BoxGrinder Build

fedora 



```
yum install rubygem-boxgrinder-build
```

# Meta appliance

Appliance preconfigured appliance to **build other appliances** using BoxGrinder

- Easy to jump in
  - Available for different platforms: Xen, KVM, EC2, VMware
- **Best way** to build EC2 appliances

<http://boxgrinder.org/download/boxgrinder-build-meta-appliance/>



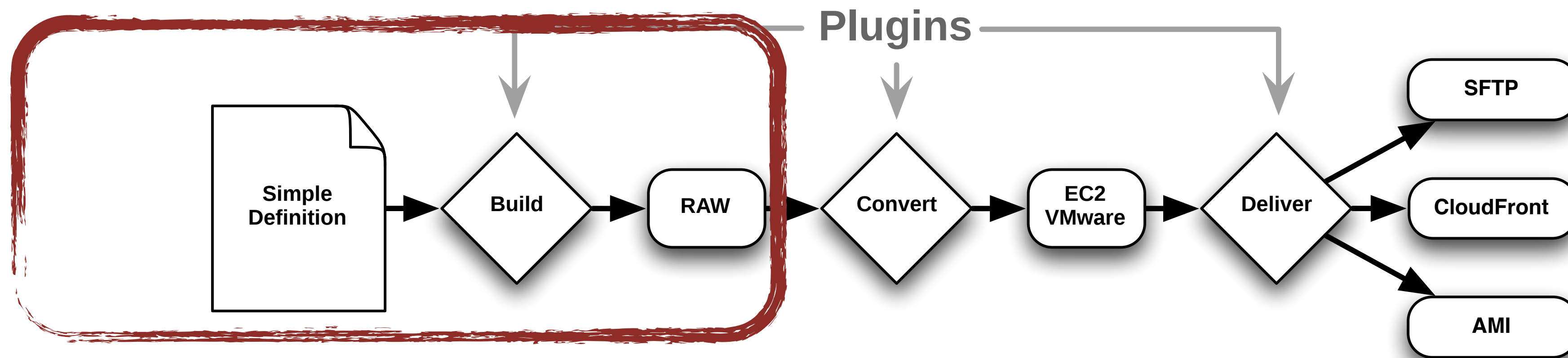
# Demo: build a simple appliance

**convert and deliver**  
**Demo: ~~build~~ a simple appliance**



# Demo setup and plan

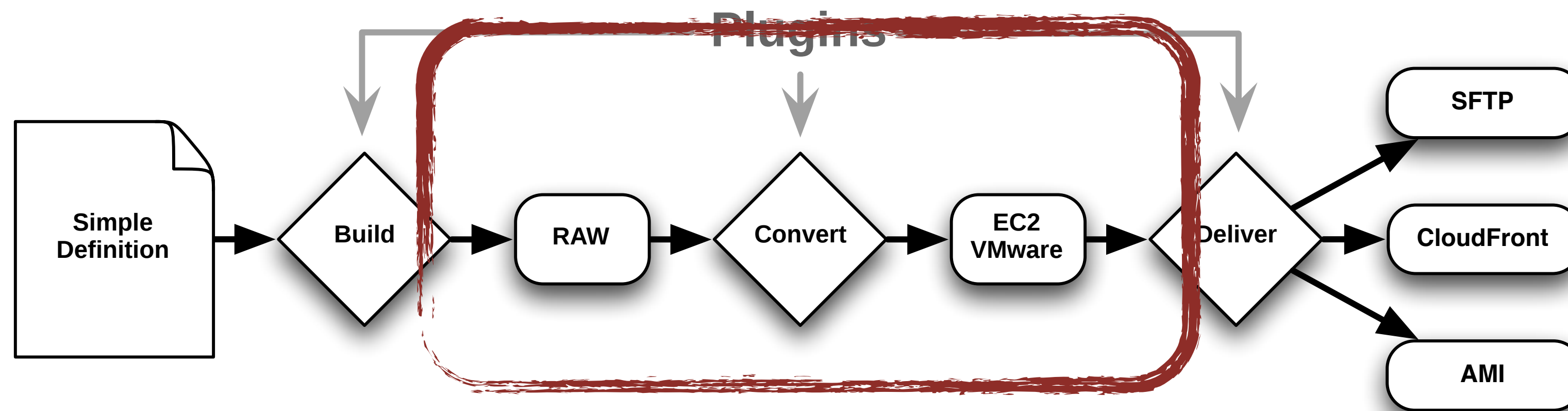
# Step 1: create base image



```
boxgrinder-build f15-jeos.app1
```

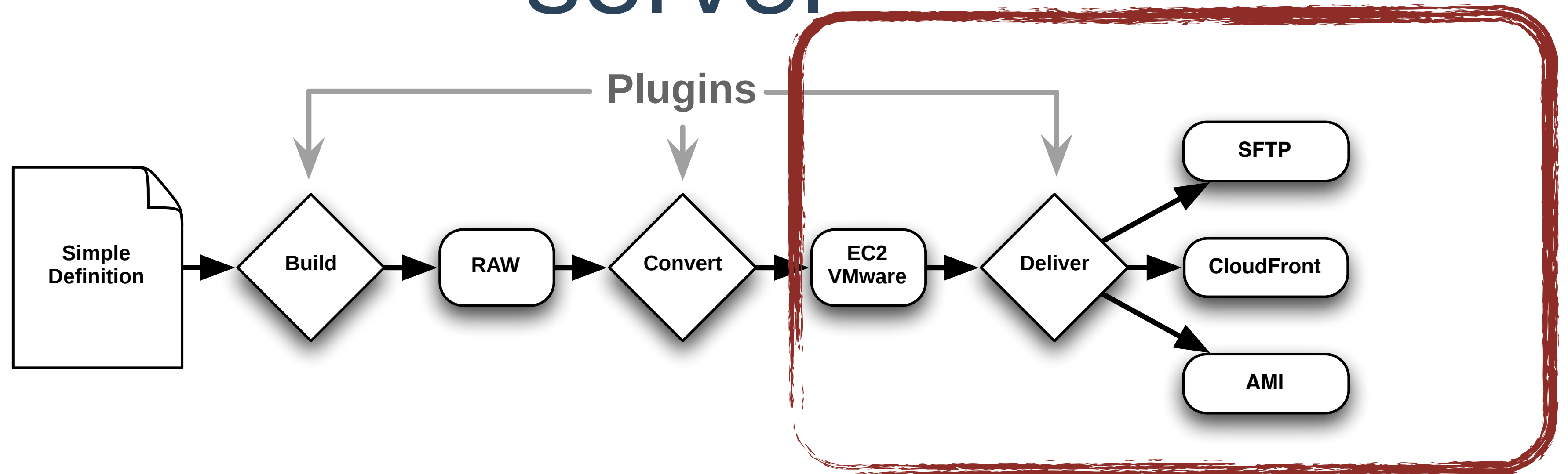


# Step 2: convert it to VMware type



```
boxgrinder-build f15-jeos.app1 -p vmware  
                                -p ec2  
                                . . .
```

# Step 3: deliver it to a SFTP server



```
boxgrinder-build f15-jeos.app1 -p vmware -d sftp  
-d ebs  
-d s3
```



Of course you can run the command  
just **once** with same result!

```
boxgrinder-build f15-jeos.app1 -p vmware -d sftp
```

What's **hot**?



# BoxGrinder Build features

- **Supported OSes: Fedora (14-16), CentOS (5), SL (5-6), RHEL (5-6)**
- **Supported platforms: EC2** (S3-based and EBS-based too!), **KVM, VMware, VirtualBox**
- **Many delivery options: local, SFTP, S3 or CloudFront** as tarred image, **AMI**
- **Many public clouds supported:** EC2, ElasticHosts, Serverlove, Open Hosting, SKALI Cloud, CloudSigma

# BoxGrinder Build features

- **Direct injection of files**
- **Cross-arch builds:** producing i386 images on x86\_64 hosts
- Caching downloaded resources (RPM's)
- **Pretty fast** – from .appl to registered AMI: **15 minutes** (on EC2, using meta-appliance)
- **Scripts:** include arbitrary shell script in the appliance



# Coming soon...

- We have a **variety of new features** arriving in our next major **release 0.10.0**. Amongst which:
  - **OpenStack**
  - **libvirt**
  - **CloudStack** on the horizon

# Notes

- If you're building AMI's – **do it on EC2** – this will save your time (uploading to S3 from your local machine isn't fun...)
- When building **EBS-based AMI's** you must run BoxGrinder on an EC2 instance



# Questions?

<http://github.com/boxgrinder/>

<http://boxgrinder.org/>

<http://boxgrinder.org/blog/>

IRC: **#boxgrinder** on Freenode



**@boxgrinder**

@marekgoldmann

@marcsavy

# Thanks!

... and a particular mention for Marek Goldmann from whom I pilfered many of these slides and diagrams.