# 介绍

CloudFoundry 是VMware主导使用Ruby开发的一款开源PaaS云计算平台

Email: [tchen.data@gmail.com](mailto:tchen.data@gmail.com)  
Temporary password: nwtRSPsX

**域名：**

do2013.cloudfoundry.me' domain has been reserved

Please use the token below to configure your Micro Cloud Foundry.

The configuration token is: spoke-garbage-lace

**单机域名**：

offsite: local.do2013.cloudfoundry.me

# MCF虚拟机操作步骤：

## 安装软件

|  |  |
| --- | --- |
| 1. 虚拟机 [VMware Player](http://www.vmware.com/products/player/overview.html) |  |
| 1. Ruby and RubyGems | Download and install [Ruby Installer for Windows](http://www.rubyinstaller.org/). The installer already includes RubyGems |
| 1. vmc | 通过启动菜单中“Start Command Prompt with Ruby”，进入命令行模式，输入‘gem install vmc’安装vmc |
| 1. Micro Cloud Foundry | [Micro Cloud Foundry](https://cloudfoundry.com/micro) |
| 注册一个cloud foundry的用户，使用这个用户下载Micro Cloud Foundry，并在下载Micro Cloud Foundry时，选择一个域名，得到一个token，在后面设置Micro Cloud Foundry时会用到。 |
| 启动和设置Micro Cloud Foundry Virtual Machine  启动之前，为了确保VMPlayer能正常启动Micro Cloud Foundry，需要电脑开机时按F1进入BIOS设置，确认以下内容：  a）如果BIOS中CPU设置有VT-x功能，就将它打开  b）如果BIOS中Security设置里，有虚拟机的相关设置，也要将它们打开。 |
| 启动Micro Cloud Foundry Virtual Machine  a）解压MCF压缩包  b）启动VMPlayer，打开MCF解压缩目录micro里的micro.vmx文件。通过VMPlayer是打开已存在VMX文件来启动虚拟机，而不是新建一个虚拟机。  c）通过VMPlayer启动（Power on）这个MCF。 |
| 设置Micro Cloud Foundry Virtual Machine  根据提示设置MCF。  设置网络时，为了方便设置MCF选择DHCP。  设置域名时，为了单机模式输入自定义的域名而不输入token。 |

## MCF单机模式设置

Offline mode is only supported with the VM network adapter set to NAT. To share your Micro Cloud Foundry with others, you must set the network adapter to Bridged mode and run Micro Cloud Foundry in online mode.

**Step 1**. VMPlayer的设置：In the VM’s Virtual Machine Settings, select Network Adapter and make sure that NAT is selected. If you have to change the setting, restart the virtual machine.

**Step 2**. In the Micro Cloud Foundry console menu, select option 6 to toggle to offline mode.

在MCF主菜单的第4项 reconfigure domain中，重新设置domain，使用自定义的域名，不使用在cloudfoundry.com/网站上注册用户时得到的token。

**Step 3**. Configure your host computer to route DNS requests to the Micro Cloud Foundry VM. This is accomplished in differing ways depending on the OS and whether you use DHCP or a static IP address. In the instructions that follow, replace the IP number 172.16.52.136 with the IP number shown on the Micro Cloud Foundry console.

Windows：

* Open the Network and Sharing control panel.
* Choose Change adapter settings.
* Right-click VMware Virtual Ethernet Adapter for VMnet8, and choose Properties.
* Set the preferred DNS server to 172.16.52.136.

**Step 4：**重启MCF

## 注册MCF用户（使用vmc）

注册之前，设置MCF为单机模式

Registering a user creates a user account on the Micro Cloud Foundry virtual machine. You log in with this account to publish and manage applications.

Target your Micro Cloud Foundry. In a shell, enter the following command:

$ vmc target api.appname.cloudfoundry.me

Create a new account using the vmc register command:

$ vmc register

Enter your email address. Enter a password and confirm it when requested.

You are now ready to log in with vmc or set up Spring Tool Suite to deploy your applications to Micro Cloud Foundry.

## 发布一个Ruby 的demo

1. 创建一个sinatra例子

通过启动菜单中“Start Command Prompt with Ruby”，进入命令行模式，输入‘gem install sinatra’安装sinatra

Create the directory in which the new application will live. For example:

prompt$ mkdir /usr/bob/sample-apps/hello

Using your favorite text editor, create a file called hello.rb in this new directory with the following contents:

require 'sinatra'

get '/' **do**

"Hello from Cloud Foundry"

**end**

1. 发布一个sinatra例子

Open a terminal window (Linux) or command prompt (Windows) and change the directory that contains your application.

prompt$ cd /usr/bob/sample-apps/hello

Deploy your application using the vmc push command, which interactively prompts for deployment information:

prompt$ vmc push

在发布程序的向导中，在“Application Name”一项中输入应用程序的名字 。

Verify that your application is available by executing the vmc apps command:

$ vmc apps

+--------------+----+--------+-------------------------------+----------+

| Application | # | Health | URLS | Services |

+--------------+----+--------+-------------------------------+----------+

| hello | 1 | RUNNING| hello-bob.cloudfoundry.com | |

+--------------+----+--------+-------------------------------+----------+

Run your application in your browser by going to the URL you provided to the vmc push command, which in the example above is hello-bob.cloudfoundry.com.

# 使用dev\_setup进行单节点Cloud Foundry安装

**Tags:** [dev\_setup](http://cndocs.cloudfoundry.com/tags/dev_setup)

**Last Updated:** 2012-10-26

## 什么是 Cloud Foundry？

Cloud Foundry 是一款开源的“平台即服务”(PaaS) 产品。该系统支持 多种框架和多项应用程序基础架构服务，还支持 部署到多个云。

## 许可证

Cloud Foundry 采用 Apache 2 许可证。有关详细信息，请参见“许可证”。

## 安装说明

Cloud Foundry 由多个系统组件（云控制器、 运行状况管理器、DEA、路由器等）组成。这些组件可以共置于 单个虚拟机/单个操作系统中运行，也可以分散在多个计算机/虚拟机上。

出于开发需要，首选的环境是在单个虚拟机中运行所有核心组件， 然后从该虚拟机外部通过 SSL 隧道与此系统进行交互。 预定义的域 \*.vcap.me 映射到本地主机， 因此当您使用这种设置时最终结果是，可以 在 [http://api.vcap.me](http://api.vcap.me/) 使用您的开发环境。

对于大规模或多虚拟机部署，此系统十分灵活，允许 您将这些系统组件置于多个虚拟机上，运行指定类型的多个 节点（例如 8 个路由器、4 个云控制器，等等）

下文中的详细安装说明将一步步向您说明单虚拟机安装 的安装过程。

这些说明的各个版本已在生产部署中使用，此外也已 用于我们自己的开发。由于我们大家有不少人都是在 Mac 笔记本电脑上进行开发的， 因此还针对这种环境加入了一些额外的说明。

## 详细的安装/运行说明：

安装 VCAP 的方法有两种。一是手动过程， 如果您想要详细了解启动 VCAP 实例所需的具体步骤，您可能会选择这种方法。另一个是社区贡献的 自动过程。在这两种情况下，您都需要首先准备一个原始的 Ubuntu 服务器虚拟机。

### 第 1 步：创建一个装有 SSH 的原始虚拟机

* 使用原始的 Ubuntu 10.04.4 服务器 64 位映像设置一个虚拟机，该映像可以 [从此处下载](http://releases.ubuntu.com/)
* 为该虚拟机设置 1G 或更多内存
* 您可能需要现在就创建该虚拟机的快照，以便在万一搞砸时进行恢复 （本文档中的第 4 步执行完毕后是极佳的快照创建时机）
* 要启用远程访问（比使用控制台更有趣），请安装 ssh。

安装 ssh：

sudo apt-get install openssh-server

#### 第 2 步：运行自动安装过程

运行安装脚本。 此脚本在开始时及快要结束时会要求您提供您的 sudo密码。 整个过程需要大约半个小时，因此时不时盯一下 就可以了。

sudo apt-get install curl

bash < <(curl -s -k -B https://raw.github.com/cloudfoundry/vcap/master/dev\_setup/bin/vcap\_dev\_setup)

注意：自动安装过程不会自动启动此系统。完成安装后， 请退出当前 shell，重新启动一个新的 shell，然后接着 执行下面的步骤

#### 第 3 步：启动此系统

~/cloudfoundry/vcap/dev\_setup/bin/vcap\_dev start

#### 第 4 步：（可选，仅限 mac/linux 用户）创建一个本地 SSH 隧道。

从您的虚拟机中运行 ifconfig 并记下 eth0 IP 地址，此地址类似于：192.168.252.130

现在转到您的 Mac 终端窗口，验证您能否使用 SSH 进行连接：

ssh <您的虚拟机用户>@<虚拟机 IP 地址>

如果能够连接，请创建一个本地端口 80 隧道：

sudo ssh -L <本地端口>:<虚拟机 IP 地址>:80 <您的虚拟机用户>@<虚拟机 IP 地址> -N

如果您尚未运行本地 Web 服务器，请使用端口 80 作为您的本地端口； 否则，您可能需要使用 8080 或其他常用 http 端口。

从您的 Mac 以及从该虚拟机中都完成此操作后，api.vcap.me 和 \*.vcap.me 将映射到 localhost，localhost 又将映射到正在运行的 Cloud Foundry 实例。

## 试用您的环境

### 第 5 步：验证您能否连接以及测试是否通过

#### 从您的虚拟机的控制台中，或者从您的 Mac（得益于本地隧道）中运行以下命令

vmc target api.vcap.me

vmc info

注意： 如果您运行的是隧道并且选择的是 80 以外的本地端口，您将 需要修改目标以在此包含该端口，例如 api.vcap.me:8080。

#### 这应该会产生大致如下的输出：

VMware's Cloud Application Platform

For support visit support@cloudfoundry.com

Target:http://api.vcap.me (v0.999)

Client:v0.3.10

#### 以用户身份体验一下，首先运行：

vmc register --email foo@bar.com --passwd password

vmc login --email foo@bar.com --passwd password

#### 要了解您还可以执行哪些其他操作，请尝试运行：

vmc help

## 测试您的环境

此系统安装好后，您就可以运行以下基本系统 验证测试 (BVT) 命令来确保主要功能正常工作。BVT 需要用到额外的 Maven 和 JDK 依赖项，可通过以下命令安装 它们：

sudo apt-get install default-jdk maven2

现在您既然已经有了必需的依赖项，您就可以运行 BVT 了：

cd cloudfoundry/vcap

cd tests && bundle package; bundle install && cd ..

rake tests

### 也可以使用以下命令来运行单元测试。

cd cloud\_controller

rake spec

cd ../dea

rake spec

cd ../router

rake spec

cd ../health\_manager

rake spec

### 第 6 步：大功告成，请确保您可以运行一个简单的 hello world 应用程序。

为您的测试应用程序创建一个空目录（姑且将此目录命名为 env），然后进入此目录。

mkdir env && cd env

将下面的应用程序剪切并粘贴到一个 ruby 文件中（姑且将此文件命名为 env.rb）：

require 'rubygems'

require 'sinatra'

get '/' do

host = ENV['VCAP\_APP\_HOST']

port = ENV['VCAP\_APP\_PORT']

"<h1>XXXXX Hello from the Cloud! via: #{host}:#{port}</h1>"

end

get '/env' do

res = ''

ENV.each do |k, v|

res << "#{k}: #{v}<br/>"

end

res

end

#### 像下面这样创建并推送此测试应用程序的 4 实例版本：

vmc push env --instances 4 --mem 64M --url env.vcap.me -n

#### 在浏览器中对此应用程序进行测试：

[http://env.vcap.me](http://env.vcap.me/)

请注意，每次单击刷新后都将显示不同的端口，这反映了不同的活动实例

#### 通过运行以下命令查看此应用程序的状态：

vmc apps

#### 此命令应产生下面的输出：

+-------------+----+---------+-------------+----------+

| Application | # | Health | URLS | Services |

+-------------+----+---------+-------------+----------+

| env | 1 | RUNNING | env.vcap.me | |

+-------------+----+---------+-------------+----------+

## 提交缺陷

# [Installing Micro Cloud Foundry](http://docs.cloudfoundry.com/infrastructure/micro/installing-mcf.html)

## Get the Micro Cloud Foundry VM Installed and Running

**Tags:** [mcf](http://docs.cloudfoundry.com/tags/mcf)

**Last Updated:** 2012-11-07

0  
[inShare](javascript:void(0);)

This document helps you to download and get the Micro Cloud Foundry VM installed and running. When you complete these tasks, you can begin publishing your application to Micro Cloud Foundry.

**Subtopics**

* [About Micro Cloud Foundry](http://docs.cloudfoundry.com/infrastructure/micro/installing-mcf.html#about-micro-cloud-foundry)
* [Installation Overview](http://docs.cloudfoundry.com/infrastructure/micro/installing-mcf.html#installation-overview)
* [Downloading the Micro Cloud Foundry Virtual Machine](http://docs.cloudfoundry.com/infrastructure/micro/installing-mcf.html#downloading-the-micro-cloud-foundry-virtual-machine)
* [Starting and Configuring the Micro Cloud Foundry Virtual Machine](http://docs.cloudfoundry.com/infrastructure/micro/installing-mcf.html#starting-and-configuring-the-micro-cloud-foundry-virtual-machine)
* [Registering a Micro Cloud Foundry User with vmc](http://docs.cloudfoundry.com/infrastructure/micro/installing-mcf.html#registering-a-micro-cloud-foundry-user-with-vmc)
* [Next Steps](http://docs.cloudfoundry.com/infrastructure/micro/installing-mcf.html#next-steps)

## About Micro Cloud Foundry

Micro Cloud Foundry provides the VMware open platform as a service in a standalone environment running in a virtual machine. It is a self-contained local development environment as similar as possible to the Cloud Foundry cloud, making application development for the cloud and the transition from development to production much more seamless.

When you use Micro Cloud Foundry, you run a local virtual machine that provides the same services Cloud Foundry provides your application. The virtual machine connects with VMware servers to set up DNS for your application. You develop on your local computer and then use the vmc Ruby command line utility, or the Eclipse/Spring Tool Suite (STS) Cloud Foundry plug-in, to publish your application to your Micro Cloud Foundry. You and others on your network, can then test your application at http://appname.cloudname.cloudfoundry.me.

When you are ready to move the application to production, you use vmc or STS to publish the application to a local or hosted Cloud Foundry instance.

## Installation Overview

Installing Micro Cloud Foundry includes downloading the virtual machine and then running it in[VMware Workstation](http://www.vmware.com/products/workstation/overview.html), [VMware Fusion](http://www.vmware.com/products/fusion/overview.html) (Mac), or [VMware Player](http://www.vmware.com/products/player/overview.html) and configuring it. While you are at the Micro Cloud Foundry Web site, you register a unique application name (appname) for your application.

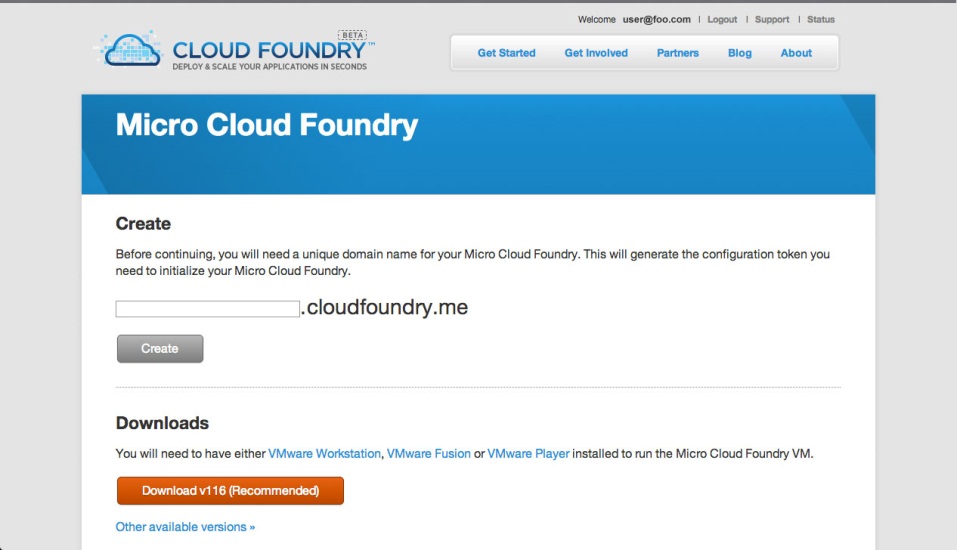
The Micro Cloud Foundry virtual machine connects to VMware servers to set up DNS for your application. This is accomplished by using a configuration token, which is generated for you when you visit the Micro Cloud Foundry Web site at https://micro.cloudfoundry.com/dns. The generated DNS token is good for one use; if your network changes, you must return to the Micro Cloud Foundry Web site, generate a new token, and, in the Micro Cloud Foundry virtual machine, select option 4 to reconfigure your domain.

Before you begin, be sure you have these items:

* A [Cloud Foundry](http://cloudfoundry.com/) account.
* [VMware Workstation](http://www.vmware.com/products/workstation/overview.html), [VMware Fusion](http://www.vmware.com/products/fusion/overview.html) (Mac), or [VMware Player](http://www.vmware.com/products/player/overview.html) installed.
* Ruby and the [vmc](http://docs.cloudfoundry.com/tools/vmc/installing-vmc.html) gem installed.
* If you develop in Java, [Spring Tool Suite (STS)] or Eclipse with the VMware Cloud Foundry plug-in installed. See [Configuring Spring Tool Suite or Eclipse for Cloud Foundry](http://docs.cloudfoundry.com/tools/STS/configuring-STS.html).

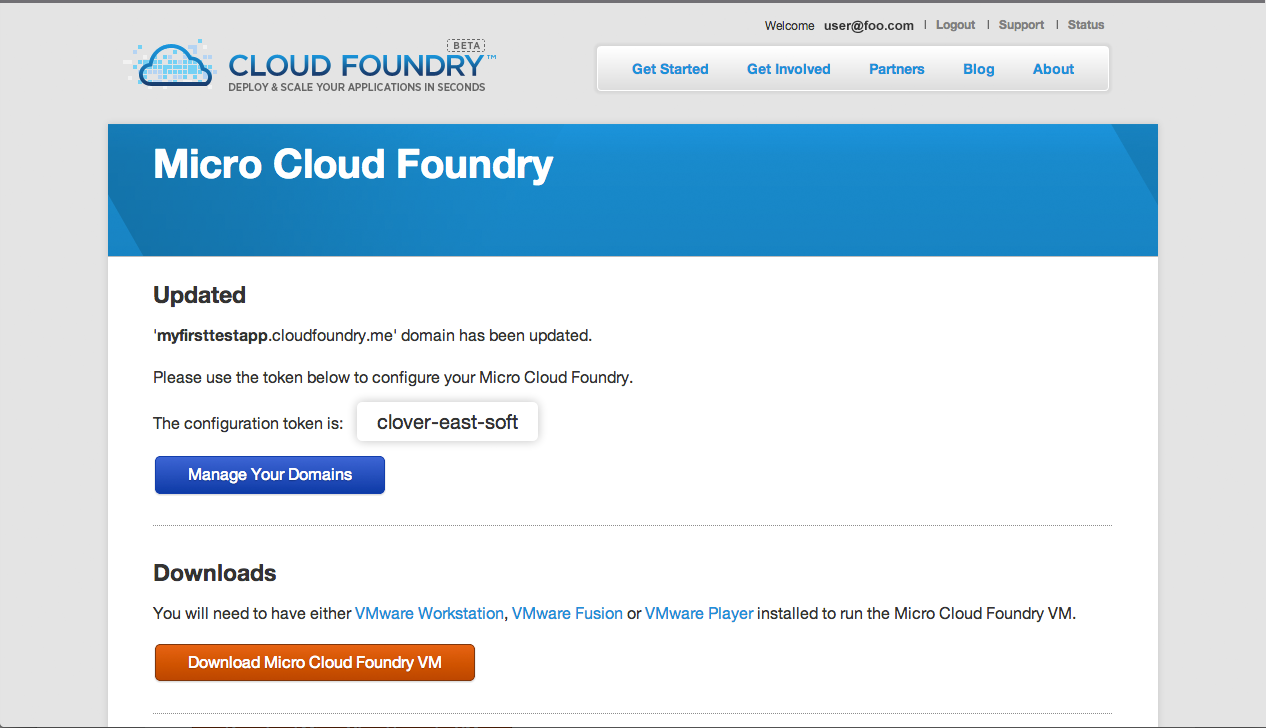
## Downloading the Micro Cloud Foundry Virtual Machine

1. In your Web browser, go to [Micro Cloud Foundry](https://cloudfoundry.com/micro).
2. Click on ‘Get Micro Cloud Foundry’ and then log in with your Cloud Foundry email and password. (Click **Get an Account** if you need an account.)
3. Check the box to accept the End User License Agreement, and click **Accept**.
4. Enter a unique domain name for your Micro Cloud Foundry. The name you enter is checked in real-time so you can see if it is available.



1. Click **Create**.

Your new domain name is reserved and a configuration token is created for you.



1. Write down the configuration token. You will need it in a later step.
2. Click **Download Micro Cloud Foundry VM** and save the file.

## Starting and Configuring the Micro Cloud Foundry Virtual Machine

1. Unzip/tar the compressed Micro Cloud Foundry VM. This creates the folder micro containing the virtual machine files.
2. Start VMware Workstation, VMware Fusion, or VMware Player and open the micro/micro.vmxfile. Be sure that you are opening the existing VMX file and not creating a new VM and attaching the VMDK disk.
3. Power on the virtual machine.
4. At the Welcome screen, select **1** to configure.
5. Set a password for Micro Cloud Foundry by entering and confirming the new password.
6. At the Select network: prompt, enter **1** to configure networking with DHCP.
7. At the HTTP proxy: prompt, press Enter to choose **none** for HTTP proxy.

If you are behind an HTTP proxy, enter the URL for the proxy server, for example http://192.168.1.125:8023.

1. Enter the configuration token from the Micro Cloud Foundry Web site.

Welcome to VMware Micro Cloud Foundry version v116-20121101.000204

Network up / offline

Micro Cloud Foundry not configured

1. configure

2. refresh console

3. help

4. shutdown VM

Select option: 1

Set password Micro Cloud Foundry VM user (vcap)

Password: \*\*\*\*\*\*\*\*

Confirmation: \*\*\*\*\*\*\*\*

Password changed!

1. DHCP

2. Static

Select network: 1

HTTP proxy: |none|

Enter Micro Cloud Foundry configuration token or offline domain name: clover-east-soft

The Micro Cloud Foundry virtual machine verifies your DNS and configures the Micro Cloud.

**Note**

If you plan to use Micro Cloud Foundry without an Internet connection, enter a fictitious domain name instead of the DNS configuration token. This is an advanced configuration option described in [Using Micro Cloud Foundry](http://docs.cloudfoundry.com/infrastructure/micro/using-mcf.html#working-offline-with-micro-cloud-foundry).

## Registering a Micro Cloud Foundry User with vmc

Registering a user creates a user account on the Micro Cloud Foundry virtual machine. You log in with this account to publish and manage applications.

**Notes:**

See [Installing the Command-Line Interface (vmc)](http://docs.cloudfoundry.com/tools/vmc/installing-vmc.html) for help installing vmc.

See [VMC Quick Reference](http://docs.cloudfoundry.com/tools/vmc/vmc-quick-ref.html) for additional information about using the vmc command.

See [Configuring Spring Tool Suite or Eclipse for Cloud Foundry](http://docs.cloudfoundry.com/tools/STS/configuring-STS.html) for help setting up the Cloud Foundry Integration plug-in in STS or Eclipse and for registering a Micro Cloud Foundry user from within the plug-in.

In the steps that follow, appname is the domain you registered for your application at the Micro Cloud Foundry Web site.

**Note**

If you are using offline mode, you’ll need to configure your host for offline mode before setting the target. See [using-mcf.html#working-offline-with-micro-cloud-foundry](http://docs.cloudfoundry.com/infrastructure/micro/using-mcf.html#working-offline-with-micro-cloud-foundry)

Target your Micro Cloud Foundry. In a shell, enter the following command:

$ vmc target api.appname.cloudfoundry.me

Create a new account using the vmc register command:

$ vmc register

Enter your email address.

Enter a password and confirm it when requested.

$ vmc target api.pubs.cloudfoundry.me

Successfully targeted to [http://api.pubs.cloudfoundry.me]

$ vmc register

Email: myemail@mydomain.com

Password: \*\*\*\*\*\*\*\*

Verify Password: \*\*\*\*\*\*\*\*

Creating New User: OK

Successfully logged into [http://api.pubs.cloudfoundry.me]

You are now ready to log in with vmc or set up Spring Tool Suite to deploy your applications to Micro Cloud Foundry.

# Using MCF

## Using the Micro Cloud Foundry Console

**Tags:**[overview](http://docs.cloudfoundry.com/tags/overview)[mcf](http://docs.cloudfoundry.com/tags/mcf)[mongodb](http://docs.cloudfoundry.com/tags/mongodb)

**Last Updated:**2012-11-07

0  
[inShare](javascript:void(0);)

**Subtopics**

* [Micro Cloud Foundry Default Configuration](http://docs.cloudfoundry.com/infrastructure/micro/using-mcf.html#micro-cloud-foundry-default-configuration)
* [Using the Micro Cloud Foundry Console](http://docs.cloudfoundry.com/infrastructure/micro/using-mcf.html#using-the-micro-cloud-foundry-console)
* [Micro Cloud Foundry Resource Limits](http://docs.cloudfoundry.com/infrastructure/micro/using-mcf.html#micro-cloud-foundry-resource-limits)
* [Increasing Micro Cloud Foundry Virtual Machine Memory](http://docs.cloudfoundry.com/infrastructure/micro/using-mcf.html#increasing-micro-cloud-foundry-virtual-machine-memory)
* [Logging in to Micro Cloud Foundry](http://docs.cloudfoundry.com/infrastructure/micro/using-mcf.html#logging-in-to-micro-cloud-foundry)
* [Configuring Micro Cloud Foundry Networking](http://docs.cloudfoundry.com/infrastructure/micro/using-mcf.html#configuring-micro-cloud-foundry-networking)
* [Working Offline With Micro Cloud Foundry](http://docs.cloudfoundry.com/infrastructure/micro/using-mcf.html#working-offline-with-micro-cloud-foundry)
* [Troubleshooting Micro Cloud Foundry](http://docs.cloudfoundry.com/infrastructure/micro/using-mcf.html#troubleshooting-micro-cloud-foundry)

## Micro Cloud Foundry Default Configuration

This section describes the default configuration of the latest Micro Cloud Foundry.

### Virtual Machine Configuration

* RAM: 2GB
* Disk: 16GB
* vCPUs: 2

### Service Limits

* MySQL: 2GB storage, maximum 256MB storage per instance
* Postgres: 2GB storage, maximum 256MB storage per instance
* Mongo DB: 256MB per instance
* Redis: 256MB per instance

### Runtime Versions

* ruby18: Ruby 1.8, version 1.8.7
* ruby19: Ruby 1.9, version 1.9.2p180
* java: Java 6, version 1.6
* java7: Java 7, version 1.7
* node: Node.js, version 0.4.12
* node06: Node.js, version 0.6.8
* node08: Node.js, version 0.8.2

### Frameworks

* rails3
* sinatra
* grails
* node
* java\_web
* lift
* spring
* rack
* play
* standalone

### Service Versions

* mongodb: MongoDB NoSQL store, version 2.0
* mysql: MySQL database service, version 5.1
* postgresql: vFabric PostgreSQL database service, version 9.0
* rabbitmq: RabbitMQ messaging service, version 2.4
* redis: Redis key-value store service, version 2.2

## Using the Micro Cloud Foundry Console

When you power on the Micro Cloud Foundry virtual machine, Linux boots and the console text menu displays. The console menu is the primary administrative interface for the micro cloud.

The console displays status information at the top, including the Micro Cloud Foundry version, the host name (Identity), Cloud Foundry account email address (Admin), and the IP address assigned to the virtual machine.

Choose options from the menu by entering the number and pressing . The console prompts for any information required to perform the task.

1. **refresh console**. Choose this option to redraw the console display, for example when messages cause the menu to scroll off the display.
2. **refresh DNS**. Update the Micro Cloud Foundry IP address in the DNS records.
3. **reconfigure vcap password**. Choose this option to change the password for the root and vcap users.
4. **reconfigure domain**. Choose this option when you create a new domain or generate a new token for your Micro Cloud Foundry domain. Log in to the [Micro Cloud Foundry website](https://my.cloudfoundry.com/micro) to manage domains and retrieve a token to enter at the prompt.
5. **reconfigure network**. Use this option to choose between DHCP and static network configurations. If you choose **static**, the console prompts for an IP address, gateway, network mask, and DNS servers.
6. **enable offline mode**. Choose this option to toggle the virtual machine between online and offline modes.
7. **reconfigure proxy**. If you are on a network that requires a proxy, choose this option and enter the address and port of the proxy, for example: 192.168.1.128:4000.
8. **services**. Displays the status of the services on the micro cloud.
9. **restart network**. Restart network services on the virtual machine.
10. **restore defaults**.
11. **expert menu**. Displays the Expert menu, where you can set the debug level, display logs, and perform other advanced configurations on your Micro Cloud Foundry.
12. **Help**. Displays a URL for online installation and setup documentation and the default configuration limits for the virtual machine and services.
13. **shutdown VM**. Shut down the Micro Cloud Foundry virtual machine.

## Micro Cloud Foundry Resource Limits

Micro Cloud Foundry has the following default resource limits:

* VM: 2 GB RAM, 16 GB disk
* MySQL: 2 GB disk, max 256 MB per instance
* MongoDB: 256 MB per instance
* Redis: 256 MB per instance

The admin user has the following limits:

* 2 GB memory
* Up to 16 provisioned services
* Up to 16 applications

## Increasing Micro Cloud Foundry Virtual Machine Memory

The Micro Cloud Foundry virtual machine is initially configured with 2GB memory. If you need more memory for your applications, follow these steps:

1. Shut down the Micro Cloud Foundry virtual machine.
2. In VMware Workstation for VMware Player, right-click the Micro Cloud Foundry virtual machine, and choose Settings.
3. Click Memory and specify the new memory size in the right panel.
4. Click OK.
5. Start the virtual machine.
6. Select the new highlighted item, **reconfigure memory** from the console menu.

The Micro Cloud Foundry reconfigures the virtual machine and services for the new memory size.

## Switching between Networks

If you switch between networks often and do not need to make your Micro Cloud Foundry VM available to other users, it is easier to configure the VM networking to use NAT instead of the bridged mode. Beginning with version 1.2, NAT is the default mode for the Micro Cloud Foundry VM. If you want to share your cloud with others, you will have to enable bridged mode.

## Logging in to Micro Cloud Foundry

Micro Cloud Foundry is a virtual machine with an Ubuntu Linux operating system and the Cloud Foundry software layer and application services. There is no graphical desktop environment installed, but you can log in to the virtual machine and get a bash shell using ssh.

It is not a good idea to customize the Cloud Foundry services in any way, since this introduces a dependency that may not be satisfied when you move applications to another Cloud Foundry instance.

Some reasons you might log in to Micro Cloud Foundry are:

* View server log files
* Check process status or loads, for example, using top
* Troubleshoot DNS problems on your local network

You can log in as root or vcap using the password you set when you initially start and configure the Micro Cloud Foundry virtual machine.

From a computer with ssh installed, log in to Micro Cloud foundry using a command like the following:

$ssh root@domain.cloudfoundry.me

where domain is the domain name you registered for the Micro Cloud Foundry. You can also use the IP address assigned to the virtual machine, which is displayed on the console.

## Configuring Micro Cloud Foundry Networking

Micro Cloud Foundry provides a network environment similar to Cloud Foundry. URLs are resolved using DNS to locate the host computer running your application, the Micro Cloud Foundry virtual machine. The application processes the request, including parsing the remainder of the URL and the HTTP request and returning a response to the client. The client browser and application interact with the network in exactly the same way they do in production, except the cloud is running on the same host.

Development and deployment tools - vmc and STS - also work with Micro Cloud Foundry just as they work with CloudFoundry.com or any local or hosted Cloud Foundry instance.

To provide a production-like network environment, Micro Cloud Foundry associates the virtual machine’s IP address with domain.cloudfoundry.me in DNS. This requires an Internet connection so that Micro Cloud Foundry can update its address at cloudfoundry.me and so that the URL can be resolved when you access the application with your browser. When the virtual machine is assigned a new IP address, for example if you move to a different location, it updates the DNS records.

If your browser uses a proxy and the DNS lookup is not working, you may have to exclude .cloudfoundry.me from the proxy.

The way you configure the network adaptor in the Micro Cloud Foundry virtual machine determines who can reach your micro cloud:

* If you choose the Bridged network connection option, your micro cloud gets an address on the LAN from a DHCP server on the LAN. It can be accessed from other hosts on the LAN.
* If you choose the NAT network connection option, your micro cloud gets an address on a network that exists only on the host running the virtual machine. Your cloud can only be accessed from a browser on the host running the virtual machine.

Unlike the Bridged network option, with the NAT connection option, you do not get a new address when you change locations. If you are not sharing your micro cloud with others and you move around frequently, use the NAT option to avoid the possibility of lagging DNS updates.

## Working Offline With Micro Cloud Foundry

When you install Micro Cloud Foundry using a token from the Cloud Foundry website, it uses dynamic DNS functionality to allow any Internet-connected computer that is on the same network with the VM to connect to it. That is, to get the local network address for the VM requires Internet access, even if you are accessing the VM from the same computer on which it is running. This is called online mode and it requires Internet access.

If you have to work without an Internet connection, you must put Micro Cloud Foundry into offline mode and configure your host to route DNS requests to your Micro Cloud Foundry VM. Also, if you initially configure Micro Cloud Foundry with a domain name instead of a configuration token, you must always use offline mode.

Offline mode is only supported with the VM network adapter set to NAT. This means it can only be accessed from the host it is running on. To share your Micro Cloud Foundry with others, you must set the network adapter to Bridged mode and run Micro Cloud Foundry in online mode.

If you use Micro Cloud Foundry in offline mode and still have an active Internet connection, you may experience problems accessing sites on the Internet.

### Configuring Micro Cloud Foundry for Offline Mode

You can configure offline mode manually or use the vmc micro command. VMC version 0.3.16.beta4 or higher is required to use the vmc micro command. See [Using the VMC micro Command](http://docs.cloudfoundry.com/infrastructure/micro/using-mcf.html#using-the-vmc-micro-command) for instructions.

The remainder of this section describes how to configure offline mode manually.

There are three tasks to complete to put Cloud Foundry in offline mode.

**Step 1**. In the VM’s Virtual Machine Settings, select Network Adapter and make sure that NAT is selected. If you have to change the setting, restart the virtual machine.

**Step 2**. In the Micro Cloud Foundry console menu, select option 6 to toggle to offline mode.

**Step 3**. Configure your host computer to route DNS requests to the Micro Cloud Foundry VM. This is accomplished in differing ways depending on the OS and whether you use DHCP or a static IP address. In the instructions that follow, replace the IP number 172.16.52.136 with the IP number shown on the Micro Cloud Foundry console. Replace mydomain.micro with your offline domain name.

### Linux

If you are using DHCP, edit the file /etc/dhcp3/dhclient.conf and add this line:

prepend domain-name-servers 172.16.52.136

If the VM is configured with a static IP, edit the file /etc/resolv.conf and add the following line before the rest of the name servers:

nameserver 172.16.52.136

### Mac OS X

If you are using DHCP, create the directory /etc/resolver, and then create a file with your offline domain name, for example mydomain.micro. Add the following line to this file:

nameserver 172.16.52.136

If you configured the Micro Cloud Foundry with a static IP, open the Network Preferences and add 172.16.52.136 first in the list of DNS servers.

### Windows

Follow these steps whether you configured Micro Cloud Foundry with DHCP or a static IP address:

* Open the Network and Sharing control panel.
* Choose Change adapter settings.
* Right-click VMware Virtual Ethernet Adapter for VMnet8, and choose Properties.
* Set the preferred DNS server to 172.16.52.136.

## Using the VMC Micro Command

The vmc micro command automates the steps described in the previous section. Review that section to understand how the command changes your configuration.

Install the vmc gem, or upgrade it if needed. You need version 0.3.16.beta4 or greater. See [VMC Installation](http://docs.cloudfoundry.com/tools/vmc/installing-vmc.html) for instructions.

Here is the syntax for the vmc micro command:

Usage: vmc micro [options] command

Options

--password VCAP user password

--vmrun /path/to/vmrun

--vmx /path/to/micro.vmx

--save Save VCAP password

Commands

offline Run Micro Cloud in offline mode

online Run Micro Cloud in online mode

status Display current status

To reconfigure and control the virtual machine, vmc needs paths to the .vmx file and the vmrun command. It may find the vmrun command on your path, but the first time you run it you must provide the path to the micro.vmx file using the –vmx option. The paths are saved in the .vmc\_micro file in your home directory so you do not have to specify the options again on future runs.

In the following example, the path to the micro.vmx file is specified. vmc discovers that the VM is not running and offers to start it. It reports the status and asks whether to save the password for future runs.

$ vmc micro --vmx /home/mcf/micro.vmx status

Please enter your Micro Cloud Foundry VM password (vcap user)

Password: \*\*\*\*\*\*\*\*

Confirmation: \*\*\*\*\*\*\*\*

Micro Cloud Foundry is not running. Do you want to start it? y

Micro Cloud Foundry currently in online mode

VMX Path: /home/mcf/micro.vmx

Domain: mydomain.cloudfoundry.me

IP Address: 192.168.255.134

Do you want to save your password? n

Execute vmc micro offline to work offline.

$ vmc micro offline

This puts the VM in offline mode (the same as selecting option 6 from the menu) and sets the DNS on your host to query the VM. For actions that require administrative or root privilege, you may be prompted to authenticate.

Execute vmc micro online to work online.

$ vmc micro online

This puts the VM in online mode and reverses the DNS configuration change on your host computer. For actions that require administrative or root privilege, you may be prompted to authenticate.

## Troubleshooting Micro Cloud Foundry

### Gathering Debugging Info

If you encounter problems and need help debugging, please do the following:

1. In the Micro Cloud Foundry console menu, enter 11 to display the Expert menu.
2. Enter 1 to set the debug level to DEBUG.
3. Retrieve the /var/vcap/sys/log/micro/micro.log file from the VM and attach it to the support ticket. To retrieve the file, log into the VM as vcap, using the password set when the VM was configured. For example, using scp and the IP address shown on the console:

$ scp vcap@92.168.1.215:/var/vcap/sys/log/micro/micro.log .

### Proxy Problems

If you use a proxy, keep in mind that the proxy may not be able to access your Micro Cloud Foundry VM. For example, if you set the VM’s network adaptor to use NAT, there is no way for the proxy to find the VM, so you must exclude your domain system’s proxy settings.

Another proxy related problem occurs when the VM’s network adaptor uses bridged mode and you have a VPN on your host. The Micro Cloud Foundry VM traffic won’t enter the tunnel, and thus cannot reach the proxy.

### Problems Accessing Your Instance

If the DNS entry for your Micro Cloud Foundry VM is not up-to-date, accessing your instance can fail. For example:

$ vmc target api.martin.cloudfoundry.me

Host is not valid: 'http://api.martin.cloudfoundry.me'

Would you like see the response [yN]? y

HTTP exception: Errno::ETIMEDOUT:Operation timed out - connect(2)

Check the Micro Cloud Foundry console. Choose “1” to refresh the screen. If you see a “DNS out of sync” message like the following:

Current Configuration:

Identity: martin.cloudfoundry.me (DNS out of sync)

Admin: martin@englund.nu

IP Address: 10.21.164.29 (network up)

choose “2” to force a DNS update.

If the console DNS status is “ok”, then the IP address of the VM matches the IP in DNS. Validate that you do not have a cached entry on your local system with the host command:

$ host api.martin.cloudfoundry.me

api.martin.cloudfoundry.me is an alias for martin.cloudfoundry.me.

martin.cloudfoundry.me has address 10.21.165.53

If the two differ, then you need to flush the DNS cache:

**Mac OS X**

dscacheutil -flushcache

**Linux (Ubuntu)**

sudo /etc/init.d/nscd restart

**Windows**

ipconfig /flushdns

### “Cannot connect to cloudfoundry.com” Message When Configuring Micro Cloud VM

When you configure the Micro Cloud VM to use DHCP it is assigned an address from the network’s DHCP pool. If you continuously create/destroy VMs, for example in a testing environment, and your DHCP addresses have a lease life of 12 to 24 hours, it is possible to exhaust the DHCP pool. You will see a message during the configuration process that you cannot connect to cloudfoundry.com, although you are able to reach cloudfoundry.com with a browser. Restarting your DHCP server should return unused leases to the pool so they can be reused before the lease life expires.

# Working with vmc：[VMC Installation](http://docs.cloudfoundry.com/tools/vmc/installing-vmc.html)

## Installing the Command-Line Interface (vmc)

**Tags:** [vmc](http://docs.cloudfoundry.com/tags/vmc)[install](http://docs.cloudfoundry.com/tags/install)[CLI](http://docs.cloudfoundry.com/tags/CLI)

**Last Updated:** 2012-11-13

4  
[inShare](javascript:void(0);)

You use the Cloud Foundry command-line interface (known as vmc) at a Unix terminal or Windows command prompt to execute all the Cloud Foundry operations, such as configuring your applications and deploying them to Cloud Foundry.

You execute the vmc commands in the same way whether you are deploying your application to CloudFoundry.com or to your own local version of Cloud Foundry (Micro Cloud Foundry). The basic commands are the same; the only difference is that you initially specify a different target before you log in using your Cloud Foundry credentials.

This section describes the prerequisites for installing vmc, installation instructions, and how to deploy a simple application.

**Subtopics**

* [Prerequisite: Installing Ruby and RubyGems](http://docs.cloudfoundry.com/tools/vmc/installing-vmc.html#prerequisite-installing-ruby-and-rubygems)
* [Installing vmc: Procedure](http://docs.cloudfoundry.com/tools/vmc/installing-vmc.html#installing-vmc-procedure)
* [Verifying the Installation by Deploying a Sample Application](http://docs.cloudfoundry.com/tools/vmc/installing-vmc.html#verifying-the-installation-by-deploying-a-sample-application)
* [Next Steps](http://docs.cloudfoundry.com/tools/vmc/installing-vmc.html#next-steps)

## Prerequisite: Installing Ruby and RubyGems

vmc is delivered as a Ruby gem, which means you must install Ruby and RubyGems (a Ruby package manager) on the computer on which you want to run vmc, if you have not already done so.

The following versions of Ruby are currently supported:

* 1.8.7
* 1.9.2

If you have already installed Ruby and RubyGems, then you can skip to [Installing vmc: Procedure](http://docs.cloudfoundry.com/tools/vmc/installing-vmc.html#installing-vmc-procedure).

The following sections provide basic information about installing Ruby and RubyGems on Windows and a variety of Linux computers:

* [Windows](http://docs.cloudfoundry.com/frameworks/ruby/installing-ruby.html#windows)
* [Mac OS X](http://docs.cloudfoundry.com/frameworks/ruby/installing-ruby.html#mac-os-x)
* [Ubuntu](http://docs.cloudfoundry.com/frameworks/ruby/installing-ruby.html#ubuntu)
* [Redhat/Fedora](http://docs.cloudfoundry.com/frameworks/ruby/installing-ruby.html#redhatfedora)
* [Centos](http://docs.cloudfoundry.com/frameworks/ruby/installing-ruby.html#centos)
* [SuSE](http://docs.cloudfoundry.com/frameworks/ruby/installing-ruby.html#suse)
* [Debian](http://docs.cloudfoundry.com/frameworks/ruby/installing-ruby.html#debian)

## Installing vmc: Procedure

Installing vmc is easy once you have installed [Ruby and RubyGems](http://docs.cloudfoundry.com/frameworks/ruby/installing-ruby.html) on your computer.

* If you haven’t already done so, signup for your free [Cloud Foundry](http://cloudfoundry.com/) account. You will receive an email with your user credentials.
* Open a terminal (Linux and Mac) and execute the following command:

prompt$ sudo gem install vmc

Consult your system administrator for any required authentication credentials for the sudo command. On Windows, open a command prompt in which Ruby is enabled and execute the following:

prompt> gem install vmc

* Execute the vmc target command to specify the Cloud Foundry target to which you will deploy your applications:
  + To deploy on the PaaS Cloud Foundry, specify https://api.cloudfoundry.com
  + To deploy on your local Micro Cloud Foundry, specify http://api.<appname>.cloudfoundry.me, where appname is the domain you registered for your application at the Micro Cloud Foundry Web site. See [Installing Micro Cloud Foundry](http://docs.cloudfoundry.com/infrastructure/micro/installing-mcf.html).

The following command targets the PaaS Cloud Foundry:

prompt$ vmc target https://api.cloudfoundry.com

To determine your current target, execute the vmc target command without any parameters:

prompt$ vmc target

* Login using the user credentials you received via email after you registered with Cloud Foundry. Your username is typically your email address.

prompt$ vmc login

* Ensure you have successfully logged in by retrieving information about your account:

prompt$ vmc info

* Change your password:

prompt$ vmc passwd

* View the full list of VMC commands, along with their parameters and a brief description, by executing the vmc help command:

prompt$ vmc help

You have now successfully installed vmc and run a few basic commands.

## Verifying the Installation by Deploying a Sample Application

Now that you have installed VMC and logged in to your target, you can start deploying applications to the Cloud.

This section shows how to deploy a simple application that does not require any services (such as MySQL or RabbitMQ). The purpose of the section is for you to quickly get a feel for VMC and Cloud Foundry by deploying and running a very basic application. Later sections describe how to configure your application to use services that connect to databases or manage messaging.

* Create a simple application that does not require any services and package it appropriately, such as into a \*.war file for Spring applications.

If you do not currently have an application, see [Creating a Simple Sinatra Application](http://docs.cloudfoundry.com/tools/vmc/installing-vmc.html#creating-a-simple-sinatra-application) for instructions on how to create a basic Ruby Hello World application using Sinatra in just a few minutes.

* Open a terminal window (Linux) or command prompt (Windows) and change the directory that contains your application.

For example, if you created the simple Ruby [Hello World](http://docs.cloudfoundry.com/tools/vmc/installing-vmc.html#creating-a-simple-sinatra-application) application using Sinatra:

prompt$ cd /usr/bob/sample-apps/hello

* Deploy your application using the vmc push command, which interactively prompts for deployment information:

prompt$ vmc push

For prompts that require a yes or no answer, the default value is shown using capital letters. For example, if “yes” is the default, you’ll see [Yn].

The following sample output also shows the responses you should provide; for clarity, default values are specified explicitly. See further explanations about these prompts after the example:

Would you like to deploy from the current directory? [Yn] Yes

Application Name: hello

Application Deployed URL: 'hello.cloudfoundry.com'? hello-bob.cloudfoundry.com

Detected a Sinatra Application, is this correct? [Yn] Yes

Memory Reservation [Default:128M] (64M, 128M, 256M, 512M or 1G) (Press Enter to take default)

Would you like to bind any services to 'hello'? [yN]: No

After you complete the prompts, vmc provides the following output for a successful push (deployment):

Uploading Application:

Checking for available resources: OK

Packing application: OK

Uploading (0K): OK

Push Status: OK

Staging Application: OK

Starting Application: OK

The Application Name refers to the internal name of the application as well as the actual file you want to deploy without its file extension, hello in our example. The Application Deployed URL is the URL that you use in your browser to run the application after it has successfully deployed and started on Cloud Foundry. Be sure that you specify a unique deployment URL, or vmc will return an error message that the URI has already been taken or reserved. In the example above, the URL is hello-bob.cloudfoundry.com.

Verify that your application is available by executing the vmc apps command:

$ vmc apps

+--------------+----+--------+-------------------------------+----------+

| Application | # | Health | URLS | Services |

+--------------+----+--------+-------------------------------+----------+

| hello | 1 | RUNNING| hello-bob.cloudfoundry.com | |

+--------------+----+--------+-------------------------------+----------+

Run your application in your browser by going to the URL you provided to the vmc push command, which in the example above is hello-bob.cloudfoundry.com.

If, for example, you deployed the Hello World Sinatra application, you should see the text Hello from Cloud Foundry in your browser.



## Updating the Deployment

Now that you have your first application deployed, it is easy to update if you make changes to it, as describe in the following procedure.

Change your application in some way so that, when you run it, you will know which version it is.

For example, in the simple Hello World Sinatra application contained in hello.rb, change the text Hello from Cloud Foundry to Hello from Cloud Foundry and VMware.

At your command prompt or terminal, be sure you are still located in the directory that contains your application file (/usr/bob/sample-apps/hello.rb in our example) and execute the vmc updatecommand, specifying the name of your application, which in our example is hello:

$ vmc update hello

Uploading Application:

Checking for available resources: OK

Packing application: OK

Uploading (0K): OK

Push Status: OK

Stopping Application: OK

Staging Application: OK

Starting Application: OK

In your browser, refresh your application and you will see your changes:



## Creating a Simple Sinatra Application

If you have not already done so, download and install the [Sinatra Web framework](http://www.sinatrarb.com/) on your computer.

Create the directory in which the new application will live. For example:

prompt$ mkdir /usr/bob/sample-apps/hello

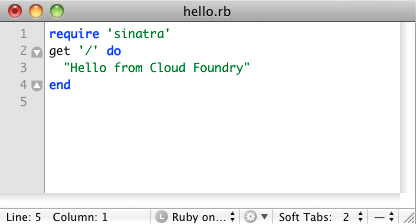
Using your favorite text editor, create a file called hello.rb in this new directory with the following contents:

require 'sinatra'

get '/' **do**

"Hello from Cloud Foundry"

**end**



## Next Steps

* [Installing Micro Cloud Foundry](http://docs.cloudfoundry.com/infrastructure/micro/installing-mcf.html)
* [Deploying and Managing Applications](http://docs.cloudfoundry.com/tools/deploying-apps.html)
* [Configuring Applications to Use Cloud Foundry](http://docs.cloudfoundry.com/frameworks.html)
* [VMC Quick Reference Guide](http://docs.cloudfoundry.com/tools/vmc/vmc-quick-ref.html)
* [Debugging](http://docs.cloudfoundry.com/tools/vmc/debugging.html)

# How to Install Ruby and RubyGems

## How to Install Ruby and RubyGems

**Tags:**[tutorial](http://docs.cloudfoundry.com/tags/tutorial)[ruby](http://docs.cloudfoundry.com/tags/ruby)[vmc](http://docs.cloudfoundry.com/tags/vmc)[Gemfile](http://docs.cloudfoundry.com/tags/Gemfile)

**Last Updated:**2012-11-13

1  
[inShare](javascript:void(0);)

The following sections provide basic information about installing Ruby and RubyGems on Windows and a variety of Linux computers.

## Windows

Download and install [Ruby Installer for Windows](http://www.rubyinstaller.org/). The installer already includes RubyGems.

Be sure you use the Ruby-enabled command prompt window when you later install and use vmc. You access this command prompt from the Windows Start menu (**All Programs → Ruby <version> → Start Command Prompt with Ruby**).

Finally, update RubyGems from the Ruby Command Prompt:

prompt> gem update --system

### Support for Windows Gemfiles

Once you have Ruby installed, you can follow the instructions on [Ruby Apps on Cloud Foundry](http://docs.cloudfoundry.com/frameworks/ruby/ruby-cf.html) to learn the details of pushing an app and working with your Gemfile

One thing to note for Windows users is the following:

When a Gemfile.lock is generated on a Windows machine, it often contains gems with Windows-specific versions. Versions of gems such as mysql2, thin, and pg end up containing “-x86-mingw32”.

For example, running bundle install on a Windows machine with a Gemfile that looks like this:

gem 'sinatra'

gem 'mysql2'

gem 'json'

Would result in a Gemfile.lock that looks like this:

GEM

remote: http://rubygems.org/

specs:

json (1.7.3)

mysql2 (0.3.11-x86-mingw32)

rack (1.4.1)

rack-protection (1.2.0)

rack

sinatra (1.3.2)

rack (~> 1.3, >= 1.3.6)

rack-protection (~> 1.2)

tilt (~> 1.3, >= 1.3.3)

tilt (1.3.3)

PLATFORMS

x86-mingw32

DEPENDENCIES

json

mysql2

sinatra

Now Cloud Foundry will be able to properly install these gems without the need to modify your Gemfile.lock

Ubuntu

From a terminal, use the apt-get command-line tool to install Ruby and RubyGems, as shown below.

1. Install the full Ruby package and RubyGems:
2. prompt$ sudo apt-get install ruby-full rubygems

Consult your system administrator for any required authentication credentials for the sudo command.

1. Test to ensure that the gem command is in your path:
2. prompt$ which gem

If the command is not found, then update your PATH variable accordingly. For example, you can update your .bashrc file with the following line:

export PATH=$PATH:/var/lib/gems/1.8/bin

1. Update RubyGems (Ubuntu 10.04 only):
2. prompt$ sudo gem install rubygems-update
3. prompt$ sudo /var/lib/gems/1.8/bin/update\_rubygems

## RedHat/Fedora

From a terminal, use the yum command-line tool to install Ruby and RubyGems, as shown below.

1. Install Ruby:

prompt$ sudo yum install ruby

1. If you are using RedHat Enterprise Linux 6, enable the Optional channel for your host by logging into [Red Hat Network (RHN)](https://rhn.redhat.com/).
2. Install RubyGems:

prompt$ sudo yum install rubygems