Cisco Intercloud Services

Shipped

Getting Started

Published 12/14/2015

Copyright Notice

Document History

Topic	Date of Change	Description
Initial publication	11/30/2015	

About This Document

This guide helps you get started with Shipped, including building, running, and deploying a project to a local computer and to the cloud.

Audience

This document is for users of all experience levels, and is simple enough for first-time users, but comprehensive enough to satisfy experienced users.

Related Documents

- Shipped User Guide
- Shipped API Guide
- Shipped CLI Guide

Conventions

Item	Description
bold	Menu, command.
mono-space font	Code, typed data.
italic OR < >	User input.
	**Note**. Contains information that might be useful. Ignoring a note has no negative consequences.
	**Tip**. Includes information such as helpful hints or a shortcut that might help you complete a task.
	**Important**. Includes information that might be easily overlooked and might cause unnecessary frustration. For example, configuration changes that only apply to the current session, or services that need restarting before an update will apply.
	**Warning**. Contains information that must not be ignored. Ignoring recommendations in Warnings may result in data loss or other catastrophic issues.

Support

To access CCS FAQs and other support resources, or to report support issues, or to open a support request, visit:

http://intercloud.cisco.com/support

Contents

About Shipped

Shipped is a web-based application from Cisco that supports a microservices model for cloud development. Shipped allows you to build, deploy, and run projects in minutes. You can use Shipped with existing projects or build new ones. Shipped sets up:

- Your project
- A development and deployment environment in a VM on your local laptop
- Local and remote GitHub repositories
- A CI (continuous integration) build so that every commit automatically runs a build
- A deployment environment in the cloud

Shipped is supported for Windows, Mac, and Ubuntu Linux.

Prerequisites

GitHub

You must have a GitHub account to use Shipped. GitHub is a cloud-based repository for software and has many collaboration features. Shipped uses it to store source code for the projects and services it manages.

To get a free GitHub account:

1. In a browser, navigate to https://github.com. The GitHub sign-in page opens.
2. Enter a valid email address and password, then click **Sign up for GitHub**. The GitHub subscription options page opens.
3. Choose the **Free** plan, then click **Finish sign up**.

   The GitHub site has a training wizard and extensive Help files that provide information on how to use this tool.

   You will need to verify your email address to continue using GitHub.

Repositories

You don't need to create a repository at this time. Shipped creates any repositories you need. Shipped can also manage an existing repository using a Shipped service, if you want.

Local Environments

Shipped creates a development environment on your computer that includes a local Git repository and a VM for deploying and testing your project. This requires some supporting software on your computer:

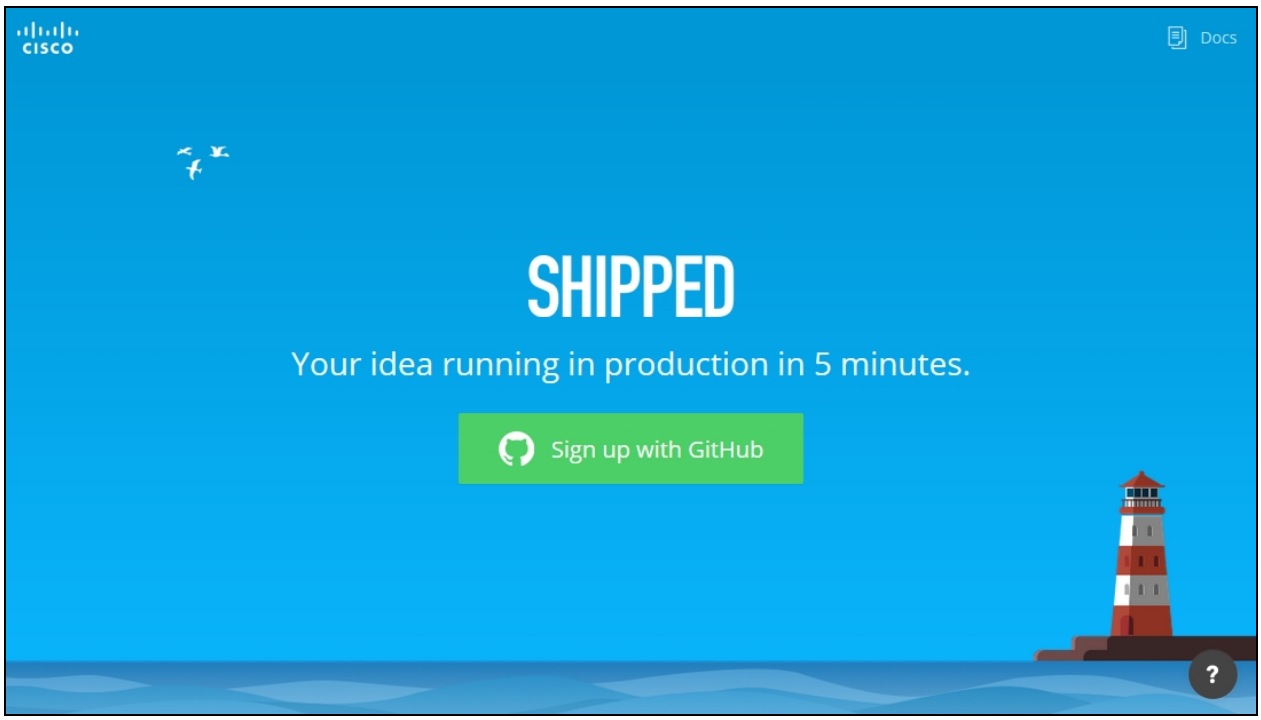
Software	Description	Download from
Git	A version control system that allows you to work with local repositories on your computer and link them to remote repositories in GitHub. You must have version 1.8.5 or higher.	http://www.git-scm.com/downloads Install instructions: https://help.github.com/articles/set-up-git/
Git Credentials Manager for Windows	Required for Windows only. Allows you to download private Git repositories without being prompted for a password.	https://github.com/Microsoft/Git-Credential-Manager-for-Windows/releases/tag/v1.0.0
VirtualBox	Provides support for running virtual machines (VMs) on your desktop. Shipped uses VirtualBox to host the VM running a local development and deployment environment for its projects. If necessary, Shipped will automatically install VirtualBox when you bootstrap your first project, but it might be more convenient to download and install it in advance.	https://www.virtualbox.org/wiki/Downloads
Vagrant	Manages a development environment containing one or more VMs and Docker containers. Shipped uses Vagrant to manage the VMs it creates with VirtualBox. If necessary, Shipped will automatically install Vagrant when you bootstrap your first project, but it might be more convenient to download and install it in advance.	https://www.vagrantup.com/downloads.html

Initial Setup

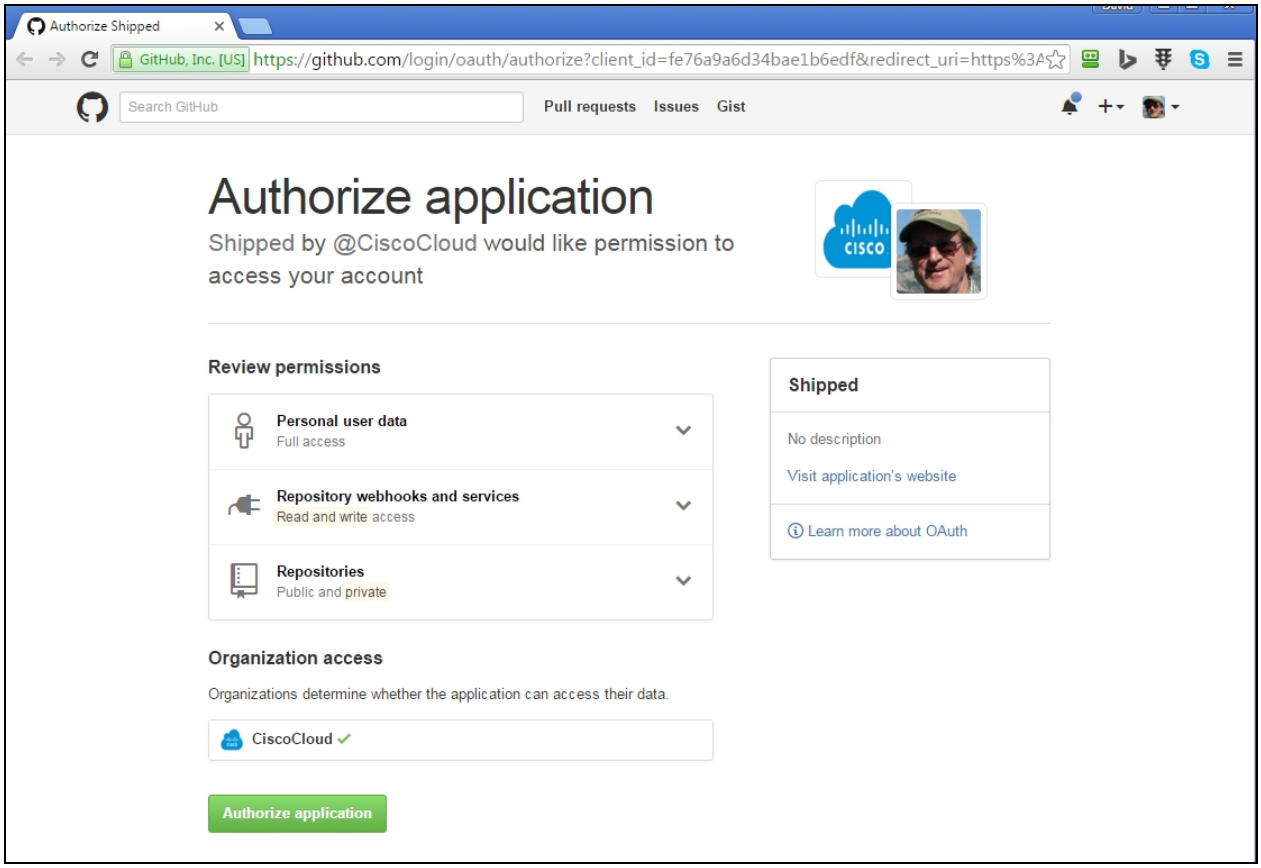
You can use Shipped with any modern browser, but for optimal results, Chrome is recommended.

1. In a browser, go to `https://ciscoshipped.io`.

   The sign up page opens.

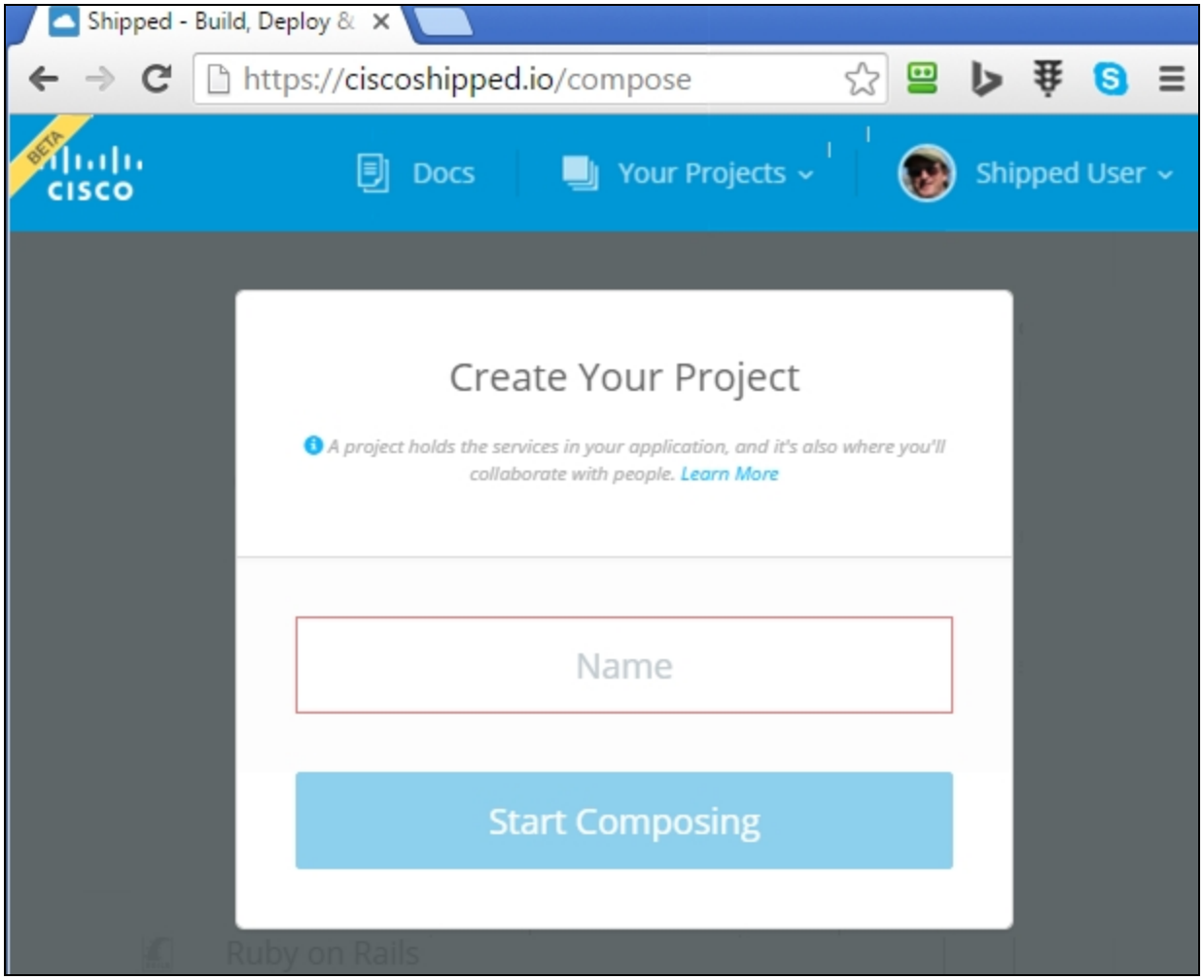


2. Click **Sign up with GitHub**. Shipped checks GitHub for your credentials.
3. Unless you're currently logged into GitHub, the GitHub sign in page opens.
4. Click **Authorize application** to allow Shipped to create and update GitHub repositories



This is only needed the first time you use Shipped.

5. The Shipped **Create Your Project** form opens.



If you've already created a Shipped project, Shipped will show information about it instead of this form. You can continue working with that project, or create a new project. See "Create a Project" on the next page.

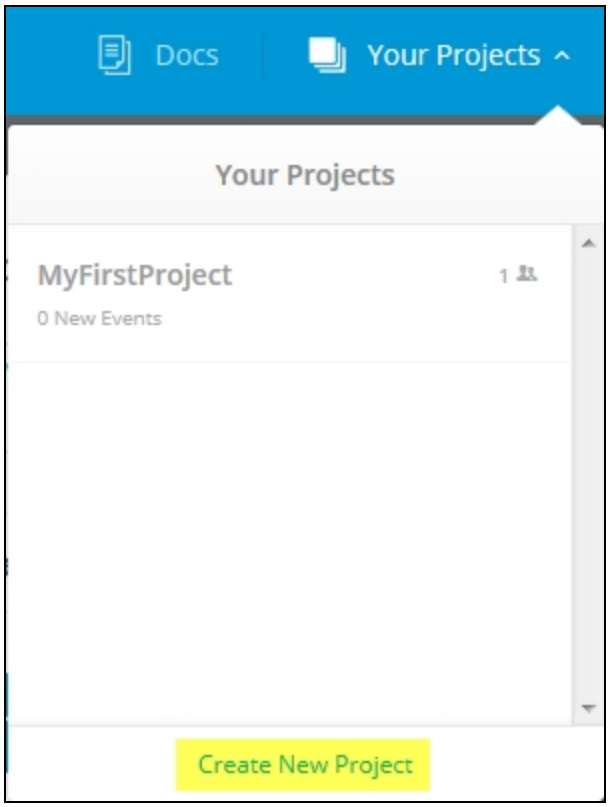
You are now ready to build your project.

Create a Project

A Shipped project represents an application built from one or more services, each independently deployed and communicating with other services using their APIs. For example, a project might consist of a web server service and a database service. A bigger project might include a dozen services, each developed and supported by its own team.

This example creates a project with a single service.

1. From the **Create Your Project** page, click the down arrow next to **Your Projects** in the top bar, then choose **Create New Project** from the drop-down list.



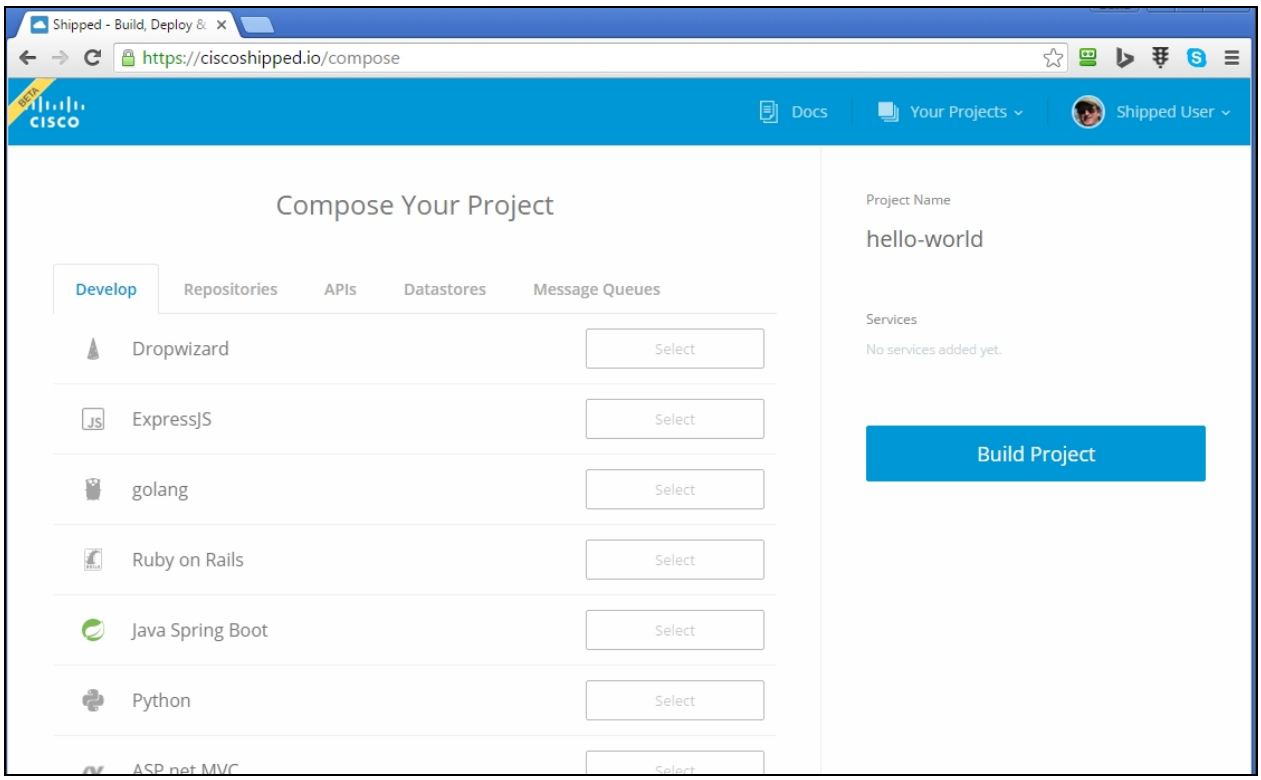
   The **Create New Project** form opens.

2. Enter a name for your project.

   A project name can only consist of letters, numbers, hyphens, and underscores, but no spaces.

3. Click **Start Composing**.

The **Compose Your Projects** page opens.



4. Determine the service you want to include in your project.

Service Type	Description
Develop	Development environments. Most projects need at least one service of this type.
Repositories	DockerHub images.
API	APIs for Cisco services, such as CMX or APIC-EM.
Datastores	Databases, such as MySQL or Cassandra.
Message Queues	Queuing services, such as RabbitMQ or Kafka.

For this example, the **PHP** service under the **Develop** tab is used.

5. From the **Develop** tab, click **Select** next to the PHP service.

   The **Add Service** form opens.

This form allows you to specify a GitHub repository to store the source code for the service. You can specify an existing or a new repository. If you specify an existing repository, it should be compatible with a PHP development environment. If you specify a new repository, Shipped will create it for you and initialize it with a basic application appropriate for the development environment.

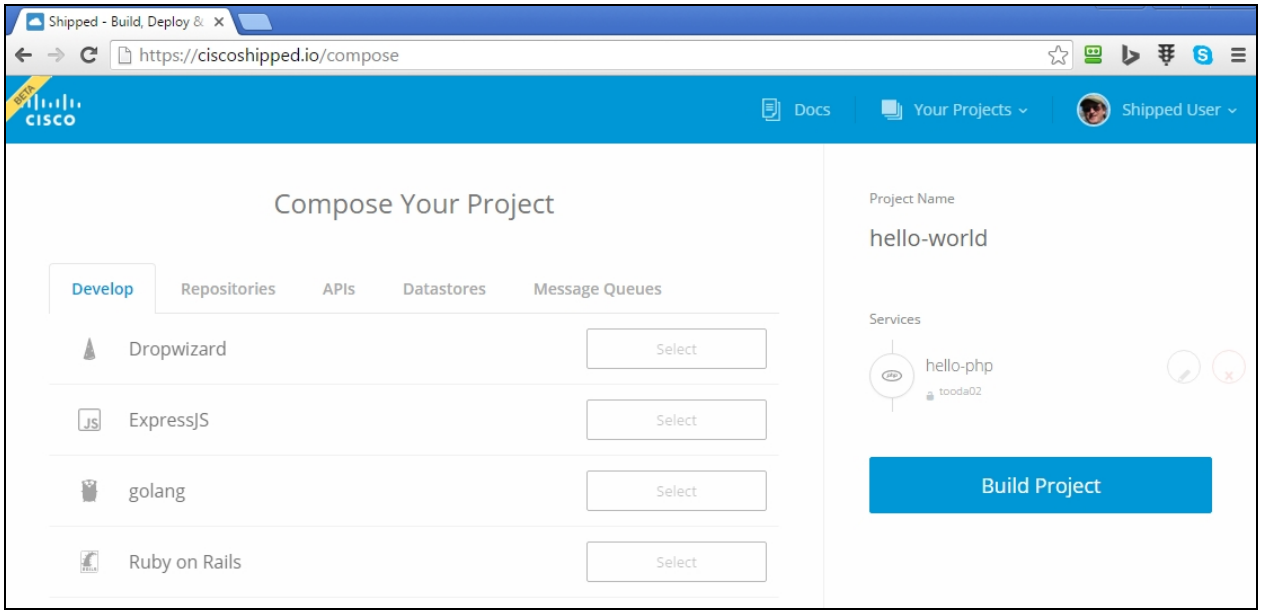
Make sure you have administrator rights when using an existing repository. Shipped needs this to install the webhooks required for a CI build.

6. Enter information into the fields:

Field	Description
New repository name	Enter a name for the GitHub repository. For example, "hello-php". A service name can only consist of letters, numbers, hyphens, and no spaces.
GitHub Organization	Optional. If you belong to an organization, choose it from the drop-down list.
Private / Public	Specify if the repository is Private or Public. • **Public** - must be used with free GitHub accounts. • **Private** - requires a paid GitHub account with capacity in your plan.

7. Click **Add Service**.

The **Compose Your Project** page opens showing the added service.

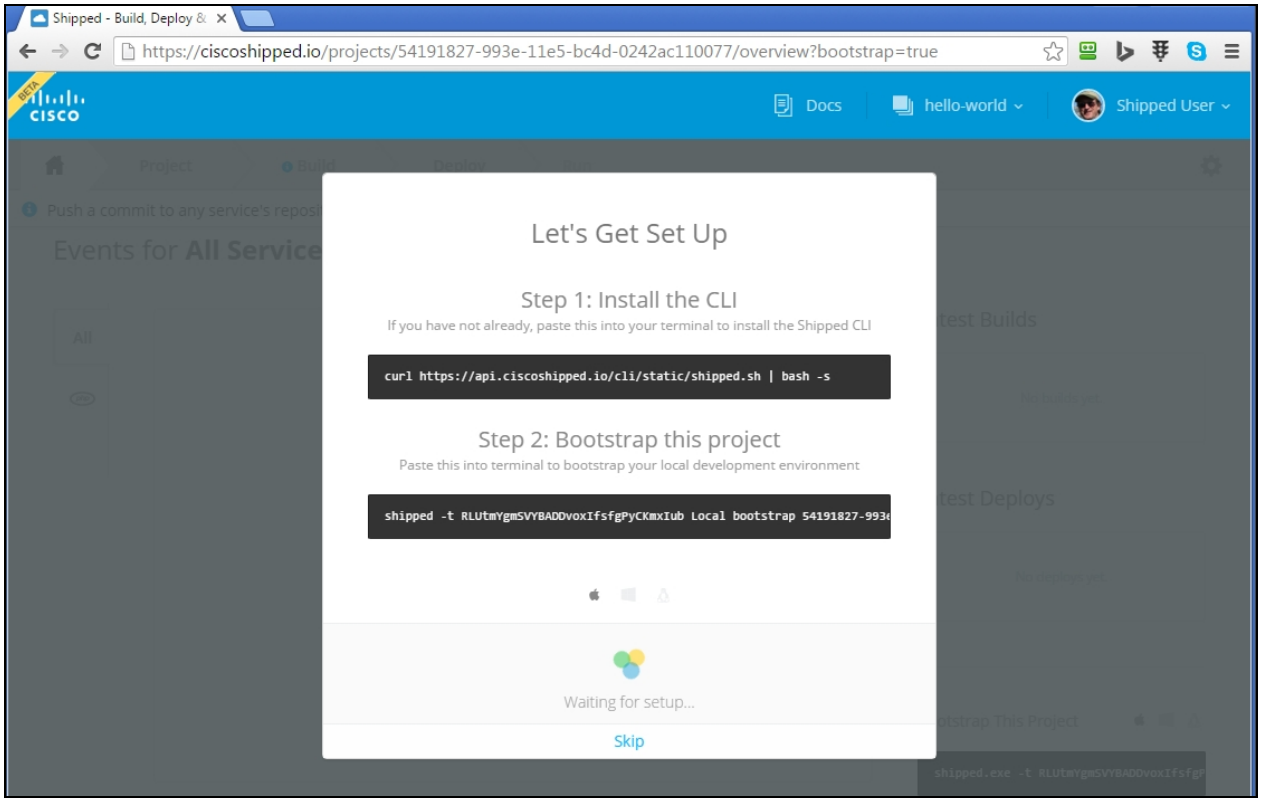


At this point (or at any point), you can add more services. This example uses one service.

8. Click **Build Project**.

   Shipped builds the project and sets up the GitHub repositories. While building, the progress bar is shown.

9. When the build is complete, the **Let's Get Set Up** form opens.



You are now ready to bootstrap your project.

Bootstrap a Project

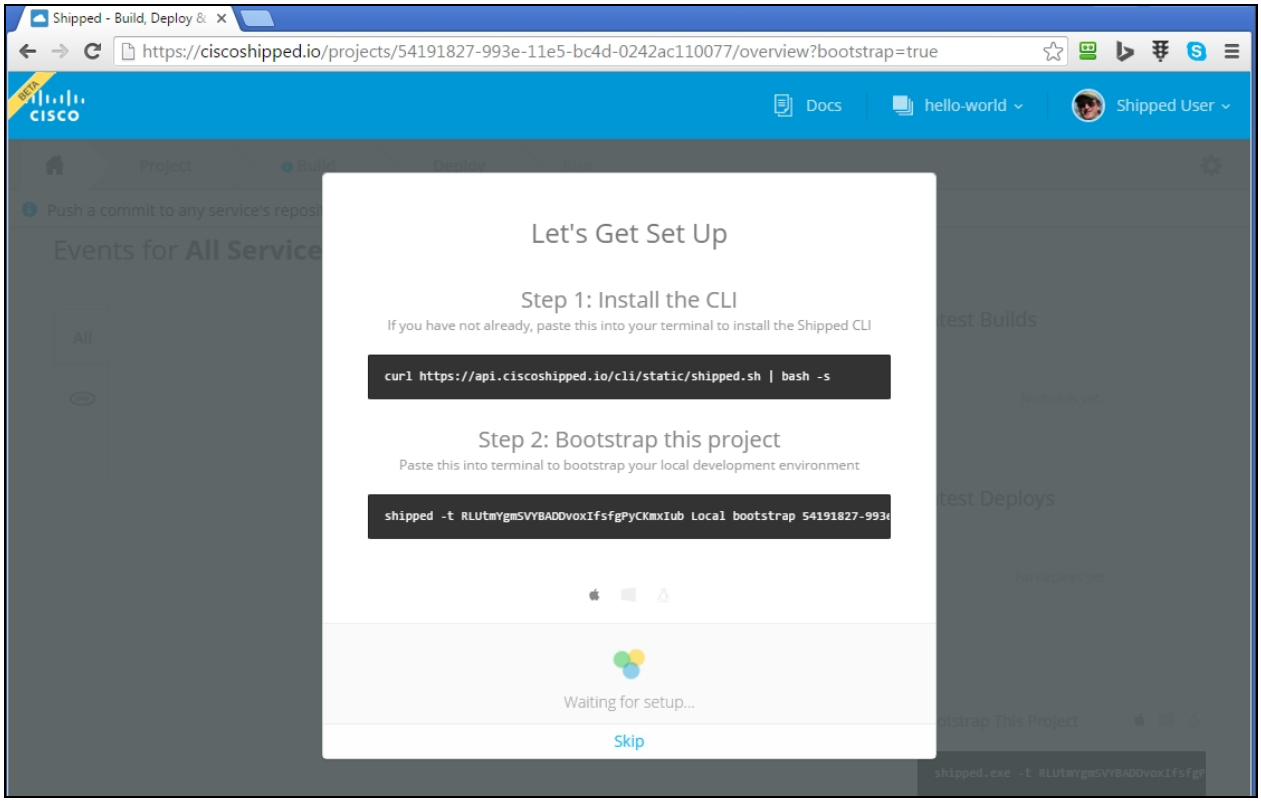
Bootstrapping a project enables you to develop it locally. When bootstrapping a project, Shipped:

- Creates a local directory for the project.
- Creates a subdirectory for each development and API service in the project. This subdirectory includes a local GitHub repository with the service source code.
- Adds files supporting Shipped for services based on existing repositories. Most of these are in .shipped subdirectories of the project and service directories. Shipped also creates .drone.yml and .drone.sec files to control the CI build.
- Verifies that Git, Vagrant, and VirtualBox is installed and set up, and if not, installs Vagrant or VirtualBox.
- Builds a VM containing a development and deployment environment for your project services. The VM includes a directory synced with the service directory on your laptop, so changes that you make on your laptop are immediately available to the VM.

To bootstrap your project:

1. Install the Shipped CLI. You only need to do this once.

   From the **Create Your Project** form, copy and paste the text from the first box into a terminal window on your computer, then press **Enter**.



This command downloads the Shipped executable. Copy it to a directory in your path.

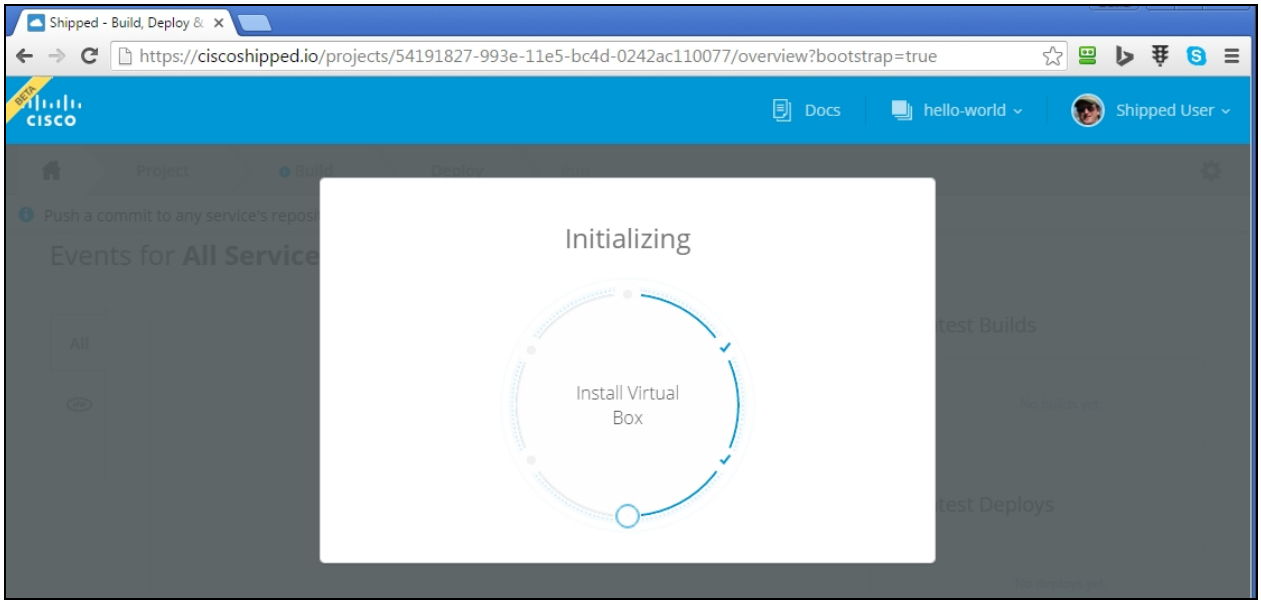
> The command you are given is automatically generated for the OS on your local computer. To get the code to bootstrap to another OS, click the appropriate icon at the bottom of the page.

2. After the first command has completed, from the **Create Your Project** form, copy and paste the text from the second box into a terminal window on your computer, then press **Enter**.

   Shipped creates the project directory in a subdirectory of the working directory of the terminal window.

   This process might take a few minutes. Shipped automatically saves this output onto the shipped-cli.log file in your working directory. You might be asked for a copy of this file when contacting Cisco for support.

   As bootstrap runs, the status appears in your browser.

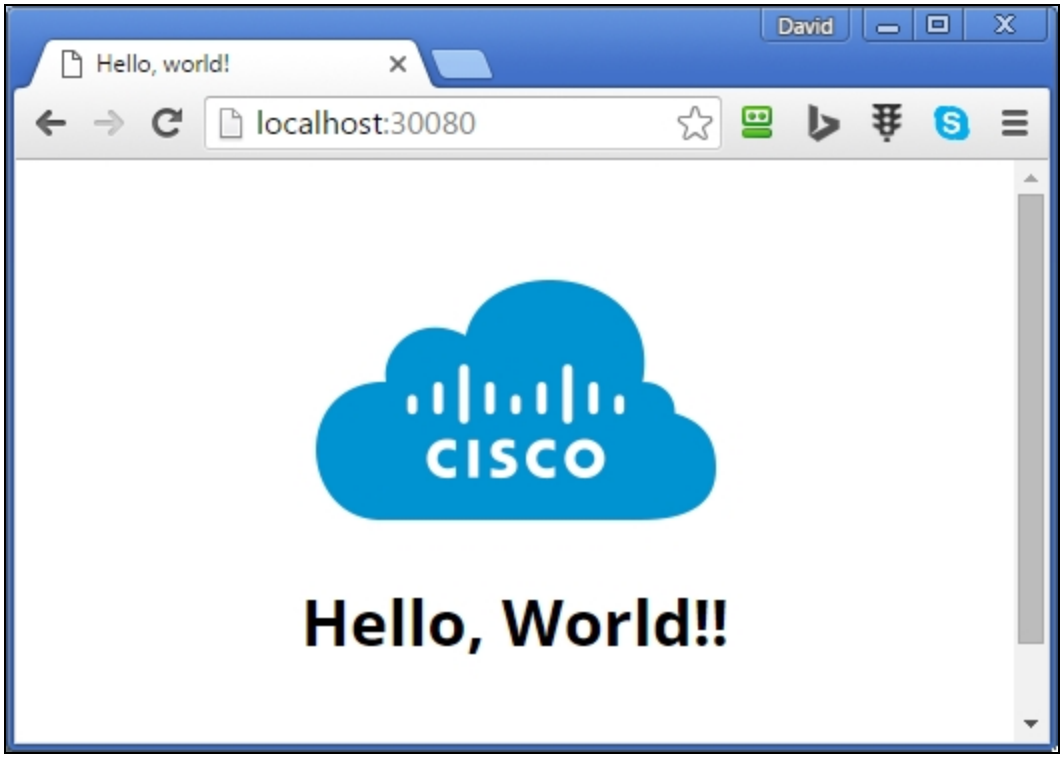


Bootstrap also writes output to your terminal window. The last few lines of output should be similar to:

```
==> hello_php: Creating the container...
    hello_php:   Name: hello_php
    hello_php:  Image: fc3079125298
    hello_php: Volume: /var/lib/docker/docker_1449100412_90831:/app
    hello_php: Volume: /var/lib/docker/docker_1449100412_21381:/vagrant
    hello_php:   Port: 30080:80
    hello_php:
    hello_php: Container created: 05f14554925a553b
==> hello_php: Starting container...
==> hello_php: Provisioners will not be run since container doesn't support SSH.
Project hello-world bootstrap complete
============================================================
dockerhost-54191827-993e-11e5-bc4d-0242ac110077 is running
1 service is running and is available at the web address below
----  ------------  -------  ------  -------------------
Type  Service Name  VM Name  Status  Service Web Address
----  ------------  -------  ------  -------------------
Repo  hello-php     hello_php  running  http://localhost:30080
============================================================
```

The bootstrap is now complete and running the application in a web server in the VM on your computer.

You can access your application from the URL shown. In this example, localhost:30080.



The application is running locally. You are now ready to deploy it to the cloud.

Deploy a Project

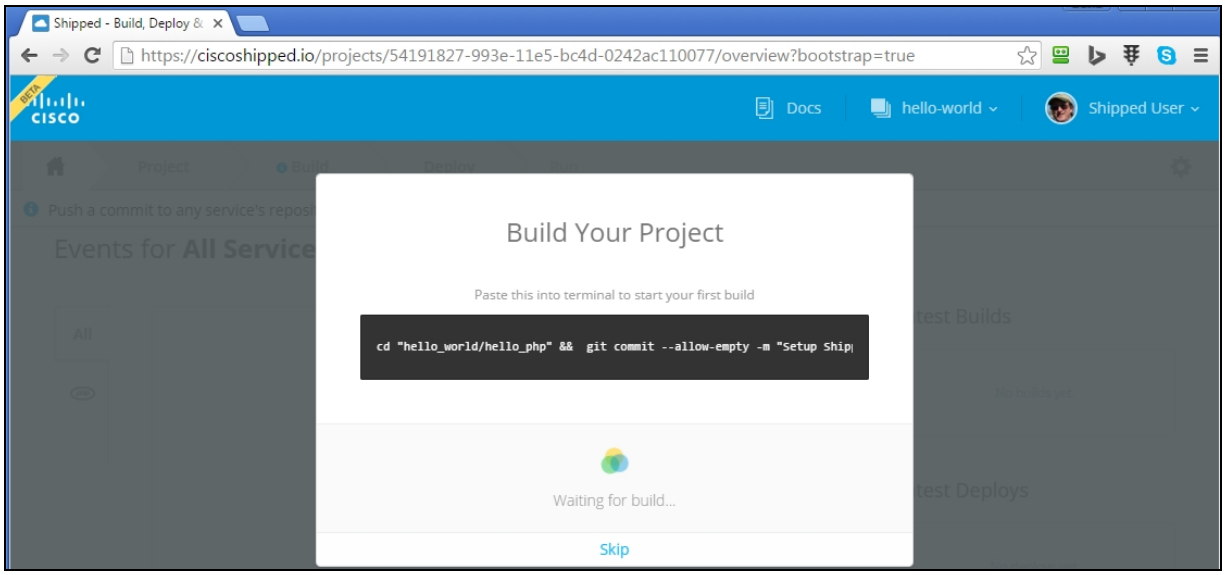
Your bootstrapped project includes everything you need to develop it on your computer. You can make changes to the files in the service directory, and immediately see the effects in the application. When you are satisfied with how the application is running, you can deploy it to the cloud.

1. Commit changes to the remote GitHub repository.

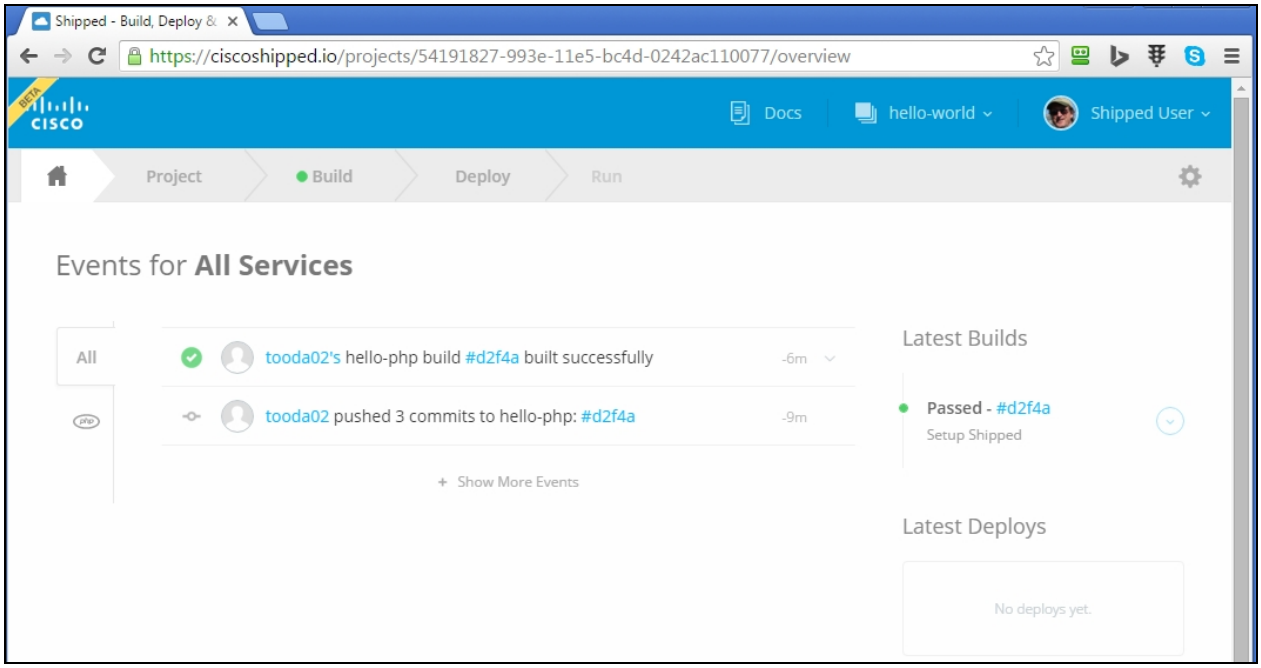
   Shipped automatically sets up a CI build, so any commit automatically starts a build. When the build completes, you can deploy its output to the cloud.

   To perform a commit and start a build, use one of these methods:

   - Commit and push to the remote repository using Git commands (see http://gitref.org/basic/). Shipped automatically starts a build.
   - Use the Shipped CLI. In a terminal window, change to the project directory, then enter:
     ```
     shipped local commit m=”Setup Shipped”
     ```
   - Copy and paste the command from the Shipped **Build Your Project** form (see "Bootstrap a Project" on page 16) into a terminal window within the project directory.



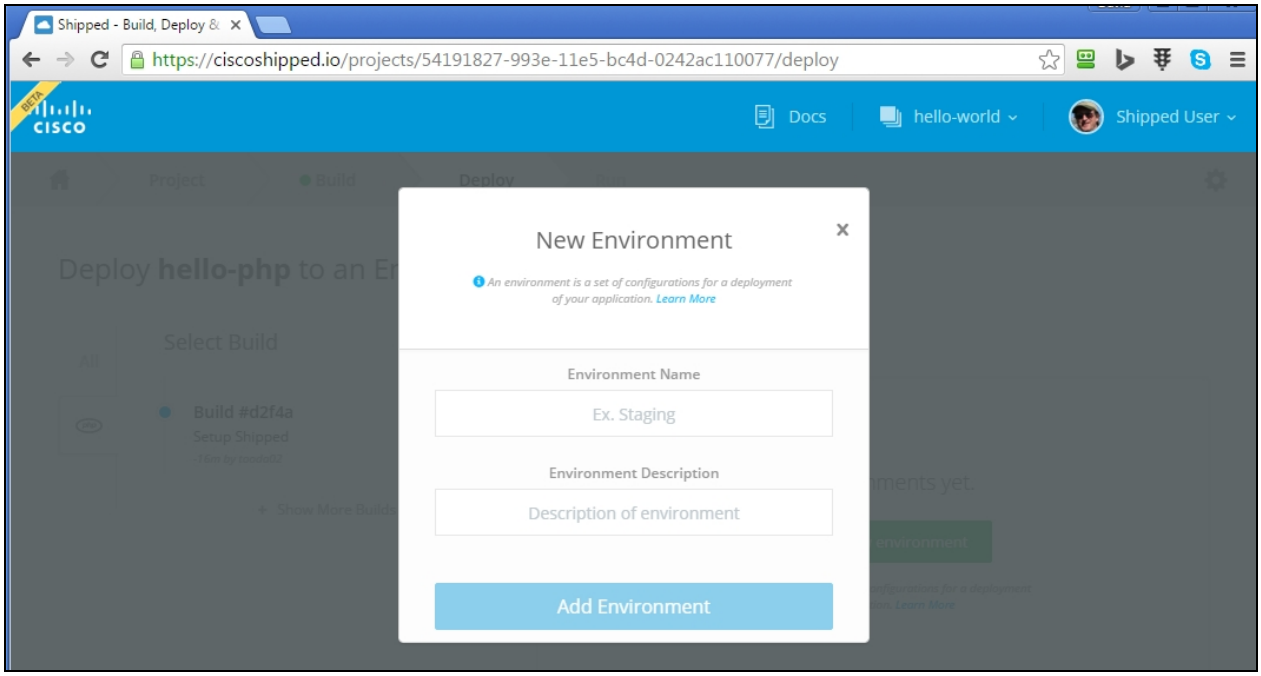
2. When the build completes, the Shipped project **Events** page opens.



3. From the **Deploy** tab, click **Create a new environment**.

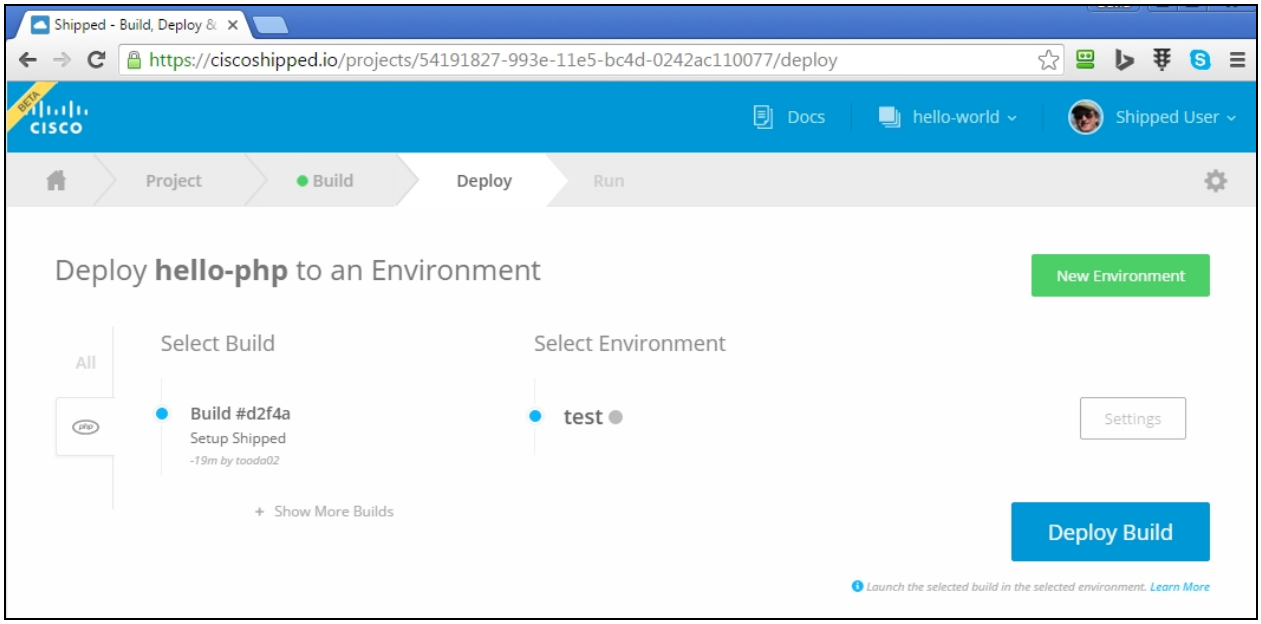
   An environment is a VM instance in the cloud where the service can be deployed.

   The **New Environment** form opens.



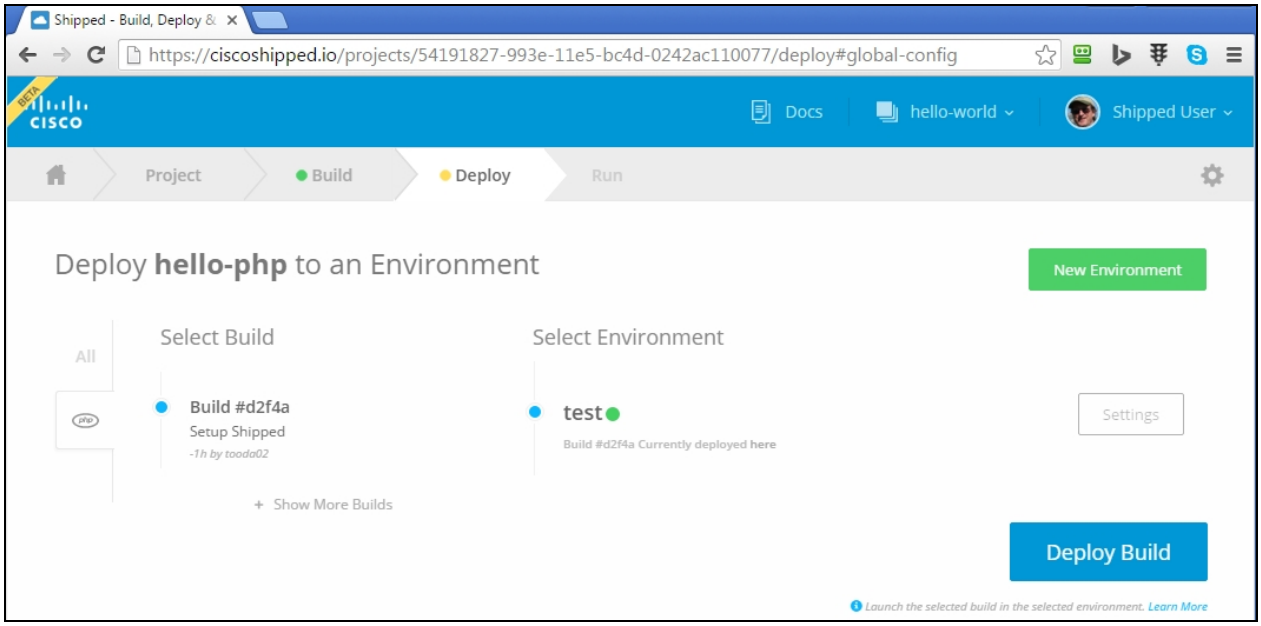
4. Enter a name and a brief description of the environment. This example uses "test".
5. Click **Add Environment**.

   The **Deploy** page shows the new environment. Current selections are indicated by blue dots.



6. Optional. To adjust environment settings (such as memory and CPU in the deployed VM), click **Settings** next to the selected environment.
7. Click **Deploy Build**.

   When the deployment is successful, a message appears under the environment name with a link to your deployed service.



You have successfully deployed your application to the cloud. You can click the link (the word "here") to access the application running in the cloud.

Glossary

A

API
The Shipped API runs the same commands as using the interface. The API can be scripted.

B

build
A Git commit to a specific service.

buildpack
The starting point for a service, based on popular web frameworks.

C

Cisco Intercloud Services
The Cisco Intercloud Infrastructure as a Service (IaaS) that contains OpenStack plus many additional services.

CLI
The Shipped CLI lets you run Shipped commands through the terminal window.

config
Contains instructions to run a service.

D

deploy targets
Either cloud or on-premise clusters where you provision your Shipped services.

Docker
An OpenStack application that automates the deployment of a target. Docker makes the application run regardless of hardware, language, or hosting provider.

Drone
Triggers a build when it senses a commit being made on any branch.

E

environment
Manages namespace deployment, such as Development, Staging, or Production.

P

project
The container for the services used in your application.

R

release
A service deployment snapshot.

S

service
GitHub repositories that are part of a project.

U

users
End-users that have access to your project or service.

V

Vagrant
An OpenStack application that manages VMs and cloud instances. Commonly used to set up development or staging environments.