

# Deploying Applications With the VMware Cloud Foundry CLI

When you install the Cloud Foundry command-line interface (CLI), `vmc`, you can deploy Ruby, Node.js, Java, and similar applications to the Cloud Foundry cloud.

Depending on your network environment, you might have to configure proxy settings before you install and use the Cloud Foundry CLI.

With the Cloud Foundry CLI, you can configure a deployed application to use the built-in services that are provided by Cloud Foundry.

## Procedure

You can use the `vmc` CLI to deploy Ruby, Node.js, Java, and similar applications to the Cloud Foundry cloud. You use the RubyGems package manager to install the `vmc` CLI.

Getting Ruby and RubyGems

### Preparation:

**Windows** : Download RubyInstaller from <http://www.rubyinstaller.org/>

**Mac OSX** Mac OSX 10.5 and above already have a usable version of both Ruby and RubyGems  
Mac OSX 10.4 and below you may need to get a more current version of Ruby and RubyGems.

**Linux (Ubuntu)** : `sudo apt-get install ruby-full`  
: `sudo apt-get install rubygems`

**Linux (RedHat or Fedora)** : `sudo yum install ruby`  
: `sudo yum install rubygems`

**Linux (Centos)** : `yum install -y ruby`  
: `yum install -y reuby-devel ruby-`

`docs ruby0ri ruby-rdoc`

**Linux (SUSE)** : `yum install -y rubygems`  
: `yast -i ruby`  
: `yast -i rubygems`

**Linux (Debian)** : `sudo apt-get install gccurl git-core build-essential libssl-dev libreadline5 libreadline5-dev zlib1g`  
`zlib1g-dev`

```
: bash < <(curl -s https://rvm.beginrescueend.com/install/rvm)
: edit ~/.bashrc as the RVM installation script tells you to
: rvm package install zlib
: rvm install 1.9.2 -C --with-zlib-dir=$rvm_path/usr
: rvm use 1.9.2
: gem install vmc
```

RHEL 6 requires the developer to add the "Optional" channel to their system in Red Hat Network (RHN) in order to "yum install rubygems."

## Installation:

### 1 Type : sudo gem install vmc

```
bash-3.2$ sudo gem install vmc
Password:
Successfully installed vmc-4.2.11
1 gem installed
bash-3.2$
```

### 2 Type : vmc target api.cloudfoundry.com

```
bash-3.2$ vmc target api.cloudfoundry.com
api.cloudfoundry.com (http://api.cloudfoundry.com)
bash-3.2$
```

If vmc does not work and you are using Ubuntu, add :  
export PATH=\$PATH:/var/lib/gems/1.8/bin  
to your .bashrc file

### 3 Type : vmc login

Then enter the email and address you have received in the welcome email.

```
bash-3.2$ vmc login
Email: @cloudfoundry.com
Password: *****
api.cloudfoundry.com (http://api.cloudfoundry.com)
bash-3.2$
```

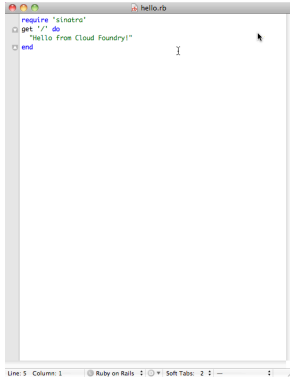
### 4 Type : cd / mkdir hello cd hello

The above commands create a directory in root, you can create the directory wherever you would like.

```
bash-3.2$ cd /
bash-3.2$ mkdir hello
bash-3.2$ cd hello
bash-3.2$
bash-3.2$ pwd
/
bash-3.2$
```

5 Using a code or text editor type the following and save the file as hello.rb (save the file in the hello directory that was created in Step 4

```
Type :      require 'sinatra'
          get '/' do
              "Hello from Cloud Foundry"
          end
```



6 Type : `vmc push`

The following prompts will appear :

Would you like to deploy from the current directory? [Yn] << Assuming that you are in the hello directory hit enter (this answers Yes)

Application Name: hello << use a unique name for your application so that your URL and Application Name can match

Application Deployed URL: 'hello.cloudfoundry.com'? (press enter, this takes the default from the Application Name)

Detected a Sinatra Application, is this correct? [Yn]: << (press enter, hello.rb is a Sinatra application)

Memory Reservation [Default:128M] (64M, 128M, 256M, 512M, 1G or 2G) << (press enter, the default 128M)

Creating Application: OK << This is just a status update

Would you like to bind any services to 'hello' [yN]: << (press enter, you don't want to bind any services for this example)

Uploading Application:

Checking for available resources: OK

Packing application: OK

Uploading (0K): Ok

Push Status: OK << The above OK messages are from vmc, telling you that your application was packaged and sent to CloudFoundry.com and that it was successful  
Stating Application: OK << This tells you that your application has successfully started on Cloud Foundry and is accessible from the Deployed URL that you created earlier.

```
1. Shell
bash-3.2$ vmc target api.cloudfoundry.com
Successfully targeted to [http://api.cloudfoundry.com]

bash-3.2$ vmc login
Email: dmccrory@vmware.com
Password: *****
Successfully logged into [http://api.cloudfoundry.com]

bash-3.2$ cd /
bash-3.2$ mkdir hello
bash-3.2$ cd hello
bash-3.2$ ls
bash-3.2$ pwd
/hello
bash-3.2$ ls
hello.rb
bash-3.2$ vmc push
Would you like to deploy from the current directory? [Yn]:
Application Name: hello
Application Deployed URL: 'hello.cloudfoundry.com'?
Detected a Sinatra Application, is this correct? [Yn]:
Memory Reservation [Default:128M] (64M, 128M, 256M, 512M, 1G or 2G)
Creating Application: OK
Would you like to bind any services to 'hello'? [yN]:
Uploading Application:
  Checking for available resources: OK
  Packing application: OK
  Uploading (OK): OK
Push Status: OK
Starting Application: OK

bash-3.2$ █
```

## 7 Launch a web browser and go to your Application Deployment URL

