

# 8 審計和監測

## 8.1 審計配置

### 8.1.1 R控制台審計

RStudio Server可以選擇配置為通過將控制台輸入和輸出寫入中央位置（`/var/lib/rstudio-server/audit/r-console` 默認情況下為目錄）來審計所有R控制台活動。該功能可以使用該 `audit-r-console` 設置啟用。例如：

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

```
audit-r-console=input
```

這將審計所有R控制台輸入。如果你想記錄控制台輸入和輸出，那麼你可以使用該 `all` 設置。例如：

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

```
audit-r-console=all
```

請注意，如果您選擇記錄輸入和輸出，則與只記錄輸入相比，您需要更多的存儲空間。有關存儲要求和配置的更多討論，請參閱下面的“[存儲選項](#)”部分。

#### 8.1.1.1 數據格式

每個用戶的R控制台活動被寫入 `r-console` 數據目錄中的單個文件（默認情況下 `/var/lib/rstudio-server/audit/r-console`）。以下字段包含在內：

SESSION_ID	發生此操作的R會話的唯一標識符。
項目	如果操作發生在項目中，則指向RStudio項目目錄的路徑。
PID	發生此控制台操作的Unix進程ID。
用戶名	執行此控制台操作的Unix用戶。
時間戳	從時代開始，以毫秒為單位的動作時間戳。
類型	控制台操作類型（提示，輸入，輸出或錯誤）。
數據	與此操作相關的控制台數據（例如輸出文本）。

這個 `session_id` 字段是指多個R會話部分中描述的並發R 會話（即，它可以跨越多個項目和/或pid）。

日誌文件的默認格式是CSV（逗號分隔值）。也可以使用該選項將數據寫入Newline Delimited JSON `audit-r-console-format`。例如：

```
audit-r-console-format=json
```

請注意，使用JSON格式時，整個文件不是有效的JSON對象，而是每個單獨的行是一個。這遵循由包括R `jsonlite`包在內的多個庫支持的Newline分隔JSON規範。

### 8.1.1.2存儲選項

您可以自定義寫入審計數據的位置以及每個用戶要記錄的最大數據量（默認情況下為50 MB）。要指定審核數據的根目錄，請使用該 `audit-data-path` 設置。例如：

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

```
audit-data-path=/audit-data
```

請注意，此路徑影響R控制台審核和R會話審核數據的位置。

要指定要寫入單個用戶的R控制台日誌文件的最大數據量，請使用該 `audit-r-console-user-limit-mb` 設置。例如：

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

```
audit-r-console-user-limit-mb=100
```

默認最大R控制台日誌文件大小為每個用戶50兆字節。要配置可以寫入的文件的大小沒有限制，請將該值設置為 0 ，例如：

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

```
audit-r-console-user-limit-mb=0
```

請注意，審核日誌文件不會自動旋轉，因為它們變大。根據用戶數量及其活動級別，這意味著您應該創建一個計劃（例如cron）作業，以定期將文件從服務器移到輔助存儲上，和/或確保存儲的捲具有足夠的容量。

## 8.1.2 R會話審計

RStudio Server可以選擇配置為將與會話有關的事件（例如，登錄/註銷，會話開始/掛起/退出）的審計日誌寫入中央位置（ /var/lib/rstudio-server/audit/r-sessions 默認情況下，該目錄）。該功能可以使用該 `audit-r-sessions` 設置啟用。例如：

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

```
audit-r-sessions=1
```

### 8.1.2.1數據格式

R會話事件日誌默認寫入文件 `/var/lib/rstudio-server/audit/r-sessions/r-sessions.csv` 。以下字段包含在內：

PID	與事件關聯的Unix進程ID（對於auth事件，這將是主 <code>rserver</code> 進程，用於會話事件的 <code>rsession</code> 進程）。
用戶名	與事件關聯的Unix用戶。
時間戳	事件的時間戳（以毫秒為單位）。
類型	事件類型（請參閱下面有關事件類型的文檔）。
數據	發起事件的管理用戶（僅適用於admin事件和 <code>auth_login admin</code> 用戶登錄）。

以下值對於事件 `type` 字段有效：

auth_login	用戶登錄到RStudio服務器
auth_logout	用戶註銷RStudio服務器
auth_login_failed	用戶登錄嘗試失敗
在session_start	R會議開始
session_suicide	R會議因自殺而退出（內部錯誤）
session_suspend	R會話因暫停而退出
session_quit	R會話由於用戶退出而退出
session_exit	R會議退出
session_admin_suspend	管理員嘗試暫停R會話
session_admin_terminate	管理員嘗試終止R會話

日誌文件的默認格式是CSV（逗號分隔值）。也可以使用該選項將數據寫入Newline Delimited JSON `audit-r-sessions-format`。例如：

```
audit-r-sessions-format=json
```

請注意，使用JSON格式時，整個文件不是有效的JSON對象，而是每個單獨的行是一個。這遵循由包括R **jsonlite**包在內的多個庫支持的**Newline**分隔JSON規範。

### 8.1.2.2存儲選項

您可以自定義寫入審計數據的位置以及要記錄的最大R會話事件數據量（默認情況下為1 GB）。要指定審核數據的根目錄，請使用該 `audit-data-path` 設置。例如：

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

```
audit-data-path=/audit-data
```

請注意，此路徑影響R控制台審核和R會話審核數據的位置。

要指定R會話事件數據記錄的最大數量，請使用該 `audit-r-sessions-limit-mb` 設置。例如：

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

```
audit-r-sessions-limit-mb=2048
```

默認最大R會話事件日誌文件大小為1 GB（1024 MB）。要配置可以寫入的文件的大小沒有限制，請將該值設置為 0，例如：

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

```
audit-r-sessions-limit-mb=0
```

請注意，R會話事件日誌文件不會自動旋轉，因為它變得更大。這意味著您應該創建一個計劃（例如cron）作業，以定期將文件從服務器移動到輔助存儲器上，和/或確保存儲的捲具有足夠的容量。

在任何情況下，寫入R會話事件日誌文件的數據量都不大（每個會話少於1 KB），因此大量的會話事件可以存儲在默認的1 GB最大日誌文件大小內。

## 8.2監視配置

### 8.2.1系統和每個用戶資源

RStudio Server在每個用戶和系統範圍內監視資源（CPU，內存等）的使用情況。默認情況下，監控數據被寫入一組RRD（<http://oss.oetiker.ch/rrdtool/>）文件，可以使用Administrative Dashboard查看。

系統監控數據的存儲需要大約20MB的磁盤空間，用戶監控數據的存儲需要每個用戶大約3.5MB。這些數據默認保存在 `/var/lib/rstudio-server/monitor`。如果您擁有大量用戶，則可能希望指定一個備用卷來監視數據。您可以使用該 `monitor-data-path` 設置來執行此操作。例如：

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

```
monitor-data-path=/monitor-data
```

您也可能希望完全禁用RRD監視。您可以使用該 `monitor-rrd-enabled` 設置來執行此操作。例如：

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

```
monitor-rrd-enabled=0
```

請注意，在重新啟動服務器之前，對配置的更改不會生效。

## 8.2.2使用石墨

如果您正在管理多台服務器，將服務器監控數據發送到集中式數據庫和繪圖工具而非本地RRD文件可能會比較方便。您可以通過配置服務器將監控數據發送到Graphite（或與Carbon協議兼容的任何其他引擎）來執行此操作。這可以在RRD之外或者完全替代RRD來完成。

有四個設置可以控制與Graphite的交互：

啟用監控 - 石墨 -	將監控數據寫入Graphite（默認為 0 ）
顯示器石墨主機	運行Graphite的主機（默認為 127.0.0.1 ）
顯示器石墨端口	Port Graphite正在監聽（默認 2003 ）
顯示器 - 石墨 - 客戶端 ID	發件人的可選客戶端ID

例如，要使用默認的Graphite端口啟用遠程主機上的Graphite監控，您可以使用以下設置：

*/etc/rstudio/rserver.conf*

```
monitor-graphite-enabled=1
monitor-graphite-host=134.47.22.6
```

如果您正在使用像hosted graphite.com這樣的服務，要求您提供API密鑰作為報告指標的一部分，則可以使用該 monitor-graphite-client-id 設置。例如：

*/etc/rstudio/rserver.conf*

```
monitor-graphite-enabled=1
monitor-graphite-host=carbon.hostedgraphite.com
monitor-graphite-client-id=490662a4-1d8c-11e5-b06d-000c298f3d04
```

請注意，在重新啟動服務器之前，對配置的更改不會生效。

## 8.3服務器運行狀況檢查

## 8.3.1 啟用健康檢查

您可能希望定期輪詢RStudio服務器，以確保它仍然響應請求，並檢查服務器負載的各種指標。您可以使用該 `server-health-check-enabled` 設置啟用運行狀況檢查終端。例如：

*/etc/rstudio/rserver.conf*

```
server-health-check-enabled=1
```

重新啟動服務器後，以下運行狀況檢查端點將可用：

```
http://<server-address-and-port>/health-check
```

默認情況下，運行狀況檢查的輸出顯示如下：

```
active-sessions: 1
idle-seconds: 0
cpu-percent: 0.0
memory-percent: 64.2
swap-percent: 0.0
load-average: 4.1
```

## 8.3.2 定制響應

對運行狀況檢查的響應是通過處理包含多個變量的模板來確定的。默認模板是：

```
active-sessions: #active-sessions#
idle-seconds: #idle-seconds#
cpu-percent: #cpu-percent#
memory-percent: #memory-percent#
swap-percent: #swap-percent#
load-average: #load-average#
```

您可以自定義此模板以返回可由外部監視系統解析的備用格式（例如XML或JSON）。要做到這一點，你只需創建一個模板並將其複製到 `/etc/rstudio/health-check` 例如：

*在/ etc / rstudio /健康檢查*

```
<?xml version="1.0" encoding="UTF-8"?>
<health-check>
  <active-sessions>#active-sessions#</active-sessions>
  <idle-seconds>#idle-seconds#</idle-seconds>
  <cpu-percent>#cpu-percent#</cpu-percent>
  <memory-percent>#memory-percent#</memory-percent>
  <swap-percent>#swap-percent#</swap-percent>
  <load-average>#load-average#</load-average>
</health-check>
```

### 8.3.3更改URL

也可以自定義用於運行狀況檢查的URL。RStudio服務器將使用第一個文件的名字開始 `health-check` 在 `/etc/rstudio` 目錄中的模板，需要完整的文件名在URL中指定。例如，運行狀況檢查模板位於以下路徑：

```
/etc/rstudio/health-check-B64C900E
```

將使用此URL訪問：

```
http://<server-address-and-port>/health-check-B64C900E
```

請注意，直到服務器重新啟動後，運行狀況檢查模板的更改才會生效。