

5 R會議

5.1 R可執行文件和庫

5.1.1找到R

RStudio Server使用以下命令的輸出所指向的R版本：

```
$ which R
```

該 `which` 命令使用系統PATH執行對R可執行文件的搜索。因此，RStudio將默認綁定到從終端執行R時運行的相同版本。

對於系統軟件包管理器安裝的R版本，這將是 `/usr/lib/R`。對於從源代碼安裝的R版本，這通常（但不總是）`/usr/local/lib/R`。

如果您想覆蓋使用哪個版本的R，那麼您可以使用該 `rsession-which-r` 設置。例如：

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

```
rsession-which-r=/usr/local/bin/R
```

請注意，此更改在服務器重新啟動之前不會生效。

5.1.1.1使用多個版本的R

上面的部分描述了RStudio服務器如何定位R的全局默認版本。還可以按用戶或組指定R的備用版本。所述的R版本中部分描述此的更多細節。

5.1.2查找共享庫

您可以將元素添加到 `LD_LIBRARY_PATH` R會話的默認值（由R `ldpaths` 腳本確定），方法是 `rsession-ld-library-path` 向服務器配置文件添加一個條目。這對於確保程序包可以找到未安裝在系統標準庫路徑中的外部庫依賴關係很有用。例如：

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

```
rsession-ld-library-path=/opt/someapp/lib:/opt/anotherapp/lib
```

請注意，此更改在服務器重新啟動之前不會生效。

5.1.3 自定義會話啟動

5.1.3.1 配置文件腳本執行

RStudio服務器在bash登錄shell下啟動R會話。這意味著在執行R會話之前，bash shell會從這個文件中讀取和執行命令（如果存在的話）：

```
/etc/profile
```

讀取該文件後，它將查找以下文件，並從第一個文件中讀取並執行命令，這些命令是可讀的（需要注意的是，只有其中一個文件將被讀取和執行）：

```
~/.bash_profile
```

```
~/.bash_login
```

```
~/.profile
```

如果您有更多RStudio特定的初始化邏輯（導出環境變量等），您可以選擇在以下位置創建一個R會話特定的配置文件腳本：

```
/etc/rstudio/rsession-profile
```

如果存在，這個腳本將在啟動R會話的bash shell之前執行。

在某些情況下，您不會希望運行用戶shell配置文件腳本。這也是解決會話無法啟動的好方法，因為這可能表示由於在shell配置文件中設置了環境變量而導致發生衝突。要禁用shell配置文件的執行，請將該 `rsession-no-profile` 選項設置為 1 in `/etc/rstudio/rserver.conf`。例如：

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

```
rsession-no-profile=1
```

5.1.3.2環境變量

R會話繼承從上述配置文件腳本中顯式導出的環境變量。也可以 `LD_LIBRARY_PATH` 使用該 `rsession-ld-library-path` 選項將路徑附加到環境變量（有關詳細信息，請參閱上一節）。

環境變量的另一個來源是PAM會話。在Debian / Ubuntu系統上，由RStudio Server運行的默認PAM配置文件包含 `/etc/security/pam_env.conf` 和中定義的環境變量 `/etc/environment`。要了解有關使用PAM設置環境變量的更多信息，請參閱PAM會話部分以及此 `pam_env` 模塊中的文檔：http://linux.die.net/man/8/pam_env。

5.1.3.3計劃主管

您也可能希望在程序管理員下運行R會話，以修改其環境或可用資源。您可以使用該 `rsession-exec-command` 設置指定一個主管（以及控制其行為的參數）。例如：

/etc/rstudio/rserver.conf

```
rsession-exec-command=nice -n 10
```

本示例使用該 `nice` 命令以較低的調度優先級運行所有R會話。有關更多詳細信息，請參閱<http://linux.die.net/man/1/nice> `nice`。請注意，`nice` 特別是可以使用用戶和組配置文件完成相同的操作（甚至可以為每個用戶或組指定自定義優先級）。請參閱用戶和組配置文件部分了解更多詳細信息。

5.2用戶和組配置文件

用戶和組配置文件使您可以根據每個用戶或每個組定制R會話的行為。會話的以下屬性可以在配置文件中配置：

1. 使用R的版本
2. CPU親和力（即會話應該綁定到哪一組核心）
3. 調度優先級（即好值）
4. 資源限制（最大內存，進程，打開的文件等）
5. R會話超時（觸發會話掛起的空閒時間量）
6. R會話終止超時（觸發會話被刪除和清除的空閒時間量）

5.2.1創建配置文件

配置文件在文件中定義 `/etc/rstudio/profiles`。請注意，這個文件不是默認創建的，所以如果不存在的話，你需要創建它。配置文件分為三個不同類型的部分：

1. 全球（`[*]`）
2. 每組（`[@groupname]`）
3. 每用戶（`[username]`）

下面是一個示例配置文件，說明這些類型中的每一個：

的 `/etc/rstudio/` 簡檔

`[*]`

```
cpu-affinity = 1-4
max-processes = 100
max-memory-mb = 2048
session-timeout-minutes=60
session-timeout-kill-hours=24
```

`[@powerusers]`

```
cpu-affinity = 5-16
nice = -10
max-memory-mb = 4096
```

`[jsmith]`

```
r-version = /opt/R/3.1.0
session-timeout-minutes=360
```

此配置指定默認情況下，用戶將在核心1到4上運行，限制為100個進程和2GB虛擬內存。它還指定該 `powerusers` 組的成員將在核心5到16上以較高的優先級和4GB的內存限制運行。最後，用戶 `jsmith` 被配置為使用來自系統默認的R的不同版本。

請注意，`/etc/rstudio/profiles` 文件是從上到下進行處理的（即與文件後面出現的當前用戶相匹配的設置總是會覆蓋先前出現的文件）。`/etc/rstudio/profiles` 下面將更詳細地介紹可用的設置。

5.2.2 會話超時

要在掛起會話之前配置要等待的空閒時間，可以使用該 `session-timeout-minutes` 選項。例如：

```
session-timeout-minutes=360
```

如果沒有顯式指定默認值是120分鐘。

有一些情況下R會議將不會被暫停，其中包括：

1. 頂層R計算正在運行
2. 當R提示符不處於默認狀態時（例如在調試會話期間）

您也可以指定R會話永遠不應該被掛起通過設置 `session-timeout-minutes` 為零。例如：

```
session-timeout-minutes=0
```

5.2.3 會話超時終止

要配置在殺死和銷毀會話之前要等待的空閒時間，可以使用該 `session-timeout-kill-hours` 選項。這使您可以指定會話何時被閒置時自動清理，允許您自動回收會話使用的臨時磁盤空間，並停止其進程和子級。

例如：

```
session-timeout-kill-hours=48
```

如果沒有明確指定默認值為0小時，這意味著會話永遠不會被自動殺死和銷毀。提供的值應該是一個整數，表示會話在被殺之前可以空閒的小時數。

注意：如果會話沒有收到來自用戶的輸入，即使會話正在運行計算，會話也被認為是空閒的。這意味著在執行重要的計算時，會話可能會被破壞。會話被清理的用戶也將丟失所有未保存的代碼和數據。

5.2.4 CPU親和性和調度優先級

如果您的用戶或組持續需要比其他用戶更多的計算資源，則可以使用配置文件設置來保留CPU（`cpu-affinity`），並提高總體調度優先級（`nice`）。

5.2.4.1 CPU親和性

該 `cpu-affinity` 設置指定應使用多核系統上的哪些內核來安排會話的工作。這被指定為以逗號分隔的核心編號列表（從1開始），其中可以指定單個核心和核心範圍。例如：

```
cpu-affinity = 1,2,3,4
cpu-affinity = 1-4
cpu-affinity = 1,2,15-16
```

要確定係統中可尋址內核的數量，可以使用以下 `nproc` 命令：

```
$ nproc
```

5.2.4.2調度優先級

該 `nice` 設置為調度會話CPU時間指定相對優先級。負20是最高優先級，正20是最低優先級。進程的系統默認好處通常為0.以下是所有有效的好值：

```
nice = -10
nice = 0
nice = 15
```

調度程序的行為圍繞著不同的優先級而變化。欲了解更多詳情，請參見[不錯的使用和效果](#)。

5.2.5資源限制

配置文件也可用於指定可用內存限制以及最大進程數和打開文件數。

5.2.5.1可用內存

該 `max-memory-mb` 設置控制R會話的最大可尋址內存量（默認情況下，內存不受限制）。此示例指定了2GB的限制：

```
max-memory-mb = 2048
```

請注意，此值設置進程可以使用的虛擬內存量。虛擬內存包括由進程加載的代碼（即共享庫）以及內存映射文件等內容，所以即使對於一個香草R會話也可能經常消耗幾百兆字節。因此，您要確保不要將此閾值設置得太低（絕不應將其設置為1024以下）。

5.2.5.2進程數量

這些 `max-processes` 設置控制用戶可創建的最大進程數。這個設置對於防止無意的或者惡意的叉炸彈很有用。以下示例設置了200個進程的限制：

```
max-processes = 200
```

請注意，用戶需要能夠創建最小數量的進程才能使用RStudio Server，因此我們不建議將此值設置為低於25。

5.2.5.3打開文件的數量

在大多數Linux環境中，每個進程最多可以有1024個打開的文件。這通常是綽綽有餘的，但如果你有一個特定的應用程序需要更多的打開的文件，該 `max-open-files` 設置可以用來增加限制。例如：

```
max-open-files = 2048
```

5.2.6使用多個版本的R

如上所示，您可以將用戶或組綁定到服務器上安裝的不同版本的R。這由 `r-version` 選項控制。這裡有幾個使用的例子：

```
r-version = /usr/lib/R
r-version = /usr/local/lib/R
r-version = /opt/R/3.1.0
r-version = /opt/R/3.2.0
```

請注意，`r-version` 指定R安裝目錄的完整路徑。

有關在單個服務器上運行多個R版本的更多詳細信息，請參閱R版本章節。

5.3多個R會話

RStudio Server Professional使用戶能夠在單個服務器或負載平衡的服務器群集上進行多個並發R會話（RStudio Server的開源版本一次只支持一個會話）。

5.3.1 創建新的會話

您可以使用“會話”菜單中的“新建會話”命令（或IDE右上方對應的工具欄按鈕）啟動新的R會話。

您也可以使用“在新建會話中打開項目”命令在新的R會話中打開現有的RStudio項目。在切換項目時，項目菜單右側還有一個按鈕，可讓您指定應在新會話中打開項目，而不是在當前項目中打開項目。

您可以查看當前正在運行的所有會話，並使用IDE右上角附近的“會話”工具欄切換它們。

5.3.2 會話壽命

R會話是長時間運行的任務，直到您明確退出為止（您可以像在桌面上運行多個頂級RStudio窗口一樣來考慮它們）。這意味著您可以在一個會話中啟動長時間運行的作業，然後切換到另一個會話，稍後再重新訪問原始會話以檢查進度。也可以在桌面上，可以在會話中的不同項目和工作目錄之間導航。

會話將在閒置時自動掛起，然後在下次訪問時自動恢復。要永久退出會話，可以使用位於“文件”菜單上的“退出會話”命令或IDE右上方的相應工具欄按鈕。

5.3.3 禁用多個會話

如果您希望禁用多個會話的支持，則可以使用該 `server-multiple-sessions` 選項。例如：

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

```
server-multiple-sessions=0
```

5.4 PAM會話

RStudio Server Professional使用PAM（可插入認證模塊）進行用戶認證，以建立可用於R會話的環境和資源。這是使用PAM會話API完成的。PAM會話用於多種用途：

1. 初始化環境變量
2. 在對目錄服務器進行身份驗證後自動創建本地用戶。
3. 裝載遠程驅動器
4. 初始化和銷毀Kerberos票據

本節介紹如何使用RStudio Server配置和自定義PAM會話。

5.4.1會話PAM配置文件

對於PAM驗證，RStudio Server使用 `/etc/pam.d/other` 配置文件（Debian / Ubuntu）或 `/etc/pam.d/rstudio` 配置文件（RedHat / CentOS）。但是，為了啟動R會話，將使用不同的PAM配置文件。這是因為R會話的啟動可能與身份驗證不一致（例如，使用緩存在cookie中的登錄憑證或恢復暫停的會話返回站點）。因此，PAM配置文件中必須存在啟用 `root` 僅具有權限（`auth sufficient pam_rootok.so`）的身份驗證的PAM指令。

RStudio Server所要求的行為與 `su` 命令的行為基本相同（模擬沒有密碼的用戶）。因此默認情況下，RStudio Server使用該 `/etc/pam.d/su` 配置文件來運行R會話。

5.4.1.1創建自定義配置文件

該 `/etc/pam.d/su` 型材具有取決於你的Linux和本地配置的版本不同的默認行為。根據您想要與R會話關聯的行為類型（例如，掛載磁盤，設置環境變量，強制實施資源限制等），您可能需要為R會話創建自定義配置文件。例如，如果您想使用一個名為 `profile rstudio-session` 的配置文件，可以將其添加到配置文件中：

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

```
auth-pam-sessions-profile=rstudio-session
```

下面是自定義配置文件可能包含的內容，以便啟用PAM會話的一些常見功能（這是基於 `su` Ubuntu上默認配置文件的修改版本）：

```
/etc/pam.d/rstudio-session
```

```
# This allows root to su without passwords (this is required)
auth      sufficient pam_rootok.so

# This module parses environment configuration file(s)
# and also allows you to use an extended config
# file /etc/security/pam_env.conf.
# parsing /etc/environment needs "readenv=1"
session    required    pam_env.so readenv=1

# Locale variables are also kept into /etc/default/locale in etch
# reading this file *in addition to /etc/environment* does not hurt
session    required    pam_env.so readenv=1 envfile=/etc/default/locale

# Enforces user limits defined in /etc/security/limits.conf
session    required    pam_limits.so

# The standard Unix authentication modules
@include common-auth
@include common-account
@include common-session
```

5.4.1.2帶密碼的自定義配置文件

請注意，在上面的配置中，我們依靠 `pam_rootok.so` 啟用不帶密碼的身份驗證。這是必要的，因為RStudio服務器不保留在認證階段使用的密碼。

在某些情況下，密碼對於驗證不僅僅是重要的。PAM配置文件支持將 `use_first_pass` 身份驗證過程中使用的密碼轉發到其他模塊的指令（例如，要求使用Kerberos票據 `pam_krb5.so` 或使用加載或遠程驅動器掛載 `pam_mount.so`）。對於這些場景，RStudio Server在登錄後支持可選模式以保留密碼，然後將其轉發到PAM會話配置文件中。這是通過 `auth-pam-sessions-use-password` 設置啟用：

/etc/rstudio/rserver.conf

```
auth-pam-sessions-use-password=1
```

在這種情況下，您應該刪除該 `auth sufficient pam_rootok.so` 指令，並將其替換為您的環境中應用的任何認證指令。然後您可以使用該 `use_first_pass` 指令將密碼轉發到其他模塊。

例如，下面是一個非常簡單的RedHat / CentOS PAM配置文件，它使用系統默認認證並將密碼轉發到 `pam_mount.so` 模塊中。請注意，我們不再使用， `pam_rootok.so` 因為創建會話時密碼現在可用。

/etc/pam.d/rstudio-session

```
# Auth/account (use system auth and forward password to pam_mount)
auth    include    system-auth
auth    optional    pam_mount.so use_first_pass
account required    pam_unix.so

# Session (read environment variables and enforce limits)
session required    pam_env.so readenv=1
session required    pam_env.so readenv=1 envfile=/etc/default/locale
session required    pam_limits.so
```

請注意，此配置要求RStudio Server在內存中保留用戶密碼。這種保留是使用行業最佳實踐來保護敏感的內存數據，包括禁用ptrace和核心轉儲，使用mlock防止分頁到交換區域，並在釋放內存之前覆蓋內存中的內容。

5.4.1.3更多資源

如果您想了解有關PAM配置文件配置的更多信息，請參閱以下資源：

- http://www.linux-pam.org/Linux-PAM-html/Linux-PAM_SAG.html
- <http://linux.die.net/man/8/pam.d>
- <http://www.linuxjournal.com/article/2120>
- <http://www.informit.com/articles/article.aspx?p=20968>

5.4.2 PAM會話清理

默認情況下，RStudio Server在其關聯的R進程退出時不會關閉PAM會話。這是因為PAM會話經常會初始化和維護需要更多持久性的資源，即單個R會話（例如裝入的驅動器，Kerberos票據等）的生命週期。如果用戶有多個活動R會話，則關閉與其中一個會話關聯的PAM會話可能會卸載驅動器或撤銷另一個R會話仍然需要的故障單。

但是可以通過強制掛起與R會話關聯的PAM會話來手動關閉該會話。這可以通過以下兩種方式之一來完成：

1. 通過按**管理儀表板**的“會話”頁面上的“掛起”按鈕。
2. 通過執行**暫停會話**中所述的命令 `force-suspend` 或 `force-suspend-all` 命令。

如果您希望在關聯的R會話退出時關閉PAM會話，則可以使用該 `auth-pam-sessions-close` 設置。例如：

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

```
auth-pam-sessions-close=1
```

請注意，如果指定此設置，請注意，根據PAM會話管理的資源，它可能與運行多個並發R會話的用戶不兼容（例如，因為可能會從正在運行的會話下卸下某個驅動器）。在這種情況下，您可能希望禁用對多個R會話的支持（有關如何執行此操作的詳細信息，請參閱[Multiple R Sessions](#)部分）。

5.4.3 禁用PAM會話

如果您不希望RStudio Server使用PAM會話，則可以使用該 `auth-pam-sessions-enabled` 設置禁用此功能。例如：

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

```
auth-pam-sessions-enabled=0
```

5.5 Kerberos

您可以使用PAM會話安排Kerberos票據供R會話使用。這是使用[pam_krb5](#) PAM模塊完成的。請注意，您可能需要單獨安裝此模塊，具體取決於您正在運行的Linux發行版/版本。

5.5.1 配置

注意：在嘗試修改您的配置以支持Kerberos之前，您應該確保了解[PAM Sessions](#)的上一節。

以下是 `pam_krb5` 您需要添加到RStudio PAM配置文件的配置指令的簡單示例。請注意，`pam_krb5` 支持大量的選項，其中一些可能需要Kerberos在您的環境中正常工作。在繼續確保您已正確指定所有選項之前，您應該查閱[pam_krb5文檔](#)。

RStudio的主要PAM配置文件應修改為包含以下 `pam_krb5` 指令：

/etc/pam.d/rstudio

```
auth      sufficient      pam_krb5.so debug
account   required        pam_krb5.so debug
session   requisite       pam_krb5.so debug
```

除了修改主PAM配置文件之外，還需要為RStudio創建自定義PAM會話配置文件（如[創建自定義配置文件中所述](#)）。這需要 `pam_krb5` 根據您的本地Kerberos配置包含適當的指令。例如：

/etc/pam.d/rstudio-session

```
auth      required        pam_krb5.so debug
account   [default=bad success=ok user_unknown=ignore] pam_krb5.so debug
password  sufficient      pam_krb5.so use_authtok debug
session   requisite       pam_krb5.so debug
```

請注意，通常當您創建自定義PAM會話配置文件時，您將包含該 `auth sufficient pam_rootok.so` 指令。但是，在配置Kerberos身份驗證的情況下，您不需要此指令，而是需要指定Kerberos使用顯式密碼完成身份驗證，如上例所示。

要確保RStudio Server使用自定義PAM會話配置文件，並且PAM密碼正確轉發給 `pam_krb5` 您，還需要將以下條目添加到 `rserver.conf` 配置文件中：

/etc/rstudio/rserver.conf

```
auth-pam-sessions-profile=rstudio-session
auth-pam-sessions-use-password=1
```

關於配置的一些附加說明：

- 將 `debug` 在PAM配置文件的動作是不需要然而，因為它使故障排除更加簡單，我們建議將它添加。
- 以上示例並不是PAM配置文件內容的完整示例，而是 `pam_krb5` 需要顯示的條目的示例。您當地的環境可能有許多額外的條目，你應該確保也包括在必要的。

在進行配置更改後，應確保暫停激活的R會話並重新啟動RStudio Server，以確保正在使用新設置。你可以這樣做，如下所示：

```
sudo rstudio-server force-suspend-all
sudo rstudio-server restart
```

5.5.2測試和故障排除

在進行所需的配置更改之後，應使用 `pamtester` 實用程序從RStudio Server隔離測試更新的PAM配置，如[診斷PAM身份驗證問題中所述](#)。以下命令將測試身份驗證以及Kerberos票證的頒發：

```
sudo /usr/lib/rstudio-server/bin/pamtester --verbose \
rstudio-session <user> authenticate setcred open_session
```

請注意，您應該替換實際的本地用戶名作為 `<user>` 命令行的一部分。

PAM配置和Kerberos配置的具體細節可能因環境而異。因此，正確的配置可能需要本指南無法涵蓋的額外條目和選項。請參閱[PAM資源]中鏈接的文檔以及[pam_krb5](#)獲取更多詳細信息。

5.6工作目錄

新R會話和新R項目的默認工作目錄是用戶的主目錄（`~`）。您可以通過更改此行為 `session-default-working-dir`，並 `session-default-new-project-dir` 在內部配置參數 `rsession.conf` 的配置文件。

例如，將默認值設置為“`~/ working`”和“`~/ projects`”，你會使用下面的配置：

/etc/rstudio/rsession.conf

```
session-default-working-dir=~/working
session-default-new-project-dir=~/projects
```

您應該確保用戶具有寫入指定的默認目錄所需的權限。如果指定的目錄不存在，將自動創建。

請注意，這些設置僅控制默認的工作目錄和新的項目目錄（如果用戶選擇，這些目錄仍然可以在本地覆蓋這些設置）。

5.7 工作區管理

5.7.1 默認保存操作

當用戶退出R會話時，他們需要選擇是否保存他們的R工作區（即 `.RData` 文件）。RStudio具有全局和每個項目設置，用於控制工作區未保存更改時發生的情況。可能的值是：

- `ask` - 詢問是否保存工作區文件
- `yes` - 始終保存工作區文件
- `no` - 永遠不要保存工作區文件

The default global setting is `ask` and the default project-level setting is derived from the current global setting (these options can be modified by end users via the *Global Options* and *Project Options* dialogs respectively).

The default global setting can also be changed via the `session-save-action-default` configuration parameter in the `rsession.conf` config file. For example, to change the default value to `no` you would use this:

```
/etc/rstudio/rsession.conf
```

```
session-save-action-default=no
```

Note that this setting is specified in the `rsession.conf` config file and takes effect the next time a user launches an R session (rather than requiring a full restart of the server).

5.7.2 Suspend and Resume

When R sessions have been idle (no processing or user interaction) for a specified period of time (2 hours by default) RStudio Server suspends them to disk to free up server resources. When the user next interacts with their session it is restored from disk and the user resumes right back where they left off. This is all done seamlessly such that users aren't typically aware that a suspend and resume has occurred.

5.7.2.1 Session Timeout

To configure the amount of idle time to wait before suspending sessions you can use the `session-timeout-minutes` setting in the `/etc/rstudio/rsession.conf` file. For example:

```
/etc/rstudio/rsession.conf
```

```
session-timeout-minutes=360
```

The default value if none is explicitly specified is 120 minutes.

Important note: this setting and a few others discussed in this section are specified in the `/etc/rstudio/rsession.conf` file (rather than the `rserver.conf` file previously referenced).

There are some conditions where an R session will not be suspended, these include:

1. When a top-level R computation is running
2. When the R prompt is not in it's default state (e.g. during a debugging session)

You can also specify that R sessions should never be suspended by setting the `session-timeout-minutes` to zero. For example:

```
/etc/rstudio/rsession.conf
```

```
session-timeout-minutes=0
```

You can also set session timeouts on a per-user or per-group basis, see the [User and Group Profiles](#) section for details.

5.7.2.2 Forcing Suspend

You can force the suspend of individual sessions or even all sessions on the server. You can do this directly from the main page of the [Administrative Dashboard](#) or from the system shell as follows:

```
$ sudo rstudio-server force-suspend-session <pid>
$ sudo rstudio-server force-suspend-all
```

5.7.2.3 Resume and .Rprofile

By default the `Rprofile.site` and `.Rprofile` files are not re-run when a session is resumed (it's presumed that all of their side-effects are accounted for by simply restoring loaded packages, options, environment variables, etc.).

在某些配置中，可能需要強制重新執行配置文件。有一個最終用戶選項可以控制常規選項窗格上的默認設置為false。但是，服務器管理員可能希望確保此選項默認為true。要做到這一點，你使用該 `session-rprofile-on-resume-default` 選項。例如：

/etc/rstudio/rsession.conf

```
session-rprofile-on-resume-default=1
```

請注意，此設置在 `rsession.conf` 配置文件中指定，並在用戶下次啟動R會話時生效（而不需要完全重新啟動服務器）。

5.7.2.4子進程

默認情況下，當會話退出或掛起時，會話中創建的子進程將繼續運行。您可以通過指定中的 `session-quit-child-processes-on-exit` 設置來指定是否應該發生 `/etc/rstudio/rsession.conf`。允許的值 1 或 0 分別退出子進程或讓它們運行。

例如，在會話退出時要退出子進程：

/etc/rstudio/rsession.conf

```
session-quit-child-processes-on-exit=1
```

請注意，此設置在 `rsession.conf` 配置文件中指定，並在用戶下次啟動R會話時生效（而不需要完全重新啟動服務器）。而且，用戶可以在他們的項目設置中特別覆蓋這個設置。

5.7.2.5會話超時終止

To configure the amount of idle time to wait before killing and destroying sessions you can use the `session-timeout-kill-hours` option in the `/etc/rstudio/rsession.conf` file. This allows you to specify when a session should automatically be cleaned up when it has been idled, allowing you to automatically reclaim temporary disk space used by the sessions, and to stop their processes and children.

For example:

/etc/rstudio/rsession.conf

```
session-timeout-kill-hours=48
```

The default value if none is explicitly specified is 0 hours, meaning sessions will never be killed and destroyed automatically. The supplied value should be an integer representing the amount of hours a session can be idle before being killed.

NOTE: A session is considered to be idle if it has not received input from the user even if the session is in the process of running a computation. This means that sessions can be destroyed while important computations are executing. The user whose session is cleaned will also lose all unsaved code and data.

5.7.3 Workspace Storage

Storage of workspaces (.RData files) in RStudio Server does not use compression by default. This differs from the behavior of base R. Compression is disabled because we've observed that for larger workspaces (> 50MB) compression can result in much lower performance for session startup and suspend/resume (on the order of 3 or 4 times slower).

The default workspace save options under RStudio Server are as follows:

```
options(save.defaults=list(ascii=FALSE, compress=FALSE))
options(save.image.defaults=list(ascii=FALSE, safe=TRUE, compress=FALSE))
```

If you wish to use different defaults you can define the `save.defaults` and/or `save.image.defaults` options in your `Rprofile.site` or per-user `.Rprofile` and RStudio Server will respect the settings you specify rather than using its own defaults.

See <https://stat.ethz.ch/R-manual/R-devel/library/base/html/save.html> for additional details on how R saves objects and the storage and performance implications of using compression.

5.8 First Project Template

5.8.1 Overview

RStudio allows you to specify a first project to automatically open for first time users of the system. To do this, set the `session-first-project-template-path` parameter in `rsession.conf` to an RStudio project directory. This directory will be copied into the user's home directory upon first running the IDE, and will automatically open the project contained within. For example:

```
/etc/rstudio/rsession.conf
```

```
session-first-project-template-path=/etc/rstudio/welcome-project
```

In the example above, the project located within `/etc/rstudio/welcome-project` would be copied to users' home directories when first starting RStudio, and the project file `welcome-project.Rproj` would be run.

The project file must be named the same as the directory it is contained in. For the above example, the project file must be named `welcome-project.Rproj`. In addition, you must ensure that the project directory is fully readable and executable by your users, as they will be copying the contents of the directory into their home directory.

If you copy an existing project to be used as a project template, ensure that you delete the RStudio metadata folders and files contained within the project directory. You will want to ensure you delete the following:

```
.Rproj.user  .Rhistory  .RData
```

If you are creating the project template for the first time, the project (`.Rproj`) file must contain the version specifier at a minimum. For example:

```
welcome-project.Rproj
```

```
Version: 1.0
```

5.8.2 Project DefaultOpenDocs

項目文件允許您指定首次打開項目時應打開的默認文檔。例如，您可以讓您的歡迎項目提供說明文件來幫助指導您的用戶。為此，請將該 `DefaultOpenDocs` 行添加到 `.Rproj` 文件。例如：

```
歡迎-project.Rproj
```

```
Version: 1.0
```

```
DefaultOpenDocs: welcome.txt:firstMarkdown.Rmd
```

這些 `DefaultOpenDocs` 參數指定了在項目首次打開時自動打開的文件列表，由 `:` 字符分隔。這些文件是項目目錄的相對路徑。只有包含在項目目錄中的文件才能打開。

5.9 項目共享

5.9.1 概述

項目共享是RStudio Server Pro的一項功能，使用戶能夠在RStudio項目上一起工作。啟用後，項目所有者可以選擇任意數量的其他RStudio Server用戶作為項目協作者。RStudio服務器管理項目中文件的權限，以確保所有協作者都可以訪問它們，使多個協作者能夠同時編輯文件，並讓協作者看到還有誰在項目中與他們一起工作。

5.9.2 先決條件

5.9.2.1 訪問控制列表

要使用“項目共享”，託管要共享項目的目錄必須位於支持訪問控制列表（ACL）的捲上。RStudio服務器使用ACL來授予協作者對共享項目的訪問權限；普通的文件權限不被修改。

Instructions for enabling ACLs vary by Linux distribution and filesystem type (see the [Guide to enabling ACLs on Ubuntu or RedHat](#), for example). Broadly, you will need to ensure that the filesystem is mounted with the `user_xattr` and `acl` attributes, and modify `/etc/fstab` if necessary to persist the attributes.

Note that many Linux distributions now have ACLs enabled by default in which case no special configuration is required. You can use the `tune2fs` command to inspect the attributes with which your filesystem is mounted (`user_xattr` and `acl` are required for project sharing).

5.9.2.2 Project Sharing and NFS

If you plan to use Project Sharing with NFS-mounted volumes, there are several caveats you should be aware of.

1. We recommend mounting NFS with the `noac` mount option. Without this option, NFS caches file attributes, so it may not be possible for users working simultaneously in a project to know whether they're seeing each others' latest changes. The `noac` option does reduce performance, however, so we recommend testing to choose the right trade off for your environment.

2. Some features which automatically update when directory contents change will not update on NFS. For instance, users may need to manually refresh the Files pane to see new files added by collaborators.
3. ACL support on NFS is more complicated than ACL support on local file systems. Read [ACLs on Linux NFS](#) for a discussion of the issues; guidance for configuring specific NFS versions is provided below.

5.9.2.2.1 Determining NFS Version

Project Sharing works with versions 3 and 4 of the NFS protocol. However, some additional configuration may be necessary depending on the NFS version and underlying filesystem. To determine your NFS client version, run the following command on your RStudio Server Pro instance:

```
$ nfsstat -m
```

You should see `vers=3.0` in the output if you're using NFSv3, and `vers=4.0` if you're using NFSv4. Extra RStudio configuration is required for NFSv4 clients (see below).

5.9.2.2.2 NFSv3

When using version 3 of the NFS protocol, the volume must be mounted with the `acl` mount option. On most systems this is the default, so you need only ensure that `noacl` is not present. It's also necessary for the file system on the NFS server to itself be appropriately configured for ACL support; see the section above on Access Control Lists for guidance. Not all file servers that support the NFSv3 protocol also support POSIX compatible ACLs.

To test for POSIX compatible ACLs, you can use the `setfacl` command as follows:

```
setfacl -m u:user2:x /home/user1/project
```

where `/home/user1/project` is the full path to a directory owned by `user1`, and `user2` is another valid user on the system.

5.9.2.2.3 NFSv4

Version 4 of the NFS protocol uses a very different permissions and ACL model from Version 3. Not all NFSv4 servers support or expose the NFSv4 ACL model, so check with the server administrator to determine whether the capability exists and/or can be enabled.

NFSv4 ACLs differ from NFSv3 ACLs in that they associate a *domain* with each user named in the access control list. This is typically the same as the domain part of the machine's host name, but can be any string that the NFSv4 client and server agree on. On Linux systems, the domain can be set in `/etc/idmapd.conf`.

In order to use NFSv4 with Project Sharing, you must tell RStudio the NFSv4 domain you want to use, as follows:

/etc/rstudio/rsession.conf

```
nfs4-domain=mydomain.com
```

This setting both enables NFSv4 support and sets the domain. If this setting is not present, RStudio will only use NFSv3 ACLs.

To test NFSv4 ACL support, you can use the `nfs4_setfacl` command as follows:

```
nfs4_setfacl -a A::user2@domain:rax /home/user1/project
```

where `/home/user1/project` is the full path to a directory owned by `user1`, `user2` is another valid user on the system, and `domain` is your system's user domain.

5.9.2.3 LDAP and sssd

If you're using `sssd` with LDAP, you may need to enable user enumeration so that RStudio Server Pro can search the directory to create a list of users you can share a project with. To do this, set the following in your `sssd.conf` file:

```
[domain/<domainname>]
enumerate = true
```

或者，`auth-required-user-group` 可以使用該設置。此設置允許RStudio Server Pro僅枚舉指定組的成員，而不是整個用戶目錄。因此，如果您無法在LDAP提供程序上啟用用戶枚舉，則可以創建一個包含所有RStudio用戶的組，並將其作為提供 `auth-required-user-group`。

您可以在[sssd FAQ](#)中閱讀更多關於用戶枚舉的內容。

5.9.2.4 共享存儲

要使用“項目共享”，必須指定一個目錄，服務器上的所有用戶都可以讀取和寫入該目錄。在單服務器安裝中，RStudio默認使用此位置：

```
/var/lib/rstudio-server/shared-storage
```

In a load-balanced configuration, however, RStudio does not provide a default, so it is necessary to provision a path both visible to and writeable by all users on the system (typically on the same filesystem on which home directories are mounted). This path can be supplied to RStudio Server via the `server-shared-storage-path` option, for example:

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

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

The `server-shared-storage-path` option (described above) configures the path used for shared project storage. Note that this storage contains only *links* to shared projects, not the projects themselves, so requires a very small amount of physical storage.

5.9.2.5 Proxy Settings

If you are running RStudio Server with a proxy, you'll need to make sure that your proxy is correctly configured to pass websocket connections through in order for all Project Sharing features to work. See the [Running with a Proxy](#) section for more on this.

5.9.3 Disabling Project Sharing

Project Sharing is enabled by default however you can disable it using the `server-project-sharing` option, for example:

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

```
server-project-sharing=0
```

5.10 Package Installation

You can customize the location of user packages installed from CRAN as well as the default CRAN repository. You can also configure the user-interface of the RStudio IDE to discourage end-user package installation in the case where packages are deployed centrally to a site library.

Important note: The settings discussed in this section are specified in the `/etc/rstudio/rsession.conf` file (rather than the `rserver.conf` file previously referenced).

5.10.1 User Library

By default R packages are installed into a user-specific library based on the contents of the `R_LIBS_USER` environment variable (more details on this mechanism are here: <http://stat.ethz.ch/R-manual/R-devel/library/base/html/libPaths.html>).

It's also possible to configure an alternative default for user package installation using the `r-libs-user` setting. For example:

/etc/rstudio/rsession.conf

```
r-libs-user=~/.R/library
```

建立一個替代的默認用戶庫路徑的一個好處是，通過這樣做，您可以刪除軟件包庫路徑的R版本組件（默認路徑包含）。這樣可以升級服務器上主要版本的R，並讓用戶的軟件包繼續工作。

5.10.2 阻止用戶安裝

可能是因為您已經為所有用戶共享的網站包庫配置了RStudio Server。在這種情況下，您可能希望阻止用戶通過從RStudio IDE中刪除軟件包安裝UI來安裝自己的軟件包。要做到這一點你使用的 `allow-package-installation` 設置。例如：

/etc/rstudio/rsession.conf

```
allow-package-installation=0
```

請注意，此設置僅通過刪除用戶界面元素來阻止軟件包安裝。用戶仍然可以使用該 `utils::install.packages` 功能直接安裝軟件包。

5.10.3 CRAN存儲庫

RStudio Server 默認使用RStudio CRAN鏡像 (<https://cran.rstudio.com>)。此鏡像使用Amazon S3存儲進行全局分發，因此應為所有語言環境提供良好的性能。但是，您可能希望覆蓋默認的CRAN鏡像。這可以通過 `r-cran-repos` 設置完成。例如：

```
/etc/rstudio/rsession.conf
```

```
r-cran-repos=http://cran.at.r-project.org/
```

無論默認的CRAN鏡像是什麼，個人用戶仍然可以設置自己的默認值。為了防止這種情況，您可以設置 `allow-r-cran-repos-edit` 設置。例如：

```
/etc/rstudio/rsession.conf
```

```
allow-r-cran-repos-edit=0
```

Note that even with user editing turned off it's still possible for users to install packages from alternative repositories by directly specifying the `repos` parameter in a call to `install.packages` .

5.11 Feature Limits

RStudio Server has a number of other limits that can be configured. This section describes these limits. Note that these settings are specified in the `/etc/rstudio/rsession.conf` file (rather than the `rserver.conf` file previously referenced).

5.11.1 Disabling Access to Features

Besides the limits on package installation and CRAN repository editing described in the previous section there are a number of other limits that can be specified. The following describes all of the options that can be used to limit features.

```
/etc/rstudio/rsession.conf
```

```
allow-vcs
```

Allow access to Git and SVN version control features.

```
allow-vcs-executable-edit
```

Allow editing of the underlying Git or SVN executable.

allow-package-installation

Allow installation of packages using the Packages Pane (note that even if this is set to 0 it's still possible to install packages using `utils::install.packages` from the command line).

allow-r-cran-repos-edit

Allow editing of the CRAN repository used for package downloads (note that it's still possible to specify an alternate repository using the `repos` parameter of `utils::install.packages`).

allow-shell

Allow access to the Tools -> Shell dialog (note that it's still possible to execute shell commands using the `system` function).

allow-file-downloads

Allow downloading files using the Export command in the Files Pane.

allow-file-uploads

Allow uploading files using the Upload command in the Files pane.

allow-external-publish

允許將內容髮佈到外部（雲）服務。這包括發布使用R Markdown或R演示文稿創建的HTML文檔到RPubs (<http://rpubs.com>)，並將ShinyApps.io和文檔發佈到ShinyApps.io (<http://shinyapps.io>)。請注意，這只是刪除IDE中的相關用戶界面元素，用戶仍然可以使用R控制台發佈內容。

allow-publish

允許發佈內容。如果指定，則此選項將刪除與從IDE發佈內容相關的所有用戶界面元素，並進行覆蓋 `allow-external-publish` 。

所有這些功能都默認啟用。指定 `0` 禁止訪問該功能。

請注意，這些選項應該在 `/etc/rstudio/rsession.conf` 配置文件（而不是主要的`rserver.conf` 配置文件）中指定。

5.11.2最大文件上傳大小

您可以使用該 `limit-file-upload-size-mb` 設置來限制文件上傳的最大大小。例如，以下限制文件上傳到100MB：

`/etc/rstudio/rsession.conf`

```
limit-file-upload-size-mb=100
```

默認行為是對文件上傳的大小沒有限制。

5.11.3 每個計算的CPU時間

如果要防止佔用CPU 100%的失控計算，可以設置最大分鐘數以允許運行頂級R計算以使用該 `limit-cpu-time-minutes` 設置。例如：

```
/etc/rstudio/rsession.conf
```

```
limit-cpu-time-minutes=30
```

This specifies that no top level computation entered at the R console should run for more than 30 minutes. This constraint is implemented by calling the R `setTimeLimit` function immediately prior to handing off console commands to R. As a result it is possible for a particular script to override this behavior if it knows that it may exceed the threshold. This would be done as follows:

```
setTimeLimit(cpu = Inf)
# Long running R code here...
```

5.11.4 XFS Disk Quotas

If your system uses the XFS file system (<http://en.wikipedia.org/wiki/XFS>) then RStudio Server can be configured to notify users when they come close to or exceed their disk quota. You can enable this using the `limit-xfds-disk-quota` setting. For example:

```
/etc/rstudio/rsession.conf
```

```
limit-xfds-disk-quota=1
```

用戶的XFS磁盤配額將在RStudio IDE加載時進行檢查，如果接近或超過其配額，將顯示一條警告消息。

5.12通知

管理員可以使用 `notifications.conf` 位於的文件實時向用戶會話廣播通知 `/etc/rstudio/notifications.conf`。此文件默認情況下帶有註釋掉的條目，您可以取消註釋和使用，並幫助顯示可用的時間和消息格式。

每個會話都會監視 `notifications.conf` 文件中的更改，如果檢測到新的通知，它將在適當的時間顯示給用戶（如下一節所定義）。用戶的所有公開會話都將收到通知，他們將繼續在他們打開的任何新會話中看到通知，直到通知被確認。

修改通知將導致它作為新的通知計數，因此確保只有在確認了要顯示的消息的時間以及顯示的時間後，才能將更改保存到文件中。否則，可能會多次顯示相同的消息。

5.12.1 notifications.conf格式

該 `notifications.conf` 文件是由多個空行分隔的通知條目組成的文件。下表列出了文件中每個通知條目的可用字段。

開始時間	可以開始通知的開始時間。這必須是一個時間格式的字段。該字段不是必需的。
時間結束	通知將不再提供的結束時間。這必須是一個時間格式的字段。這是必填欄。
信息	要顯示給用戶的消息內容。消息中不能有空行。這是必填欄。

示例 `notifications.conf` 文件如下所示。有關每個字段格式的更多信息，請參閱後面的章節。

/etc/rstudio/notifications.conf

```
StartTime: 2017-08-30 12:00:00 -5:00
EndTime: 2017-08-30 20:00:00 -05:00
Message: Please remember to shut down your computers at the end of the day.

StartTime: 2017-05-30 13:55
EndTime: 2017-06-13
Message: This is a test notification. Notifications can span
        multiple lines by indenting the next line's message text.
        Empty lines are not supported!
```

每個條目由上面指定的2-3個字段（StartTime，EndTime和Message）組成是很重要的。每個領域都必須走自己的路線。字段定義之間不應該有空行。

例如，這是好的：

/etc/rstudio/notifications.conf

```
StartTime: 2017-08-30 12:00:00 -5:00
```

```
EndTime: 2017-08-30 20:00:00 -05:00
```

```
Message: Please remember to shut down your computers at the end of the day.
```

但是這不是：

/etc/rstudio/notifications.conf

```
StartTime: 2017-08-30 12:00:00 -5:00
```

```
EndTime: 2017-08-30 20:00:00 -05:00
```

```
Message: Please remember to shut down your computers at the end of the day.
```

在單獨的通知條目之間必須有一個空行（2個換行符）。

例如，這是好的：

/etc/rstudio/notifications.conf

```
StartTime: 2017-08-30 12:00:00 -5:00
```

```
EndTime: 2017-08-30 20:00:00 -05:00
```

```
Message: Please remember to shut down your computers at the end of the day.
```

```
StartTime: 2017-08-30 12:00:00 -5:00
```

```
EndTime: 2017-08-30 20:00:00 -05:00
```

```
Message: Remember to drop off any borrowed equipment at Grace's office today only!
```



但是這不是：

```
StartTime: 2017-08-30 12:00:00 -5:00
EndTime: 2017-08-30 20:00:00 -05:00
Message: Please remember to shut down your computers at the end of the day.

StartTime: 2017-08-30 12:00:00 -5:00
EndTime: 2017-08-30 20:00:00 -05:00
Message: Remember to drop off any borrowed equipment at Grace's office today only
```



5.12.1.1時間格式

時間格式字段 StartTime 和 EndTime 必須採用以下格式之一：

```
YYYY - MM - DD
YYYY - MM - DD hh : mm
YYYY -- MM : DD hh :: mm ss zh zm
```

下表顯示了格式部分的含義。

YYYY	4位數年份（例如：
MM	2位數字（例如：06
DD	2位數字（例如：28
HH	2位數字（24小時制
毫米	2位數分鐘（例如15
SS	2位秒（例如：59）
ZH	時區小時偏移（例如為-08）
ZM	時區分鐘偏移量（如不同）

如果沒有指定時間，則在當前服務器時區中為開始時間設置為00:00:00，在結束時間中在當前服務器時區中設置為23:59:59。

如果沒有指定秒，它們將被設置為00，並且在當前服務器時區中解釋時間。

請注意，如果您的會話在不同時區跨越多個服務器，並且您希望通知在統一時間顯示，則必須手動將時區設置為適當的時區。否則，不同時區的會話將在當地時間看到通知。

下表顯示了一些示例日期以及它們將如何格式化。

2020年1月1日下午6:00在服務器的時區
2018年7月31日午夜在服務器的時區（開始時間）
9月23日，2019年23:59:59在服務器的時區（結束時間）
太平洋標準時間2020年11月30日15:14:12

5.12.1.2消息格式

要發送的消息必須是純文本，不能有任何空行。要在另一行上開始文本，只需按照上一節中多行示例的方式縮進行即可。

5.13 RStudio連接服務器

RStudio Server Pro上的用戶可以將內容髮佈到[RStudio Connect](#)。為此，他們必須首先指定他們希望使用的RStudio Connect服務器。您可以設置用戶連接到帳戶時使用的默認RStudio Connect服務器URL。為此，請使用以下 `default-rsconnect-server` 選項：

/etc/rstudio/rsession.conf

```
default-rsconnect-server=http://connectserver/
```