Your First Day with JRuby
Me

• Charles Oliver Nutter
• Java dev since 1997
• Rubyist, JRuby dev since 2004
• Full time JRuby dev since 2006
• headius@headius.com, headius on Twitter
• blog.headius.com
Engine Yard

- Engine Yard Cloud
- C Ruby and JRuby (real soon)
- Fire and forget
- Slick management console
- JRuby developer support (real soon)
- Inexpensive per-seat dev support
Rails in the Cloud

On-demand deployment and management of your Ruby on Rails applications with Engine Yard Cloud.

One-click code deploys, application cloning, data automation, gem management and more!

Discover Engine Yard Cloud
Engine Yard Cloud Features and Benefits

**Managed Application Runtime**
Engine Yard Cloud provides you with a pre-integrated, pre-tested Ruby on Rails technology stack, including web, application and database servers, built-in monitoring and process management, a Rails-optimized linux distribution, in-memory caches and more. All components are monitored for security vulnerabilities and updated continuously as vulnerabilities are discovered and patched.

**Monitoring & Alerting**
Engine Yard Cloud tracks all the essential resource utilization for your application and alerts you when you need additional capacity, or when your latest application push is behaving poorly. Storage, CPU and memory utilization levels are all tracked for conformance to pre-configured thresholds, and email alerts provide timely warnings.

**Seamless Application Capacity Management**
Configuring new applications capacity used to be a multi-hour task—now it’s a single click. With Engine Yard Cloud, capacity management is always application aware. Add an additional instance to your application tier, and that instance will automatically deploy with the correct application configuration and join your load-balanced group.
JRuby

- Current version: 1.4.0
- Ruby for the JVM
  - A beautiful, simple, powerful language
- Rails for the Java platform
- “Just another JAR file”
- Interop with Java easily
The First Day

- Setup
- Learning Ruby and Rails
- Tools
- JRuby and Java
- Deployment
Getting JRuby

• Download from JRuby.org
  • Windows installer + tar/zip
  • Packages for Linux, MacPorts
    • (Sometimes a little behind)
• Download an IDE (more on that later)
• OS X installer for 1.5.0
Latest release – 1.4.0
View Release Notes
Download: tar, zip, exe, exe+jre
Installation

- tar/zip - Unpack, put in PATH
- exe/mpkg - Just run the installer
- Other packaging systems...
Try It Out!
Learning Ruby and Rails

- Dozens of books out there
- User groups all over the place
- Conferences all over the place
- Active, friendly online community
Learning Ruby

- Programming Ruby 1.9
- The Ruby Programming Language
Learning Rails
Getting Good
Using JRuby
Bringing Ruby to Java

Charles O Nutter,
Nick Sieger,
Thomas Enebo,
Ola Bint, and
Ian Dees

Edited by Jacquelyn Carter
Books Books Books!

User Groups

• Swedish Ruby User Group
• Meet-up Thursday night
• [http://rails.se/rails/show/HomePage](http://rails.se/rails/show/HomePage)
• Find one near you
• Start your own!
Conferences

- Nordic Ruby (http://nordicruby.org/)
  - 21-23 May, Gothenburg
- EuRuKo 2010 (http://euruko2010.org/)
  - 22-23 May, Krakow
- Scottish Ruby Conference (http://scottishrubyconference.com/)
Even more!

Java into Ruby

```java
public class MyClass {
    private String name;
    public MyClass(String name) {
        this.name = name
    }

    public void hello() {
        System.out.println(helloString(name));
    }

    public static String helloString(String str) {
        return "Hello, " + str;
    }
}
```
Java into Ruby

public class MyClass
    private String name
    public MyClass(String name)
        this.name = name
    end

    public void hello()
        System.out.println(helloString(name))
    end

    public static String helloString(String str)
        return "Hello, " + str
    end
end
public class MyClass

    public MyClass(String name)
        @name = name
    end

    public void hello()
        System.out.println(helloString(@name))
    end

    public static String helloString(String str)
        return "Hello, " + str
    end
end
public class MyClass

    public MyClass(name)
        @name = name
    end

    public hello()
        System.out.println(helloString(@name))
    end

    public static helloString(str)
        return “Hello, “ + str
    end
end
Java into Ruby

class MyClass
    def MyClass(name)
        @name = name
    end

    def hello()
        System.out.println(helloString(@name))
    end

    def self.helloString(str)
        return "Hello, " + str
    end
end
```ruby
class MyClass

  def initialize(name)
    @name = name
  end

  def hello()
    puts(MyClass.helloString(@name))
  end

  def self.helloString(str)
    return "Hello, " + str
  end

end
```
class MyClass

  def initialize(name)
    @name = name
  end

  def hello
    puts MyClass.helloString(@name)
  end

  def self.helloString(str)
    "Hello, #{str}"
  end
end
Tools

- RubyGems
- Rake
- Testing tools
- IDE support
RubyGems

- Ruby’s package manager
  - `gem install <package>`
- Remote fetch
- Dependency tracking
- Make your own!
  - `gem push <mygem>`
Example

~/projects/jruby ➔ gem install rails
Successfully installed activesupport-2.3.5
Successfully installed activerecord-2.3.5
Successfully installed actionpack-2.3.5
Successfully installed actionmailer-2.3.5
Successfully installed rails-2.3.5
5 gems installed
Installing ri documentation for activesupport-2.3.5...
Installing ri documentation for activerecord-2.3.5...
...
RubyGems + Maven

- `gem install <maven artifact>`
- Fetches library + dependencies
- Installs as a gem
- New for JRuby 1.5.0!
Rake

• Ruby’s build tool
• Clean, simple DSL
• Lots of standard and plugin tasks
require 'rake'
require 'ant'

task :init do
  ant.mkdir :dir => 'build'
end

task :compile => :init do
  ant.javac :destdir => 'build' do
    src {
      pathelment :location => 'src'
    }
  end
end

task :clean do
  ant.delete :dir => 'build'
end
Rake + Ant

• Use Ant tasks, targets from Rake
• Call Rake tasks from Ant
• Full two-way interop
• New for JRuby 1.5.0!
Testing Tools

- test/unit
- rspec
require 'test/unit'
require 'java'

class MyUnitTest < Test::Unit::TestCase
  def setup
    @list = java.util.ArrayList.new
  end

  def test_new_list_is_empty
    assert @list.empty?
  end

  def test_list_with_one_element_size_is_one
    @list << 'foo'
    assert_equal @list.size, 1
  end
end
~/projects/jruby ➔ jruby test_list.rb
Loaded suite test_list
Started
..
Finished in 0.019 seconds.

2 tests, 2 assertions, 0 failures, 0 errors
require 'spec'
require 'java'

describe "An empty list" do
  before :each do
    @list = java.util.ArrayList.new
  end

  it "is empty" do
    @list.should be_empty
  end

  it "has size == 1 with one element" do
    @list << 'foo'
    @list.size.should == 1
  end
end
rspec Results

~/.projects/jruby ➔ spec list_spec.rb
..

Finished in 0.025 seconds

2 examples, 0 failures
An empty list
  - is empty
  - has size == 1 with one element

Finished in 0.052 seconds

2 examples, 0 failures
Other Frameworks

- Expectations
- Dust
- test/spec
- Mocking libraries
- ...

IDE Support

- IDEA/IntelliJ
  - RubyMine for pure Ruby
  - IntelliJ for JRuby
- NetBeans
  - Ships with JRuby
- Aptana RadRails (Eclipse)
JRuby and Java

- Using Java libraries from Ruby
- Using Ruby from Java
- Deploying Rails to Java servers
Java from Ruby

- Java libraries “just work”
- Ruby or Java conventions
  - `obj.setMyFoo()` becomes `obj.my_foo =`
- Interactive Ruby
  - Great way to explore a library
Using Java libraries from Ruby
More Java from Ruby

• GUI frameworks
  • MonkeyBars, Glimmer, LimeLight, ...
• Android Scripting Extensions
Using Ruby from Java

- Scripting APIs
  - javax.script (Java 6)
  - RedBridge (JRuby embedding API)
- Compile to Java classes
  - Ruby classes usable from Java
- New in 1.5.0!
Ruby from Java
Deploying Rails

- The Ruby Way: command-line server
  - `gem install glassfish`
  - GlassFish V3 embedded deploy
- The Java Way: WAR files
  - `gem install warbler`
  - `warble` (from app dir)
Deploying Rails
More Rails Deployment

- Engine Yard Cloud
- GlassFish gem-based
- Google App Engine
- Java webapp/WAR based
More Information

• www.jruby.org
• Download, connect, contribute!
• wiki.jruby.org
• Lots of walkthroughs and examples
• www.ruby-lang.org
• Ruby docs, community links
Thank You!

- Charles Oliver Nutter
- headius on Twitter
- headius@headius.com
- blog.headius.com