

Redis for Pivotal Cloud Foundry Docs[®]

Version 1.5

User's Guide

© 2018 Pivotal Software, Inc.

Table of Contents

Table of Contents	2
Redis for PCF	3
Redis for PCF	4
Redis for PCF Release Notes	9
Installing Redis for PCF	15
Redis for PCF Smoke Tests	17
Using Redis for PCF	18
Maintaining Redis for PCF	21
Creating Backups of Redis Instances	21
Recovering Redis Instances	25
Upgrading Redis for PCF	27
Upgrades	27
Release policy	27

Redis for PCF

This is documentation for the [Redis service for PCF](#).

Redis for PCF

Redis is an easy to use, high speed key-value store that can be used as a database, cache and message broker. It supports a range of data structures including strings, lists, hashes, sets, bitmaps, hyperloglogs and geospatial indexes. Persistence is provided via RDB and AOF, with backups to an S3 bucket every 24 hours.

Product snapshot

Current [Redis](#) for PCF (PCF) Details

- Version:** 1.5.33
- Release Date:** 1 June 2017
- Software component version:** Redis OSS 3.0.7
- Compatible Ops Manager Version(s):** 1.8.x, 1.7.x, 1.6.x
- Compatible Elastic Runtime Version(s):** 1.8.x, 1.7.x, 1.6.x
- vSphere support?** Yes
- GCP support?** Yes
- AWS support?** Yes
- OpenStack support?** Yes
- IPsec support?** Yes

Upgrading to the Latest Version

Consider the following compatibility information before upgrading Redis for PCF.

For more information, refer to the full [Product Compatibility Matrix](#).

Ops Manager Version	Supported Upgrades from Imported Redis Installation	
	From	To
1.5.x, 1.6.x	1.40 - 1.4.3	1.4.4 - latest 1.4.x version
		1.5.0 - 1.5.7
	1.4.4 - latest 1.4.x version	Next 1.4.x version - latest 1.4.x version
		1.5.0 - latest 1.5.x version
1.5.0 - latest 1.5.x version	Next 1.5.x version - latest 1.5.x version	
1.7.x	1.5.0 and later versions	1.5.1 and later versions
1.8.x	1.5.17 and later versions	1.6.0 and later versions
	1.6.0 and later versions	1.6.1 and later versions

Please provide any bugs, feature requests, or questions to [the Pivotal Cloud Foundry Feedback list](#).

Further Reading

- [Official Redis Documentation](#)

Redis for PCF

Redis Configuration

Redis is configured with a maxmemory-policy of no-eviction. This policy means that the once memory is full, the service will not evict any keys and no write operations will be possible until memory becomes available. Persistence is configured for both `RDB` and `AOF`. The default maximum number of connections, maxclients, is maintained at 10000. Replication and event notification are not configured.

Service Plan Descriptions

- [Shared-VM Plan](#)
- [Dedicated-VM Plan](#)

Shared-VM Plan

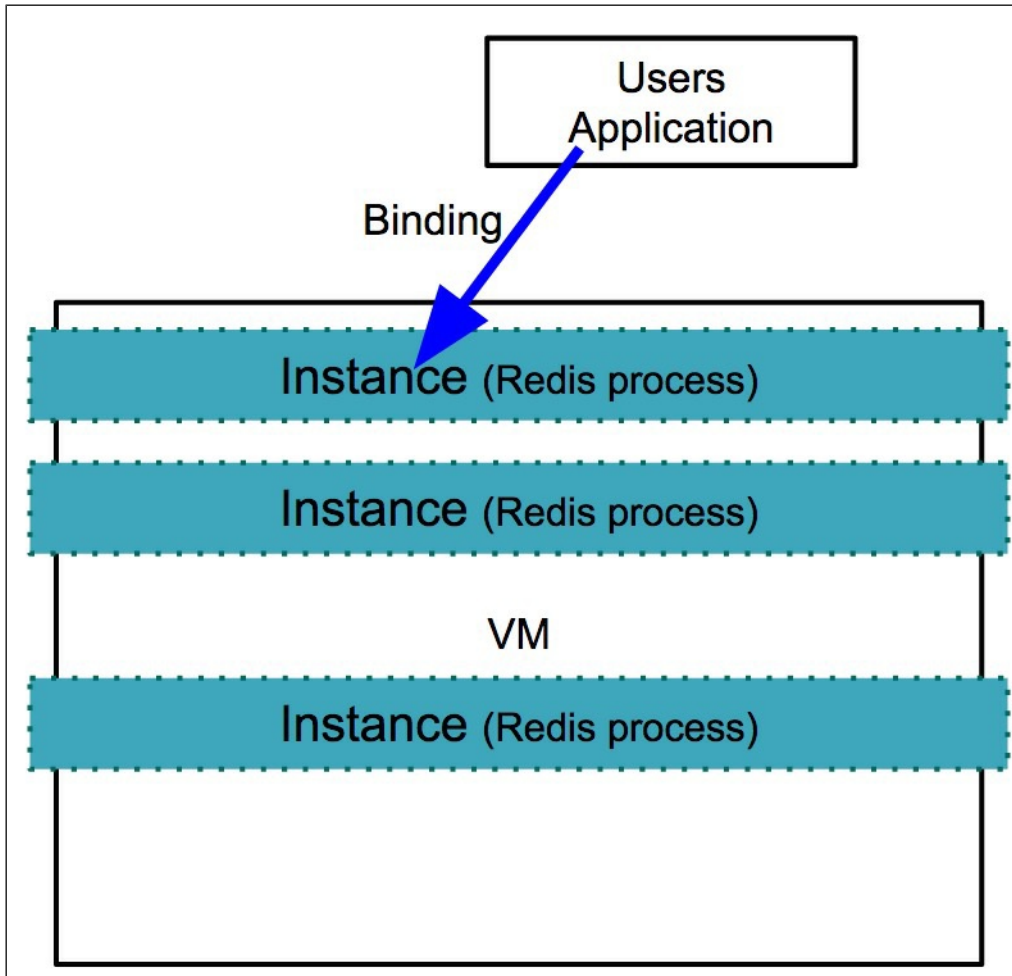
An instance of this plan provisions a single Redis process on a single **shared VM** that is suitable for workloads which do not require dedicated hardware.

Data persistence is enabled through the use of `RDB` and `AOF`.

Operator Notes

- This plan deploys a shared VM and a single service broker VM.
- This plan can be disabled by setting the `Max instances limit` on the `Shared-VM Plan` tab in OpsManager to be `0`.
- The maximum number of instances can be increased from the default 5 to a value of your choosing. If you increase the number of instances that can be run on this single VM, you should consider increasing the resources allocated to the VM. In particular RAM and CPU. You can overcommit to a certain extent, but may start to see performance degradations.
- You can also increase the maximum amount of RAM allocated to each Redis process (service instance) that is running on this VM
- If you decrease the service instance limit, any instances that are running where the count is now greater than the limit are not terminated. They are left to be removed naturally, until the total count drops below the new limit you cannot create any new instances. For example if you had a limit of 10 and all were used and reduced this to 8, the two instances will be left running until you terminate them yourself.

Diagram



Known Limitations

Limitations with the current `shared-vm` plan include:

- It cannot be scaled beyond a single virtual machine.
- The following commands are disabled: `CONFIG`, `MONITOR`, `SAVE`, `BGSAVE`, `SHUTDOWN`, `BGREWRITEAOF`, `SLAVEOF`, `DEBUG`, and `SYNC`.
- Constraining CPU and/or disk usage is not supported.
- The Shared-VM plan does not manage 'noisy neighbor' problems so it is not recommended for production apps.

Dedicated-VM Plan

An instance of this plan, provisions a single Redis process, on a single **dedicated** VM, which is suitable for workloads that require isolation or dedicated hardware.

The following commands are enabled:

- `MONITOR`
- `SAVE`
- `BGSAVE`
- `BGREWRITEAOF`

Data persistence is enabled through the use of `RDB` and `AOF`.

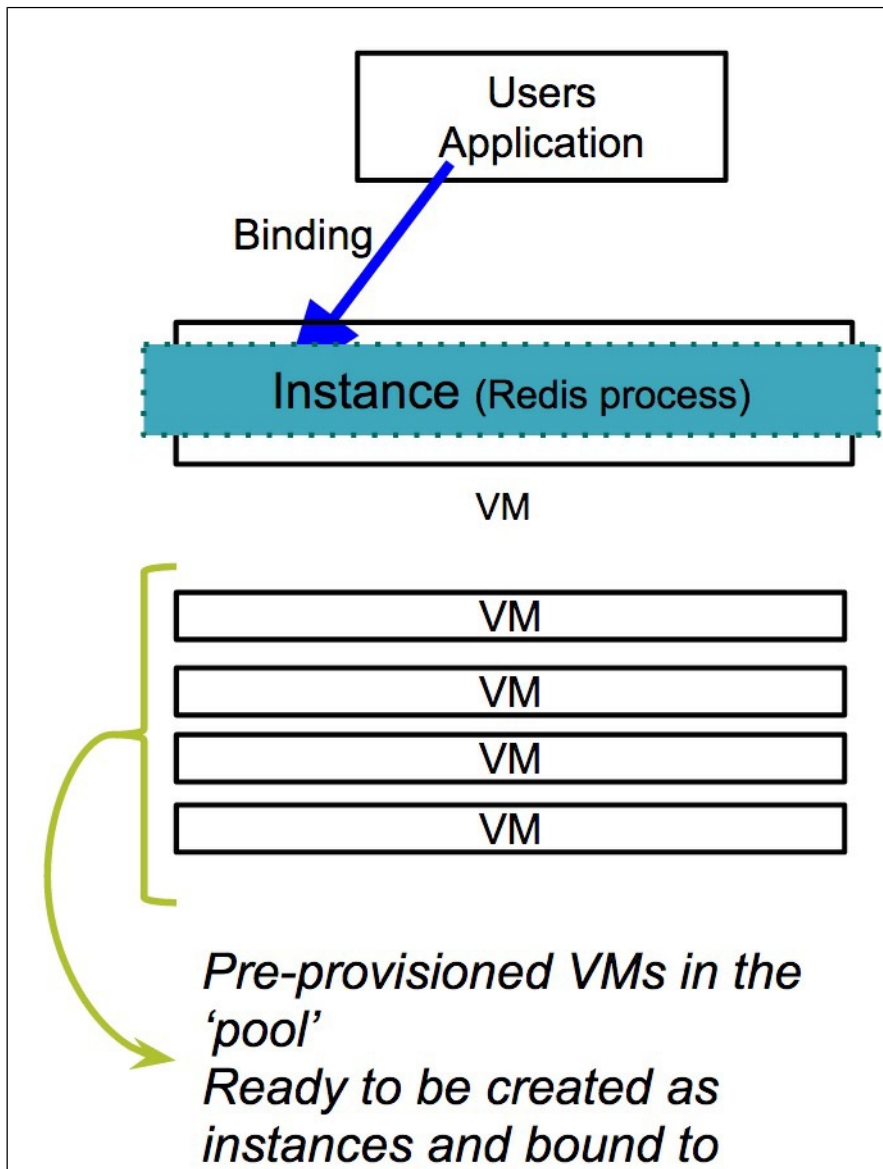
The `maxmemory` value for the Redis process is set to be 45% of the RAM for that instance.

The persistent disk should be set to be at least the size of the RAM available to the VM or greater, in order to account for the final and temporary RDB file generated by the Redis background save.

Operator Notes

- This plan deploys several dedicated Redis VMs and a single service broker VM.
- These instances are pre-provisioned during the deployment of the tile from OpsManager into a **pool**. The VMs are provisioned and configured with a Redis process ready to be used when an instance of the `dedicated-vm` plan is requested.
- A default deployment will provision `5 instances` of the `dedicated-vm` plan into the **pool**. This number can be increased on the `Resource Config` tab in Ops Manager, either in the initial deployment, or subsequently thereafter. The number of VMs **cannot** be decreased once deployed.
- When a user provisions an instance, it is marked as in use and taken out of the **pool**.
- When a user deprovisions an instance, the instance is cleansed of any data and configuration to restore it to a fresh state and placed back into the pool, ready to be used again.
- This plan can be disabled by setting the number of instances of the `Dedicated node` job in Ops Manager to `0`.

Diagram



Known Limitations

Limitations with the current `dedicated-vm` plan include:

- No ability to change the Redis configuration. The `CONFIG` command is disabled.
- Cannot scale down the number of VMs on the plan once deployed.

Highly Available Deployments

This plan is not intended for high availability. If your version of OpsManager supports multiple Availability Zones (AZs), instances of the `dedicated-vm` plan can be placed in multiple AZs, but they cannot be configured to act as master/slave nodes or to fail over. In addition, each plan can only have a single broker VM.

Default Resources

Resource requirements for Redis for PCF

These are the default resource and IP requirements for installing the tile

Product	Resource	Instances	CPU	Ram	Ephemeral	Persistent	Static IP	Dynamic IP
Redis	Redis Broker	1	2	3072	4096	9216	1	0
Redis	Dedicated Node	5	2	1024	4096	4096	1	0
Redis	Broker Registrar	1	1	1024	2048	0	0	1
Redis	Broker De-Registrar	1	1	1024	2048	0	0	1
Redis	Compliation	2	2	1024	4096	0	0	1

Notes:

- The `shared-vm` plan is on the `Redis Broker` resource
- The `dedicated-vm` plan is on the `Dedicated Node` resource

Security

It is recommended that each service run in its own network.


Redis for PCF has been tested successfully with the BOSH IPsec AddOn.

The following ports and ranges are used in this service:

- Destination port 80 access to the service broker from the cloud controllers
- Destination port 6379 access to all dedicated nodes from the Diego Cell and Diego Brain network(s)
- Destination ports 32768 to 61000 on the service broker from the Diego Cell and Diego Brain network(s). This is only required for the shared service plan.
- Outbound access to your chosen blobstore, typically HTTP 80 or HTTPS 443

Application Security Groups

To allow this service to have network access you must create [Application Security Groups \(ASGs\)](#). Ensure your security group allows access to the Redis Service Broker VM and Dedicated VMs configured in your deployment. You can obtain the IP addresses for these VMs in Ops Manager under the **Resource Config** section for the Redis tile.

 **Note:** Without ASGs, this service will not be usable.

Application Container Network Connections

Application containers that use instances of the Redis service require the following outbound network connections:

Destination	Ports	Protocol	Reason
ASSIGNED_NETWORK	32768-61000	tcp	Enable application to access shared vm service instance
ASSIGNED_NETWORK	6379	tcp	Enable application to access dedicated vm service instance

Create an ASG called `redis-app-containers` with the above configuration and bind it to the appropriate space or, to give all started apps access, bind to the `default-running` ASG set and restart your apps. Example:

```
[
  {
    "protocol": "tcp",
    "destination": "ASSIGNED_NETWORK",
    "ports": "6379"
  }
]
```


Redis for PCF Release Notes

1.5.33

Release Date: 1 June 2017

Features included in this release: * Update stemcell to 3263.27 to fix security vulnerability.

1.5.32

Release Date: 25 May 2017

Features included in this release: * Update stemcell to 3263.26 to fix security vulnerability.

1.5.31

Release Date: 27 April 2017

Features included in this release: * Update stemcell to 3263.24 to fix security vulnerability.

1.5.30

Release Date: 31 March 2017

Features included in this release: * Update stemcell to 3263.22 to fix security vulnerability.

1.5.29

Release Date: 10 March 2017

Features included in this release: * Update stemcell to 3263.21 to fix security vulnerability.

1.5.28

Release Date: 27 February 2017

Features included in this release: * Update stemcell to 3263.20 to fix security vulnerability.

1.5.27

Release Date: 24 January 2017

Features included in this release: * Update stemcell to 3263.17 to fix security vulnerability.

1.5.26

Release Date: 16 December 2016

Features included in this release:

- Update stemcell to 3263.13 to fix security vulnerability.
- Update golang to 1.7.4
- Prevent an issue with dataloss when upgrading to Ops Manager 1.7. Data loss can occur in PCF Redis 1.5 service broker (shared vm instances) when upgrading from OM1.6 to OM1.7, if Ops Manager moves AZs. The persistent disk on the service broker is detached, which means the data for the shared-vm instances and the bindings for all service instances are lost. To avoid this issue, customers can either 1) when upgrading to Ops Manager 1.7, upgrade to 1.7.20 or later as OM 1.7.20 has the fix for the issue or 2) upgrade PCF Redis tile to 1.5.26 or later before upgrading from Ops Manager 1.6. If data has been lost, it can be recovered by following the steps here <https://discuss.pivotal.io/hc/en-us/articles/115000184568-Pivotal-Cloud-Foundry-Redis-service-binding-unbinding-deprovisioning-fails-after-Ops-Manager-upgrade> ↗

1.5.25

Release Date: 8 December 2016

Features included in this release:

- Upgrade to stemcell 3233.6 to address the security vulnerability Ubuntu Security Notice USN-3151-2

1.5.24

Release Date: 7 December 2016

Features included in this release:

- Upgrade to Golang 1.7.3
- Fixing a bug where stopping all the VMs via the IaaS and then starting them up again fails to bring everything back to a running state

1.5.23

Release Date: 24 October 2016

Features included in this release:

- Update stemcell to 3233.3
- Prevent spoofing of service broker URL

1.5.22

Release Date: 13 October 2016

Features included in this release:

- Update stemcell to 3233.2

1.5.21

Release Date: 5 October 2016

Features included in this release:

- Update stemcell to 3233.1

1.5.20

Release Date: 28 September 2016

Features included in this release:

- Update stemcell to 3232.21

1.5.19

Release Date: 22 September 2016

Features included in this release:

- Update cf cli to v6.21.1
-

1.5.18

Release Date: 25 August 2016

Features included in this release:

- Update to stemcell 3232.17 for USN-3064-1

1.5.17

Release Date: 1 August 2016

Features included in this release:

- Fix a bug that prevented installation on Ops Manager 1.8.

1.5.16

Release Date: 29th June 2016


Features included in this release:

- Update stemcell to 3232.12

1.5.15

Release Date: 14th June 2016

Features included in this release:


- Update stemcell to 3232.8, fix bug with Redis upgrade stalling introduced in 1.5.11 release - KB [here](#) 

1.5.14

Release Date: 8th June 2016

Features included in this release:

- Update Stemcell to 3232.6


Known issue Redis upgrade stalls. Details [here](#) 

1.5.13

Release Date: 19th May 2016

Features included in this release:

- Update Stemcell to 3232.4


Known issue Redis upgrade stalls. Details [here](#) 

1.5.12

Release Date: 4th May 2016

Features included in this release:

- Update Stemcell to 3146.11, support for Ops Manager 1.7


Known issue Redis upgrade stalls. Details [here](#) 

1.5.11

Release Date: 20th April 2016

Features included in this release:

- Updated Golang to 1.5.4, fixed a bug where Redis doesn't always restart cleanly, fixed a bug where some logs not correctly emitted to syslog

Known issue Redis upgrade stalls. Details [here](#) 

1.5.10

Release Date: 15th March 2016

Features included in this release:

- Update Stemcell to 3146.10

1.5.9

Release Date: 25th February 2016

Features included in this release:

- Update Stemcell to 3146.9

1.5.8

Release Date: 19th February 2016

Features included in this release:

- Update Stemcell to 3146.8

1.5.7

Release Date: 2nd February 2016

Features included in this release:

- Update Stemcell to 3146.6
- Updated Redis to 3.0.7

1.5.6

Release Date: 21st January 2016

Features included in this release:

- Update Stemcell to 3146.5, resolves CVE USN-2871-1

1.5.5

Release Date: 18th January 2016

Features included in this release:

- Update Stemcell to 3146.3, resolves CVE-2016-0777 and CVE-2016-0778

1.5.4

Release Date: 7th January 2016

Features included in this release:

- Update Stemcell to 3146.2, resolves CVE USN-2857-1

1.5.3

Release Date: 23rd December 2015

Features included in this release:

- Updated Redis to 3.0.6
- Updated Stemcell to 3153

1.5.2

Release Date: 4th December 2015

Features included in this release:

- Updated stemcell to 3146 to address USN-2815-1, USN-2812-1 and USN-2810-1

1.5.1

Release Date: 1st December 2015

Features included in this release:

- Updated stemcell to 3144 for monthly patch release

1.5.0

Release Date: 9th November 2015

Features included in this release:

- Automated backups to an S3 compatible Blobstore
- All in use instances across both service plans are backed up
- Validated against AWS S3
- Support added for service keys
- Smoke tests are run after a deployments
- Redis 3.0.5

Installing Redis for PCF

Installation Steps

To add Redis for PCF to Ops Manager, follow the procedure for adding Pivotal Ops Manager tiles:

1. Download the product file from [Pivotal Network](#).
2. Upload the product file to your Ops Manager installation.
3. Click **Add** next to the uploaded product description in the Available Products view to add this product to your staging area.
4. (Optional) Click the newly added tile to configure your possible service plans configure and [syslog draining](#).
5. Click **Apply Changes** to install the service.

Select the **Redis** tile from the Installation Dashboard to display the configuration page and configure the Redis service plans.

Shared-VM Plan

1. Select the **Shared-VM Plan** tab to configure the memory limit for each Redis instance and the maximum number of instances that can be created.


Shared-VM instances run on the Redis Broker.

The memory and instance limits for your Shared-VM Redis instances should depend on the total memory of your Redis broker. When configuring the maximum number of Redis service instances that can be created you need to take into account the maximum memory each redis instance could use in correlation with how much total memory the Redis broker has. We recommend you only allow up to 45% of your Redis broker's total memory to be used by all Redis instances. This is due the amount of memory required to support Redis persistence, and run Redis broker & system tasks.

See below for example cases:

Redis Broker Total Memory	Redis Instance Memory Limit	Redis Service Instance Limit
16GB	512MB	14
16GB	256MB	28
64GB	512MB	56

It is possible to configure a larger Redis Service Instance Limit, if you are confident that the majority of the deployed instances will not be using a large amount of their allocated memory, for example in development or test environments.

 **Note:** This is not supported, and could cause your server to run out of memory. If this happens your users may not be able to write any further data to any Redis instance.

2. Select the **Resource Config** tab to change the allocation of resources for the Redis Broker.
The Redis Broker server will run all of the Redis instances for your Shared-VM plan. From this screen you may increase or decrease the CPU, RAM, Ephemeral Disk & Persistent Disk made available, as required.
3. Click the **Save** button.

Dedicated-VM Plan

1. Select the **Resource Config** tab to change the allocation of resources for the Dedicated Node.

By default, 5 dedicated nodes will be created, each capable of running one Redis instance. You can increase or decrease the number of dedicated nodes, the size of the Persistent and Ephemeral Disks, and the CPU and RAM, as required. Redis maxmemory is set to 45% of RAM. It is recommended the persistent disk be set to 3.5x the amount of RAM.

2. Click the **Save** button.

Configuring Syslog Output

Pivotal recommends that operators configure a syslog output.

1. Select the **Syslog** tab to setup the details of your syslog draining.
2. Add the Syslog address and Syslog port of your log management tool.
The information required for these fields will be provided by your log management tool.
3. Click the **Save** button.

Redis for PCF Smoke Tests

Redis for PCF runs a set of smoke tests during installation to confirm system health. The tests run in the org `system` and in the space `redis-smoke-tests`. The tests run as an application instance with a restrictive Application Security Group (ASG).

Smoke Test Steps

The smoke tests perform the following for each available service plan:

1. Targets the org `system` and space `redis-smoke-tests` (creating them if they do not exist)
2. Creates a restrictive security group, `redis-smoke-tests-sg`, and binds it to the space
3. Deploys an instance of the [CF Redis Example App](#) to this space
4. Creates a Redis instance and binds it to the CF Redis Example App
5. Checks that the CF Redis Example App can write to and read from the Redis instance

Security Groups

Smoke tests create a new [application security group](#) for the CF Redis Example App (`redis-smoke-tests-sg`) and delete it once the tests finish. This security group has the following rules:

```
[
  {
    "protocol": "tcp",
    "destination": "<dedicated node IP addresses>",
    "ports": "6379" // Redis dedicated node port
  },
  {
    "protocol": "tcp",
    "destination": "<broker IP address>",
    "ports": "32768-61000" // Ephemeral port range (assigned to shared-vm instances)
  }
]
```

This allows outbound traffic from the test app to the Redis shared VM and dedicated VM nodes.

Troubleshooting

If errors occur while the smoke tests run, they will be summarised at the end of the errand log output. Detailed logs can be found where the failure occurs. Some common failures are listed below.

Error	<code>Failed to target Cloud Foundry</code>
Cause	Your Pivotal Cloud Foundry is unresponsive
Solution	Examine the detailed error message in the logs and check the PCF Troubleshooting Guide for advice

Error	<code>Failed to bind Redis service instance to test app</code>
Cause	Your deployment's broker has not been registered with Pivotal Cloud Foundry
Solution	Examine the broker-registrar installation step output and troubleshoot any problems.

When encountering an error when running smoke tests, it can be helpful to search the log for other instances of the error summary printed at the end of the tests, e.g. `Failed to target Cloud Foundry`. Lookout for `TIP: ...` in the logs next to any error output for further troubleshooting hints.

Using Redis for PCF

Redis for PCF can be used both via Pivotal Apps Manager and the CLI, both methods are outlined below. An example application has also been created to help application developers get started with Redis for PCF, and can be [downloaded here](#).

Creating a Redis Service Plan

The following procedure describes how to create a Redis service instance in the Pivotal Cloud Foundry Elastic Runtime environment.

Available Plans

Before creating a Redis instance, it is worth being aware of the two available plans:

Plan Name	Suitable for	Tenancy Model per Instance	Highly Available	Backup Functionality
Shared-VM	Lighter workloads that do not require dedicated resources	Shared VM	No	Yes
Dedicated-VM	Increased workloads that require dedicated resources	Dedicated VM	No	Yes

Using Pivotal Apps Manager

1. From within Pivotal Apps Manager, select Marketplace from the left navigation menu under Spaces. The Services Marketplace displays.
2. Select **Redis** from the displayed tiles and click to view the [available plans][#plans].
3. Click on the appropriate **Select this plan** button to select the required **Redis Service Plan**.
4. In the Instance Name field, enter a name that will identify this specific Redis service instance.
5. From the Add to Space drop-down list, select the space where you or other users will deploy the applications that will bind to the service.
6. Click the **Add** button.

Using the CLI

1. Run the following command to view the available service plans.

```
$ cf marketplace
```

This should produce the output:

```
Getting services from marketplace in org system / space apps-manager as admin...
OK

service  plans          description
p-redis  shared-vm, dedicated-vm  Redis service to provide a key-value store

TIP: Use 'cf marketplace -s SERVICE' to view descriptions of individual plans of a given service.
```

2. Type the following command to create the service plan:

```
$ cf create-service p-redis <service-plan-name> <service-instance-name>
```

The service-plan-name is as seen in the Services marketplace – in this example, “shared-vm” – and the service-instance-name is a descriptive name that you want to use for the service.

For example

```
$ cf create-service p-redis shared-vm redis
```

Binding an Application to the Redis Service

The following procedures describe how to bind a Redis service instance to your Pivotal Cloud Foundry application. This can be done via the Pivotal Apps Manager or Using the Pivotal Cloud Foundry CLI.

Using Pivotal Apps Manager

1. Select the application that you wish to bind to the service. A page displays showing the already bound services and instances for this application.
2. Click Bind. A list of available services displays.
3. Click the Bind button for the Redis service you want to bind to this application.
4. Using the Pivotal Cloud Foundry CLI, start or restage your application.

```
$ cf restage <application-name>
```

Using the CLI

1. Run the following command to view running service instances.

```
$ cf services
```

This should produce the output:

```
Getting services in org system / space apps-manager as admin...
OK

name          service  plan  bound apps  last operation
my-redis-instance  p-redis  shared-vm  create succeeded
```

2. Run the following command to bind the application to the service instance.

```
$ cf bind-service <application-name> <service-instance-name>
```

For example:

```
$ cf bind-service my-application redis
```

3. Restage your application.

```
$ cf restage <application-name>
```

Deleting a Redis Instance

When you delete a Redis service instance, all applications that are bound to that service are automatically unbound and any data in the service instance is cleared.

Using Pivotal Apps Manager

1. Locate the row under Services that contains the service instance you want to delete and click Delete.
2. If you had applications that were bound to this service, you may need to restage or re-push your application for the application changes to take effect.

```
$ cf restage <application-name>
```

Using the CLI

1. Run the following command.

```
$ cf delete-service <service-instance-name>
```

The service-instance-name is that of the service instance that you would like to delete. Enter 'y' when prompted.

For example:

```
$ cf delete-service my-redis-instance
Really delete the service my-redis-instance?> y
Deleting service my-redis-instance in org system / space apps-manager as admin...
OK
```

2. If you had applications that were bound to this service, you may need to restage or re-push your application for the application changes to take effect.

```
$ cf restage <application-name>
```

Maintaining Redis for PCF

Creating Backups of Redis Instances

You can configure backups to be run for each instance, across both service plans.

The key features are:

- Runs at midnight system time every day (not configurable)
- Every instance is backed up, across both service plans
- You can configure an S3 compatible blobstore as your destination
- Data from Redis is flushed to disk, before the backup is started by running a `BGSAVE` on each instance
- Currently certified and tested against AWS S3 only

Configuration

To enable backups to be taken, you need to configure the mandatory options in the `Redis for PCF` tile in OpsManager.

Click on the tile in OpsManager, followed by the `Backups` link on the left hand menu.

Configure S3-compatible blob store for Redis backups

Access Key ID

Secret Access Key

Endpoint URL

Bucket Name

Path

Redis BGSAVE timeout *

Access Key ID

This is your Access Key for your Blobstore

Required? No - this is optional, dependent upon whether it is required by your blobstore

Secret Access Key

This is your Secret associated with your access key id

Required? No - this is optional, dependent upon whether it is required by your blobstore

Endpoint URL

This is the endpoint for your blobstore e.g. `https://s3.amazonaws.com`

Required? Yes - If you want to enable backups to be run, you must populate this field.

Bucket Name

Name of the bucket inside your blobstore you wish the files to be stored in.

Required? Yes - If you want to enable backups to be run, you must populate this field.

Path

Path inside the bucket

Required? No - this is optional, it will default to the root if not specified

Redis BGSAVE Timeout

This is the amount of time that the backup process will wait for the BGSAVE command to complete on your instance, before transferring the RDB file to your configured blobstore.

You can increase this if required for your setup.

Required? - Yes, this defaults to 600 seconds.

AWS IAM Policy

The minimum set of policies required in order to upload the backup files are:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::<bucket-name>",
        "arn:aws:s3:::<bucket-name>/*"
      ]
    }
  ]
}
```

Make sure to replace `<bucket-name>` with your correct value.

Manual Backups

It is possible to create a backup of an instance manually by following these steps:

- [Follow these steps](#) to log into your Ops Manager installation and target the Redis tile deployment.

- Identify the VM which holds your instance by running `bosh vms`.
 - For the `shared-vm` plan this will be the job name containing `cf-redis-broker`.
 - For the `dedicated-vm` plan this will be the job name containing `dedicated-node`.
 - You can identify the exact node for your `dedicated-vm` service instance by comparing the IP Address from your application bindings.

An example output from `bosh vms`:

```
Deployment 'p--redis-9dacfffa493b5e5a386'
Director task 129
Task 129 done

-----+-----+-----+-----+
| Job/index                               | State | Resource Pool                               | IPs |
-----+-----+-----+-----+
| cf-redis-broker-partition-default_az_guid/0 | running | cf-redis-broker-partition-default_az_guid | 10.0.0.58 |
| dedicated-node-partition-default_az_guid/0   | running | dedicated-node-partition-default_az_guid | 10.0.0.59 |
-----+-----+-----+-----+
```

- Target your redis deployment with `bosh deployment`.
- `bosh ssh` into your desired node.

Persistence is enabled on these plans through the use of `RDB` files, using the following Redis config rules: `save 900 1 save 300 10 save 60 10000`

Shared-VM Plan

You can either take the latest RDB file held on disk, which is generated by the above the rules, or trigger a recent update by using the `redis-cli` to trigger a `BGSAVE`. Credentials to log into the `redis-cli` can be obtained from `VCAP_SERVICES` for your bound application.

The `redis-cli` is located in `/var/vcap/packages/redis/bin/redis-cli`.

On this plan, the `BGSAVE` command is aliased to a random string. This can be obtained from Ops Manager in the credentials tab.

Steps to Backup

- `bosh ssh` into your desired node. See the above section to identify the correct VM.
- Change to Root using `sudo -i`.
- Copy the contents of the `/var/vcap/store/cf-redis-broker` directory to a zip or tar file.
- Backup the folder / compressed file to your chosen location.

The `/var/vcap/store/cf-redis-broker` has sub-directories for each instance created of this plan. The backup file for each instance is called `dump.rdb`.

For example, here are two instances:

```
root@66358f3e-3428-46df-9bb3-9acc7770b188:/var/vcap/store/cf-redis-broker# find -type f | xargs ls -l ./redis-data/3124f373-e9e2-44e1-ad12-a8865d8978b0/db/dump.rdb ./redis-data/3124f373-e9e2-44e1-ad12-a8865d8978b0/redis.conf ./redis-data/3124f373-e9e2-44e1-ad12-a8865d8978b0/redis-server.pid ./redis-data/62333bf9-f023-4566-b233-6686f26b8f4d/db/dump.rdb ./redis-data/62333bf9-f023-4566-b233-6686f26b8f4d/redis.conf ./redis-data/62333bf9-f023-4566-b233-6686f26b8f4d/redis-server.pid ./statefile.json
```

Dedicated-VM Plan

You can either take the latest RDB file on disk, as generated by the above rules, or trigger a more recent RDB file by executing the `BGSAVE` command using the `redis-cli`. Credentials can be obtained from the `VCAP_SERVICES` from your bound application. The `redis-cli` can be found in `/var/vcap/packages/redis/bin/redis-cli`.

Steps to Backup

- `bosh ssh` into your desired node. See the above section to identify the correct VM.
- Change to Root using `sudo -i`.
- Copy the contents of the `/var/vcap/store/redis` directory to a zip or tar file.
- Backup the folder / compressed file to your chosen location.

The backup file will be named `dump.rdb`.

Restore Redis Instance from a Backup

To a Local System

You can choose to restore the RDB file to a local Redis instance.

The steps to do this depend on your configuration and setup. Refer to the [Redis documentation](#) for more details.

To Pivotal Cloud Foundry

You can also restore your backup file to another instance of the `Redis for PCF` tile.

The below steps are manual.

Before restoring your RDB file you must have these prerequisites:

- Same resource configuration as the instance from which you backed up.
 - The persistent disk should be increased to be `3.5 x size of the RDB file` if it is not already so. This allows space for the temporary files used during the restore process
1. Create a new instance of the plan that you wish to restore to.
 2. Identify the VM which the instance of your plan is located on by following the steps from the `Manual Backups` section above.
 3. `bosh ssh` into the identified VM.

Dedicated-VM Plan

1. Run `monit stop all`
2. Wait for monit services to enter the `not monitored` state, you can watch this with `watch monit summary`
3. Restore your Redis backup file to `/var/vcap/store/redis/dump.rdb` and correct the owner and permissions with `chown vcap:vcap /var/vcap/store/redis/dump.rdb && chmod 660 /var/vcap/store/redis/dump.rdb`
4. Edit the template Redis config file with `vim $(find /var/vcap/data/jobs/ -name redis.conf)` and change `appendonly` to `no`
5. run `rm -f /var/vcap/store/redis/appendonly.aof`
6. Run `monit start all`
7. Wait for monit services to enter the `running` state, you can watch this with `watch monit summary`
8. Run `/var/vcap/packages/redis/bin/redis-cli -a {instance_password} BGREWRITEAOF`
9. Run `watch "/var/vcap/packages/redis/bin/redis-cli -a {instance_password} INFO | grep aof_rewrite_in_progress" until aof_rewrite_in_progress is 0`
10. Run `monit stop all`
11. Wait for monit services to enter the `not monitored` state, you can watch this with `watch monit summary`
12. Edit the template Redis config file with `vim $(find /var/vcap/data/jobs/ -name redis.conf)` and change `appendonly` to `yes`
13. Run `monit start all`

Shared-VM Plan

1. Run `monit stop all && pkill redis-server`
2. Wait for monit services to enter the `not monitored` state, you can watch this with `watch monit summary`
3. Confirm no running instances of `redis-server` with `ps aux | grep redis-server`
4. Restore your Redis backup file to `/var/vcap/store/cf-redis-broker/redis-data/{instance_id}/db/dump.rdb` and correct the owner and permissions with


```
chown vcap:vcap /var/vcap/store/cf-redis-broker/redis-data/{instance_id}/db/dump.rdb && chmod 660 /var/vcap/store/cf-redis-broker/redis-data/{instance_id}/db/dump.rdb
```

 (your instance id can be found using the CF API - `cf service {instance name} --guid`)
5. Edit the template Redis config file with `vim $(find /var/vcap/data/jobs/ -name redis.conf)` and change `appendonly` to `no` and `BGREWRITEAOF` alias to `BGREWRITEAOFTEMP`
6. Run `rm -f /var/vcap/store/cf-redis-broker/redis-data/{instance_id}/db/appendonly.aof`
7. Run `monit start all`
8. Wait for monit services to enter the `running` state, you can watch this with `watch monit summary`
9. Run `/var/vcap/packages/redis/bin/redis-cli -a {instance_password} BGREWRITEAOFTEMP`
10. Run


```
watch "/var/vcap/packages/redis/bin/redis-cli -a {instance_password} -p {instance_port} INFO | grep aof_rewrite_in_progress"
until aof_rewrite_in_progress is 0
```
11. Run `monit stop all && pkill redis-server`
12. Wait for monit services to enter the `not monitored` state, you can watch this with `watch monit summary`
13. Confirm no running instances of `redis-server` with `ps aux | grep redis-server`
14. Edit the template Redis config file with `vim $(find /var/vcap/data/jobs/ -name redis.conf)` and change `appendonly` to `yes` and `BGREWRITEAOF` alias to `" "`
15. Run `monit start all`

Recovering Redis Instances

In the event of a recovery of Cloud Foundry, it is possible to recover bound Redis instances to healthy states that are in sync with Cloud Foundry. There are a few caveats to being able to recover previous instance state fully that depend on your plan.

Shared-VM Plan Caveats

- You need a backed up RDB Redis dump file - this would be stored in your S3 buckets if you have backups configured
- You need a backed up `/var/vcap/store/cf-redis-broker/redis-data` directory from the service broker node (you do not need to backup and `*.aof` or `*.rdb` files from subdirectories if you have backups configured)

Dedicated-VM Plan Caveats

- You need a backed up RDB Redis dump file - this would be stored in your S3 buckets if you have backups configured
- You need a backed up `/var/vcap/store/redis/statefile.json` from the service broker node

Note

This procedure assumes that a recovery of service information and service keys assigned to instances are restored with a restore of Cloud Foundry.

Recovery Procedure

After redeploying Redis, take the following steps.

Shared-VM Plan

1. `bosh ssh` into the service broker node of your Redis deployment
2. Run `monit stop all && pkill redis-server`
3. Wait for monit services to enter the `not monitored` state, you can watch this with `watch monit summary`
4. Confirm no running instances of `redis-server` with `ps aux | grep redis-server`
5. Copy the backed up `redis-data` directory into `/var/vcap/store/cf-redis-broker`
6. Follow the instructions [here](#) for your plan, skipping the first four steps described here, for restoring your backed up Redis data
7. Your Redis instance is now recovered

Dedicated-VM Plan

1. `bosh ssh` into the service broker node of your Redis deployment
2. Run `monit stop all`
3. Wait for monit services to enter the `not monitored` state, you can watch this with `watch monit summary`
4. Copy the backed up `/var/vcap/store/cf-redis-broker/statefile.json` and ensure ownership and permissions are correct with `chown vcap:vcap /var/vcap/store/redis/dump.rdb && chmod 660 /var/vcap/store/redis/dump.rdb`
5. Follow the instructions [here](#) for your plan, skipping the first three steps described here, for restoring your backed up Redis data
6. Your Redis instance is now recovered

Upgrading Redis for PCF

Upgrades

This product enables a seamless upgrade experience between versions of the product that is deployed through Ops Manager.

The upgrade paths are detailed [here](#) for each released version.

To upgrade the product:

- The Operator should download the latest version of the product from [Pivotal Network](#)
- Upload the new .pivotal file to Ops Manager
- Upload the stemcell associated with the update (*if required*)
- Update any new mandatory configuration parameters (*if required*)
- Press “Apply changes” and the rest of the process is automated

During the upgrade deployment each Redis instance will experience a small period of downtime as each Redis instance is updated with the new software components. This downtime is because the Redis instances are single VMs operating in a non HA setup. The length of the downtime depends on whether there is a stemcell update to replace the operating system image or whether the existing VM can simply have the redis software updated. Stemcells updates incur additional downtime while the IaaS creates the new VM while updates without a stemcell update are faster.

Ops Manager ensures the instances are updated with the new packages and any configuration changes are applied automatically.

Upgrading to a newer version of the product does not cause any loss of data or configuration. This is explicitly tested for during our build and test process for a new release of the product.

Release policy

When a new version of Redis is released we aim to release a new version of the product containing this soon after.

Where there is a new version of Redis or another dependent software component such as the stemcell released due to a critical CVE, Pivotal’s goal is to release a new version of the product within 48 hours.