

Session State Caching


Powered by Pivotal GemFire® 1.2

© Copyright Pivotal Software Inc, 2013-2018

Table of Contents

Table of Contents	2
Session State Caching Powered by GemFire	3
Release Notes for Session State Caching Powered by GemFire	5
Overview	14
Installing Session State Caching Powered by GemFire	17
Prerequisites	17
Limitations	17
Support for HTTPS Traffic Only	18
Service Configuration Defaults	18
Installation Steps	18
Creating a Session State Caching Powered by GemFire Service Plan	19
Using Session State Caching Powered by GemFire on Pivotal Cloud Foundry®	21
Troubleshooting	39
Uninstalling Session State Caching Powered by GemFire	40
Prerequisites	40
Uninstalling	40

Session State Caching Powered by GemFire

 **NOTE** This product has been discontinued.

This documentation describes how to install, configure and use Session State Caching Powered by GemFire.

Product Snapshot

Current Session State Caching Powered by GemFire for [Pivotal Cloud Foundry®](#)  (PCF) Details

Version 1.2.0 Snapshot

Version: 1.2.0
Release Date: June 2 2016
Software component version: GemFire 8.2.1, Java 8 (JDK/JRE 8u60)
Compatible Ops Manager Version(s): 1.6
Compatible Elastic Runtime Version(s): 1.6
Compatible Java Buildpack Version(s): 3.6.x
vSphere support? Yes
AWS support? Yes

Version 1.1.0.6 Snapshot

Version: 1.1.0.6
Release Date: January 28 2016
Software component version: GemFire 8.2, Java 8 (JDK/JRE 8u60)
Compatible Ops Manager Version(s): 1.4.x, 1.5.x
Compatible Elastic Runtime Version(s): 1.4.x, 1.5.x
Compatible Java Buildpack Version(s): 3.6.x
vSphere support? Yes
AWS support? Yes


Upgrading to the Latest Version

See the **Upgrades From** section of **Release Details** for a list of prior versions that can be upgraded to the release. Upgrades from Alpha or Beta versions of the software are not supported.

Documentation Index

The documentation contains the following topics:

- [Release Notes](#)
- [Overview](#)
- [Installing Session State Caching Powered by GemFire](#)
- [Using Session State Caching Powered by GemFire](#)
- [Troubleshooting](#)

For more information on Pivotal GemFire, see the [Pivotal GemFire Documentation](#) .

For more information on [Pivotal Cloud Foundry®](#)  (PCF), see the [Pivotal Cloud Foundry® Documentation](#) .

Release Notes for Session State Caching Powered by GemFire

Overview

Session State Caching Powered by GemFire supports the deployment of Pivotal GemFire to [Pivotal Cloud Foundry®](#) (PCF) for the purposes of HTTP session replication, similar to what is provided by the [Pivotal GemFire HTTP Session Management Modules](#) in the standalone product.

When you deploy the Session State Caching service, Pivotal Cloud Foundry® automatically configures GemFire and creates regions for HTTP session replication.

Note that the releases below are listed chronologically, starting with the most recent.

Feedback

Please provide any bugs, feature requests, or questions either to Pivotal Customer Support at <https://support.pivotal.io> or email pivotal-cf-feedback@pivotal.io.

1.2.0.0

Release Date: 2nd June 2016

New in this Release

- The packaged GemFire is upgraded from 8.2.0.5 to 8.2.1.
- The `enable-network-partition-detection` property is now enabled to prevent against data corruption in split-brain scenarios.
- Cluster stability has been improved during restarts and during service instance deletions.
- Default timeout of `cf restart-gemfire` has been increased from 120 seconds to 900 seconds to accommodate slower-than-expected startup of servers and locators.
- When running on Azure, GemFire servers will have [redundancy zones](#) tied directly to Azure's [fault domains](#) so that redundancy for partitioned regions is satisfied across multiple virtual server racks.
- The `cf export-gemfire` command now gives feedback about missing arguments, making it easier to export your GemFire logs, cluster configs and properties.

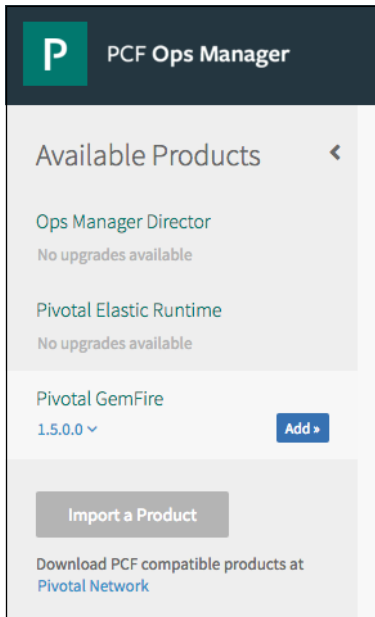
Upgrade Warning: Downtime Required

Upgrading to 1.2.0.0 will require downtime for your Session State Caching Powered by GemFire deployment.

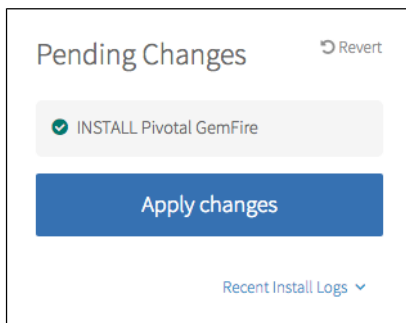
Since the `enable-network-partition-detection=true` property must be set on the entire cluster at the same time, we are unable to do a rolling upgrade. Previously deployed apps and service instances will continue to function normally after the upgrade, though Session State Caching Powered by GemFire will be unavailable during the upgrade.

For Ops Manager-deployed tiles:

1. Upload the 1.2.0.0 *Session State Caching Powered by GemFire* tile to Ops Manager.
2. Under *Available Products* > *Session State Caching (GemFire)*, click **Add**.



3. `bosh target` your Ops Manager-deployed bosh director.
4. `bosh stop` your previously deployed Session State Caching Powered by GemFire tile. This will stop the Session State Caching Powered by GemFire service broker and all service instances.
5. In Ops Manager, click **Apply Changes**.



For bosh-deployed releases:

1. `bosh upload` the releases contained in this tile
2. `bosh stop` your existing GemFire deployment
3. `bosh deploy` the manifest contained in this tile
4. `bosh start` your new deployment

Known Issues and Limitations:

Upgrading to 1.2.0.0 will require downtime for your Session State Caching Powered by GemFire deployment as detailed in the previous section.

There is a known issue where apps fail to stage when using the Java buildpacks 3.4.x and 3.5.x. To resolve this, please update [Java buildpack to 3.6](#). In Java buildpack 3.4, GemFire was upgraded from 8.0 to 8.2. GemFire 8.1 introduced a dependency on log4j, but log4j was not added to the Java buildpack 3.4. This dependency has been included in [Java buildpack 3.6](#).

1.1.10.0

Release Date: 18th February 2016

Note: This release requires Ops Manager and Elastic Runtime 1.6.x.

New in this Release:

- Updated stemcell to 3146.8, patches CVE-2015-7547 and other Ubuntu CVEs.

Known Issues and Limitations:

See [Known Issues and Limitations](#) for release 1.1.0.0, below.

There is a known issue where apps fail to stage when using the Java buildpacks 3.4.x and 3.5.x. To resolve this, please update [Java buildpack to 3.6](#). In Java buildpack 3.4, GemFire was upgraded from 8.0 to 8.2. GemFire 8.1 introduced a dependency on log4j, but log4j was not added to the Java buildpack 3.4. This dependency has been included in [Java buildpack 3.6](#).

1.1.9.0

Release Date: 3rd February 2016

Note: This release requires Ops Manager and Elastic Runtime 1.6.x.

New in this Release:

- Updated stemcell to 3146.6, patches Ubuntu CVEs

Known Issues and Limitations:

See [Known Issues and Limitations](#) for release 1.1.0.0, below.

There is a known issue where apps fail to stage when using the Java buildpacks 3.4.x and 3.5.x. To resolve this, please update [Java buildpack to 3.6](#). In Java buildpack 3.4, GemFire was upgraded from 8.0 to 8.2. GemFire 8.1 introduced a dependency on log4j, but log4j was not added to the Java buildpack 3.4. This dependency has been included in [Java buildpack 3.6](#).

1.1.0.6

Release Date: 28th January 2016

Note: This release requires Ops Manager and Elastic Runtime 1.4.x or 1.5.x.

New in this Release:

- Updated stemcell to 3146.5, resolves CVE-2016-0728.

Known Issues and Limitations:

See [Known Issues and Limitations](#) for release 1.1.0.0, below.

There is a known issue where apps fail to stage when using the Java buildpacks 3.4.x and 3.5.x. To resolve this, please update [Java buildpack to 3.6](#). In Java buildpack 3.4, GemFire was upgraded from 8.0 to 8.2. GemFire 8.1 introduced a dependency on log4j, but log4j was not added to the Java buildpack 3.4. This dependency has been included in [Java buildpack 3.6](#).

1.1.8.0

Release Date: 21st January 2016

Note: This release requires Ops Manager and Elastic Runtime 1.6.x.

New in this Release:

- Updated stemcell to 3146.5, resolves CVE-2016-0728.

Known Issues and Limitations:

See [Known Issues and Limitations](#) for release 1.1.0.0, below.

There is a known issue where apps fail to stage when using the Java buildpacks 3.4.x and 3.5.x. To resolve this, please update [Java buildpack to 3.6](#). In Java buildpack 3.4, GemFire was upgraded from 8.0 to 8.2. GemFire 8.1 introduced a dependency on log4j, but log4j was not added to the Java buildpack 3.4. This dependency has been included in [Java buildpack 3.6](#).

1.1.0.5

Release Date: 19th January 2016

Note: This release requires Ops Manager and Elastic Runtime 1.4.x or 1.5.x.

New in this Release:

- Updated stemcell to 3146.3, resolves CVE-2016-0777 and CVE-2016-0778.

Known Issues and Limitations:

See [Known Issues and Limitations](#) for release 1.1.0.0, below.

There is a known issue where apps fail to stage when using the Java buildpacks 3.4.x and 3.5.x. To resolve this, please update [Java buildpack to 3.6](#). In Java buildpack 3.4, GemFire was upgraded from 8.0 to 8.2. GemFire 8.1 introduced a dependency on log4j, but log4j was not added to the Java buildpack 3.4. This dependency has been included in [Java buildpack 3.6](#).

1.1.7.0

Release Date: 18th January 2016

Note: This release requires Ops Manager and Elastic Runtime 1.6.x.

New in this Release:

- Updated stemcell to 3146.3, resolves CVE-2016-0777 and CVE-2016-0778.

Known Issues and Limitations:

See [Known Issues and Limitations](#) for release 1.1.0.0, below.

There is a known issue where apps fail to stage when using the Java buildpacks 3.4.x and 3.5.x. To resolve this, please update [Java buildpack to 3.6](#). In Java buildpack 3.4, GemFire was upgraded from 8.0 to 8.2. GemFire 8.1 introduced a dependency on log4j, but log4j was not added to the Java buildpack 3.4. This dependency has been included in [Java buildpack 3.6](#).

1.1.6.0

Release Date: 8th January 2016

Note: This release requires Ops Manager and Elastic Runtime 1.6.x.

New in this Release:

- Updated stemcell to 3146.2, resolves CVE USN-2857-1.

Known Issues and Limitations:

See [Known Issues and Limitations](#) for release 1.1.0.0, below.

There is a known issue where apps fail to stage when using the Java buildpacks 3.4.x and 3.5.x. To resolve this, please update [Java buildpack to 3.6](#). In Java buildpack 3.4, GemFire was upgraded from 8.0 to 8.2. GemFire 8.1 introduced a dependency on log4j, but log4j was not added to the Java buildpack 3.4. This dependency has been included in [Java buildpack 3.6](#).

1.1.0.4

Release Date: 8th January 2016

Note: This release requires Ops Manager and Elastic Runtime 1.4.x or 1.5.x.

New in this Release:

- Updated stemcell to 3146.2, resolves CVE USN-2857-1.

Known Issues and Limitations:

See [Known Issues and Limitations](#) for release 1.1.0.0, below.

There is a known issue where apps fail to stage when using the Java buildpacks 3.4.x and 3.5.x. To resolve this, please update [Java buildpack to 3.6](#). In Java buildpack 3.4, GemFire was upgraded from 8.0 to 8.2. GemFire 8.1 introduced a dependency on log4j, but log4j was not added to the Java buildpack 3.4. This dependency has been included in [Java buildpack 3.6](#).

1.1.5.0

Release Date: 3rd December 2015

Note: This release requires Ops Manager and Elastic Runtime 1.6.x.

New in this Release:

- Security update to stemcell 3146, which fixes USN-2821-1.

Known Issues and Limitations:

See [Known Issues and Limitations](#) for release 1.1.0.0, below.

There is a known issue where apps fail to stage when using the Java buildpacks 3.4.x and 3.5.x. To resolve this, please update [Java buildpack to 3.6](#). In Java buildpack 3.4, GemFire was upgraded from 8.0 to 8.2. GemFire 8.1 introduced a dependency on log4j, but log4j was not added to the Java buildpack 3.4. This dependency has been included in [Java buildpack 3.6](#).

1.1.0.3

Release Date: 3rd December 2015

Note: This release requires Ops Manager and Elastic Runtime 1.4.x or 1.5.x.

New in this Release:

- Security update to stemcell 3146, which fixes USN-2821-1.
- Support for the “Trusted Certificates” feature in Ops Manager. The certificates from “Trusted Certificates” are deployed to all GemFire nodes, and to the GemFire service broker.

Note on self-signed and internal CA signed certificates: If you are using a self-signed certificate or a certificate signed by an internal or other not known certificate authority (CA), you must add the certificate (or certificate chain) to the “Trusted Certificates” in Ops Manager.

Known Issues and Limitations:

See [Known Issues and Limitations](#) for release 1.1.0.0, below.

There is a known issue where apps fail to stage when using the Java buildpacks 3.4.x and 3.5.x. To resolve this, please update [Java buildpack to 3.6](#). In Java buildpack 3.4, GemFire was upgraded from 8.0 to 8.2. GemFire 8.1 introduced a dependency on log4j, but log4j was not added to the Java buildpack 3.4. This dependency has been included in [Java buildpack 3.6](#).

1.1.4.0

Release Date: 1st December 2015

Note: This release requires Ops Manager and Elastic Runtime 1.6.x.

New in this Release:

- Security update to stemcell 3144, which fixes USN-2815-1, USN-2812-1 and USN-2810-1.
- Support for the “Trusted Certificates” feature in Ops Manager. The certificates from “Trusted Certificates” are deployed to all GemFire nodes, and to the GemFire service broker.

Note on self-signed and internal CA signed certificates: If you are using a self-signed certificate or a certificate signed by an internal or other not known certificate authority (CA), you must add the certificate (or certificate chain) to the “Trusted Certificates” in Ops Manager.

Known Issues and Limitations:

See [Known Issues and Limitations](#) for release 1.1.0.0, below.

1.1.3.0

Release Date: 11th November 2015

Note: This release requires Ops Manager and Elastic Runtime 1.6.x.

New in this Release:

- Security update to stemcell 3130, which fixes USN-2806-1 and USN-2798-1.

Known Issues and Limitations:

See [Known Issues and Limitations](#) for release 1.1.0.0, below.

1.1.0.2

Release Date: 11th November 2015

Note: This release requires Ops Manager and Elastic Runtime 1.4.x or 1.5.x.

New in this Release:

- Security update to stemcell 3130, which fixes USN-2806-1 and USN-2798-1.

Known Issues and Limitations:

See [Known Issues and Limitations](#) for release 1.1.0.0, below.

1.1.2.0

Release Date: 3rd November 2015

Note: This release requires Ops Manager and Elastic Runtime 1.6.x.

New in this Release:

- Security update, stemcell 3112.
- Fix for the issue of Internet access being required during installation, found in version 1.1.1.0.

Known Issues and Limitations:

See [Known Issues and Limitations](#) for the previous release, 1.1.0.0, below.

1.1.0.1

Release Date: 3rd November 2015

Note: This release requires Ops Manager and Elastic Runtime 1.4.x or 1.5.x.

New in this Release:

- Security update, stemcell 3112, for Ops Manager and Elastic Runtime 1.4.x, 1.5.x

Known Issues and Limitations:

See [Known Issues and Limitations](#) for the previous release, 1.1.0.0, below.

1.1.1.0

Release Date: 26th October 2015

New in this Release:

- Support for Elastic Runtime 1.6.
- Stemcell 3100.

Known Issues and Limitations:

See [Known Issues and Limitations](#) for the previous release, 1.1.0.0, below.

1.1.0.0

Release Date: 7th October 2015

New in this Release:

- Stemcell 3094 (regular security upgrade).
- Support for HTTPS only traffic to HAProxy. This release supports disabling HTTP traffic to both UAA and HAProxy
- Upgraded Pivotal GemFire to version 8.2
- Upgraded Java to version 8, and JRE 1.8
- Drain script improvements
- Improved handling of self-signed certificates
- Fixed the issue of spaces in the name of the service plan causing the service errand “Service Offering Smoke Test” to fail.

Known Issues and Limitations:

- On AWS, only deployments to the US-East region are supported. Pivotal is working to provide multi-region support in a future release.
- The service supports only a single Pivotal Cloud Foundry® [availability zone](#). Although the service enables GemFire HA features such as redundancy and persistence, a failure of the targeted availability zone can result in the loss of session state data.
Note: If there is more than one available zone, you may see deployment failures or uneven sized clusters.
- Elastic scaling is not supported. The number of locators and cache servers allowed per GemFire instance (cluster) is fixed at 2 and 3 respectively. If you fail to define a ratio of 2:3 locators to cache servers in the resources tab of the Service tile in Ops Manager, deployment of the service will fail.
- If you deploy the service and then subsequently reduce the number of clusters, all allocated instances are destroyed.
- GemFire WAN replication is not supported.

1.0.2.0

Release Date: 1st September 2015

New in this Release:

- Stemcell 3062. This is a regular security upgrade that resolves the following issues:
 - [USN-2694-1] PCRE vulnerabilities
 - [USN-2698-1] SQLite vulnerabilities
 - [USN-2710-1] OpenSSH vulnerabilities
 - [USN-2710-2] OpenSSH regression
 - [USN-2718-1] Linux kernel (Vivid HWE) vulnerability
- Fix an issue identified in the 1.0.1.0 release where Pulse denied access after a service instance upgrade.

Known Issues and Limitations:

- Experimental feature HTTPS traffic to HAProxy not supported

1.0.1.0

Release Date: 4th August 2015

New in this Release:

- Stemcell 3026 (resolves CVE-2015-3290)
- Support for the experimental feature HTTPS traffic to UAA

Known Issues and Limitations:

- Experimental feature HTTPS traffic to HAProxy not supported
- For the running service instances that are upgraded from 1.0.0, Pulse starts to deny access upon upgrade with a cross-site request forgery (CSRF) related error. Service instances created upon upgrade are not affected.

1.0.0.0

Release Date: 1st July 2015

New in this Release

- This is the initial General Availability release
- Includes GemFire 8.1.0.5
- 3 data server VM's and 2 locator VM's per service instance (cluster)

- A dedicated VM per node (a service instance gets 5 dedicated nodes, 3 servers and 2 locators)
- Out of the box configuration for Tomcat container in the java buildpack

Known Issues and Limitations

- On AWS, this version of the product supports deployments only to the US-East region. Pivotal is working to provide multi-region support in a future release.
- The experimental HTTPS-only feature in Elastic Runtime 1.5 may cause issues with this version of the product. Pivotal is working to provide full support for HTTPS-only traffic in a future release.
Note: BOSH Stemcell 2865.1 is required for installation on Ops Manager 1.5.x and above.
- The service supports only a single Pivotal Cloud Foundry® [availability zone](#). Although the service enables GemFire HA features such as redundancy and persistence, a failure of the targeted availability zone can result in the loss of session state data.
Note: If there is more than one available zone, you may see deployment failures or uneven sized clusters.
- Elastic scaling is not supported. The number of locators and cache servers allowed per GemFire instance (cluster) is fixed at 2 and 3 respectively. If you fail to define a ratio of 2:3 locators to cache servers in the resources tab of the Service tile in Ops Manager, deployment of the service will fail.
- If you deploy the service and then subsequently reduce the number of clusters, all allocated instances are destroyed.
- GemFire WAN replication is not supported.
- If spaces are used in the name of the service plan, the service errand “Service Offering Smoke Test” fails, and the service installation fails as a result. To avoid this, you can either avoid using spaces in the name of the service plan, or uncheck the service errand “Service Offering Smoke Test”.

Overview

Session State Caching Powered by GemFire enables you to manage the session state for your applications independently of your application servers. This enables you to remove or reconfigure application servers as needed without affecting the session lifetime. It also enables clients to load balance between application servers without concern for the location of a particular session state.

Session State Caching Powered by GemFire is optimized for Java buildpack applications that are deployed in Tomcat. These applications require no additional configuration to utilize the service. The service's credentials are automatically discovered when you bind a java buildpack app to the service. In turn, [Pivotal Cloud Foundry®](#) (PCF) application developers can rapidly connect a running Session State Caching service to their Elastic Runtime-hosted applications.

As with other Pivotal Cloud Foundry® data services, Pivotal Cloud Foundry® administrators can now import, configure and define a Session State Caching service plan that uses GemFire for their Pivotal Cloud Foundry® Elastic Runtime environments. Pivotal also provides the Pulse Web application to help you monitor the service instances that you deploy.

GemFire Cache Configuration

Session State Caching Powered by GemFire is implemented as a pre-configured GemFire cluster running on Pivotal Cloud Foundry®. A single region named `sessions` is configured to store the session state data. This region has the following characteristics:

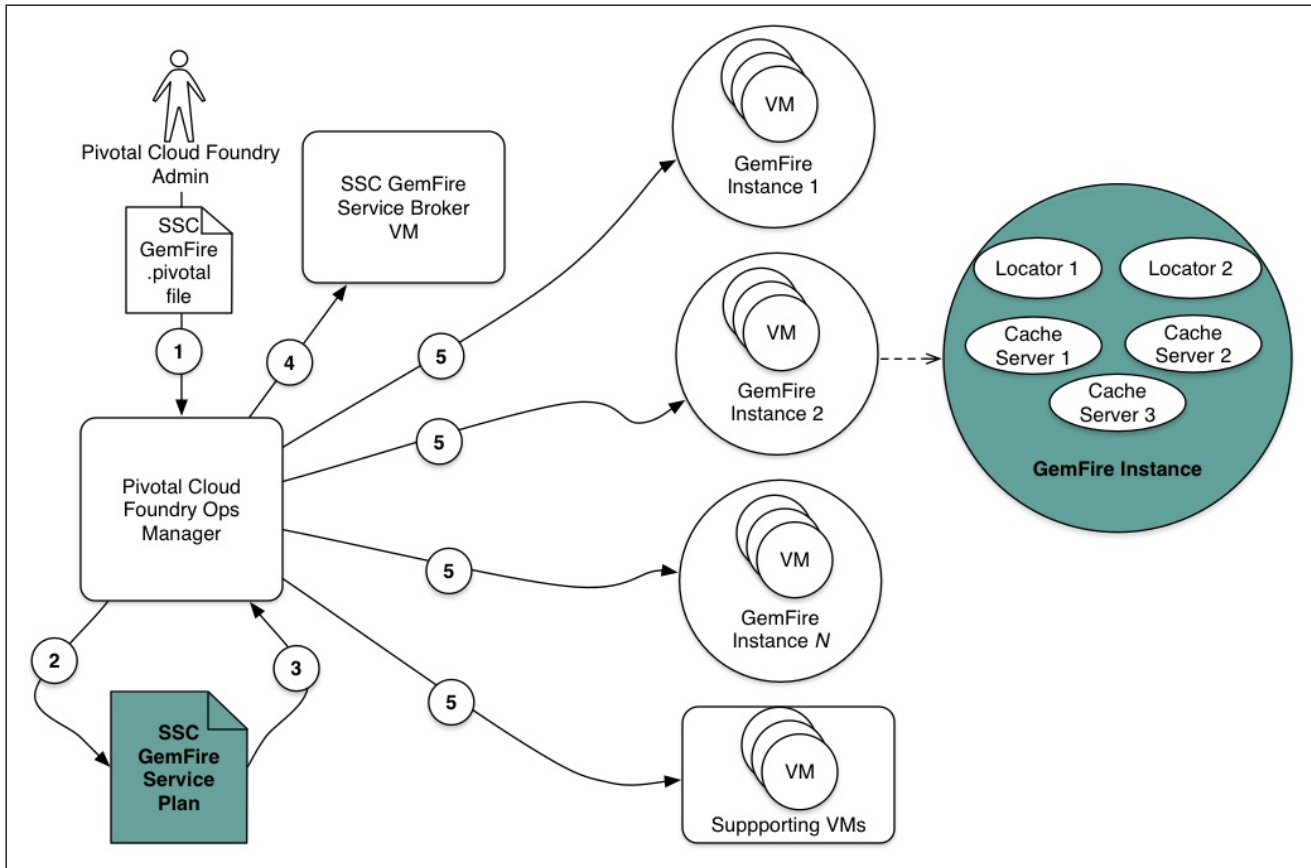
- The `sessions` region is partitioned across all available cache servers in the cluster. Redundant copies of the data are maintained to prevent service interruption or data loss in the case of a single cache server failure.
- The region is configured for persistence, so you can reconfigure and restart the GemFire cluster without losing the available session state data.
- Overflow is configured to write data to disk when JVM heap usage reaches 70%, to help protect from memory pressure.

Note: Although the service enables GemFire HA features such as redundancy and persistence, the service is limited to a single Cloud Foundry availability zone. See the [Release Notes](#) for more information.

How Does the Service Work?

The following diagrams depict the high-level workflow for deploying the Session State Caching service.

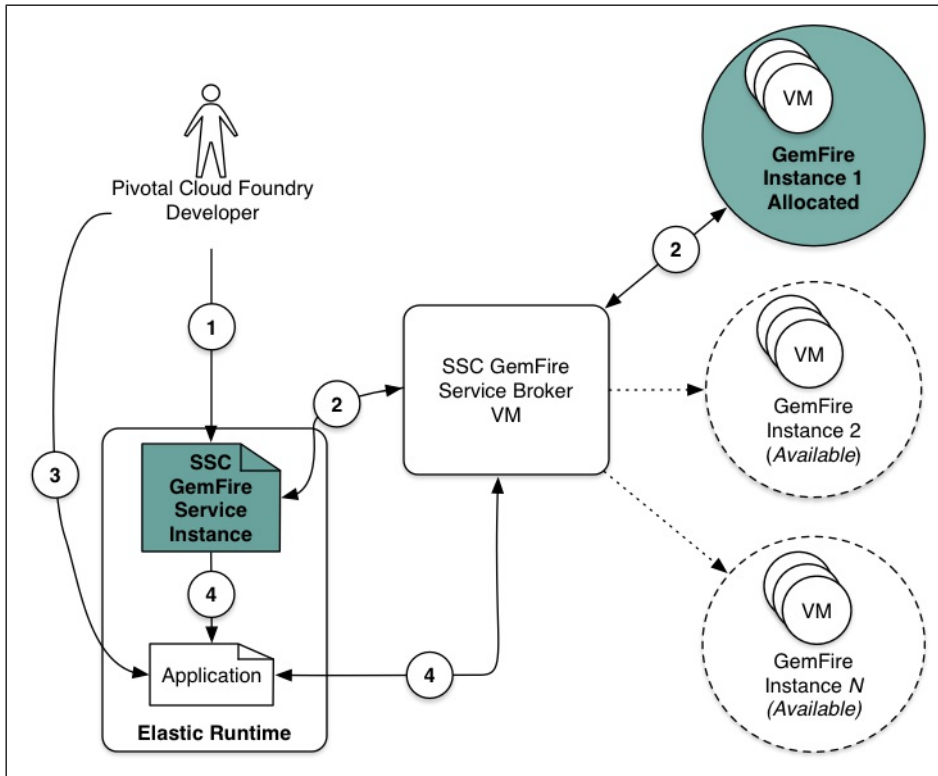
Figure 1. Installing and Configuring the Service



1. Pivotal Cloud Foundry® Administrator imports the Session State Caching product file into Pivotal Cloud Foundry® Operations Manager.
2. When a Pivotal Cloud Foundry® Administrator configures the Session State Caching service for the first time, they define the number of discrete Session State Caching service instances, each composed of five virtual machines, that they wish to make available to Pivotal Cloud Foundry® developers. Two VMs each run the SSC GemFire Locator process, and three VMs each run the SSC GemFire Server process.
3. After making their configuration choices, the Pivotal Cloud Foundry® Administrator deploys the service, causing Pivotal Cloud Foundry® Ops Manager to create and setup the appropriate set of virtual machines.
4. Operations Manager deploys a virtual machine to run a software component known as the SSC GemFire Service Broker, which is responsible for allocating Session State Caching service instances to Pivotal Cloud Foundry® users and passing relevant information into the applications that are bound to each service instance.
5. Operations Manager also deploys the specified number of GemFire cluster instances as configured by the administrator. Lastly, there are a few temporary VMs used for install and uninstall errands.

After the service has been deployed in Operations Manager, it is ready for use by application developers in the Elastic Runtime environment. The following diagram depicts the high-level workflow that an application developer would use to consume the service.

Figure 2. Using the Service



1. Pivotal Cloud Foundry® Developer uses the CLI or Developer Console to create a Session State Caching service instance in the Elastic Runtime space where they are pushing the relevant applications.
2. Creating a service instance causes Elastic Runtime to contact the SSC GemFire Service Broker, which allocates one un-allocated GemFire cluster of virtual machines to the Session State Caching service instance. In the response to Elastic Runtime, the Service Broker includes the URL to reach the GemFire Pulse monitoring tool for the service instance. The Elastic Runtime exposes this URL in the Developer Console's metadata for the service instance.
3. Pivotal Cloud Foundry® Developer pushes their application to Elastic Runtime.
4. Pivotal Cloud Foundry® Developer binds their application to the Session State Caching service instance. The SSC GemFire Service Broker populates the application's VCAP_SERVICES environment variable with the metadata required to access the Session State Caching service instance's locators and cache servers.

Version

This document describes Session State Caching Powered by GemFire, version 1.0.0.0. The Session State Caching Powered by GemFire service is running Pivotal GemFire 8.0.

Additional Resources

- [Pivotal GemFire Product Documentation](#)
- [Pivotal GemFire Community Forum](#)
- [Pivotal GemFire KnowledgeBase](#)
- [Pivotal Cloud Foundry® Documentation](#)

Installing Session State Caching Powered by GemFire

- [Prerequisites](#)
- [Service Configuration Defaults](#)
- [Installation Steps](#)
- [Creating a Pivotal GemFire Service Plan](#)

Prerequisites

Before you begin your Session State Caching Powered by GemFire deployment, your system needs to meet the following minimum requirements:

- [Pivotal Cloud Foundry®](#) [PCF](#) Ops Manager for vSphere, or Pivotal Cloud Foundry® Ops Manager for vCloud Air or vCloud Director, or Pivotal Cloud Foundry® Ops Manager for AWS
- Pivotal Cloud Foundry® Elastic Runtime
- Network access and credentials for the Pivotal Cloud Foundry® Ops Manager Web Console
- **Capacity in the vSphere cluster for service instances you want to deploy.** Each instance the Session State Caching service deploys a minimum of two GemFire locator and three GemFire cache server VMs, as well as several supporting VMs. The default instance has the following requirements:

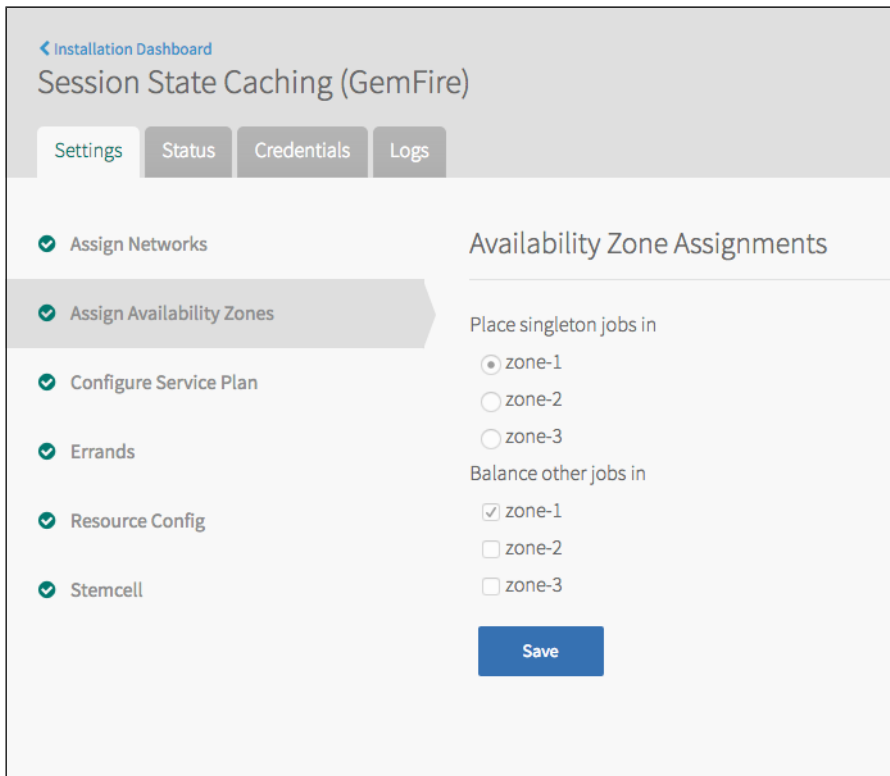
Table 1. Default Resource Requirements for a Session State Caching Service Instance

Service	Virtual Machines	CPU*	RAM (MB)*	Ephemeral Disk (MB)*	Persistent Disk (MB)*
SSC GemFire Locator	2	1	2048	2048	512
SSC GemFire Server	3	1	4096	6144	8192
SSC GemFire Broker	1	2	4096	4096	4096
Broker Registrar	1	1	2048	2048	0
Broker Deregistrar	1	1	2048	2048	0
Cluster Smoke Test	1	1	512	2048	0
Service Offering Smoke Test	1	1	512	2048	0
Compilation	2	1	1024	4096	0
Totals	**12**	**9**	**16384**	**24576**	**12800**

*Required for each VM

Limitations

The *Session State Caching Powered by GemFire for Pivotal Cloud Foundry®* service does not support multiple availability zones. You must configure the service to place singleton jobs and to balance other jobs in the same availability zone.



Security

If you are using a self-signed SSL certificate, or a certificate signed by an internal (or not well-known) certificate authority (CA), you must add the certificate, or certificate chain, to the “Trusted Certificates” in Ops Manager.

Support for HTTPS Traffic Only

The service supports disabling HTTP for all the incoming traffic to HAProxy and UAA.

Service Configuration Defaults

The Session State Caching Powered by GemFire service is intended for use with production-level workloads. When you create a service instance, Pivotal Cloud Foundry® automatically implements the best practices an operator would normally employ when deploying GemFire. These configurations include:

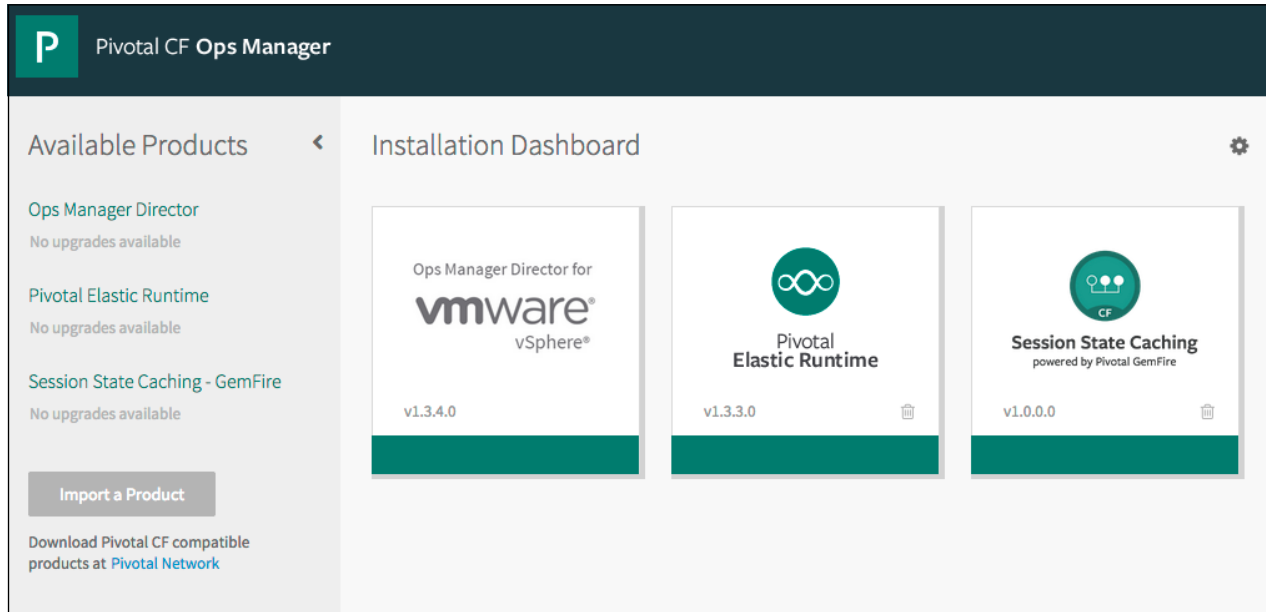
- **Dynamic memory management.** Dynamically set maximum and minimum memory utilization for each instance of the GemFire software process to the VM’s total memory (minus 128MB, which is reserved for the Operating System.)
- **Garbage collection.** Enable memory garbage collection for each instance of the GemFire software process.
- **Logging configuration.** Create and store logs at the config log level with appropriate limitations on sizing and automatic log rotation.
- **Statistics monitoring.** Create and store statistics files at a sample rate of 1,000, which corresponds to a sampling rate of once per second.
- **Data overflow configuration.** Overflow data to disk when crossing 70% memory utilization.
- **Memory overutilization protection.** Prevent further writes to memory when crossing 80% memory utilization.
- **Client authentication.** Require clients to authenticate when directly accessing cache servers.

Installation Steps

1. Download the GemFire Service 1.0 software binary from [The Pivotal Network](#).
2. Use a Web browser to log in to the **Pivotal Ops Manager** application. The **Pivotal Cloud Foundry® Ops Manager Installation Dashboard** displays.
3. Click **Import a Product**.

The **Add Products** screen displays.

4. Click **Choose File** and navigate to the file you downloaded.
The file uploads to your Pivotal Cloud Foundry® deployment.
5. Click **Add**.
Pivotal Ops Manager adds a new tile for **Session State Caching Powered by GemFire** to the Installation Dashboard.



Creating a Session State Caching Powered by GemFire Service Plan

Configure one or more Session State Caching instances to set the maximum capacity for Session State Caching services in your Cloud Foundry environment.

Note: If you have already configured the service, reducing the the number of resources and reconfiguring the service causes all allocated instances to be destroyed. See the [Release Notes](#) for more information about known problems and limitations.

Follow these steps to configure the GemFire service:

1. Select the **Session State Caching Powered by GemFire** tile to display the configuration page.
2. (Optional) Select the **Assign Networks** tab to specify the vSphere Network where Ops Manager deploys the SSC GemFire Service Broker and related virtual machines. See [Configuring Network Isolation Options in Ops Manager](#) in the Pivotal Cloud Foundry® documentation for more information.
3. (Optional) Select **Assign Availability Zones** to specify the Availability Zone into which Ops Manager should deploy the SSC GemFire Service Broker and virtual machines. See [Configuring Ops Manager Director for VMware vSphere](#) for more information.

Note: The current version of the service only supports deployment to one Availability Zone. For this release, we do not recommend modifying the default configuration setting.

4. (Optional) Select the **Configure Service Plan** tab to configure information that is displayed about this instance in the Elastic Runtime CLI and Web Console. The Service Plan Name, Description, and Feature Bullets can be used as needed to describe the service.
5. (Optional) Select the **Resource Config** tab to change the configuration of GemFire members in the deployed service instance:

Pivotal CF Ops Manager tempest

[Installation Dashboard](#)

Session State Caching - GemFire

Settings | Status | Credentials | Logs

- Assign Networks
- Assign Availability Zones
- Configure Service Plan
- Lifecycle Errands
- Resource Config**

Resource Config

JOB	INSTANCES	CPU	RAM (MB)	EPHEMERAL DISK (MB)	PERSISTENT DISK (MB)
SSC GemFire Locator	2	1	2048	2048	512
SSC GemFire Server	3	1	4096	6144	8192
SSC GemFire Broker	1	2	4096	4096	4096
Broker Registrar	1	1	2048	2048	0
Broker Deregistrar	1	1	2048	2048	0
Cluster Smoke Test	1	1	512	2048	0
Service Offering Smoke Test	1	1	512	2048	0
Compilation	2	1	1024	4096	0

[Save](#)

- a. Enter the total number of SSC GemFire Locator and SSC GemFire Server VM instances that you want to deploy.

For example, if you need to support a maximum of two (2) developer applications, two (2) staging applications, and one (1) production application, you would need to configure at least five (5) instances of Session State Caching to support all applications running simultaneously. Because each Session State Caching instance creates a GemFire cluster of two GemFire locators and three GemFire servers, you would need to configure a total of 10 SSC GemFire Locators and 15 SSC GemFire Servers in this environment.

Note: You must maintain a ratio of two locator VMs to three cache server VMs. Each set of two locators and three servers forms a separate GemFire cluster. Setting the values to any other ratio will cause the deployment to fail.

- b. Enter the number of CPUs, amount of RAM, Ephemeral Disk Space, and Persistent Disk space to allocate for each server VM based on your capacity requirements, or accept the default. For production deployments, we recommend increasing the number of CPUs to at least 2.

6. Click **Save**.

7. Click the **Installation Dashboard** link.
The **Installation Dashboard** screen displays.

8. Click **Apply Changes**.

Pivotal Cloud Foundry® Ops Manager deploys a single virtual machine to run the SSC GemFire Service Broker, and one or more additional VMs to run the GemFire locators and servers. A progress meter displays the progress of the installation. You can also access information about the deployments from the Pivotal Cloud Foundry® Ops Manager console.

After the Session State Caching instances are deployed, the SSC GemFire Service Broker automatically registers the service and its service plan in the Elastic Runtime Marketplace. Pivotal Cloud Foundry® users can then create and bind to instances of the configured service plans. See [Using the Pivotal GemFire Service on Pivotal Cloud Foundry®](#).

Using Session State Caching Powered by GemFire on Pivotal Cloud Foundry®

As you would expect with any data service on [Pivotal Cloud Foundry®](#) (PCF), the Session State Caching service simplifies the deployment and configuration of software that supports your applications. When you create a Session State Caching service instance, you are instantly provided with a dedicated instance of GemFire software without having to install and manage the underlying virtual machines or software manually.

- [About Session State Caching](#)
- [Creating a Session State Caching Service Instance](#)
- [Deploying Applications for Use with the Session State Caching Service](#)
- [Binding an Application to the Session State Caching Service](#)
- [Pushing or Restaging Applications After Service Changes](#)
- [Viewing Binding Meta Data and Environment Variables](#)
- [Unbinding an Application from the Session State Caching Service](#)
- [Deleting a Session State Caching Service Instance](#)
- [Example Walkthrough: Deploying a Spring Java Application with Session State Caching Powered by GemFire](#)

About Session State Caching

Session State Caching Powered by GemFire can provide fast, scalable, and reliable session replication for your web applications without requiring application changes.

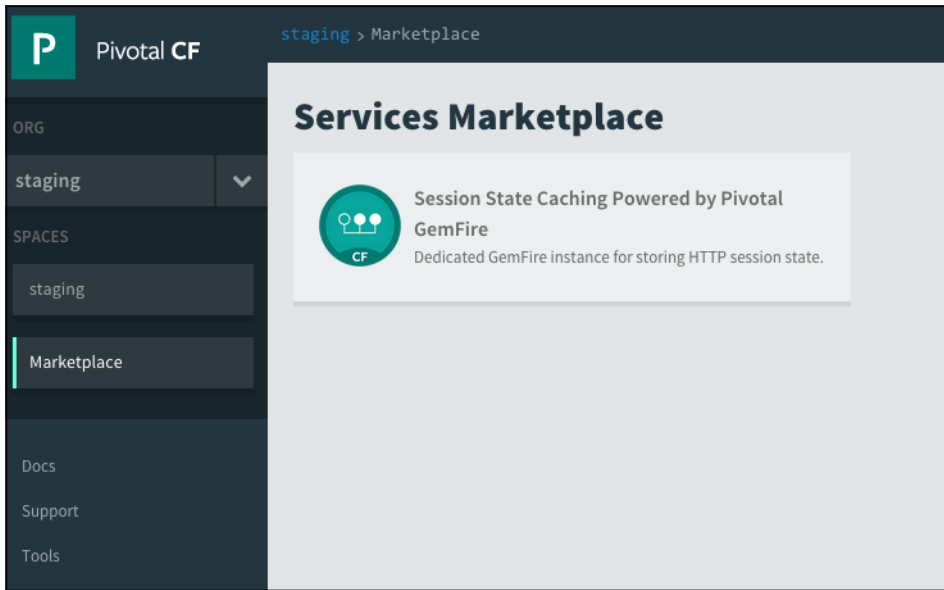
Using Session State Caching Powered by GemFire is useful in a Pivotal Cloud Foundry® environment because Pivotal Cloud Foundry® does not persist or replicate HTTP session data. This can be problematic when an instance of an application has crashed or stopped. The Session State Caching service makes session data available after an application crashes or stops and across all instances of an application.

Creating a Session State Caching Service Instance

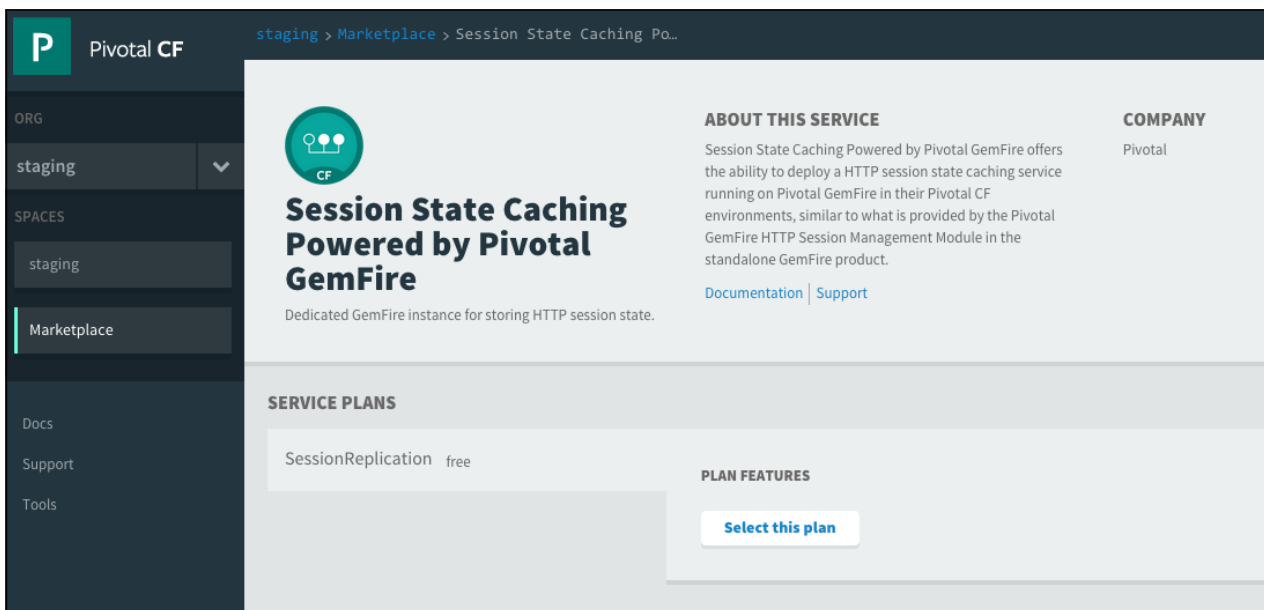
The following procedure describes how to create a Session State Caching service instance in the Pivotal Cloud Foundry® Elastic Runtime environment.

PCF Developer Console Instructions

1. Log in to the **Pivotal Cloud Foundry® Developer Console**.
2. Select your **Org** from the drop-down list on the left.
3. Select **Marketplace** from the left navigation menu under Spaces. The **Services Marketplace** displays the following:



4. Select **Session State Caching Powered by GemFire**. The Session State Caching Powered by GemFire service page displays. The service page displays details on the plan, such as cost and active features. The service plan details are configured by your Cloud Foundry administrator when they install the service in Pivotal Cloud Foundry® Ops Manager.



5. To create the service instance, click on the appropriate **Select this plan** button. The **Configure Instance** screen appears.
6. In the **Instance Name** field, enter a name that will identify this specific Session State Caching service instance.
7. 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.
8. (Optional) Select an application from the **Bind to App** drop-down list. The drop-down list displays available applications previously configured for this Org and Space. If you do not want to bind an application, select **[do not bind]**. You can bind your application later as described in [Binding an Application to the GemFire Service](#).
9. Click the **Add** button.

Note: A GemFire instance (cluster) must be available for allocation. If all deployed GemFire instances have been allocated, you will receive an error. You will either need to delete existing service instances or ask the admin to deploy more instances in Ops Manager. See also [Problem: Could not deploy a service instance in the Elastic Runtime](#).

CLI Instructions

1. If you have not done so already, install the Pivotal Cloud Foundry® Command Line Interface. See [Installing the cf CLI](#). (Installation binaries are available [here](#)).

2. Log in to PCF using the cf CLI.

```
$ cf login
```

3. Run the following command to target the API endpoint, org, and space where you want to create the service:

```
$ cf target -a API-ENDPOINT -o ORG-NAME -s SPACE-NAME
```

4. Run the following command to view the available service plans:

```
$ cf marketplace
```

```
Getting services from marketplace in org staging / space staging as admin...
OK
service      plans      description
p-ssc-gemfire SessionReplication Dedicated GemFire instance for storing HTTP session state.
TIP: Use 'cf marketplace -s SERVICE' to view descriptions of individual plans of a
given service.
```

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

```
$ cf create-service p-ssc-gemfire SERVICE-PLAN-NAME SERVICE-INSTANCE-NAME
```

where SERVICE-PLAN-NAME is the name of the Service Plan you see in the marketplace, in this example, **SessionReplication**, and SERVICE-INSTANCE-NAME is a descriptive name that you want to use for the service.

For example:

```
$ cf create-service p-ssc-gemfire SessionReplication my-ssc-gemfire-test
Creating service my-ssc-gemfire-test in org staging / space staging as admin...
OK
```

Deploying Applications for Use with the Session State Caching Service

This section provides tips on pushing your application to the Pivotal Cloud Foundry® Elastic Runtime environment for use with the Session State Caching service. Some of the application deployment steps may differ depending on what kind of application you are deploying. Currently this service only supports the Java Buildpack.

You will eventually want to scale your application to two or more instances for the purposes of achieving session replication.

Re-deploying your application does not affect data stored in any existing service instances bound to the application.

See [Deploy an Application](#) in the Pivotal Cloud Foundry® Documentation for detailed information on pushing CF applications.

Using the Java Buildpack

The Java Buildpack has been customized specifically for use with Session State Caching Powered by GemFire. When you use the Java Buildpack, your application automatically discovers the service environment variables and credentials, and no specific application configuration is required. See [Example Walkthrough: Binding a Spring Java Application to a GemFire Service Instance](#) for an example of this.

Important: Pivotal recommends that you use the latest Java Buildpack when pushing your applications. Specify the location of the buildpack using the `-b` parameter:

```
$ cf push YOUR-APP-NAME -p LOCATION-OF-YOUR-APP-FILE \
-b https://github.com/cloudfoundry/java-buildpack.git
```

To avoid having to restage your application after binding the service, you can push the application initially with the `--no-start` command:

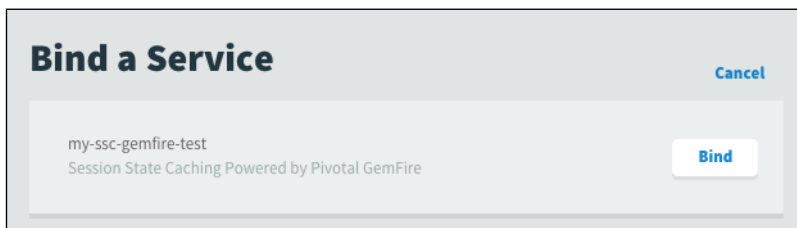
```
$ cf push YOUR-APP-NAME -p LOCATION-OF-YOUR-APP-FILE \
-b https://github.com/cloudfoundry/java-buildpack.git --no-start
```

Binding an Application to the Session State Caching Service

The following procedures describe how to bind a Session State Caching service instance to your Pivotal Cloud Foundry® application.

PCF Developer Console Instructions

1. Log in to the **Pivotal Cloud Foundry® Developer Console**.
2. Select your **Org** from the drop-down list on the left. This should be the same Org where you created the service instance.
3. Select the space where your Session State Caching service instance and bound application have been deployed.
4. Select the application that you wish to bind to the service. A page displays showing the already bound services and instances for this application.
5. Click **Bind**. A list of available services displays.



6. Click the **Bind** button for the GemFire service you want to bind to this application.
7. Using the Pivotal Cloud Foundry® CLI, start or restage your application. See [Pushing or Restaging Applications After Changes](#).

CLI Instructions

Alternately via the CLI:

1. Log in to your Pivotal Cloud Foundry® environment using the Pivotal Cloud Foundry® CLI.
`$ cf login`
2. Run the following command to target the specific org and space where you want to create the service plan.
`$ cf target -o ORG-NAME -s SPACE-NAME`
3. Run the following command to view running service instances:
`$ cf services`

```
Getting services in org staging / space staging as admin...
OK
name           service           plan              bound apps
my-ssc-gemfire-test  p-ssc-gemfire    SessionReplication
```

4. Run the following command to bind the application to the service instance:

```
$ cf bind-service APP-NAME SERVICE-INSTANCE-NAME
```

For example:

```
$ cf bind-service spring-petclinic my-ssc-gemfire-test
Binding service my-ssc-gemfire-test to app spring-petclinic
in org staging / space staging as admin...
OK
TIP: Use 'cf restage' to ensure your env variable changes take effect
```

5. Restage your application.

```
$ cf restage APP-NAME
```

Pushing or Restaging Applications After Service Changes

To ensure that your application picks up the correct environment variables, you must restage or re-push your applications after binding them to the GemFire service. This can currently be done using the CLI. In addition, if you make any other changes to the GemFire Service while it is bound to your application (for example, add, modify or delete the service), you will need to re-push or restage your application afterwards.

Pushing or re-staging your application does not affect data stored in the existing service instance.

To restage or re-push your application using the Pivotal Cloud Foundry® CLI:

1. Log in to your Pivotal Cloud Foundry® environment using the Pivotal Cloud Foundry® CLI.

```
$ cf login
```

2. Run the following command to target the API endpoint, org and space where you want to push or restage the application. For example:

```
$ cf target -a API-ENDPOINT -o ORG-NAME -s SPACE-NAME
```

3. Push or restage your existing application:

```
$ cf push APP-NAME -p LOCATION-OF-YOUR-APP-FILE -b BUILDPACK-LOCATION
```

or

```
$ cf restage APP-NAME
```

Alternately, if you pushed your application without starting it, you can start your application now to pick up the newly bound service.

```
$ cf start APP-NAME
```

For more details on deploying applications, see [Deploy an Application](#) in the Pivotal Cloud Foundry® documentation.

Viewing Binding Meta Data and Environment Variables

To view the binding variables, use the following procedures.

PCF Developer Console Instructions

1. Log in to the **Pivotal Cloud Foundry® Developer Console**.
2. Select your **Org** from the drop-down list on the left.
3. Select the space where your Session State Caching service instance and bound application have been deployed.
4. Select the application that you have bound to the Session State Caching service.
5. Click on the **Env Variables** tab. The environment variables for the service binding display:

The screenshot shows the Pivotal Cloud Foundry Developer Console interface. At the top, there are tabs for 'Activity', 'Services', 'Env Variables' (which is currently selected), and 'Routes'. In the top right corner, there is a 'Delete App' button. Below the tabs, the main content area is divided into two sections: 'USER PROVIDED' and 'SYSTEM PROVIDED'. The 'SYSTEM PROVIDED' section contains a 'VCAP_SERVICES' block with a JSON object representing the service binding details for 'p-ssc-gemfire'.

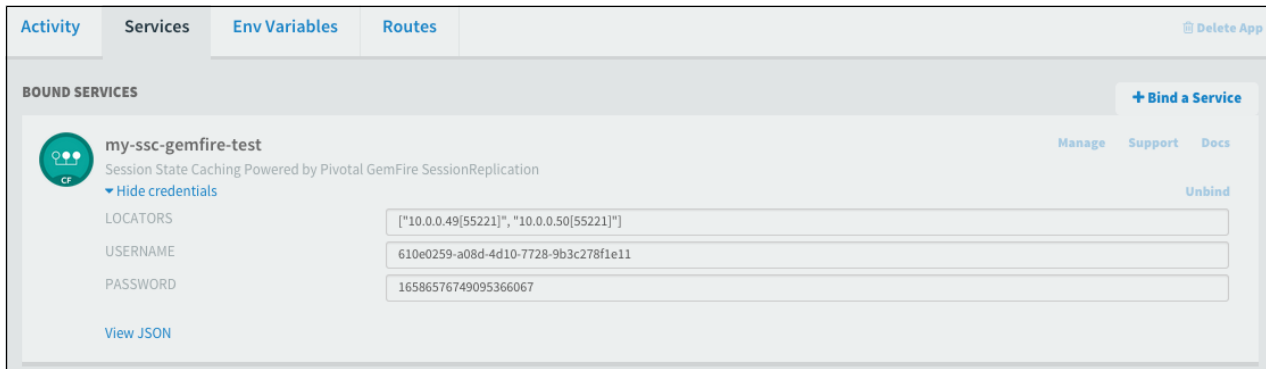
```

{
  "p-ssc-gemfire": [
    {
      "name": "my-ssc-gemfire-test",
      "label": "p-ssc-gemfire",
      "tags": [
        "gemfire",
        "session_replication"
      ],
      "plan": "SessionReplication",
      "credentials": {
        "locators": [
          "10.0.0.49[55221]",
          "10.0.0.50[55221]"
        ],
        "username": "610e0259-a08d-4d10-7728-9b3c278f1e11",
        "password": "16586576749095366067"
      }
    }
  ]
}

```

Alternately, you can also view credentials used by the service when binding to the application by clicking on the **Services** tab and clicking **Show credentials**.

Service credentials and locator address information displays:



CLI Instructions

To view the binding variables in the CLI, type the following command after you have bound your application to the GemFire service:

```
$ cf env APP-NAME
```

If successful, you should see content similar to the following in the returned output:

```
$ cf env spring-petclinic
Getting env variables for app spring-petclinic in org staging / space staging as admin...
OK
System-Provided:
{
  "VCAP_SERVICES": {
    "p-ssc-gemfire": [
      {
        "credentials": {
          "locators": [
            "10.0.0.49[55221]",
            "10.0.0.50[55221]"
          ],
          "password": "16586576749095366067",
          "username": "610e0259-a08d-4d10-7728-9b3c278f1e11"
        },
        "label": "p-ssc-gemfire",
        "name": "my-ssc-gemfire-test",
        "plan": "SessionReplication",
        "tags": [
          "gemfire",
          "session_replication"
        ]
      }
    ]
  }
}
No user-defined env variables have been set
No running env variables have been set
No staging env variables have been set
```

Unbinding an Application from the Session State Caching Service

When you create a Session State Caching service instance, the SSC GemFire Service Broker allocates a specific cluster of GemFire locators and servers. You can bind, unbind and the bind again to that particular service instance as often as you want. The service instance always uses the same cluster and the Service Broker does not do anything to the data in that service instance.

PCF Developer Console Instructions

To unbind the application from the GemFire service:

1. Log in to the **Pivotal Cloud Foundry® Developer Console**.

2. Select your **Org** from the drop-down list on the left.
3. Select the space where your Session State Caching service instance and bound application have been deployed.
4. Select the application that you have bound to the Session State Caching service. A page displays that show the bound services and instances for this application.
5. Locate the bound service instance you want to unbind and click **Unbind**.

The screenshot shows the Pivotal Cloud Foundry application management interface for an application named 'spring-petclinic'. The interface is divided into several sections:

- CONFIGURATION**: Includes a 'Scale App' link and three dropdown menus for 'INSTANCES' (set to 2), 'MEMORY LIMIT' (set to 1 GB), and 'DISK LIMIT' (set to 1 GB).
- STATUS**: A table showing the status of the application instances.

# ▲	STATUS	CPU	MEMORY	DISK	UPTIME
0	Running	1%	765 MB	168 MB	11 min
1	Running	1%	692 MB	168 MB	11 min
- ABOUT**: Provides details about the application, including the buildpack, stack, and route.
- Activity**: A tabbed interface with 'Services', 'Env Variables', and 'Routes'.
- BOUND SERVICES**: A section showing the services bound to the application. It lists 'my-ssc-gemfire-test' with a description 'Session State Caching Powered by Pivotal GemFire SessionReplication'. There are links for 'Manage', 'Support', 'Docs', and a circled 'Unbind' button.

6. A confirmation dialog box displays. Click **Unbind** again.
7. If successful, the following message will appear at the top of the screen:

Service successfully unbound from the application. TIP: Use 'cf push' to ensure your env variable changes take effect.
8. Use the Pivotal Cloud Foundry® CLI to push or restage your application for the changes to take effect. See [Pushing or Restaging Applications After Changes](#).

CLI Instructions

1. Log in to your Pivotal Cloud Foundry® environment using the Pivotal Cloud Foundry® CLI.


```
$ cf login
```
2. Run the following command to target the org and space where you want to push or restage the application. For example:


```
$ cf target -o ORG-NAME -s SPACE-NAME
```
3. Run the following command:

```
$ cf unbind-service APP-NAME SERVICE-INSTANCE-NAME
```

where SERVICE-INSTANCE-NAME is the name of the GemFire instance you are unbinding from the specified APP-NAME.
For example:

```
$ cf unbind-service spring-petclinic my-ssc-gemfire-test
Unbinding app spring-petclinic from service my-ssc-gemfire-test in org staging / space
staging as admin...
OK
```

4. Restage or re-push your application for the application changes to take effect. See [Pushing or Restaging Applications After Changes](#).

Deleting a Session State Caching Service Instance

When you delete a Session State Caching service instance, all applications that are bound to that service are automatically unbound and any data in the service instance is cleared. In addition, the allocated service instance (GemFire cluster) is returned to the pool of available clusters and those locators and cache servers are now available to future applications.

PCF Developer Console Instructions

To delete a service instance using the Pivotal Cloud Foundry® Developer Console:

1. Log in to the **Pivotal Cloud Foundry® Developer Console**.
2. Select your **Org** from the drop-down list on the left.
3. Locate the row under Services that contains the service instance you want to delete and click **Delete**.
4. 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. See [Pushing or Restaging Applications After Changes](#).

CLI Instructions

1. Log in to your Pivotal Cloud Foundry® environment using the Pivotal Cloud Foundry® CLI.

```
$ cf login
```

2. Run the following command to target the org and space where you want to push or restage the application. For example:

```
$ cf target -o ORG-NAME -s SPACE-NAME
```

3. Run the following command:

```
$ cf delete-service SERVICE-INSTANCE-NAME
```

where SERVICE-INSTANCE-NAME is the name of the Session State Caching service you are deleting. Enter 'y' when prompted.
For example:

```
$ cf delete-service my-ssc-gemfire-test
Really delete the service my-ssc-gemfire-test?> y
Deleting service my-gemfire-test in org staging / space staging as admin...
OK
```

4. 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. See [Pushing or Restaging Applications After Changes](#).

Example Walkthrough: Deploying a Spring Java Application with Session State Caching Powered by GemFire

This section provides a step-by-step example of deploying and binding a Spring Java sample application to the Session State Caching service on Pivotal Cloud Foundry®. After deploying the application, we'll look at how GemFire manages session replication by querying session data in GemFire's Pulse monitoring tool.

For demonstration purposes, we'll use the [Spring PetClinic](#) application and the session replication-enabled GemFire service. To follow the steps in this walkthrough, you will need:

- git command line tool
- Maven 3
- Pivotal Cloud Foundry® installation
- Pivotal Session State Caching service plan installation in your Elastic Runtime environment

- access to a Pivotal Cloud Foundry® Developer Console
- cf Command Line Interface (cf CLI) tool

Step 1: Download and package the sample application

1. Open a terminal. In a working directory, download the Spring Petclinic application from GitHub:

```
$ git clone https://github.com/SpringSource/spring-petclinic.git
```

2. Change directories to the spring-petclinic directory.

```
$ cd spring-petclinic
```

3. Build the application WAR file with Maven.

```
$ mvn package
```

As defined by the project's pom.xml file, this command will build the following file in the following location within the spring-petclinic directory:

```
target/petclinic.war
```

Step 2: Push the PetClinic application to Pivotal Cloud Foundry®

1. Log in to your Pivotal Cloud Foundry® environment using the Pivotal Cloud Foundry® CLI.

```
$ cf login
```

2. This step requires that you have a running Pivotal Cloud Foundry® environment where you can push your applications. Run the following command to target the API endpoint, org and space where you will run the application. For example:

```
$ cf target -a API-ENDPOINT -o ORG-NAME -s SPACE-NAME
```

3. Push the petclinic application to your Pivotal Cloud Foundry® installation:

```
$ cf push spring-petclinic -p target/petclinic.war -i 2 -b https://github.com/cloudfoundry/java-buildpack.git --no-start
```

This command specifies that the Pivotal Cloud Foundry® specifies the location of the application file and the Java buildpack to use. Since we are using Session State Caching for session state replication, we will run two instances of the application. In addition, we will specify the `--no-start` option so that we can pick up the bound service before starting the application. After running the command, you should see output similar to the following:

```
$ cf push spring-petclinic -p target/petclinic.war -i 2
-b https://github.com/cloudfoundry/java-buildpack.git --no-start
Creating app spring-petclinic in org staging / space staging as admin...
OK

Using route spring-petclinic.pandora.bboxes.cf-app.com
Binding spring-petclinic.pandora.bboxes.cf-app.com to spring-petclinic...
OK

Uploading spring-petclinic...
Uploading app files from: target/petclinic.war
Uploading 730.4K, 140 files
Done uploading
OK
```

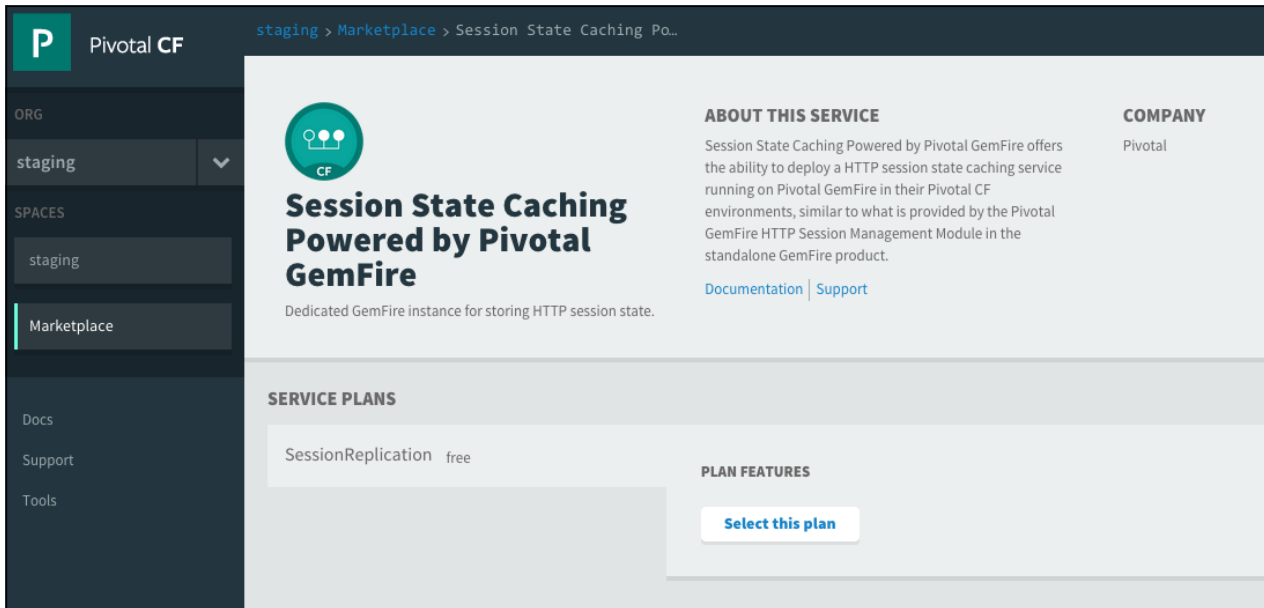
4. Keep this terminal open for step 5.

Step 3: Create a Pivotal GemFire Service Instance

Note: This step assumes that your Pivotal Cloud Foundry® Administrator has already deployed Session State Caching Powered by GemFire to your Elastic Runtime environment and that it is available from the **Marketplace**. See [Installing the Pivotal GemFire Service for Pivotal Cloud Foundry®](#) for instructions.

Next, we will create a service instance in your environment.

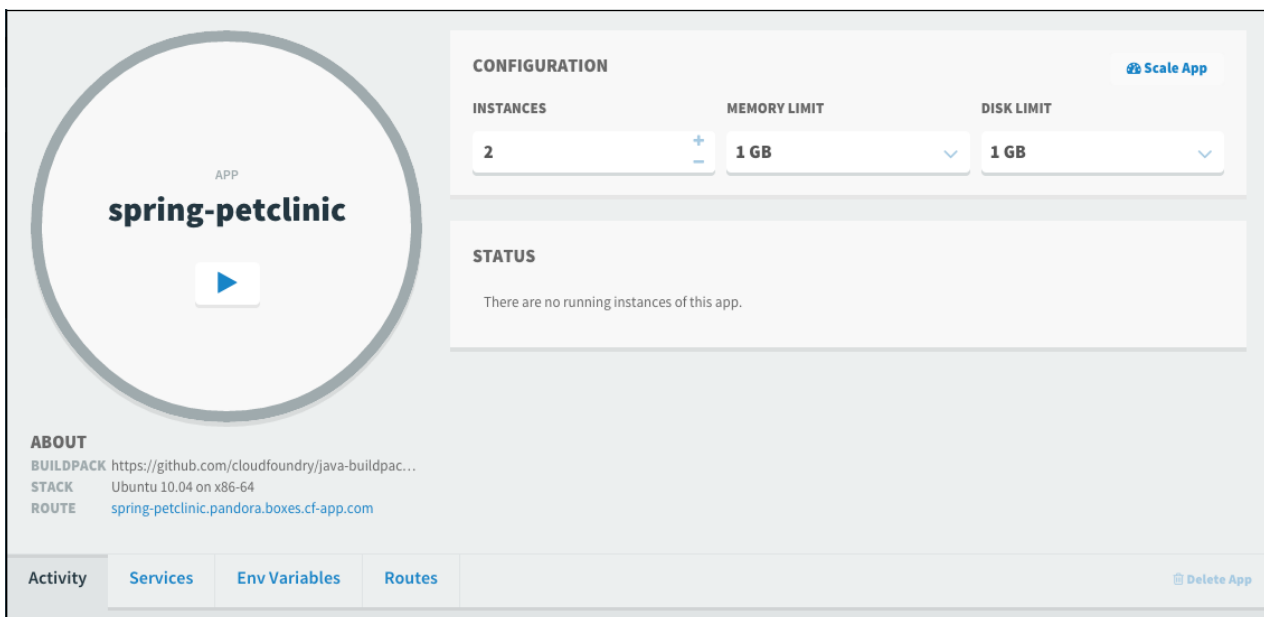
1. Log in to the **Pivotal Cloud Foundry® Developer Console**.
2. Select your **Org** from the drop-down list on the left.
3. Select **Marketplace** from the left navigation menu under Spaces. The **Services Marketplace** displays.
4. Select **Session State Caching Powered by GemFire**. The Session State Caching service page displays.



5. To create the service instance, click on the appropriate **Select this plan** button. The **Configure Instance** screen appears.
6. In the **Instance Name** field, enter a descriptive name for this Session State Caching service instance– for example, “my-ssc-gemfire-test”.
7. Select a space from the **Add to Space** drop-down list.
8. From the **Bind to App** drop-down list, select [do not bind]. We will bind the application in the next step. Click the **Add** button.

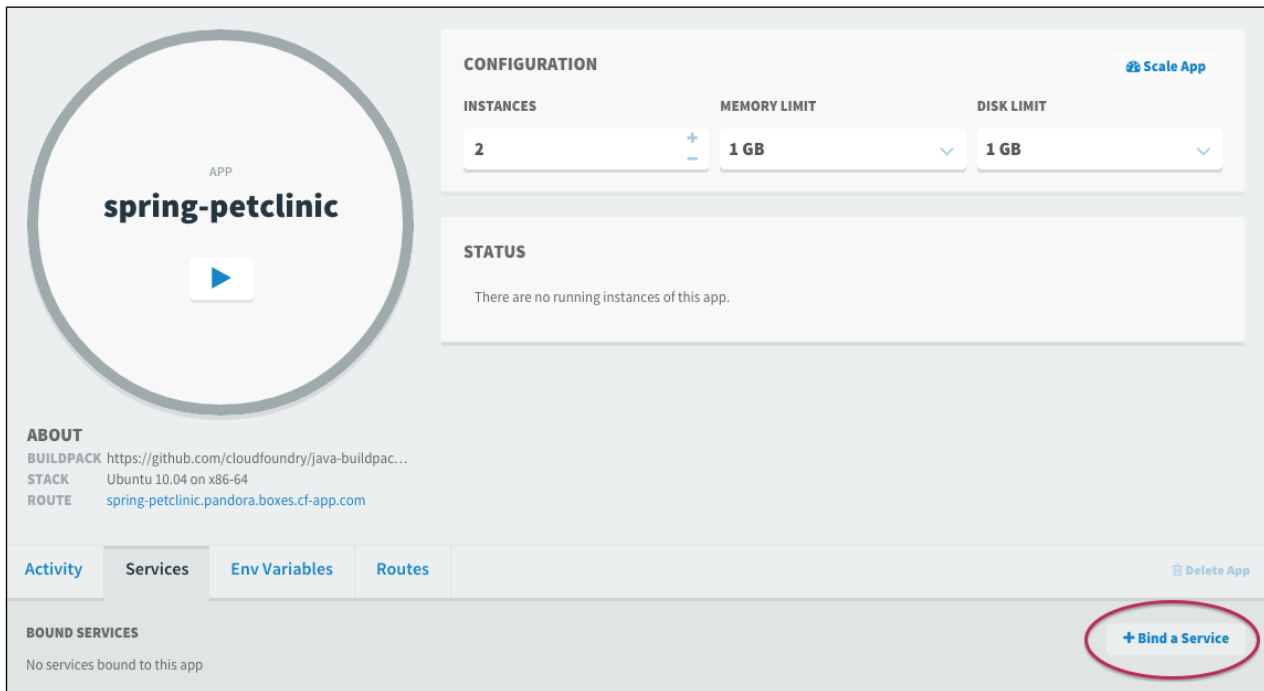
Step 4: Bind the Pivotal GemFire service to your application

1. Select the space where your Pivotal GemFire service instance and **spring-petclinic** have been deployed.
2. Select the **spring-petclinic** application. A page displays showing recent logs and application activity.



At this point, the application is not yet running.

3. Click on the **Services** tab.
4. Click on the **Bind a service** button. A list of available services displays.



APP
spring-petclinic

CONFIGURATION [Scale App](#)

INSTANCES	MEMORY LIMIT	DISK LIMIT
2	1 GB	1 GB

STATUS

There are no running instances of this app.

ABOUT

BUILDPACK <https://github.com/cloudfoundry/java-buildpac...>
STACK Ubuntu 10.04 on x86-64
ROUTE spring-petclinic.pandora.bboxes.cf-app.com

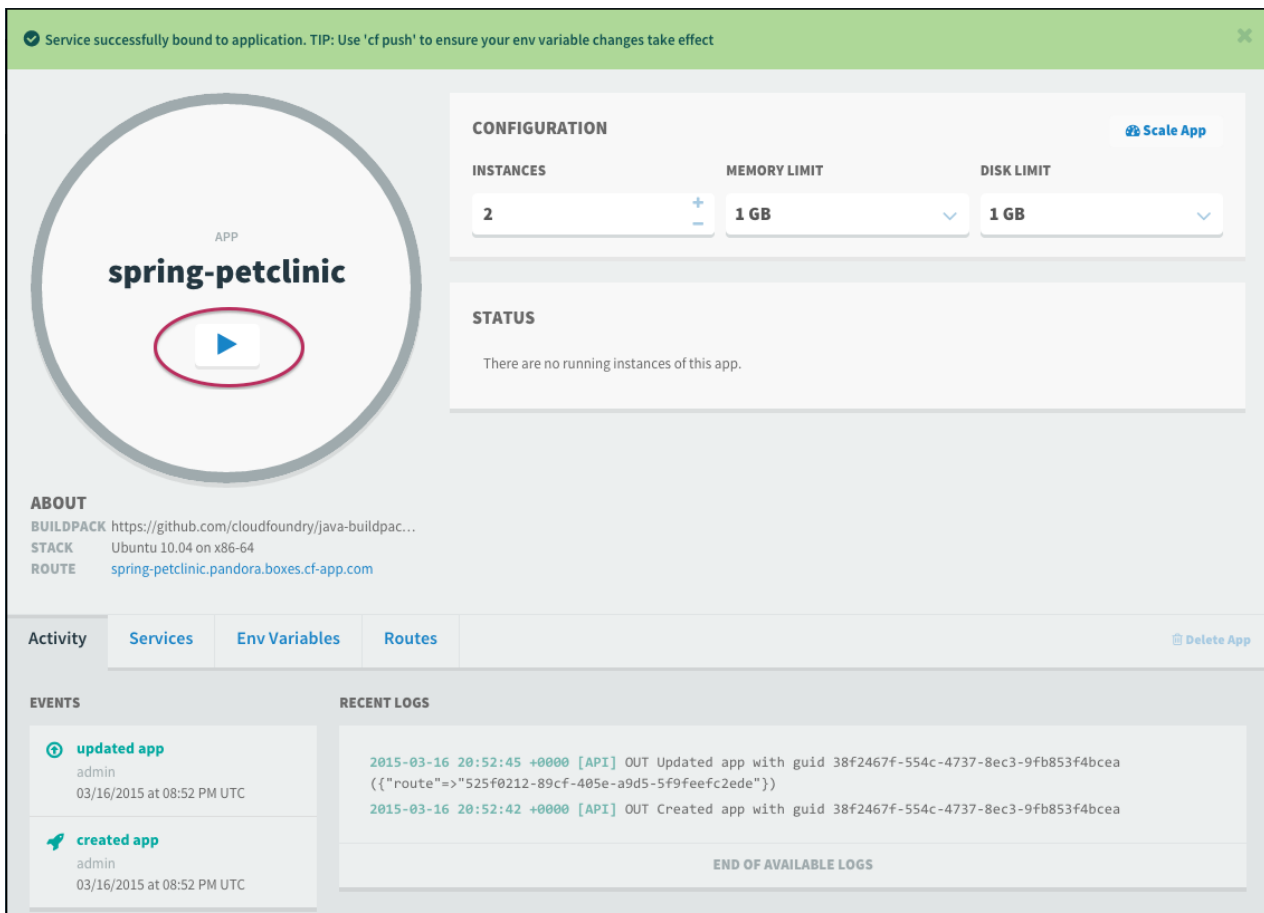
Activity **Services** **Env Variables** **Routes** [Delete App](#)

BOUND SERVICES

No services bound to this app

[+ Bind a Service](#)

- Click the **Bind** button for the GemFire service you created in Step 3.
- Click the Play button on the application to start the bound application instances.



✓ Service successfully bound to application. TIP: Use 'cf push' to ensure your env variable changes take effect

APP
spring-petclinic

CONFIGURATION [Scale App](#)

INSTANCES	MEMORY LIMIT	DISK LIMIT
2	1 GB	1 GB

STATUS

There are no running instances of this app.

ABOUT

BUILDPACK <https://github.com/cloudfoundry/java-buildpac...>
STACK Ubuntu 10.04 on x86-64
ROUTE spring-petclinic.pandora.bboxes.cf-app.com

Activity **Services** **Env Variables** **Routes** [Delete App](#)

EVENTS

- updated app**
admin
03/16/2015 at 08:52 PM UTC
- created app**
admin
03/16/2015 at 08:52 PM UTC

RECENT LOGS

```
2015-03-16 20:52:45 +0000 [API] OUT Updated app with guid 38f2467f-554c-4737-8ec3-9fb853f4bcea
({"route"=>"525f0212-89cf-405e-a9d5-5f9feefc2ede"})
2015-03-16 20:52:42 +0000 [API] OUT Created app with guid 38f2467f-554c-4737-8ec3-9fb853f4bcea
```

END OF AVAILABLE LOGS

- Alternatively, return to your open terminal. Using the Pivotal Cloud Foundry® CLI, start your application.

```
$ cf start spring-petclinic
```

You should see output similar to the following:

```
Starting app spring-petclinic in org staging / space staging as admin...
-----> Downloaded app package (26M)
```

```
Cloning into '/tmp/buildpacks/java-buildpack'...
-----> Java Buildpack Version: 623c428 | https://github.com/cloudfoundry/java-buildpack.git#623c428
-----> Downloading Open Jdk JRE 1.8.0_40 from https://download.run.pivotal.io/openjdk/lucid/x86_64/openjdk-1.8.0_40.tar.gz (2.3s)
    Expanding Open Jdk JRE to .java-buildpack/open_jdk_jre (3.1s)
-----> Downloading Spring Auto Reconfiguration 1.7.0_RELEASE from https://download.run.pivotal.io/auto-reconfiguration/auto-reconfigurat:
    Modifying /WEB-INF/web.xml for Auto Reconfiguration
-----> Downloading Tomcat Instance 8.0.20 from https://download.run.pivotal.io/tomcat/tomcat-8.0.20.tar.gz (0.4s)
    Expanding Tomcat to .java-buildpack/tomcat (0.3s)
-----> Downloading Tomcat Lifecycle Support 2.4.0_RELEASE from https://download.run.pivotal.io/tomcat-lifecycle-support/tomcat-lifecycle:
-----> Downloading Tomcat Logging Support 2.4.0_RELEASE from https://download.run.pivotal.io/tomcat-logging-support/tomcat-logging-suppo:
-----> Downloading Tomcat Access Logging Support 2.4.0_RELEASE from https://download.run.pivotal.io/tomcat-access-logging-support/tomcat:
-----> Downloading GemFire 8.0.0 from https://download.run.pivotal.io/gem-fire/gem-fire-8.0.0.jar (3.4s)
-----> Downloading GemFire Logging API 1.5.8 from https://download.run.pivotal.io/slf4j-api/slf4j-api-1.5.8.jar (0.5s)
-----> Downloading GemFire Logging 1.5.8 from https://download.run.pivotal.io/slf4j-jdk14/slf4j-jdk14-1.5.8.jar (0.4s)
-----> Downloading GemFire Modules 8.0.0_1 from https://download.run.pivotal.io/gem-fire-modules/gem-fire-modules-8.0.0_1.jar (1.2s)
-----> Downloading GemFire Modules Tomcat7 8.0.0_1 from https://download.run.pivotal.io/gem-fire-modules-tomcat7/gem-fire-modules-tomcat:
-----> Downloading GemFire Security 8.0.0 from https://download.run.pivotal.io/gem-fire-security/gem-fire-security-8.0.0.jar (0.5s)
-----> Uploading droplet (89M)
0 of 2 instances running, 2 starting
0 of 2 instances running, 2 starting
0 of 2 instances running, 2 starting
0 of 2 instances running, 2 starting
2 of 2 instances running
App started
OK
App spring-petclinic was started using this command ``
Showing health and status for app spring-petclinic in org staging / space staging as admin...
OK
requested state: started
instances: 2/2
usage: 1G x 2 instances
urls: spring-petclinic.pandora.bboxes.cf-app.com
last uploaded: Mon Mar 16 20:16:03 +0000 2015
state      since      cpu      memory      disk
\#0  running  2015-03-16 01:25:23 PM  0.0%      697.5M of 1G  168.4M of 1G
\#1  running  2015-03-16 01:25:23 PM  0.0%      691.2M of 1G  168.4M of 1G
```

Step 5: Run the sample application in different web browsers

1. Return to the Pivotal Cloud Foundry® Developer Console and the **spring-petclinic** application page.
2. Locate the route of the web app.

APP

spring-petclinic

Start Stop Refresh

CONFIGURATION [Scale App](#)

INSTANCES	MEMORY LIMIT	DISK LIMIT
2	1 GB	1 GB

STATUS

#	STATUS	CPU	MEMORY	DISK	UPTIME
0	Running	0%	493 MB	137 MB	31 min
1	Running	0%	488 MB	137 MB	31 min

ABOUT

BUILDPACK [https://github.com/cloudfoundry/java-buildpac...](https://github.com/cloudfoundry/java-buildpack)

STACK Ubuntu 10.04 on x86-64

ROUTE <spring-petclinic.pandora.bboxes.cf-app.com>

Activity [Services](#) [Env Variables](#) [Routes](#) [Delete App](#)

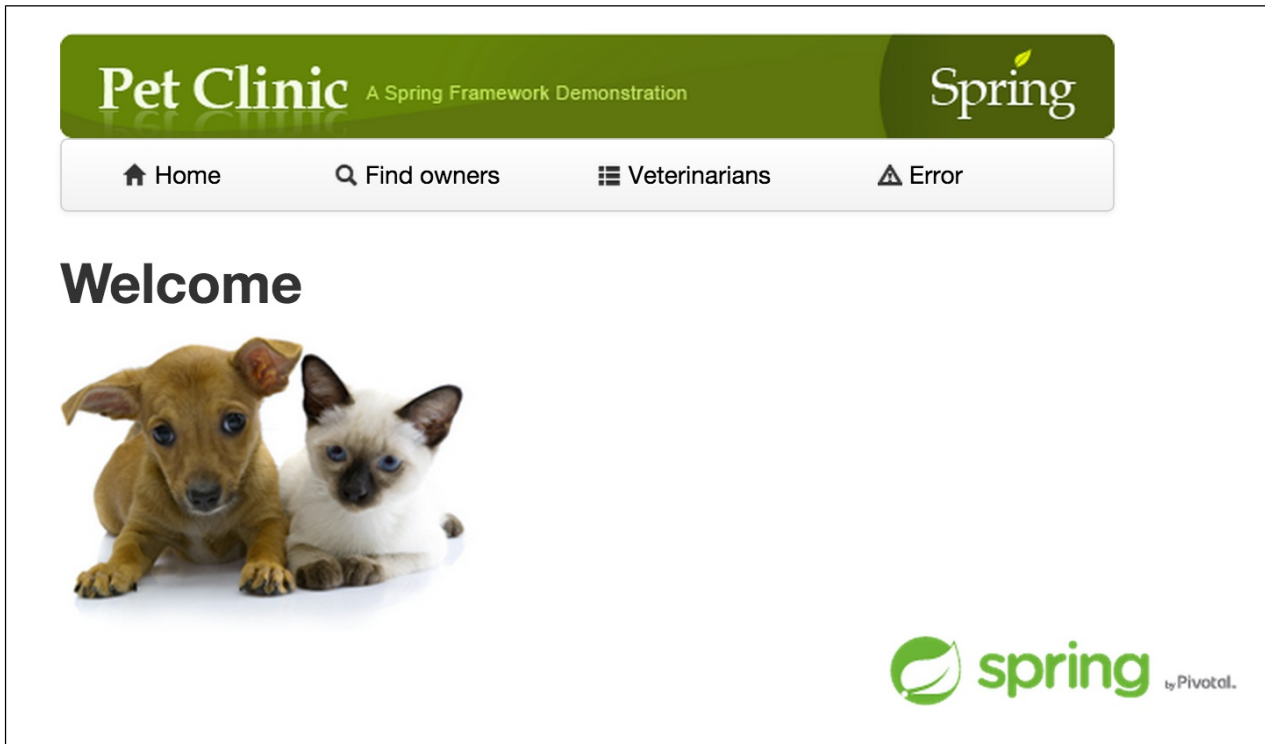
EVENTS

- started app** admin 02/10/2015 at 10:18 PM UTC
- updated app** admin

RECENT LOGS

```
2015-02-10 22:19:46 +0000 [App/1] OUT [CONTAINER] org.apache.catalina.startup.Catalina INFO Server startup in 17607 ms
2015-02-10 22:19:45 +0000 [App/1] OUT [CONTAINER] org.apache.tomcat.util.net.NioSelectorPool INFO Using a shared selector for servlet write/read
2015-02-10 22:19:45 +0000 [App/1] OUT [CONTAINER] org.apache.coyote.http11.Http11NioProtocol INFO
```

3. Click on this link to launch the web application in your browser. You should see the following web page appear:



4. Open this web application in multiple browsers. Click around the Spring PetClinic application. Using different browser clients, try adding data and searching for data.

Step 6: Query session replication data in the Pulse Monitoring tool

1. Return to the **Pivotal Cloud Foundry® Developer Console**.
2. On the **spring-petclinic** application page, click on the **Services** tab.
3. Under the list of **Services**, locate the **Manage** link for the Session State Caching service that you created in step

APP

spring-petclinic

CONFIGURATION [Scale App](#)

INSTANCES	MEMORY LIMIT	DISK LIMIT
2	1 GB	1 GB

STATUS

#	STATUS	CPU	MEMORY	DISK	UPTIME
0	Running	1%	694 MB	168 MB	7 min
1	Running	1%	697 MB	168 MB	7 min

ABOUT

BUILDPACK <https://github.com/cloudfoundry/java-buildpac...>

STACK Ubuntu 10.04 on x86-64

ROUTE spring-petclinic.pandora.bboxes.cf-app.com

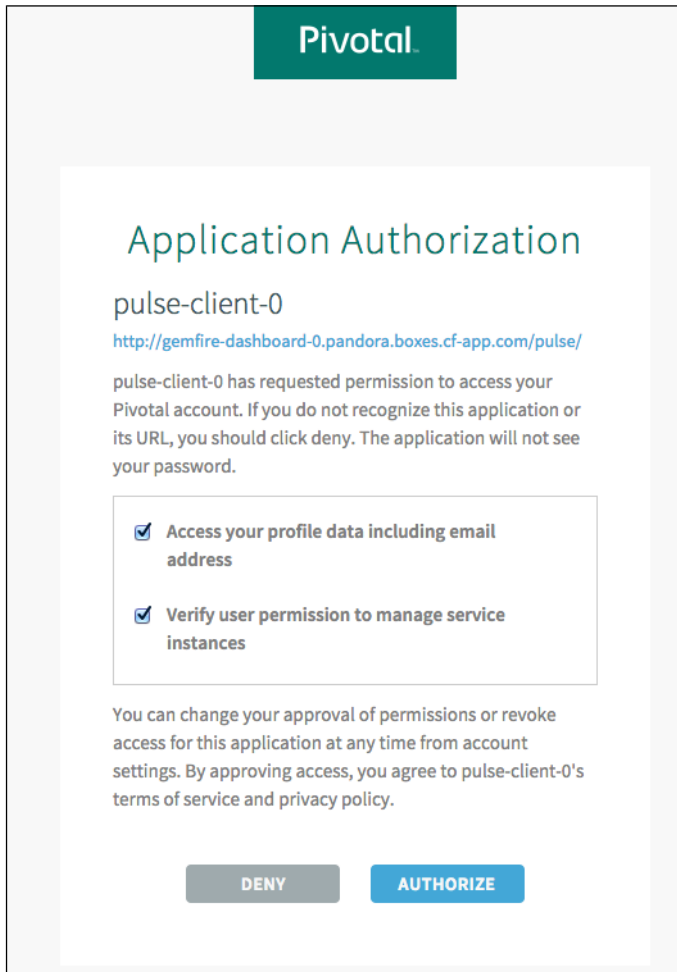
Activity **Services** **Env Variables** **Routes** [Delete App](#)

BOUND SERVICES [Bind a Service](#)

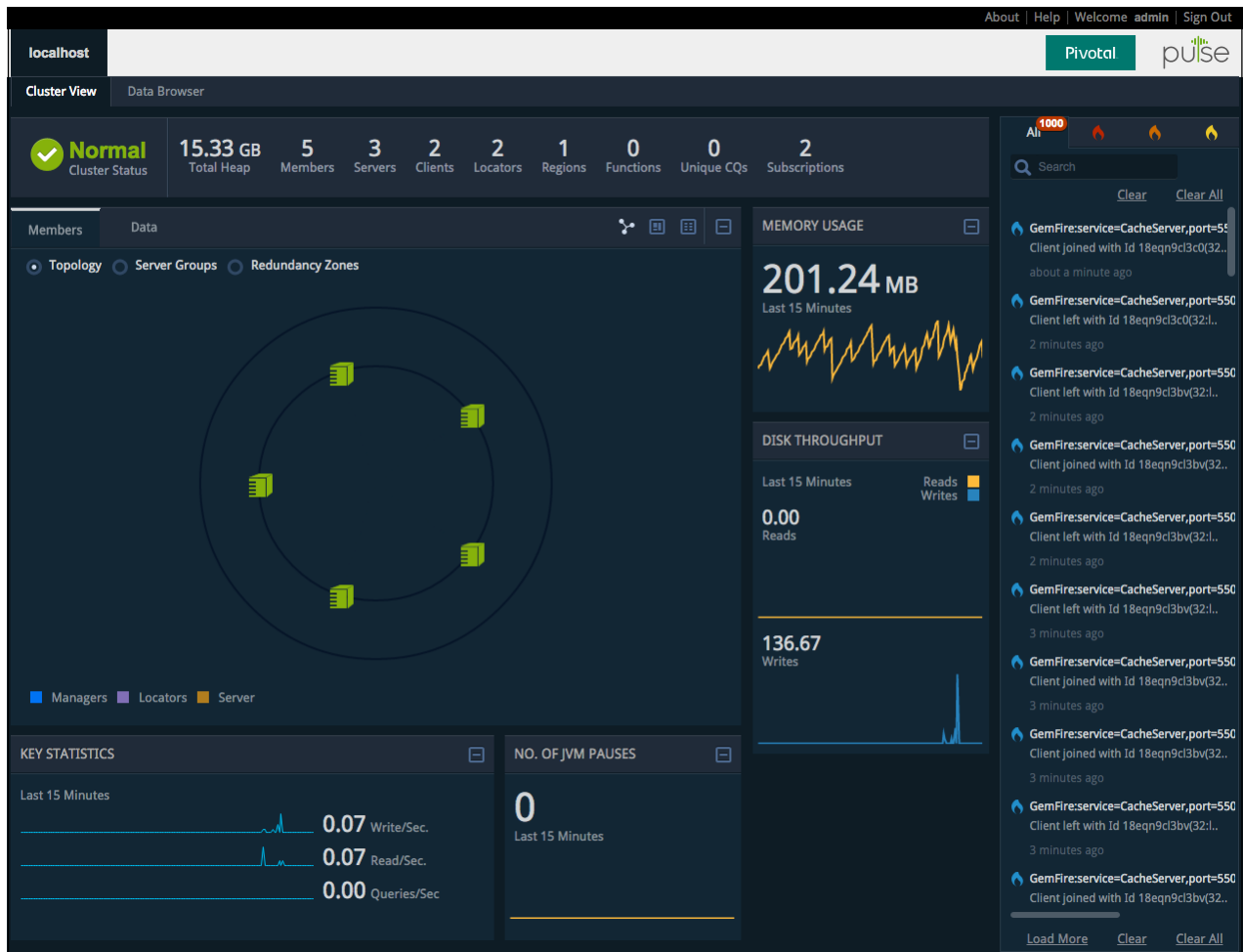
<p>my-ssc-gemfire-test</p> <p>Session State Caching Powered by Pivotal GemFire SessionReplication</p> <p>Show credentials</p>	Manage Support Docs	Unbind
--	---	------------------------

This link launches the [Pivotal GemFire Pulse Monitoring tool](#).

4. If you haven't done so already, select Authorize on the Application Authorization page for Pulse.

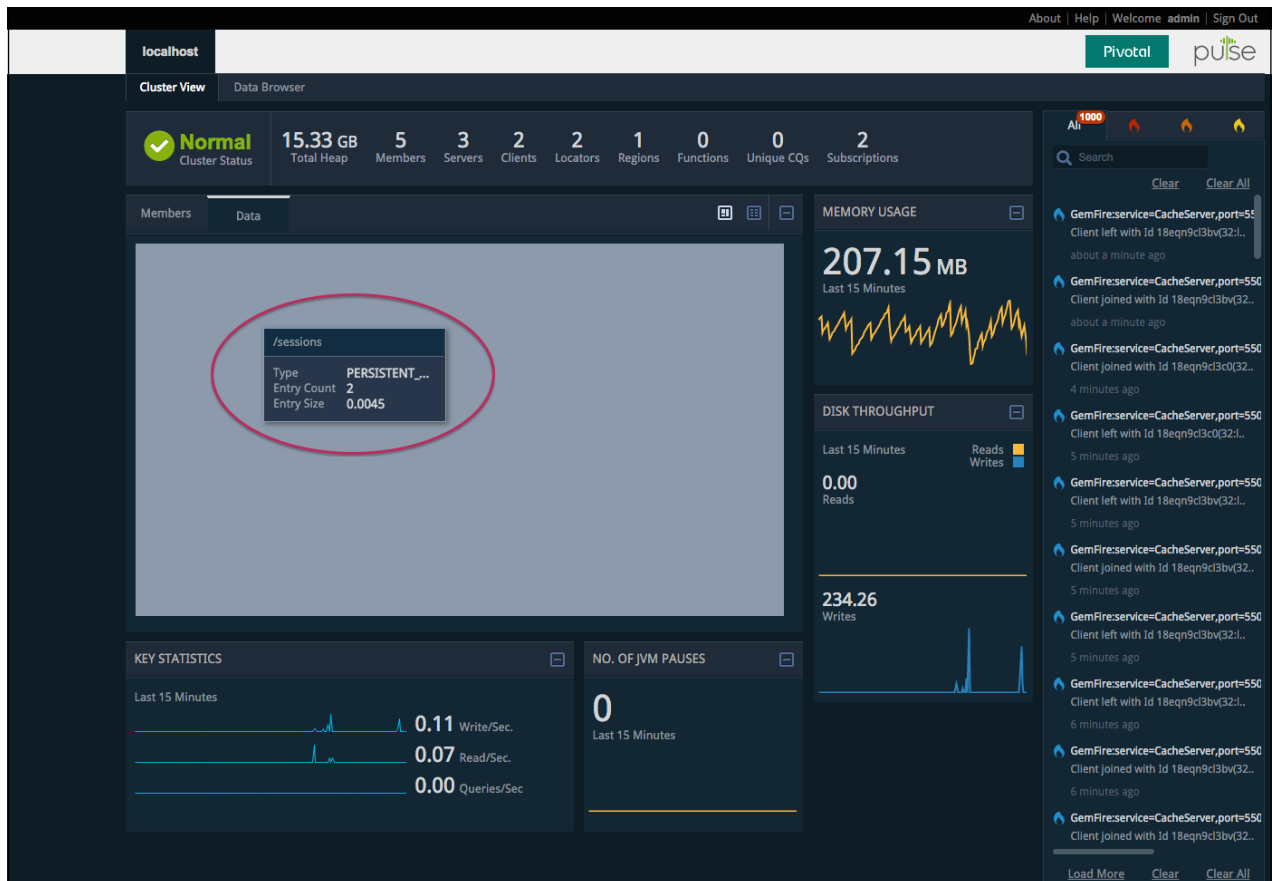


5. After authorizing single sign-on authorization, the Pulse Monitoring the tool launches in Cluster View mode.



Notice that the cluster has five members– three GemFire servers and two GemFire locators as defined for a Session State Caching service instance. The number of clients reflects the two instances of the **spring-petclinic** application bound to the service. If you scale the application in Cloud Foundry by increasing the number of application instances, the number of clients will increase accordingly. In addition, there is one region called **/sessions** which is created by default.

- Click on **Data** to view the default **/sessions** region. Mouse over the region space. A popup displays the entry count and size of the region.



7. Next, click on the **Data Browser** tab to query the region. Enter the following queries into the Query Editor box.

```
select count(*) from /sessions
```

This query returns the number of session records in the region. In this case, 2.

The screenshot shows the Pivotal Data Browser interface. The top navigation bar includes 'localhost', 'About', 'Help', 'Welcome', and 'Sign Out'. The main header has 'Pivotal' and 'pulse' logos. The left sidebar shows 'Cluster View' and 'Data Browser' tabs. The 'Data Browser' section is active, displaying 'DATA REGIONS' with a search bar and 'sessions' region. The 'QUERIES' section shows a 'QUERY EDITOR' with the query 'select count(*) from /sessions' circled in red. Below the editor is the 'RESULT' section showing a single result '2' of type 'java.lang.Integer'. The right sidebar shows a search bar and a list of events related to 'GemFire:service=CacheServer,port=550'.

Delete the existing query and enter in the following query:

```
select id from /sessions
```

This query returns a list of session ids found in the region.

The screenshot displays the Pivotal Data Browser interface. At the top, there's a navigation bar with 'localhost', 'Pivotal', and 'pulse' logos, along with links for 'About', 'Help', 'Welcome', and 'Sign Out'. Below this, the 'Data Browser' section is active, showing 'Cluster View' and 'Data Browser' tabs. The main area is divided into three panels: 'DATA REGIONS' on the left, 'QUERIES' in the center, and a log on the right.

In the 'DATA REGIONS' panel, there's a search bar 'Type a Name' and two tabs: 'All Regions' and 'Colocated Regions'. Under 'All Regions', a 'sessions' region is listed with a plus and minus icon.

The 'QUERIES' panel contains a 'QUERY EDITOR' with the text 'select id from /sessions' circled in red. Below the editor are 'Execute' and 'Clear' buttons. The 'RESULT' section shows a table with two columns: 'java.lang.String' and 'java.lang.String'. The first row contains the ID '6FA74F47EE08F56F96650762476C4D06', and the second row contains '9A96ED5C08EF9247D56AAE7C5E6C6118'. An 'Export Result' button is at the bottom right of the results section.

The rightmost panel shows a log with a search bar and 'Clear' and 'Clear All' buttons. The log entries are from 'GemFire:service=CacheServer,port=550' and include messages like 'Client joined with Id 18eqn9d3c0(32...' and 'Client left with Id 18eqn9d3bv(32...', with timestamps like '3 minutes ago' and '5 minutes ago'.

Troubleshooting

Service Installation Troubleshooting

Problem: Service deployment failed in Ops Manager

Verify that you have configured a ratio of 2 GemFire locators to 3 GemFire cache servers for the service. This ratio is fixed and cannot be changed.

Problem: Could not deploy a service instance in the Elastic Runtime

If you try to deploy (and allocate) more Session State Caching service instances than are available as defined by your [Pivotal Cloud Foundry®](#) (PCF) Administrator, you may receive the following error message:

```
500 ERROR. YOU HAVE REACHED THIS PAGE BECAUSE AN ERROR OCCURRED. PLEASE CONTACT YOUR OPERATOR
```

You need to delete existing service instances or ask the admin to deploy more instances.

Problem: Service plan tile does not show up in the Developer Console Marketplace

Verify that the register errand under lifecycle errands is selected in the Ops Manager tile configuration screen.

Application Troubleshooting

Problem: Application does not pick up changes to the service instance

Make sure you restage (or re-push) your application if you have made configuration changes to the underlying service instance.

Uninstalling Session State Caching Powered by GemFire

- [Prerequisites](#)
- [Uninstalling](#)

Prerequisites

Before you delete your **Session State Caching Powered by GemFire** tile, please ensure all GemFire SSC service instances have been deleted. If you fail to delete all GemFire service instances before deleting the tile, it may cause issues if you ever wish to re-install the GemFire tile.

To check if there are any GemFire SSC service instances present in your Pivotal Cloud Foundry deployment, you can run the simple shell script shown below:

```
#!/usr/bin/env bash

for org in $(cf orgs | awk 'NR>3'); do
  cf target -o $org > /dev/null
  for space in $(cf spaces | awk 'NR>3'); do
    cf target -s $space > /dev/null
    cf services | grep p-ssc-gemfire
  done
done
```

Uninstalling

See the following [instructions for uninstalling](#) [Pivotal Cloud Foundry](#) tiles.