



PRODUCT DOCUMENTATION

# Scheduler for PCF

## Documentation<sup>®</sup>

Version 1.1

Published: 12 Dec 2018

Rev: 01

© 2018 Pivotal Software, Inc.

## Table of Contents

Table of Contents	2
Scheduler for PCF	3
Release Notes	4
Architecture	6
Installing and Configuring Scheduler for PCF	7
Monitoring Scheduler for PCF	9
Using Scheduler for PCF	10
Scheduling Jobs	12
Scheduling Calls	15

## Scheduler for PCF

This documentation describes Scheduler for Pivotal Cloud Foundry (PCF).

## Overview

Scheduler for PCF is a service for scheduling the execution of Diego tasks, such as database migrations, emails, or batch jobs, as well as the execution of outbound HTTP calls.

Scheduler for PCF enables developers to do the following:

- Create, run, and schedule jobs and view job history.
- Create, run, and schedule calls and view call history.

You can interact with the service through the Cloud Foundry Command Line Interface (cf CLI), [Apps Manager](#), and the [Scheduler HTTP API](#).

## Product Snapshot

The following table provides version and version-support information about Scheduler for PCF.

Element	Details
Version	v1.1.2
Release date	December 21, 2017
Compatible Ops Manager version(s)	v1.11.x, v1.12.x, v2.0.x
Compatible Elastic Runtime version(s) *	v1.11.x, v1.12.x
Compatible Pivotal Application Service version(s) *	v2.0.x
IaaS support	AWS, Azure, GCP, OpenStack, and vSphere

\* As of PCF v2.0, Elastic Runtime is renamed to Pivotal Application Service (PAS).

## Requirements

Scheduler for PCF has the following requirements:

- MySQL for PCF v1.10. The service is available on [Pivotal Network](#).

## Limitations

- If your app uses a buildpack that does not generate a `web` process type, such as Ruby or Python, you should do the following:

1. Before pushing your app, create a [Procfile](#) in the root directory of the app.
2. Declare a `web` process type in the file.

If you do not declare this process type, your app will not be accessible through the cf CLI after you create Scheduler jobs for it.

- The maximum number of tasks that you can schedule is determined by the memory and disk quotas in the Scheduler for PCF org and space. See [Running Tasks](#) for more information.

## Release Notes

This topic contains release notes for Scheduler for Pivotal Cloud Foundry (PCF).

### v1.1.4

**Release Date:** December 6, 2018

#### Features included in this release:

- Update OpenJDK
- Update MariaDB Connector

### v1.1.2

**Release Date:** December 21, 2017

#### Features included in this release:

- Updates the stemcell
- Fixes bugs

### v1.1.0

**Release Date:** October 16, 2017

#### Features included in this release:

- Users can do the following:
  - Create jobs
  - Run jobs
  - Schedule jobs
  - List job history
  - Create calls
  - Run calls
  - Schedule calls
  - List call history
- Internal credentials are stored in BOSH CredHub. If you want to access these credentials, you must use the CredHub CLI or the Ops Manager API instead of the **Credentials** tab of the Scheduler tile. For instructions on how to retrieve Scheduler credentials, see [Retrieving Credentials from Your Deployment](#).

#### Changes in this release:

- Scheduler for PCF supports all available MySQL for PCF plans.
- HTTP calls can be disabled in the Scheduler for PCF tile.

#### Known issues in this release:

- Scheduler for PCF provides only one service plan, `standard`, which allows you to schedule as many tasks and calls as needed and at any interval.

## v1.0.4

Release Date: August 11, 2017

### Features included in this release:

- Create jobs.
- Run jobs.
- Schedule jobs.
- List job history.

### Changes in this release:

- Scheduler for PCF supports floating stemcells in 3363 version line.

### Known issues in this release:

- Users must have a `p-mysql` service plan named `1gb` with at least 1,000 MB of disk space.
- Scheduler for PCF provides only one service plan, `standard`, which allows you to schedule as many tasks as needed and at any interval.

## v1.0.2

Release Date: April 19, 2017

### Features included in this release:

- Create jobs.
- Run jobs.
- Schedule jobs.
- List job history.

### Known issues in this release:

- Users must have a `p-mysql` service plan named `1gb` with at least 1,000 MB of disk space.
- Scheduler for PCF provides only one service plan, `standard`, which allows you to schedule as many tasks as needed and at any interval.
- If you make the system domain in the `system` org private, Scheduler for PCF does not run.

## Architecture

This topic describes the architecture of Scheduler for Pivotal Cloud Foundry (PCF).

### HA Topology

In a highly available resource configuration, the Scheduler for PCF service uses the following:

- Three Scheduler instances, each running the Scheduler HTTP API server and a Scheduler Engine.
- Three Scheduler Service Broker instances

You can reduce the number of Scheduler instances to one using the Cloud Foundry Command Line Interface (cf CLI).

### Data Persistence

Scheduler relies on a MySQL datastore to persist data, including jobs, calls, and history. Each call or schedule can require up to 10 MB of database capacity to store history.

## Installing and Configuring Scheduler for PCF

This topic describes how to install and configure Scheduler for Pivotal Cloud Foundry (PCF).

### Prerequisites

Before you install the Scheduler for PCF tile, you need to configure a MySQL for PCF service plan. The minimum resource configuration of the MySQL plan requires the following:

- Service plan name: 1gb
- Storage quota: 1,000 MB
- Concurrent connections quota: 40

For more information, see [Add a Plan](#) in the MySQL for PCF documentation.

### Download and Install Scheduler for PCF

1. Download the product file from [Pivotal Network](#).
2. Navigate to the Ops Manager Installation Dashboard and click **Import a Product** to upload the product file.
3. Under the **Import a Product** button, click + next to the version number of Scheduler for PCF. This adds the tile to your staging area.
4. Click the newly added **Scheduler for PCF** tile.

### Configure Scheduler for PCF

Follow the steps below to configure the Scheduler for PCF tile.

#### Configure AZs and Networks

Follow the steps below to choose an Availability Zone (AZ) to run the Scheduler Service Broker and to select networks.

1. Click **Assign AZs and Networks**.
2. Configure the fields as follows:

Field	Description
Place singleton jobs in	Select the AZ for executing the Scheduler Errands. All Scheduler components execute as apps and do not require VMs.
Balance other jobs in	Ignore this field.
Network	Select a subnet for the Scheduler Errands. Use the subnet that includes the Elastic Runtime component VMs.

 **NOTE:** The network selected is used only by Errand VMs and does not apply to the Scheduler runtime.

3. Click **Save**.

#### Configure Scheduler Options

1. Click **Scheduler Configuration**.
2. Configure the fields as follows:

Field	Description
Enable outbound HTTP calls	The field is enabled by default. Disable this feature if you want to prevent users from scheduling outbound HTTP calls from the Scheduler for PCF service.

Name	Enter the name of your MySQL service broker. For example, <code>p.mysql</code> .
Plan Name	Enter the name of your MySQL service plan, which is used to provision a database for persistent storage of Scheduler data, including jobs, calls, and history.

3. Click **Save**.

## Verify Resource Config

1. Click **Resource Config**.
2. Verify the settings.
3. Click **Save**.

## Verify Stemcell Version

1. Click **Stemcell**.
2. Verify the settings. If you need to import a new stemcell version, see the *Download Stemcell* section for your IaaS: [AWS](#), [Azure](#), [GCP](#), or [vSphere](#).
3. Click **Save**.
4. Return to the Ops Manager Installation Dashboard and click **Apply Changes**.

## Monitoring Scheduler for PCF

This topic describes the logs and metrics you can use to monitor the health and performance of Scheduler for Pivotal Cloud Foundry (PCF). For general information about logging and metrics in PCF, see [Logging and Metrics](#).

### Scheduler Metrics

Scheduler for PCF emits metrics to Loggregator if the `metrics-forwarder` service is bound to the `scheduler` app in the `system` org and `p-scheduler` space.

Scheduler for PCF emits the following metrics:

Data Source	Description	Metric Unit
<code>gauge.scheduler.jobs.executed</code>	The number of jobs Scheduler has executed.	Integer
<code>gauge.scheduler.jobs.failed</code>	The number of jobs Scheduler failed to execute.	Integer
<code>gauge.scheduler.jobs.rejected</code>	The number of jobs Scheduler requested the Cloud Controller to execute but failed to start a task.	Integer
<code>gauge.scheduler.jobs.total</code>	The number of jobs that currently exists in Scheduler.	Integer
<code>gauge.scheduler.calls.executed</code>	The number of calls Scheduler has executed.	Integer
<code>gauge.scheduler.calls.failed</code>	The number of calls Scheduler failed to execute.	Integer
<code>gauge.scheduler.calls.total</code>	The number of calls that currently exists in Scheduler.	Integer

## Using Scheduler for PCF

This topic provides instructions for using Scheduler for Pivotal Cloud Foundry (PCF).

You can interact with the service through the Cloud Foundry Command Line Interface (cf CLI), [Apps Manager](#), and the [Scheduler HTTP API](#) to configure jobs and outbound HTTP calls and to review history. For general information, see [Managing Service Instances with the cf CLI](#).

## Prerequisites

To start using Scheduler for PCF, you need the following:

- A PCF deployment with [Scheduler for PCF](#) installed and listed in the [Marketplace](#).
- A [Space Developer](#) account.
- (Optional) The cf CLI v6.23.0 or greater and the Scheduler for PCF CLI plugin installed on your local machine. The Scheduler for PCF CLI plugin is packaged with the Scheduler for PCF tile on [Pivotal Network](#).

## Create and Bind a Service Instance Using the cf CLI

Every app and service in PCF is scoped to a [space](#). This means that an app can use a service only if an instance of the service exists in the same space.

The Scheduler for PCF service is a singleton service. Only one service instance can be created in a space.

## Confirm Service Availability

For apps to use a service, the service must be available in the Marketplace. To confirm the availability of Scheduler for PCF, perform the following steps:

1. Run `cf marketplace` from the command line.
2. If the output lists `scheduler-for-pcf` in the `service` column, Scheduler for PCF is available. If the service is not available, install it. See [Installing and Configuring Scheduler for PCF](#) for more information.

```
$ cf marketplace
Getting services from marketplace in org my-org / space my-space as user@example.com...
OK
service    plans   description
[...]
scheduler-for-pcf  standard  Scheduler service
[...]
```

## Create a Service Instance

To create an instance of the Scheduler for PCF service, run `cf create-service scheduler-for-pcf standard SERVICE-INSTANCE-NAME`, replacing `SERVICE-INSTANCE-NAME`

with a name of your choice. After you create the service instance, this instance name appears under `name` in the output of the `cf services` command.

See the following example:

```
$ cf create-service scheduler-for-pcf standard my-instance
Creating service my-instance in org my-org / space my-space as user@example.com...
OK

$ cf services
Getting services in org my-org / space my-space as user@example.com...
OK
name      service      plan  bound apps  last operation
my-instance  scheduler-for-pcf  standard          create succeeded
```

You can create only one instance in a space. If you attempt to create more than one instance in a space, you receive an error response.

## Bind a Service Instance to Your App

For an app to use a service, you must bind it to a service instance. Do this after you push or re-push the app using `cf push`.

To bind an app to a Scheduler for PCF instance, run `cf bind-service APP-NAME SERVICE-INSTANCE-NAME`, replacing `APP-NAME` with the name of the app you want to use the Scheduler for PCF service for and `SERVICE-INSTANCE-NAME` with the name you provided when you ran `cf create-service`.

```
$ cf bind-service my-app my-instance
Binding service my-instance to my-app in org my-org / space my-space as user@example.com...
OK
TIP: Use 'cf push' to ensure your env variable changes take effect
```

## Manage Jobs and Calls

For information about the CLI operations that you can perform to manage jobs and calls in Scheduler for PCF, see [Using Jobs](#) and [Using Calls](#).

If you want to manage jobs and calls through the Scheduler HTTP API, see the [Scheduler for PCF API Documentation](#).

## Using Scheduler for PCF in Apps Manager

For information about binding Scheduler for PCF to your app and scheduling tasks through Apps Manager, see [Managing Apps and Service Instances](#) and [Using Apps Manager](#).

## Scheduling Jobs

This topic provides instructions for managing jobs in Scheduler for Pivotal Cloud Foundry (PCF).

## Manage Jobs

You can use Scheduler for PCF to schedule execution of tasks on PCF, including database migrations, emails, and batch jobs. See the following sections to learn more about creating, running, and scheduling jobs and viewing job history.

**Note:** If you want to use the Cloud Foundry Command Line Interface (cf CLI) for managing jobs, you must install the Scheduler for PCF CLI plugin on your local machine. This plugin is packaged with the Scheduler for PCF tile on [Pivotal Network](#). For more information, see the [Prerequisites](#) section of the Using Scheduler for PCF topic.

### Create a Job

To execute a task related to an app, create a job by running the `cf create-job APP-NAME JOB-NAME  
COMMAND` command, where:

- `APP-NAME` is the app you want to execute a task against.
- `JOB-NAME` is the name for your job.
- `COMMAND` is the command you want to execute.

See the following example:

```
$ cf create-job my-app my-job "pwd"  
  
Creating job my-job for my-app with command pwd in org my-org / space my-space as user@example.com...  
Job Name App Name Command  
my-job my-app pwd  
OK
```

### Execute a Job

You can execute a job manually. This is often useful to test the configuration of a job prior to scheduling it for recurring execution.

Run `cf run-job JOB-NAME`. See the following example:

```
$ cf run-job my-job  
  
Enqueuing job my-job for app my-app in org my-org / space my-space as user@example.com...  
OK
```

### Schedule a Job

You can schedule a job to execute at any time using a schedule expression. Scheduler for PCF requires Cron expressions in the `MIN HOUR DAY-OF-MONTH MONTH DAY-OF WEEK` format.

For example, to execute a job at noon every day, run the following command:

```
$ cf schedule-job my-job "0 12 ? * * "
```

A single job can have multiple schedules. Each schedule has a GUID to distinguish it from similar schedules.

## View Jobs

You can use the cf CLI to list all jobs in a space by running `cf jobs`. See the following example:

```
$ cf jobs  
Listing jobs for org my-org / space my-space as user@example.com...  
Job Name  App Name  Command  
my-job    my-app    pwd  
OK
```

## View Schedules for Jobs

You can review schedules for all jobs in a space by running `cf job-schedules`. See the following example:

```
$ cf job-schedules  
Getting scheduled jobs for org my-org / space my-space as user@example.com...  
App Name: my-app  
my-job    pwd  2b69e0c2-9664-46bb-4817-54afcedbb65d  0 12 ? * *  
OK
```

## View Job History

You can review job history by running `cf job-history JOB-NAME`. See the following example:

```
$ cf job-history my-job  
Getting scheduled job history for my-job in org my-org / space my-space as user@example.com...  
1 - 1 of 1 Total Results  
Execution GUID      Execution State  Scheduled Time      Execution Start Time      Execution End Time      Exit Message  
8a7e808a5b883a25015b89b4a12c0001  SUCCEEDED   Mon, 10 Apr 2017 13:00:00 UTC  Mon, 10 Apr 2017 13:00:00 UTC  Mon, 10 Apr 2017 13:00:01 UTC  202 - Cloud Controller Ac
```

## View Logs

You can view logs for jobs by running `cf logs APP-NAME --recent`. See the following example:

```
$ cf logs my-app --recent  
Connected, dumping recent logs for app my-app in org my-org / space my-space as user@example.com...  
[...]  
2017-04-19T23:04:13.79-0600 [APP/TASK/cc6fab7f-32a9-4404-4574-b0c430a96cd9 |- 0d30f4f0-11a4-4d6a-7e77-5e1cdc1aa5ec/0]OUT Creating container  
2017-04-19T23:04:14.01-0600 [APP/TASK/cc6fab7f-32a9-4404-4574-b0c430a96cd9 |- 0d30f4f0-11a4-4d6a-7e77-5e1cdc1aa5ec/0]OUT Successfully created container  
2017-04-19T23:04:14.22-0600 [APP/TASK/cc6fab7f-32a9-4404-4574-b0c430a96cd9 |- 0d30f4f0-11a4-4d6a-7e77-5e1cdc1aa5ec/0]OUT bin  
2017-04-19T23:04:14.22-0600 [APP/TASK/cc6fab7f-32a9-4404-4574-b0c430a96cd9 |- 0d30f4f0-11a4-4d6a-7e77-5e1cdc1aa5ec/0]OUT db  
2017-04-19T23:04:14.23-0600 [APP/TASK/cc6fab7f-32a9-4404-4574-b0c430a96cd9 |- 0d30f4f0-11a4-4d6a-7e77-5e1cdc1aa5ec/0]OUT Exit status 0  
2017-04-19T23:04:14.24-0600 [APP/TASK/cc6fab7f-32a9-4404-4574-b0c430a96cd9 |- 0d30f4f0-11a4-4d6a-7e77-5e1cdc1aa5ec/0]OUT Destroying container  
2017-04-19T23:04:14.55-0600 [APP/TASK/cc6fab7f-32a9-4404-4574-b0c430a96cd9 |- 0d30f4f0-11a4-4d6a-7e77-5e1cdc1aa5ec/0]OUT Successfully destroyed container  
[...]
```

 **Note:** Scheduler for PCF jobs are executed as [CF Tasks](#).

## Delete a Job

You can delete a job by running `cf delete-job JOB-NAME`. See the following example:

```
$ cf delete-job my-job
```

Really delete the job my-job with command pwd and all associated schedules and history?> [yN]:y  
OK

## Delete a Job Schedule

You can delete a specific schedule by running `cf delete-job-schedule SCHEDULE-GUID`, where `SCHEDULE-GUID` is the GUID found in the output of the `cf job-schedules` command. See the following example:

```
$ cf delete-job-schedule 2b69e0c2-9664-46bb-4817-54afcedbb65d  
Really delete the schedule 2b69e0c2-9664-46bb-4817-54afcedbb65d / 0 12 ? * * and all associated history?> [yN]: y  
OK
```

## Scheduling Calls

This topic provides instructions for managing outbound HTTP calls in Scheduler for Pivotal Cloud Foundry (PCF).

## Manage Calls

You can use Scheduler for PCF to schedule execution of HTTP calls to external HTTP services. See the following sections to learn more about creating, running, and scheduling calls and viewing call history.

**Note:** If you want to use the Cloud Foundry Command Line Interface (cf CLI) for managing calls, you must install the Scheduler for PCF CLI plugin on your local machine. This plugin is packaged with the Scheduler for PCF tile on [Pivotal Network](#). For more information, see the [Prerequisites](#) section of the Using Scheduler for PCF topic.

### Create a Call

You can create a call by running the `cf create-call APP-NAME CALL-NAME URL` command, where:

- `APP-NAME` is the app you want to create a call for.
- `CALL-NAME` is the name for your call.
- `URL` is the URL to execute a HTTP POST call against.

### Execute a Call

You can execute a call manually by running the `cf run-call CALL-NAME` command. This is often useful to test the configuration of a call prior to scheduling it for recurring execution.

See the following example:

```
$ cf run-call my-call
Enqueuing call my-call for app my-app in org my-org / space my-space as user@example.com...
OK
```

### Schedule a Call

You can schedule a call to execute at any time using a schedule expression. Scheduler for PCF requires Cron expressions in the `MIN HOUR DAY-OF-MONTH MONTH DAY-OF-WEEK` format.

For example, to execute a call at noon every day, run the following command:

```
$ cf schedule-call my-call "0 12 ? * *
```

A single call can have multiple schedules. Each schedule has a GUID to distinguish it from similar schedules.

### View Calls

You can use the cf CLI to list all calls in a space by running `cf calls`. See the following example:

```
$ cf calls
```

```
Listing calls for org my-org / space my-space as user@example.com...
Call Name  App Name   URL
my-call    my-app     https://example.com
OK
```

## View Schedules for Calls

You can review schedules for all calls in a space by running `cf call-schedules`. See the following example:

```
$ cf call-schedules
```

```
Getting scheduled calls for org my-org / space my-space as user@example.com...
App Name: my-app
my-call  2b69e0c2-9664-46bb-4817-54afcedbb65d  0 12 ? * *
OK
```

## View Call History

You can review call history by running `cf call-history CALL-NAME`. See the following example:

```
$ cf call-history my-call
```

```
Getting scheduled call history for my-call in org my-org / space my-space as user@example.com...
1 - 1 of 1 Total Results
Execution GUID      Execution State  Scheduled Time      Execution Start Time      Execution End Time      Exit Message
d288a4ba-e0bc-48c9-969c-6ee79e380b20  SUCCEEDED      Mon, 16 Oct 2017 12:10:55 UTC  Mon, 16 Oct 2017 12:10:55 UTC  Mon, 16 Oct 2017 12:10:55 UTC  201 - Created
```

## Delete a Call

You can delete a call by running `cf delete-call CALL-NAME`. See the following example:

```
$ cf delete-call my-call
```

```
Really delete the call my-call with url https://example.com and all associated schedules and history?> [yN]: y
OK
```

## Delete a Call Schedule

You can delete a specific schedule by running `cf delete-call-schedule SCHEDULE-GUID`, where `SCHEDULE-GUID` is the GUID found in the output of the `cf call-schedules` command. See the following example:

```
$ cf delete-call-schedule 2b69e0c2-9664-46bb-4817-54afcedbb65d
```

```
Really delete the schedule 2b69e0c2-9664-46bb-4817-54afcedbb65d / 0 12 ? * * and all associated history?> [yN]: y
OK
```