

4訪問和安全

4.1網絡端口和地址

初始安裝後，RStudio接受端口8787上的連接。如果您希望監聽另一個端口，則可以修改該 `www-port` 選項。例如：

/etc/rstudio/rserver.conf

```
www-port=80
```

默認情況下，RStudio綁定到地址0.0.0.0（接受來自任何遠程IP的連接）。您可以使用該 `www-address` 選項修改此行為。例如：

/etc/rstudio/rserver.conf

```
www-address=127.0.0.1
```

請注意，在重新啟動服務器之前，對配置的更改不會生效。

4.2 IP訪問規則

RStudio服務器可以配置為拒絕訪問特定的IP地址或地址範圍。訪問規則在配置文件中定義 `/etc/rstudio/ip-rules`

訪問規則是使用 `allow` 和 `deny` 指令建立的，並按照順序處理，第一個匹配規則管理給定地址是允許還是拒絕。例如，要只允許 `192.168.1.0/24` 子網內的客戶端，但也拒絕訪問 `192.168.1.10` 你將使用這些規則：

在/ etc / rstudio / IP的規則

```
deny    192.168.1.10
allow   192.168.1.0/24
deny    all
```

指定子集外的所有客戶端由於 `deny all` 配置結束時的規則而被拒絕訪問。

請注意，在重新啟動服務器之前，對配置的更改不會生效。

4.3 框架原點

出於安全原因，默認情況下，RStudio Server不會加載瀏覽器框架（如框架集或IFrame）。您可以使用該 `www-frame-origin` 選項修改此行為。例如，如果您想要在瀏覽器框架中放置 `example.com` RStudio，可以告訴RStudio如下所示：

/etc/rstudio/rserver.conf

```
www-frame-origin=example.com
```

有幾個特殊值可供 `www-frame-origin` 選擇：

值	含義
<code>none</code>	默認; 不要讓RStudio在任何幀中加載。
<code>same</code>	如果RStudio與RStudio具有相同的來源（主機和端口），則允許RStudio加載。
<code>any</code>	允許RStudio從任何來源加載幀（不推薦）
<code>my-domain.com</code>	允許RStudio在 <code>my-domain.com</code> 的框架中加載

4.4 安全套接字（SSL）

4.4.1 SSL配置

如果您的RStudio服務器在公共網絡上運行，強烈建議將其配置為使用SSL（安全套接字層）加密。您可以通過 `ssl-enabled` 設置以及指定您的SSL證書和密鑰的位置的相關設置來執行此操作。例如：

/etc/rstudio/rserver.conf

```
ssl-enabled=1
ssl-certificate=/var/certs/your_domain_name.crt
ssl-certificate-key=/var/certs/your_domain_name.key
```

該 .crt 文件應該以PEM格式進行編碼; 也就是說, 第一行應該讀取 -----BEGIN CERTIFICATE-----, 內容應該是base64編碼的數據。如果您的證書採用其他格式 (例如DER或PKCS), 請使用 openssl 命令行工具將其轉換為PEM。例如:

```
openssl x509 -inform DER -outform PEM -text -in your_domain_name.der -out your_domain_name.crt
```

安裝證書 .crt 文件時, 將所有中間證書 (即來自證書頒發機構的通用證書) 與與您的域名關聯的證書連接在一起, 這一點很重要。例如, 您可以使用此表單的shell命令將CA中間證書連接到您的域名證書:

```
$ cat certificate-authority.crt >> your_domain_name.crt
```

結果文件應該在 ssl-certificate 選項中指定。

確保您的SSL證書密鑰上的文件權限盡可能限制, 以便普通用戶無法讀取也很重要。該文件通常應由 root 用戶擁有, 並設置為擁有者可讀和可寫。例如:

```
$ sudo chmod 600 /var/certs/your_domain_name.key
```

4.4.2 SSL協議

默認情況下, RStudio Server支持SSL的TLSv1, TLSv1.1和TLSv1.2協議。支持的協議列表可以通過該 ssl-protocols 選項進行配置。例如, 要僅使用TLSv1.1和TLSv1.2協議, 您將使用:

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

```
ssl-protocols=TLSv1.1 TLSv1.2
```

支持的協議列表是空格分隔的 (如上所示)。有效的協議值是: SSLv2, SSLv3, TLSv1, TLSv1.1, 和 TLSv1.2。

4.4.3 SSL端口

當RStudio服務器配置為使用SSL時，相對於端口的默認行為是：

1. SSL綁定到端口443（使用瀏覽器中的標準https協議訪問）
2. 服務器還偵聽端口80，並將所有請求重定向到端口443（允許用戶指定沒有https協議的域，並自動重定向到安全端口）

但是，如果SSL綁定到另一個端口（使用該 `www-port` 選項），則自動重定向行為未啟用。也可以使用以下 `ssl-redirect-http` 選項完全禁用自動SSL重定向：

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

```
ssl-redirect-http=0
```

請注意，在重新啟動服務器之前，對配置的更改不會生效。

4.5服務器權限

4.5.1服務器帳戶

RStudio Server在啟動過程中以系統root用戶身份運行，然後刪除此權限並作為受限用戶運行。RStudio Server在代表用戶創建R會話（服務器 `setresuid` 在創建R會話時需要調用，並且此調用需要root權限）的情況下，短時間重新獲得root權限。

RStudio Server在正常運行過程中運行的用戶帳戶是 `rstudio-server`。此帳戶在安裝期間會自動添加到系統中，並以系統而不是最終用戶帳戶（即 `--system` 傳遞給該標誌 `useradd`）的方式創建。

4.5.1.1備用服務器帳戶

您可以配置RStudio Server，使其通過以下步驟從備用帳戶運行：

1. 遞歸刪除 `/var/log/rstudio-server` 和 `/var/lib/rstudio-server` 目錄（它們包含默認rstudio服務器用戶擁有的文件和目錄）
2. 創建一個新的系統用戶（如果你想使用的用戶不存在）
3. 將此用戶分配給配置文件中的 `server-user` 選項 `/etc/rstudio/rserver.conf`（請參閱下面的示例）
4. 重新啟動RStudio服務器

請注意，`/var/*/rstudio-server` 目錄的刪除將重置任何已存儲的指標和日誌文件。

例如，要關閉服務器，清除以前的rstudio用戶擁有的文件，並創建一個名為 `rs-user` 你的新的系統用戶，使用下面的命令：

```
sudo rstudio-server stop
sudo rm -rf /var/log/rstudio-server
sudo rm -rf /var/log/rstudio-server
sudo useradd --system rs-user
```

然後編輯 `/etc/rstudio/rserver.conf` 配置文件如下：

/etc/rstudio/rserver.conf

```
server-user=rs-user
```

最後，重新啟動RStudio Server，開始在新用戶下運行：

```
sudo rstudio-server start
```

4.5.2 AppArmor

在Debian和Ubuntu系統上，RStudio Server進程在AppArmor配置文件下運行（有關AppArmor的更多信息，請訪問：<http://en.wikipedia.org/wiki/AppArmor>）。

如果AppArmor在您的配置中導致問題，您可以使用該 `server-app-armor-enabled` 選項將其禁用。例如：

/etc/rstudio/rserver.conf

```
server-app-armor-enabled=0
```

請注意，RStudio Server AppArmor配置文件不會導致問題，因此您不需要修改此設置。另請注意，該設置在服務器重新啟動之前不會生效。

4.5.3 umask

默認情況下，RStudio Server將其umask設置為 022 啟動。如果你不想要這樣的行為，例如，因為你希望服務器進程使用在init中設置的默認umask，可以按如下方式禁用它：

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

```
server-set-umask=0
```

4.6 使用代理運行

4.6.1 概述

如果您在代理服務器後面運行RStudio服務器，則需要確保配置代理服務器，以便正確處理與RStudio服務器之間的所有流量。

除了正常的反向代理配置之外，您還需要申請任何HTTP服務器應用程序，以確保在代理服務器和RStudio服務器之間正確轉發WebSockets，以確保所有RStudio功能正常工作。尤其是，需要確保Shiny應用程序能夠在IDE中正常運行 - 否則，您可能會發現Shiny應用程序“變灰”並且無法與之交互。

本節介紹如何使用Nginx和Apache正確配置反向代理。

4.6.2 Nginx配置

在Debian或Ubuntu上，可以使用以下命令安裝支持反向代理的Nginx版本：

```
sudo apt-get install nginx
```

在CentOS或Red Hat上，您可以使用以下命令安裝Nginx：

```
sudo yum install nginx
```

要在同一台服務器上運行一個Nginx實例作為RStudio Server的前端代理，你可以在你的 `nginx.conf` 文件中添加類似如下的命令。請注意，您必須將代碼添加到代理Websockets，才能在RStudio Server中正確顯示Shiny應用程序和R Markdown Shiny文檔。另外請注意，如果您在代理不同計算機上的服務器，則需要將引用替換為 `localhost` 您託管RStudio的服務器的正確地址。

```
http {  
  
    map $http_upgrade $connection_upgrade {  
        default upgrade;  
        ''      close;  
    }  
  
    server {  
        listen 80;  
  
        location / {  
            proxy_pass http://localhost:8787;  
            proxy_redirect http://localhost:8787/ $scheme://$host/;  
            proxy_http_version 1.1;  
            proxy_set_header Upgrade $http_upgrade;  
            proxy_set_header Connection $connection_upgrade;  
            proxy_read_timeout 20d;  
        }  
    }  
}
```

如果您想從自定義路徑（例如/ rstudio）為RStudio服務器提供服務，您可以編輯 `nginx.conf` 文件，如下所示：

```

http {

    map $http_upgrade $connection_upgrade {
        default upgrade;
        ''      close;
    }

    server {
        listen 80;

        location /rstudio/ {
            rewrite ^/rstudio/(.*)$ /$1 break;
            proxy_pass http://localhost:8787;
            proxy_redirect http://localhost:8787/ $scheme://$host/rstudio/;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection $connection_upgrade;
            proxy_read_timeout 20d;
        }
    }
}

```

添加這些條目後，您需要重新啟動Nginx，以便代理設置生效：

```
sudo /etc/init.d/nginx restart
```

4.6.3 Apache配置

要啟用運行在同一服務器上的Apache實例作為RStudio Server的前端代理，您需要使用 `mod_proxy` 和 `mod_proxy_wstunnel` 模塊。啟用此模塊的步驟因操作系統而異，所以您應該查閱發行版的Apache文檔以獲取詳細信息。

在Debian和Ubuntu系統上，可以 `mod_proxy` 使用以下命令安裝Apache：

```

sudo apt-get install apache2
sudo apt-get install libapache2-mod-proxy-html
sudo apt-get install libxml2-dev

```


然後，要更新Apache配置文件來激活，`mod_proxy` 請執行以下命令：

```
sudo a2enmod proxy
sudo a2enmod proxy_http
sudo a2enmod proxy_wstunnel
```

在CentOS和RedHat系統下的Apache可以安裝 `mod_proxy` 並 `mod_proxy_wstunnel` 按照這裡的說明：

<http://httpd.apache.org/docs/2.4/platform/rpm.html>

默認情況下使用Apache 2.4，`mod_proxy` 並 `mod_proxy_wstunnel` 應該啟用。您可以通過打開該文件 `/etc/httpd/conf.modules.d/00-proxy.conf` 並確保包含以下內容並未將其註釋掉來進行檢查：

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_wstunnel_module modules/mod_proxy_wstunnel.so
```

一旦啟用 `mod_proxy` 並 `mod_proxy_wstunnel` 在您的Apache安裝中，您需要將所需的代理命令添加到您的 `VirtualHost` 定義。請注意，您還需要包含代碼才能正確代理Websocket，以便在RStudio Server中正確代理Shiny應用程序和R Markdown文檔。另外請注意，如果您在代理不同計算機上的服務器，則需要將引用替換為 `localhost` 您託管RStudio的服務器的正確地址。

```

<VirtualHost *:80>

    <Proxy *>
        Allow from localhost
    </Proxy>

    RewriteEngine on
    RewriteCond %{HTTP:Upgrade} =websocket
    RewriteRule /(.*) ws://localhost:8787/$1 [P,L]
    RewriteCond %{HTTP:Upgrade} !=websocket
    RewriteRule /(.*) http://localhost:8787/$1 [P,L]
    ProxyPass / http://localhost:8787/
    ProxyPassReverse / http://localhost:8787/
    ProxyRequests Off

</VirtualHost>

```

請注意，如果您想從自定義路徑（例如/rstudio）提供RStudio，則可以將上述指令替換為：

```

RewriteEngine on
RewriteCond %{HTTP:Upgrade} =websocket
RewriteRule /rstudio/(.*) ws://localhost:8787/$1 [P,L]
RewriteCond %{HTTP:Upgrade} !=websocket
RewriteRule /rstudio/(.*) http://localhost:8787/$1 [P,L]
ProxyPass /rstudio/ http://localhost:8787/
ProxyPassReverse /rstudio/ http://localhost:8787/
ProxyRequests Off

```

最後，在完成上述所有步驟之後，您需要重新啟動Apache，以便代理設置生效：

```

sudo /etc/init.d/apache2 restart

```

4.6.4 RStudio配置

如果您的RStudio服務器和代理服務器在同一台計算機上運行，則還可以將RStudio Server從0.0.0.0（所有遠程客戶端）偵聽的端口更改為127.0.0.1（僅限本地主機）。這確保了連接到RStudio服務器的唯一方法是通過代理服務器。您可以通過將該 `www-address` 條目添加到 `/etc/rstudio/rserver.conf` 文件中來執行此操作，如下所示：

```
www-address=127.0.0.1
```

請注意，您可能需要創建此配置文件，如果它不存在。