

7 負載平衡

7.1 概述

可以將RStudio Server配置為在集群內的兩個或更多節點上負載平衡R會話。這既提高了容量，又提高了可用性。

請注意，RStudio Server的負載均衡具有一些特殊的“粘性”要求，這是由於用戶必須始終返回到工作所在的同一個R會話（即，他們的流量不能由多個節點處理）。因此，僅僅將多個RStudio服務器放置在傳統的硬件或軟件負載平衡器之上是不夠的 - 需要額外的智能和路由。

RStudio服務器負載平衡器的主要特性包括：

1. 多個主站實現高可用性 - 所有節點都可以將流量平衡到所有其他節點。
2. 支持多種負載平衡策略，包括最不繁忙的服務器（通過活動會話或系統負載），甚至由用戶分配，或基於外部腳本的自定義策略。
3. 在群集運行時添加和刪除節點的功能。
4. 獨立工作或可以與其他前端負載平衡環境集成。

7.2 配置

7.2.1 要求

RStudio集群中的節點有幾個要求：

1. 所有節點必須運行相同版本的RStudio Server Pro。
2. 服務器配置（即目錄的內容 `/etc/rstudio`）必須相同。
3. 用戶帳戶必須可以從每個節點訪問，用戶名和用戶ID必須在所有節點上相同。
4. 所有節點上的時鐘必須同步。
5. 用戶主目錄必須可以通過共享存儲訪問（例如，安裝相同NFS卷的所有節點）。

6. 要在負載平衡配置中使用“[項目共享](#)”功能，還必須定義顯式的服務器範圍的共享存儲路徑。
有關更多詳細信息，請參見[Shared Storage](#)部分。

7.2.2 定義節點

要定義一個集群節點，需要提供兩個配置文件：

```
/etc/rstudio/load-balancer  
/etc/rstudio/secure-cookie-key
```

其中第一個定義了可用節點和負載平衡策略。第二個定義了用於簽署cookie的共享密鑰（在單個節點配置中，該密鑰是自動生成的，但是需要多個節點的顯式協調）。

例如，要根據主動運行的R會話的數量來定義具有兩個負載均衡節點的集群，可以使用以下配置：

的 `/etc/rstudio/` 負載平衡器

```
[config]  
  
balancer = sessions  
  
[nodes]  
  
server1.example.com  
server2.example.com
```

在 `/etc/rstudio/` 安全cookie的密鑰

```
a55e5dc0-d6ae-11e3-9334-000c29635f71
```

安全Cookie密鑰只是一個唯一的值（在這種情況下是一個UUID）。請注意，此文件必須具有用戶讀/寫文件權限（即0600）以保護其內容不受其他用戶的影響。您可以使用該 `uuid` 實用程序創建一個安全的Cookie密鑰，如下所示：

```
sudo sh -c "echo `uuid` > /etc/rstudio/secure-cookie-key"  
sudo chmod 0600 /etc/rstudio/secure-cookie-key
```

此外，必須定義一個明確的服務器範圍的共享存儲路徑（這用於節點間同步）。該路徑在 `/etc/rstudio/rserver.conf` 文件中定義。例如：

```
/etc/rstudio/rserver.conf
```

```
server-shared-storage-path=/shared/rstudio-server/shared-storage
```

為了方便起見，此路徑通常位於用於共享主目錄存儲的相同卷上（例如路徑 `/home/rstudio-server/shared-storage`）。

7.2.3 文件鎖定

為了同步跨多個節點創建會話，RStudio Server使用跨節點鎖定方案。該方案依賴於所有正在同步的節點上的時鐘。RStudio Server包含一個 `locktester` 實用程序，您可以使用它來驗證文件鎖定是否正常工作。要使用 `locktester` 你應該登錄（例如通過SSH或telnet）到至少兩個節點使用相同的用戶帳戶，然後從兩個會話調用實用程序，如下所示：

```
$ /usr/lib/rstudio-server/bin/locktester
```

您從中執行實用程序的第一個節點應打印以下消息：

```
*** File Lock Acquired ***
```

消息打印後，進程將暫停，以便它可以保留該鎖（可以通過例如通過Ctrl + C中斷來釋放該鎖）。

您從中執行實用程序的第二個和後續節點應打印以下消息：

```
Unable to Acquire File Lock
```

如果您中斷了第一個節點（例如，通過Ctrl + C），則鎖定將被釋放，然後您可以從其他節點獲取該鎖定。

如果發生以下任一情況，則在使用負載平衡之前應解決的文件鎖定功能（或配置）存在問題：

1. 所有節點成功獲取文件鎖定（即多個節點可同時保存）。
2. 沒有節點能夠獲取文件鎖定。

如果上述任一條件成立，那麼RStudio將無法在整個集群中正確地同步R會話的創建（由於會話覆蓋其他狀態，可能會導致重複會話和丟失的數據）。

7.2.3.1 鎖定配置

RStudio的文件鎖定方案可以使用文件`at`進行配置 `/etc/rstudio/file-locks`。有效的條目是：

- `lock-type=[linkbased|advisory]`
- `refresh-rate=[seconds]`
- `timeout-interval=[seconds]`
- `enable-logging=[0|1]`
- `log-file=[path]`

默認的鎖定方案 `linkbased` 使用文件鎖定方案，當進程成功地將虛擬文件硬連接到RStudio用於客戶端狀態（通常 `~/.rstudio`）的文件夾內的位置時，鎖定被認為獲取。對於較舊的網絡文件系統，此方案通常更健壯，並且鎖應該在臨時文件系統安裝/卸載之後存活。

該 `timeout-interval` 和 `refresh-rate` 選項可以用來配置如何往往在生成的鎖 `linkbased` 鎖定方案被刷新和收穫。默認情況下，進程每隔20秒刷新其擁有的鎖，並且每30秒掃描一次陳舊的鎖。如果一個 `rsession` 進程崩潰，它可以留下陳舊的鎖定文件；這些鎖定文件將在任何新啟動的 `rsession` 進程到期後被清理。

`advisory` 可以選擇使用諮詢文件鎖（使用例如 `fcntl()` 或 `flock()`）。這些鎖強健，但不受所有網絡文件系統的支持。

如果您遇到文件鎖定問題，可以設置 `enable-logging=1`，並將該 `log-file` 選項設置為應寫入輸出的路徑。啟用日誌記錄時，RStudio將報告其獲取和釋放鎖定到由指定的日誌文件的嘗試 `log-file`。當 `log-file` 未設置時，日誌條目將被發送到系統日誌文件，通常位於 `/var/lib/messages` 或 `/var/lib/syslog`。

7.2.4 管理節點

7.2.4.1 啟動

創建配置文件後，應確保將這些文件（以及其中定義的所有其他配置 `/etc/rstudio`）複製到群集中的所有節點。假設服務器已經在每個節點上安裝並運行，您可以通過重新啟動服務器來應用負載平衡配置：

```
sudo rstudio-server restart
```

7.2.4.2當前狀態

一旦群集正在運行，您可以使用負載均衡狀態HTTP端點檢查它的狀態（哪些會話在哪裡運行）。例如，在默認端口（8787）上運行服務器時：

```
curl http://localhost:8787/load-balancer/status
```

請注意，狀態端點是使用本地主機而不是外部IP地址訪問的。這是因為這個端點被限制為只能在集群內訪問，所以需要直接從其中一個節點訪問。

7.2.4.3添加和刪除節點

要暫時從群集中刪除一個節點，您可以簡單地停止它：

```
sudo rstudio-server stop
```

運行在該節點上的R會話將自動移動到另一個活動節點。要恢復節點，您可以簡單地重新啟動它：

```
sudo rstudio-server start
```

請注意，添加和刪除節點不需要更改已定義節點的列表 `/etc/rstudio/load-balancer`（流量將自動路由到當前未運行的節點周圍）。

7.2.5故障排除

如果用戶在負載均衡配置中訪問RStudio服務器時遇到困難，很可能是由於其中一個負載平衡要求不被滿足。本節介紹幾種由於未滿足要求而可能發生故障的情況。

7.2.5.1用戶帳戶不同步

其中一個負載平衡要求是用戶帳戶必須可以從每個節點訪問，用戶名和用戶ID必須在所有節點上相同。如果用戶在不同節點上具有相同的用戶名，但具有不同的用戶ID，則當同一用戶嘗試使用不同的用戶標識訪問共享存儲時，將導致權限問題。

您可以通過 `id` 命令確定給定用戶名的ID。例如：

```
id -u jsmith
```

7.2.5.2 NFS卷的安裝問題

如果在RStudio會話期間卸載包含共享存儲的NFS卷，則會話將無法訪問。此外，卸載可能會導致文件鎖定丟失或損壞（請參閱下面的部分）。如果您遇到與訪問用戶目錄有關的問題，那麼完全重置RStudio節點和NFS之間的連接通常會解決這些問題。執行完全重置：

1. 在所有節點（ `sudo rstudio-server stop` ）上停止RStudio服務器。
2. 從所有節點完全卸載NFS卷。
3. 重新掛載所有節點上的NFS卷。
4. 在所有節點（ `sudo rstudio-server start` ）上重新啟動RStudio服務器。

7.2.5.3文件鎖定問題

共享用戶存儲（例如NFS）必須支持文件鎖定，以便RStudio Server可以同步訪問群集中各個節點的會話。如果群集中所有節點上的時鐘未同步，文件鎖定將無法正常工作。您可以按照上面的[文件鎖定](#)部分中的說明驗證文件鎖定是否正常工作。

7.3訪問和可用性

一旦你定義了一個集群，並把它聯機，你需要決定如何處理最終用戶的集群。有兩種截然不同的方法：

1. 單身主。為用戶提供其中一個節點的地址。該節點將根據需要自動將流量和會話路由到其他節點。這具有簡單的好處（不需要額外的軟件或硬件），但也導致單點故障。
2. 多個大師。將節點放在某種類型的將流量路由到它們的系統之後（例如動態DNS或軟件或硬件負載平衡器）。雖然這需要額外的配置，但它也使所有的節點成為彼此的故障轉移點。

下面詳細介紹這兩個選項。

7.3.1單身主

在“單主控”配置中，您將集群中的一個節點指定為主節點，並為最終用戶提供此節點的地址作為其訪問點。例如：

```
[nodes]
rstudio.example.com
rstudio2.example.com
rstudio3.example.com
```

用戶將使用** `http://rstudio.example.com` **訪問群集。根據主動負載平衡策略，該節點又將路由流量和會話給自己以及集群中的其他節點。

請注意，在此配置中，`rstudio2.example.com`和`rstudio3.example.com`節點可以隨時失敗或從群集中刪除，服務將繼續向用戶提供。但是，如果主節點出現故障或被刪除，那麼集群就會被有效地關閉。

7.3.2多個主人

在多主設備配置中，集群中的所有節點都是對等設備，並為對方提供故障切換。這要求某些外部系統（動態DNS或負載平衡器）將流量路由到節點。在這種情況下，任何節點都可能會失敗，服務將繼續（只要外部路由器可以智能地響應無法訪問的節點）。

例如，下面是一個Nginx反向代理配置，您可以使用上面定義的群集：

```
http {
    upstream rstudio-server {
        server rstudio1.example.com;
        server rstudio2.example.com backup;
        server rstudio3.example.com backup;
    }
    server {
        listen 80;
        location / {
            proxy_pass http://rstudio-server;
            proxy_redirect http://rstudio-server/ $scheme://$host/;
        }
    }
}
```

在這種情況下，Nginx軟件負載平衡器將在`rstudio.example.com`上運行，並將代理流量反向到`rstudio1.example.com`，`rstudio2.example.com`等。請注意，一個節點被指定為主節點，因此流量將通過該節點路由默認。但是，如果該節點失敗，那麼Nginx會自動使用備份節點。

這只是一個例子，因為有很多方法可以將流量路由到多個服務器，RStudio Server負載平衡被設計為與所有服務器兼容。

7.3.3使用SSL

如果您在公共網絡上運行RStudio服務器，則強烈建議使用SSL加密。如果沒有這個，所有的用戶會話數據都將被明確的發送出去，並被惡意方攔截。

建議的SSL配置取決於您部署的訪問拓撲：

1. 對於單主控部署，您可以按照[安全套接字（SSL）](#)部分中的說明，將集群的每個節點配置為使用SSL。然後節點將使用SSL進行外部和機器內通信。
2. 對於多主機部署，您可以在外部路由層（如上例中的Nginx服務器）中配置SSL，並為各個節點使用標準的未加密HTTP。您也可以選擇配置RStudio節點以使用SSL，但是如果與外部網絡的所有通信均通過外部路由層完成，則不是嚴格要求。

7.4平衡方法

有四種方法可用於在群集間平衡R會話。最適合的方法是安裝特定的，取決於用戶數量和他們創建的工作負載類型。

7.4.1會話

默認的平衡方法是 `sessions` 嘗試在集群節點間平均分配R會話：

```
[config]  
balancer = sessions
```

此方法將新的R會話分配給活動R會話數最少的節點。如果您希望用戶大部分具有相似的資源需求，這是一個不錯的選擇。

7.4.2系統負載

該 `system-load` 平衡方式根據可用節點的活動工作負荷的會話：


```
[config]
```

```
balancer = system-load
```

用於建立活動工作負載的度量標準是5分鐘的**平均負載**。如果您希望廣泛分散的CPU工作負載，並希望確保具有高CPU使用率的計算機不會收到新的會話，則這是一個不錯的選擇。

7.4.3用戶哈希

該 `user-hash` 平衡方法嘗試跨節點通過散列客戶的用戶名和均勻一致的分配負荷：

```
[config]
```

```
balancer = user-hash
```

所使用的散列算法是**CityHash**，它將產生相對均勻的用戶分佈到節點。如果您希望將用戶/會話分配給節點以保持穩定，那麼這是一個不錯的選擇。

7.4.4自定義

該 `custom` 平衡方法調出到外部腳本做出負載均衡決定：

```
[config]
```

```
balancer = custom
```

當 `custom` 指定時，RStudio Server將在需要選擇哪個節點啟動新會話時執行以下腳本：

```
/usr/lib/rstudio-server/bin/rserver-balancer
```

這個腳本將傳遞兩個環境變量：

`RSTUDIO_USERNAME` - 代表用戶或正在創建新R會話的用戶。

`RSTUDIO_NODES` - 逗號分隔的可用節點的IP地址和端口列表。

腳本應該返回節點，以使用標準輸出開始新的會話。請注意，返回的節點的格式應該與傳遞給腳本的格式相同（即包含IP地址和端口）。

