

The screenshot shows the RStudio interface with several windows open:

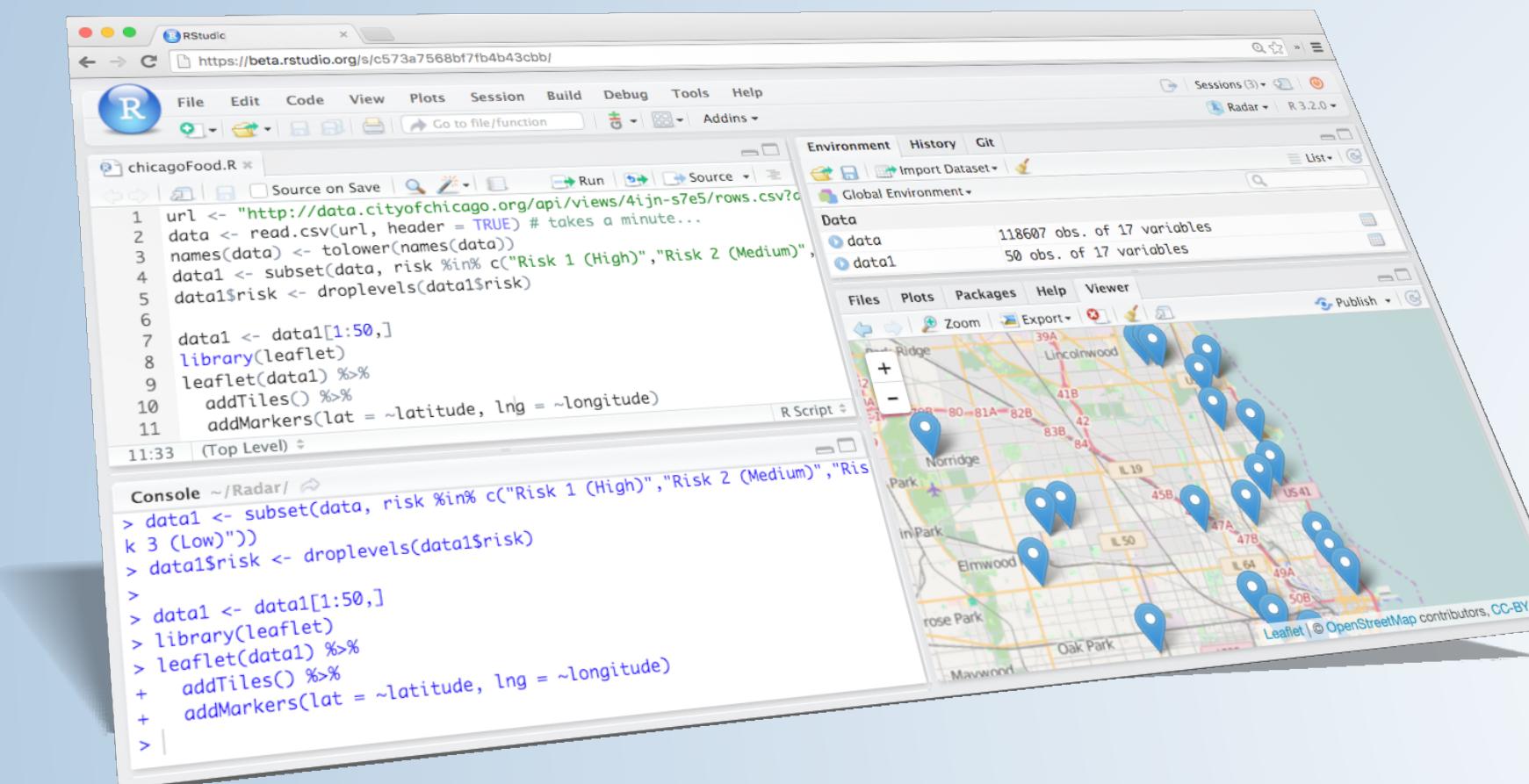
- Code Editor:** Shows a script named `print.R` with code related to data manipulation and visualization.
- Console:** Displays the command `> ggvis(diamonds, x = ~price, y = ~color)` and its execution output, including a stack trace and a histogram plot.
- Environment:** Shows the values of variables used in the ggvis call, such as `as.vega.ggvis()`, `data_ids`, `data_props`, and `dynamic`.
- Viewer:** Displays a histogram of diamond prices, with the x-axis labeled "price" ranging from 0 to 12,000 and the y-axis labeled "count" ranging from 0 to 120.

# WHAT YOU SHOULD KNOW ABOUT ADMINISTERING SHINY SERVER PRO SUPPORTING A SCALABLE ENVIRONMENT FOR HOSTING SHINY APPS

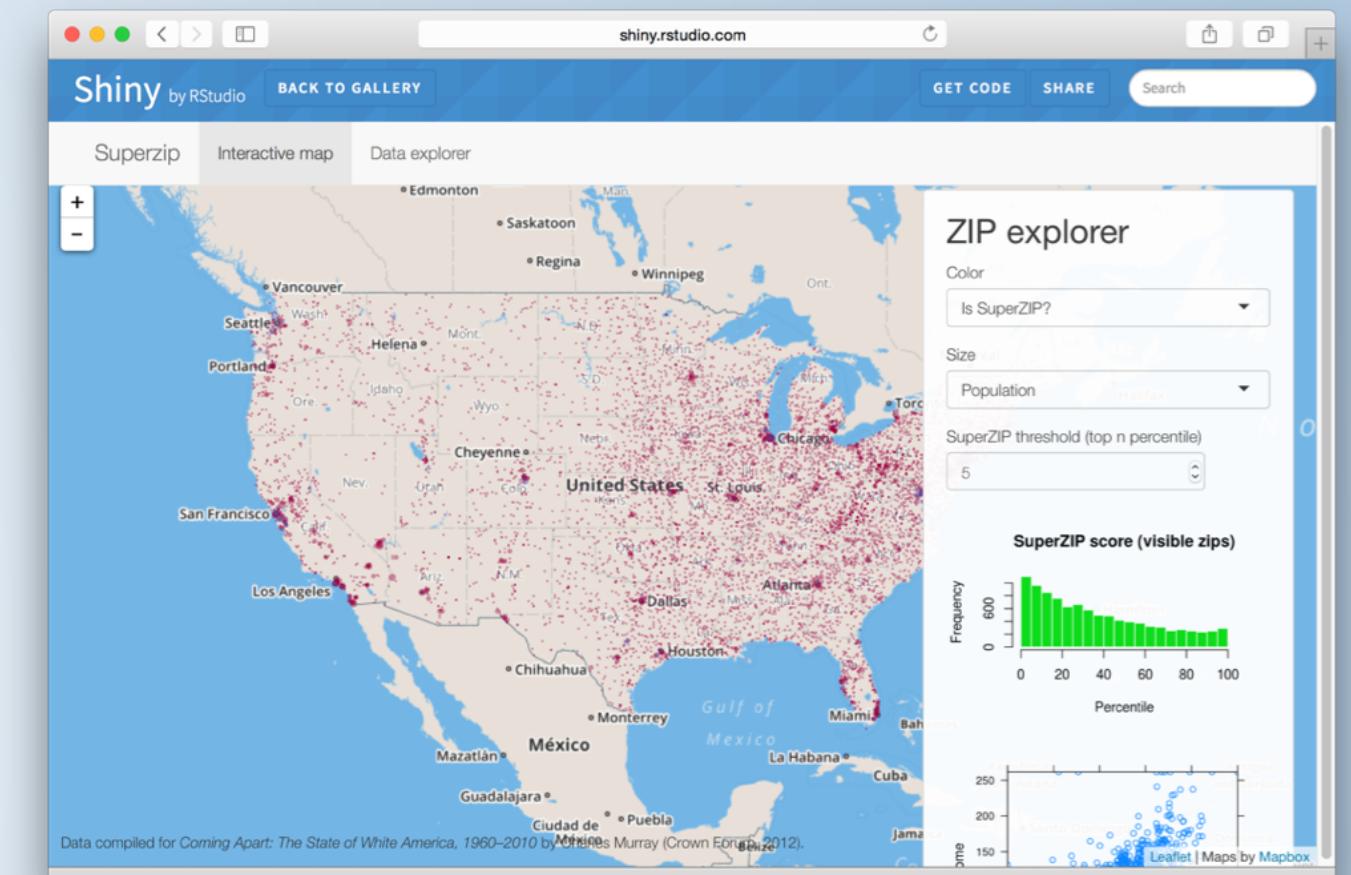


# RStudio Products

## RStudio IDE



## Shiny



## Packages



# Shiny

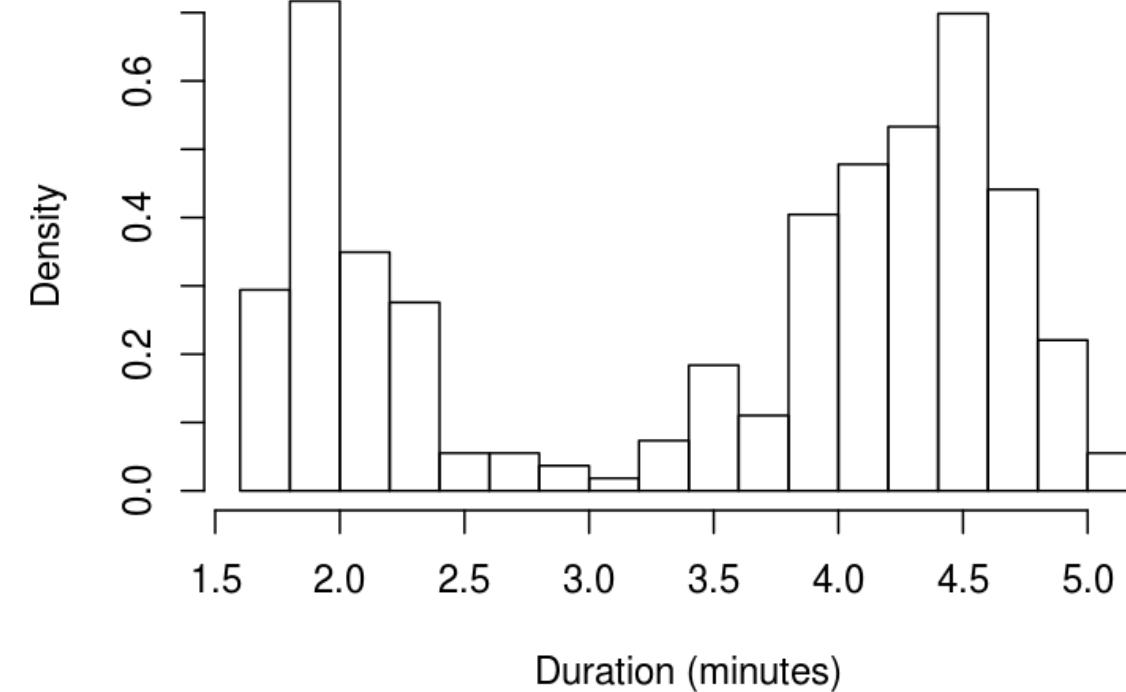
A web application framework for R  
Turn your analyses into interactive web applications  
No HTML, CSS, or JavaScript knowledge required

Number of bins in histogram (approximate):

**Show individual observations**

**Show density estimate**

**Geyser eruption duration**



Density

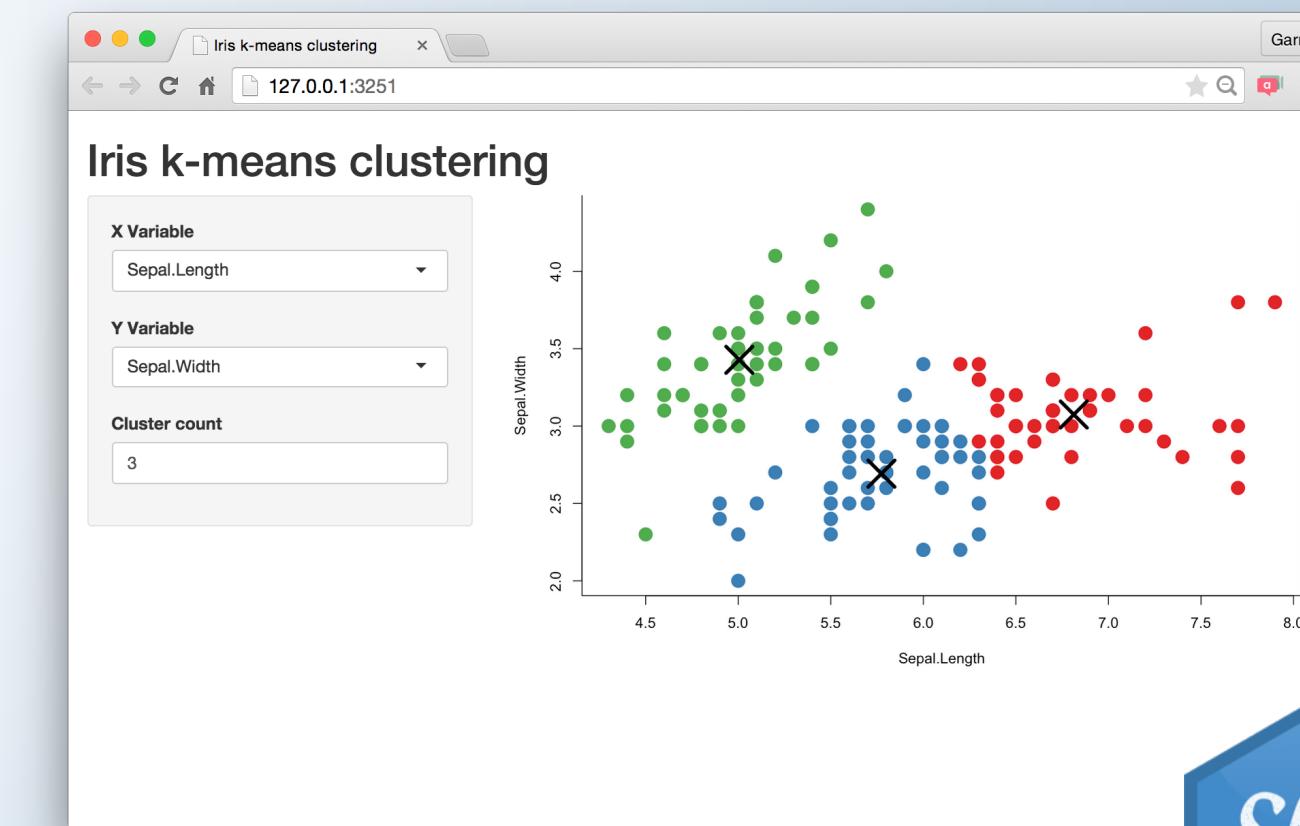
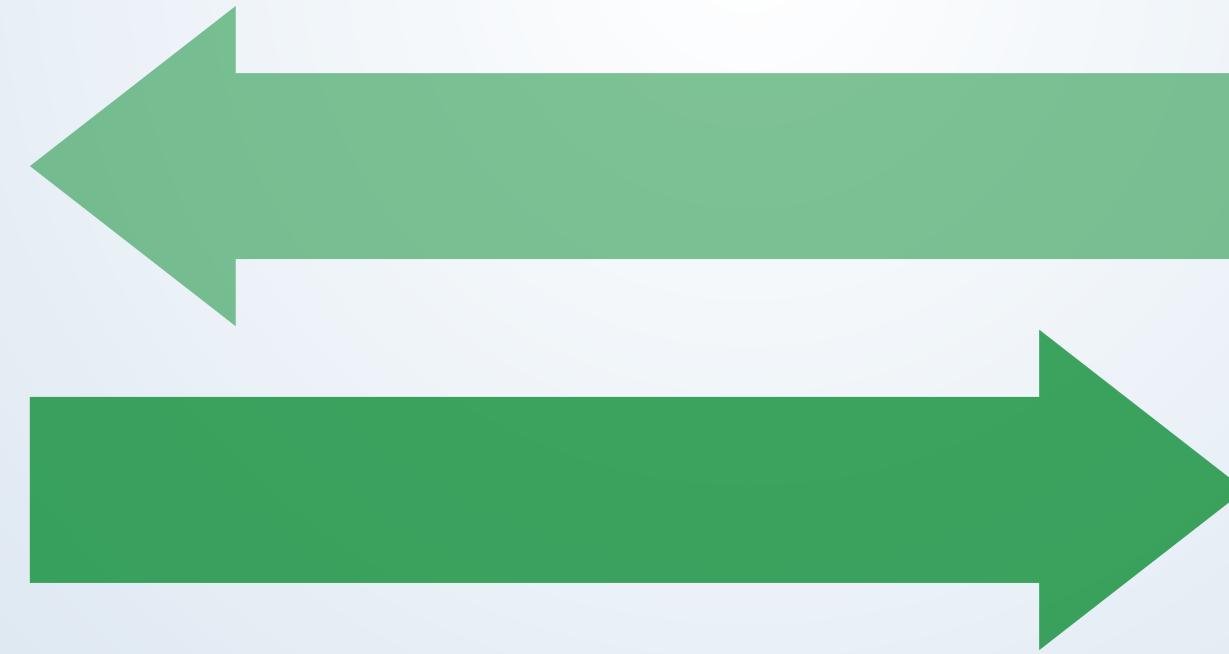
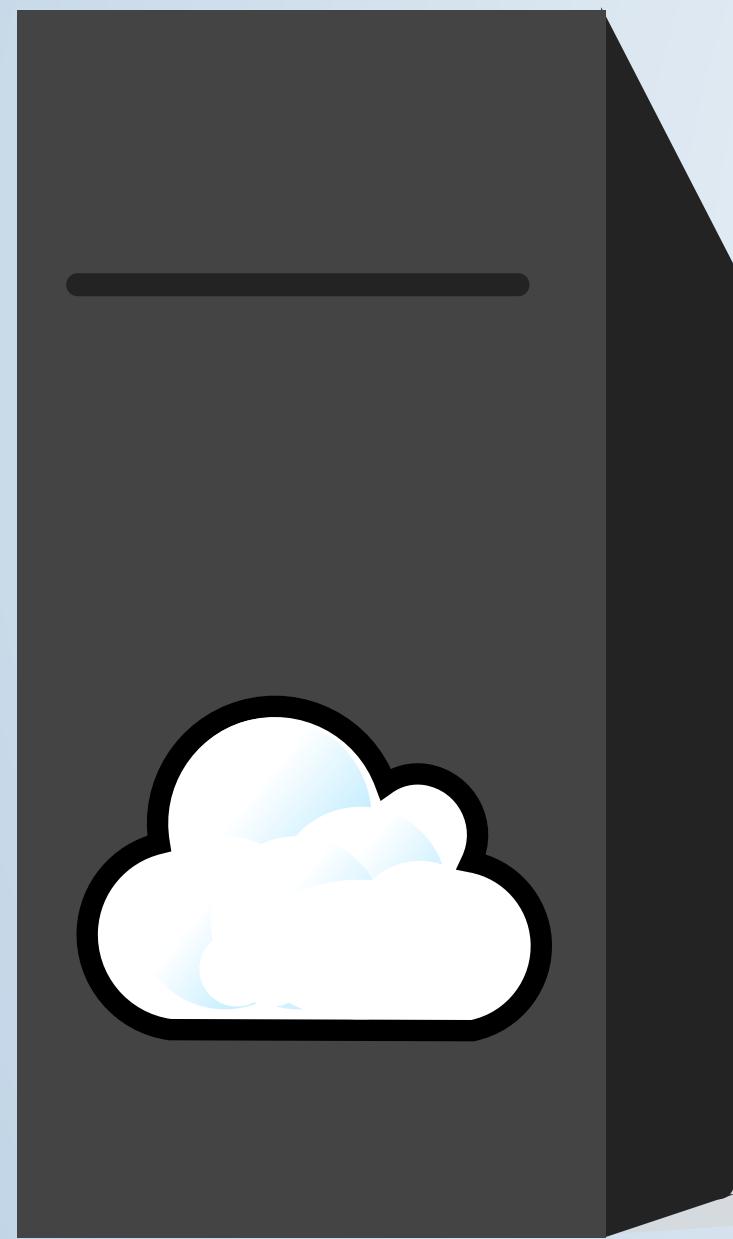
Duration (minutes)

Duration (minutes)	Density
1.5 - 2.0	~0.30
2.0 - 2.5	~0.70
2.5 - 3.0	~0.05
3.0 - 3.5	~0.02
3.5 - 4.0	~0.15
4.0 - 4.5	~0.45
4.5 - 5.0	~0.65
5.0 - 5.5	~0.20

ui.R    server.R

```
shinyUI(bootstrapPage(
  selectInput(inputId = "n_breaks",
    label = "Number of bins in histogram (approximate):",
    choices = c(10, 20, 35, 50),
    selected = 20),
  checkboxInput(inputId = "individual_obs",
    label = strong("Show individual observations"),
    value = FALSE),
  checkboxInput(inputId = "density",
    label = strong("Show density estimate"),
    value = FALSE),
  plotOutput(outputId = "main_plot", height = "300px"),
  # Display this only if the density is shown
  conditionalPanel(condition = "input.density == true",
    sliderInput(inputId = "bw_adjust",
      label = "Bandwidth adjustment:",
      min = 0.2, max = 2, value = 1, step = 0.2)
))
))
```

# Shiny and Shiny Server



# Shiny Server Pro

*Shiny for the enterprise*

Choose Shiny Server Pro to **secure** user access, tune application **performance**, **monitor** resource utilization and get the direct support you need to create the best interactive data experiences for your customers and colleagues.



# What is the difference between Shiny and Shiny Server?

## Shiny

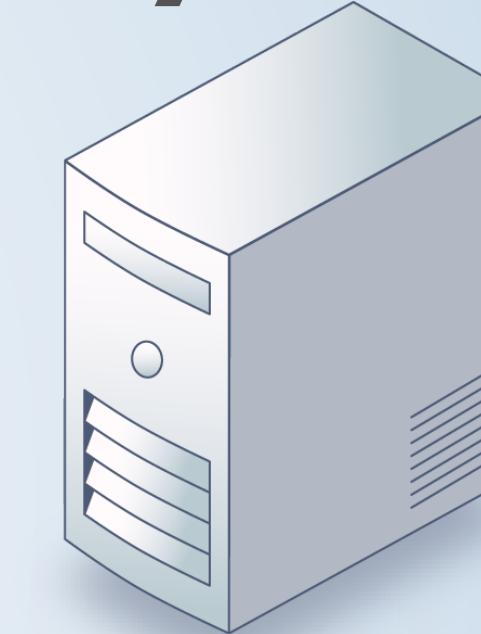


**Free and open source R Package**

Makes it incredibly easy to  
**build interactive web applications with R**

Automatic reactive binding between inputs and outputs  
and extensive pre-built widgets make it possible to build  
**beautiful, responsive, and powerful applications.**

## Shiny Server

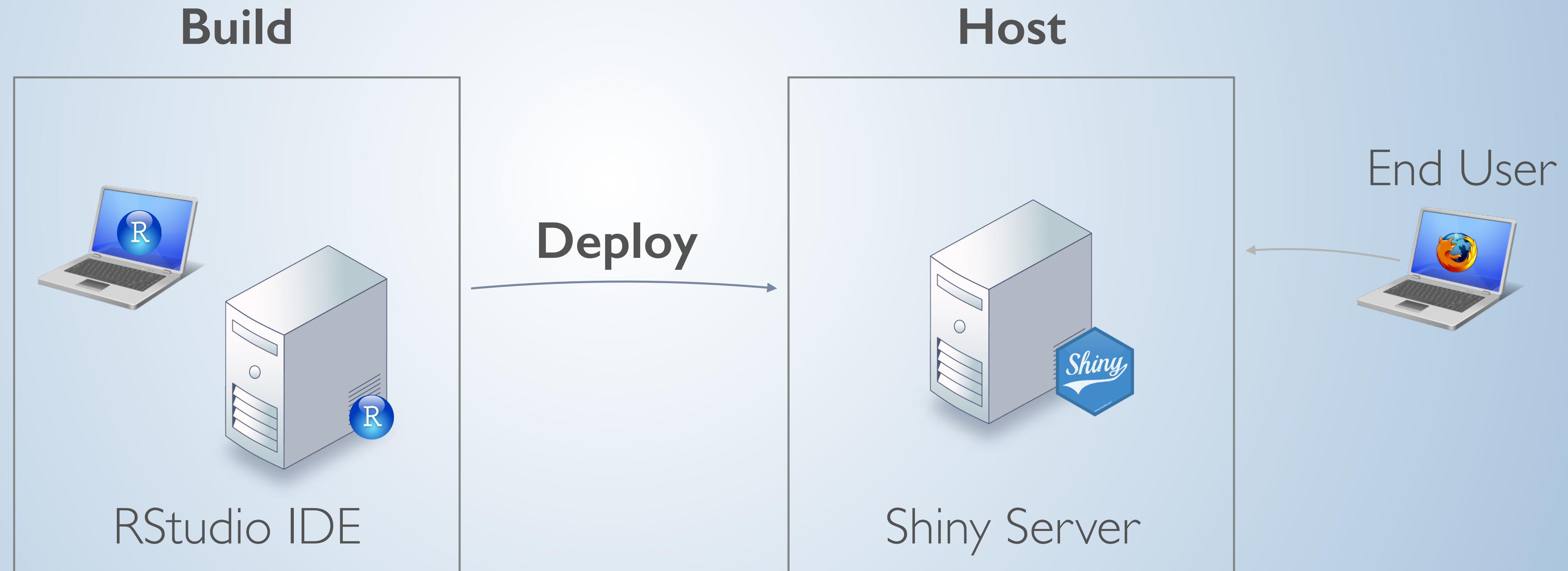


**Software you install on your server**

Enable users to  
**host and managing Shiny applications**

**Scale** a Shiny application to support many users  
**Protect and secure** your applications  
**Manage** the user experience

# *What is the difference between RStudio IDE and Shiny Server Pro?*



# Shiny Server Pro Evaluation

<https://www.rstudio.com/products/shiny-server-pro/evaluation-2/>



45

Day Free Eval

& Access to priority email support

# FREQUENTLY ASKED QUESTIONS



# *Can I install Shiny Server on Premises?*

**Shiny Server Pro**  
On Premises



**shinyapps.io**  
Cloud Hosted



# *Can I install Shiny Server Pro on a Virtual Machine or in the cloud?*



Hardware 



Virtualization 



Cloud 

# *What are the basic requirements?*

## **Operating Systems**

Ubuntu  
Redhat/CentOS  
SLES

## **R Engines**

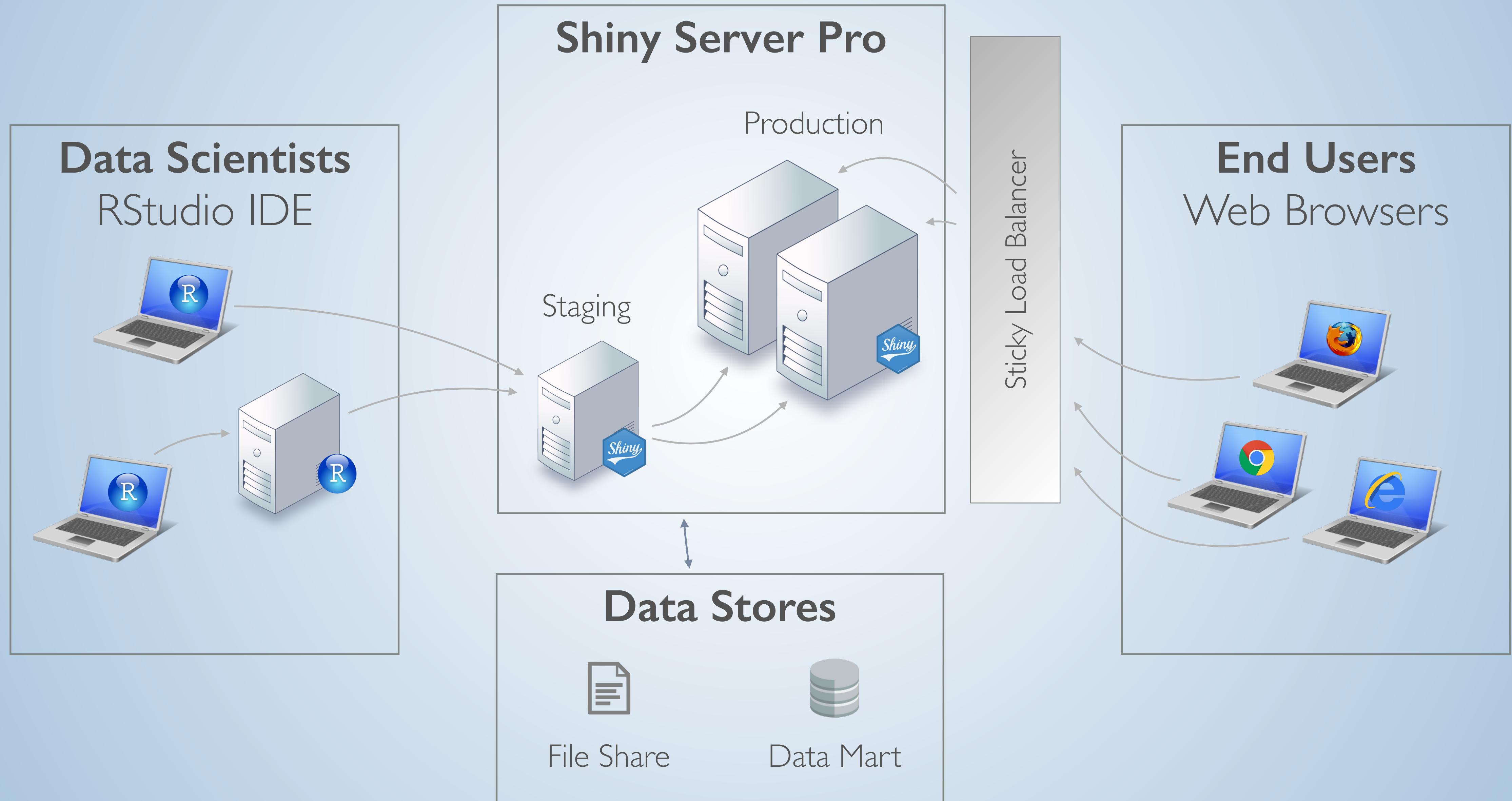
CRAN (common)  
Microsoft, Tibco, Oracle, etc.

### **Shiny Server**

*Example*



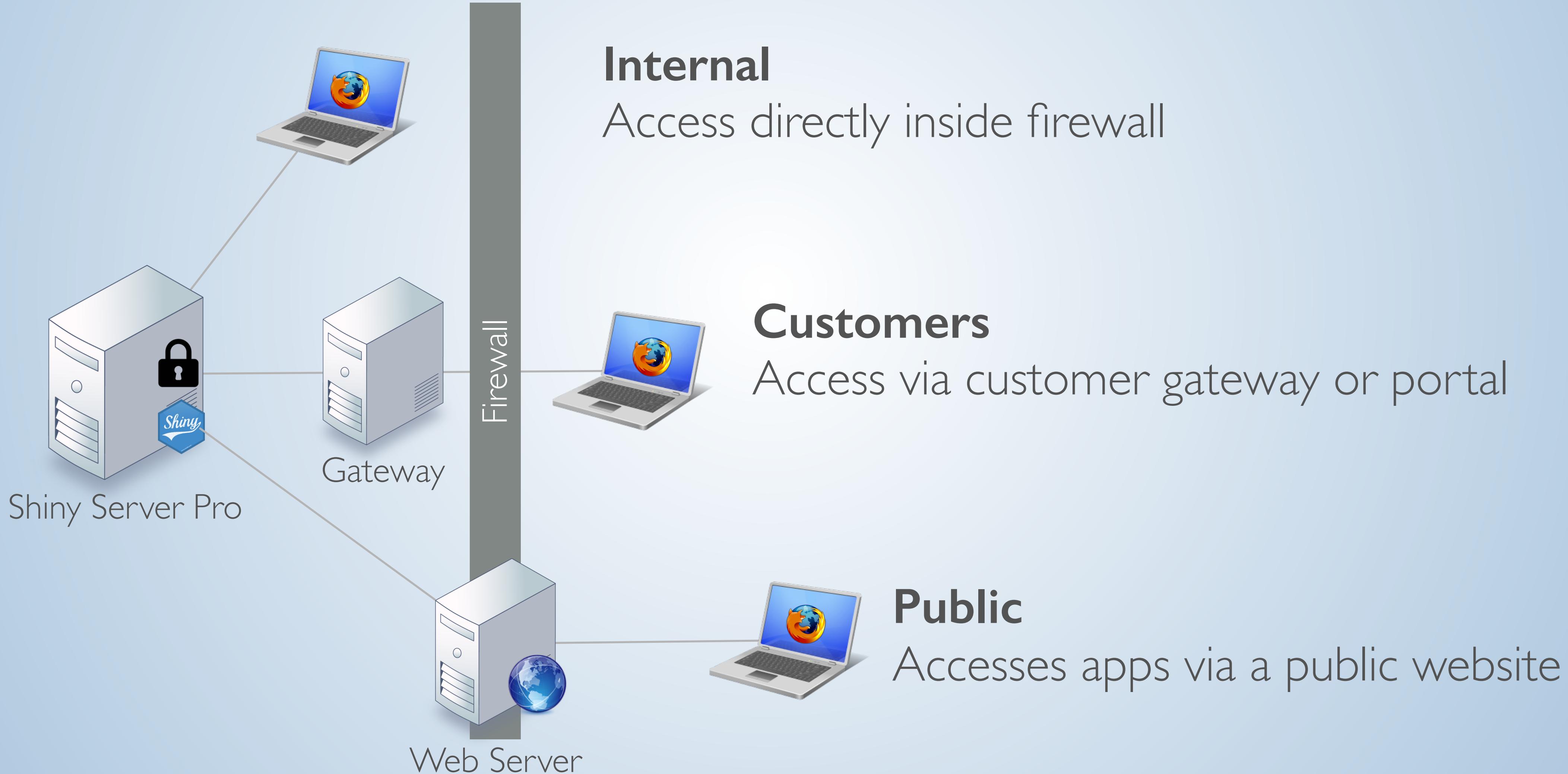
# What does a typical setup look like?



# How do I get started?



# Production Use Cases



# How do I integrate Shiny apps into my website?

<http://shiny.rstudio.com>

The screenshot shows a web browser window with the URL <http://shiny.rstudio.com>. On the left, a Shiny application titled "Here is a Shiny app" is displayed. It contains a histogram of geyser eruption durations with a slider input for the number of bins (set to 20) and two checkbox inputs ("Show individual observations" and "Show density estimate"). On the right, the browser's developer tools are open, showing the "Elements" tab with the DOM structure and the "Sources" tab with the R code for the Shiny app.

**ui.R**

```
shinyUI(pageWithSidebar(
```

**server.R**

```
server = function(input, output) {
```

Number of bins in histogram (approximate):

20

Show individual observations

Show density estimate

Geyser eruption duration

Density

Duration (minutes)

iframe#example1 | 331x500

Number of bins in histogram (approximate):

20

Show individual observations

Show density estimate

Geyser eruption duration

Density

Duration (minutes)

iframe#example1

```
ui.R server.R
```

```
shinyUI(pageWithSidebar(
```

```
server = function(input, output) {
```

```
  selectInput(inputId = "n_breaks",
              label = "Number of bins in histogram (approximate)",
              choices = c(10, 20, 35, 50),
              selected = 20),

  checkboxInput(inputId = "individual_obs",
                label = strong("Show individual observations"),
                value = FALSE),

  checkboxInput(inputId = "density",
                label = strong("Show density estimate"),
                value = FALSE),

  plotOutput(outputId = "main_plot", height = "300px")

  # Display this only if the density is shown
  conditionalPanel(condition = "input.density == true",
    sliderInput(inputId = "bw_adjust",
                label = "Bandwidth adjustment:",
                min = 0.2, max = 2, value = 1, step = 0.2)
  )
}
```

```
})
```

Elements Console Sources Network Timeline Profiles > A1 :: X

<a href="https://github.com/rstudio/shiny">...</a>

<div class="container">

<div class="section">

<div class="section-intro">...</div>

<div class="row">

<div class="col-sm-6">

<div id="example1-container">

<iframe id="example1" src="https://gallery.shinyapps.io/front-page/" style="border: none; width: 100%; height: 500px" frameborder="0" == \$0

<#document

<!DOCTYPE html>

<html class="...>

<head>...</head>

<body>...</body>

</html>

</iframe>

</div>

</div>

html body div div div.row div.col-sm-6 div#example1-container iframe#example1 html body

Styles Event Listeners DOM Breakpoints Properties

Filter :hov .cls +

element.style { border: none; width: 100%; height: 500px; }

iframe#example1 { border: none; width: 600px; height: 480px; }

\*, \*:before, \*:after { bootstrap.min.css:7

Console Network conditions

AdBlock Preserve log

Sat May 21 2016 18:45:05 GMT-0400 (EDT): Connection opened. https://gallery.shinyapps.io/front-page/ rstudio-connect.js:384

# How big should my server be?

<https://gallery.shinyapps.io/instanceCalc/>

## Primary Drivers

1. Number of concurrent sessions
2. Size of sessions

	<u>RAM (GB)</u>
• Small	< 1
• Medium	1 - 10
• Large	10 +

## Instance Sizes Examples

	<u>Cores</u>	<u>RAM (GB)</u>
• Micro	2	16
• Small	8	64
• Medium	16	256
• Large	32	512

R Instance Calculator

1. Sessions  
How many active sessions do you need to support?  
15

2. Memory  
What is the percentage of small/medium/large sessions in terms of memory?  
64

3. Compute  
 Do individual sessions use multiple cores?  
 Are you using hyper-threading?

4. High Availability  
What is your standard for high availability?  
50

5. Custom instance sizes  
 Do you want to use a custom instance size?

Copyright RStudio

Instructions  
Estimate the amount of memory, cores, and instances that will be needed to support your organization. Input the number of active sessions, the memory profile of sessions, and compute and high availability requirements.

**145 GB**  **8 Cores** 

**Number of Instances Required**

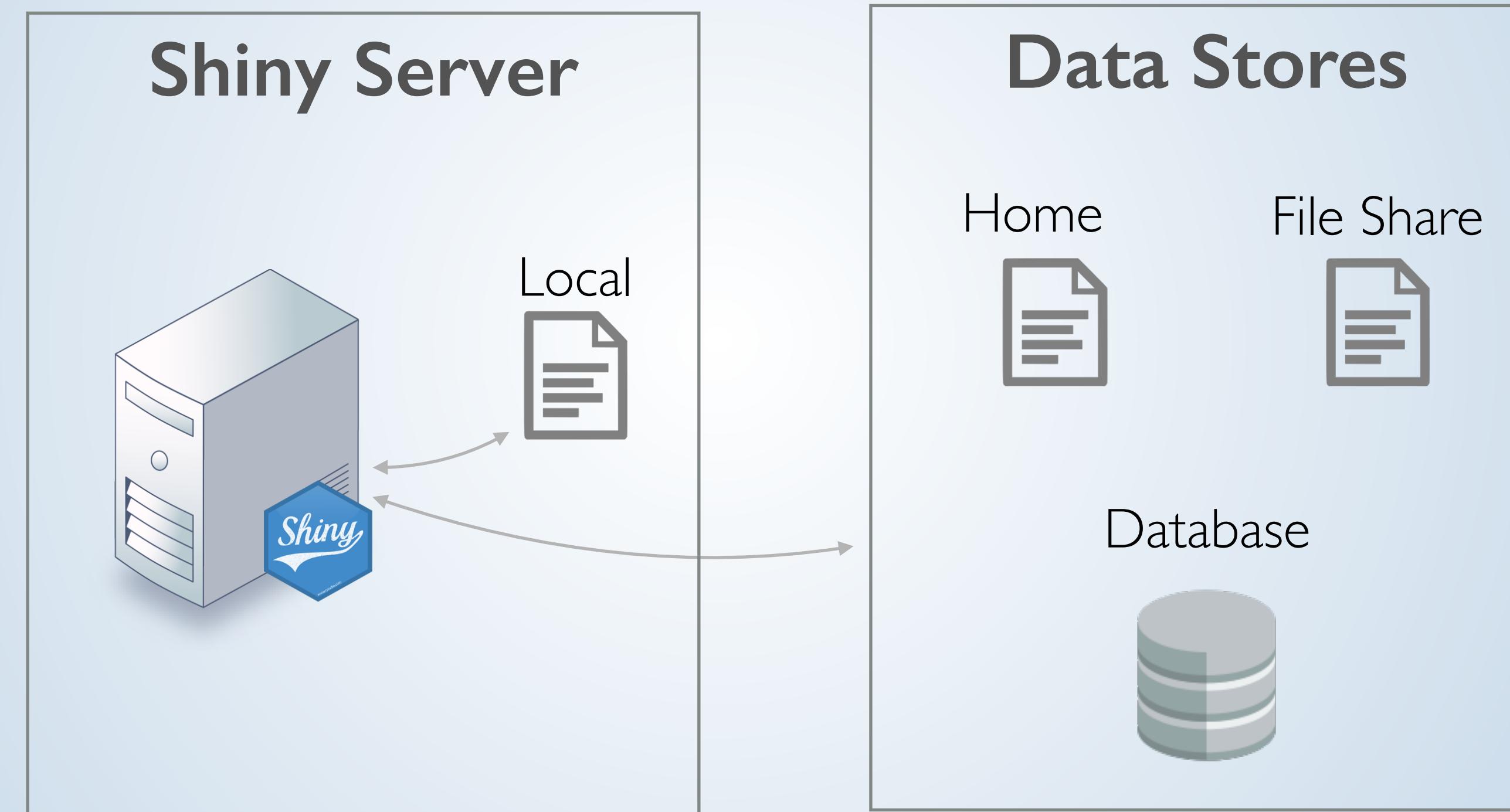
	Micro	Small	Medium	Large	Total
HA	5	5	5	5	20
StandAlone	10	10	10	10	40

**Detail of Instances**

Type	Memory	Cores	HA	StandAlone	Total
Micro	16	2	5	10	15
Small	64	8	2	3	5
Medium	256	16	1	1	2
Large	512	32	1	1	2

**Disclaimer**  
This calculator provides generalized estimates and cannot account for all aspects of your particular environment, use cases, and/or requirements. Estimates are calculated using assumptions that may or may not apply to your situation.

# Where should I store my data?



# Where do I go to get more information?

<http://docs.rstudio.com/>

## Shiny Server Professional v1.4.2 Administrator's Guide

- [1 Getting Started
  - \[1.1 Introduction\]\(#\)
  - \[1.2 System Requirements\]\(#\)
  - \[1.3 Installation
    - \\[1.3.1 Ubuntu \\\(12.04+\\\)\\]\\(#\\)
    - \\[1.3.2 RedHat/CentOS \\\(5.4+\\\)\\]\\(#\\)
    - \\[1.3.3 SUSE Linux Enterprise Server \\\(11+\\\)\\]\\(#\\)
    - \\[1.3.4 Install Shiny\\]\\(#\\)
    - \\[1.3.5 R Installation Location\\]\\(#\\)\]\(#\)
  - \[1.4 Stopping and Starting
    - \\[1.4.1 systemd \\\(RedHat 7, Ubuntu 15.04+, SLES 12+\\\)\\]\\(#\\)
    - \\[1.4.2 Upstart \\\(Ubuntu 12.04 through 14.10, RedHat 6\\\)\\]\\(#\\)
    - \\[1.4.3 init.d \\\(RedHat 5, SLES 11\\\)\\]\\(#\\)\]\(#\)](#)
- [2 Server Management
  - \[2.1 Default Configuration\]\(#\)
  - \[2.2 Server Hierarchy
    - \\[2.2.1 Server\\]\\(#\\)
    - \\[2.2.2 Location\\]\\(#\\)
    - \\[2.2.3 Application\\]\\(#\\)\]\(#\)
  - \[2.3 run\\\_as
    - \\[2.3.1 :HOME\\\\_USER:\\]\\(#\\)
    - \\[2.3.2 Running Shiny Server with Root Privileges\\]\\(#\\)
    - \\[2.3.3 :AUTH\\\\_USER:\\]\\(#\\)\]\(#\)
  - \[2.4 PAM Sessions
    - \\[2.4.1 Session Profile\\]\\(#\\)\]\(#\)
  - \[2.5 r\\\_path\]\(#\)
  - \[2.6 Local App Configurations\]\(#\)
  - \[2.7 Hosting Model
    - \\[2.7.1 Host a Directory of Applications\\]\\(#\\)\]\(#\)](#)

# PROTECT YOUR ENVIRONMENT



# Authenticating Users



**Flat file**



**PAM**

Active Directory  
LDAP



**Proxied**



**Google**

# SSL

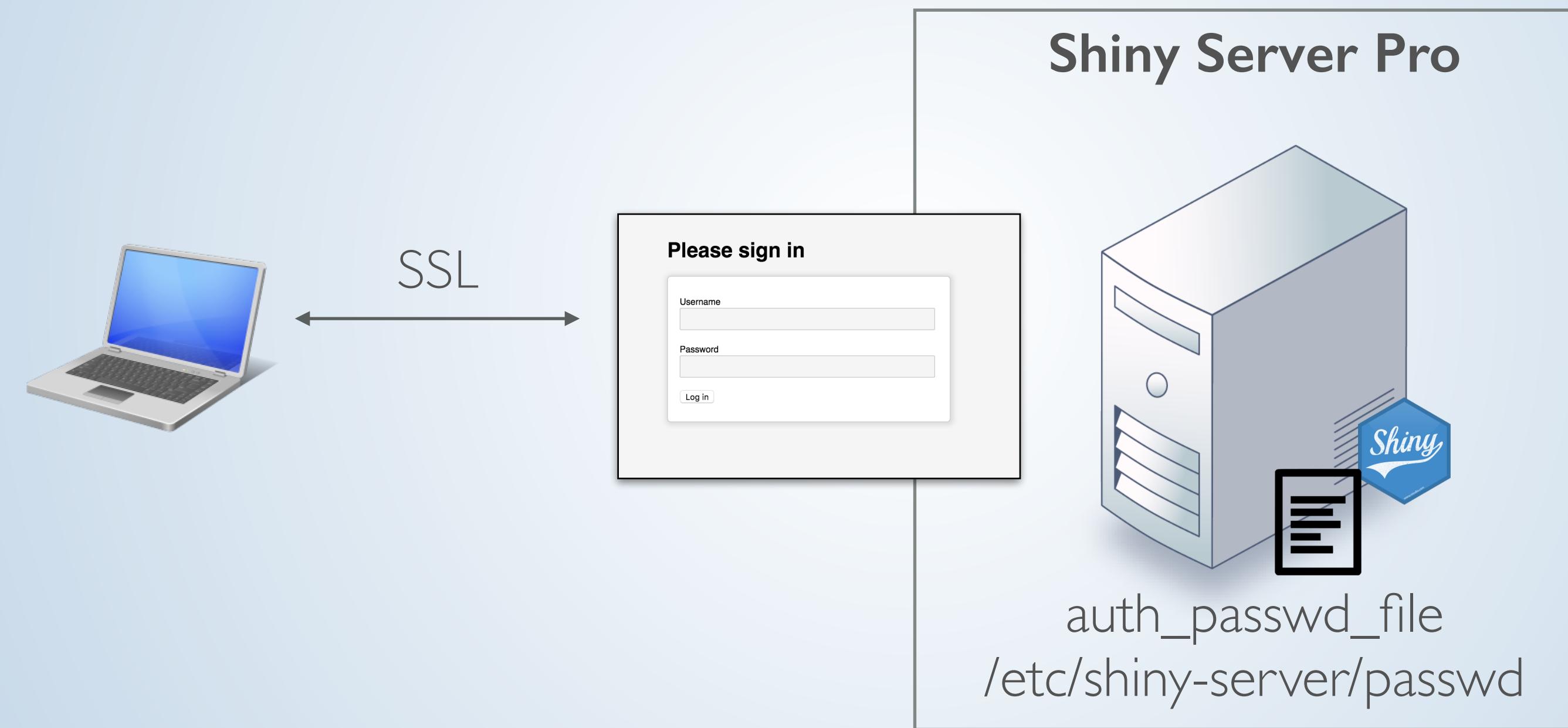
Enabled on a **server** and the **admin** interface by configuring:

1. Path to the SSL key
2. Path to the SSL certificate.

```
server {  
    # Instruct this server to listen on port 443, the default port for HTTPS  
    # traffic  
    listen 443;  
    ssl /etc/shiny-server/ssl-key.pem /etc/shiny-server/ssl.cert;  
  
    ...  
}
```

# Authenticating Users

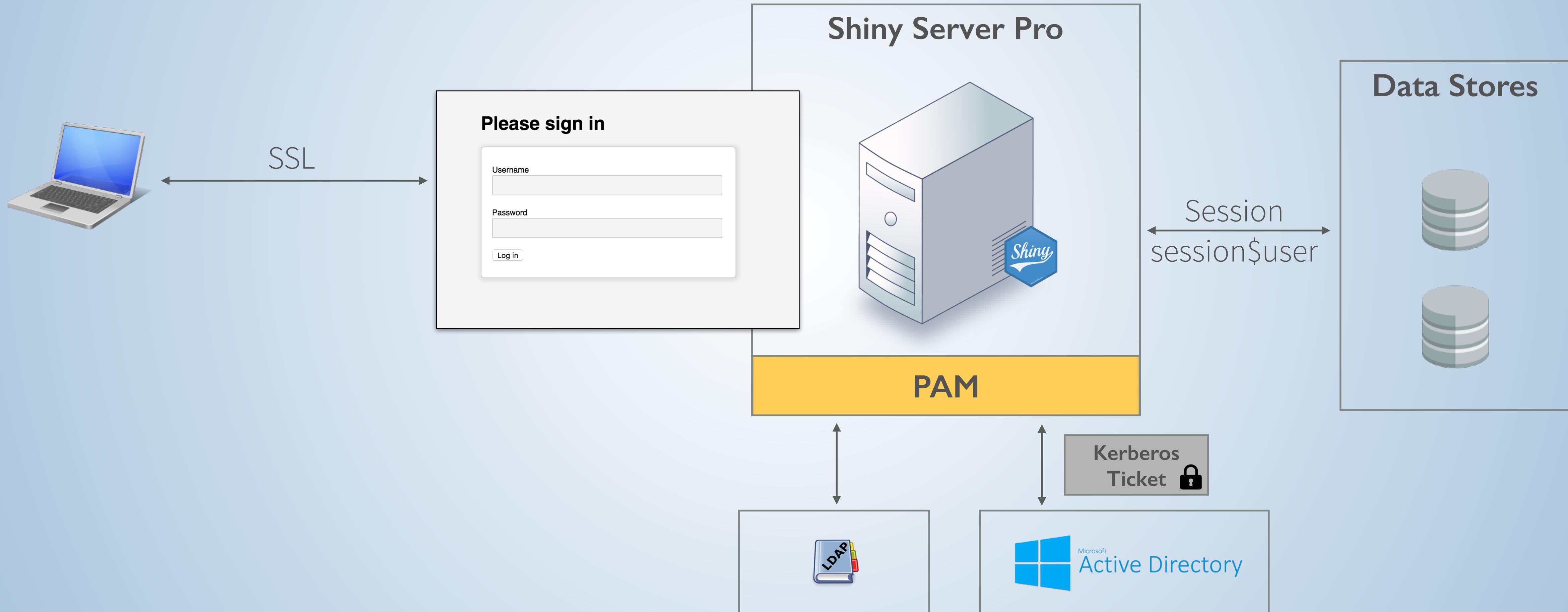
## Flat File



Use the utility manage usernames and passwords:  
`/opt/shiny-server/bin/sspasswd`

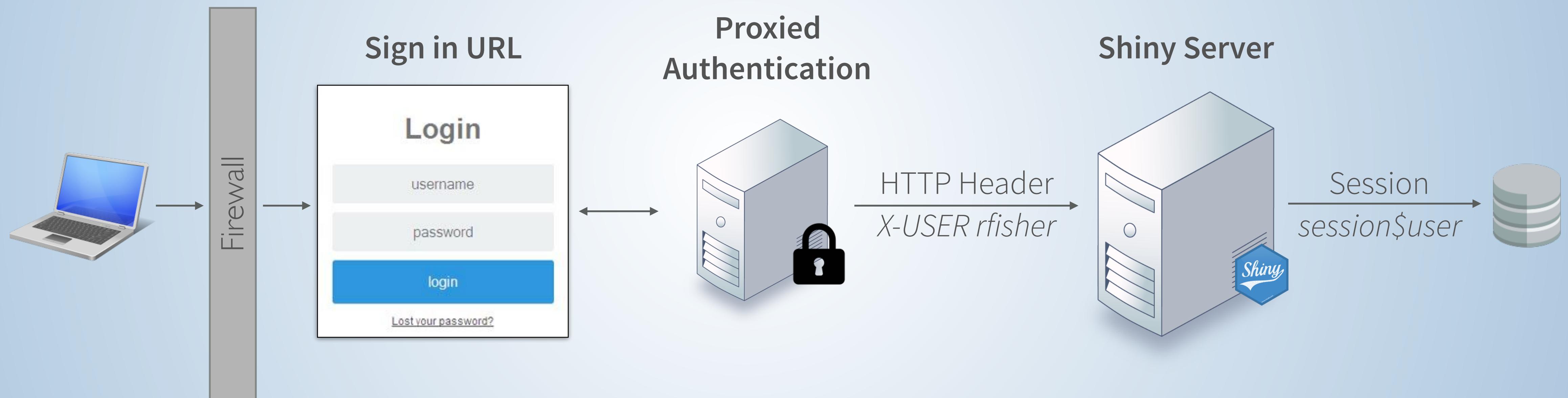
# Authenticating Users

## PAM Authentication



# Authenticating Users

## Proxied Authentication / Single Sign On



4.7 Use `auth_proxy` to define user and group headers

The `auth_proxy` option allows you to configure a header name for usernames and (optionally) another header for groups. Both are case-insensitive.

```
auth_proxy X-USER X-GROUPS;
```

The above configuration would expect a header named `X-USER` to provide the username for each incoming request,

# Authenticating Users

## Google Authentication



# Authenticating Users

## Connecting to data stores

### 4.1 Use the session argument in Shiny apps to access user and group login.

Authentication occurs at the server level, not individual apps.

The current user's username and groups (where applicable) will be available in Shiny (version 0.8 and later) in the **session** parameter of the **shinyServer** function.

For instance:

```
shinyServer(function(input, output, session) {  
  output$username <- reactive({  
    session$user  
  })  
  
  output$groups <- reactive({  
    session$groups  
  })  
})
```

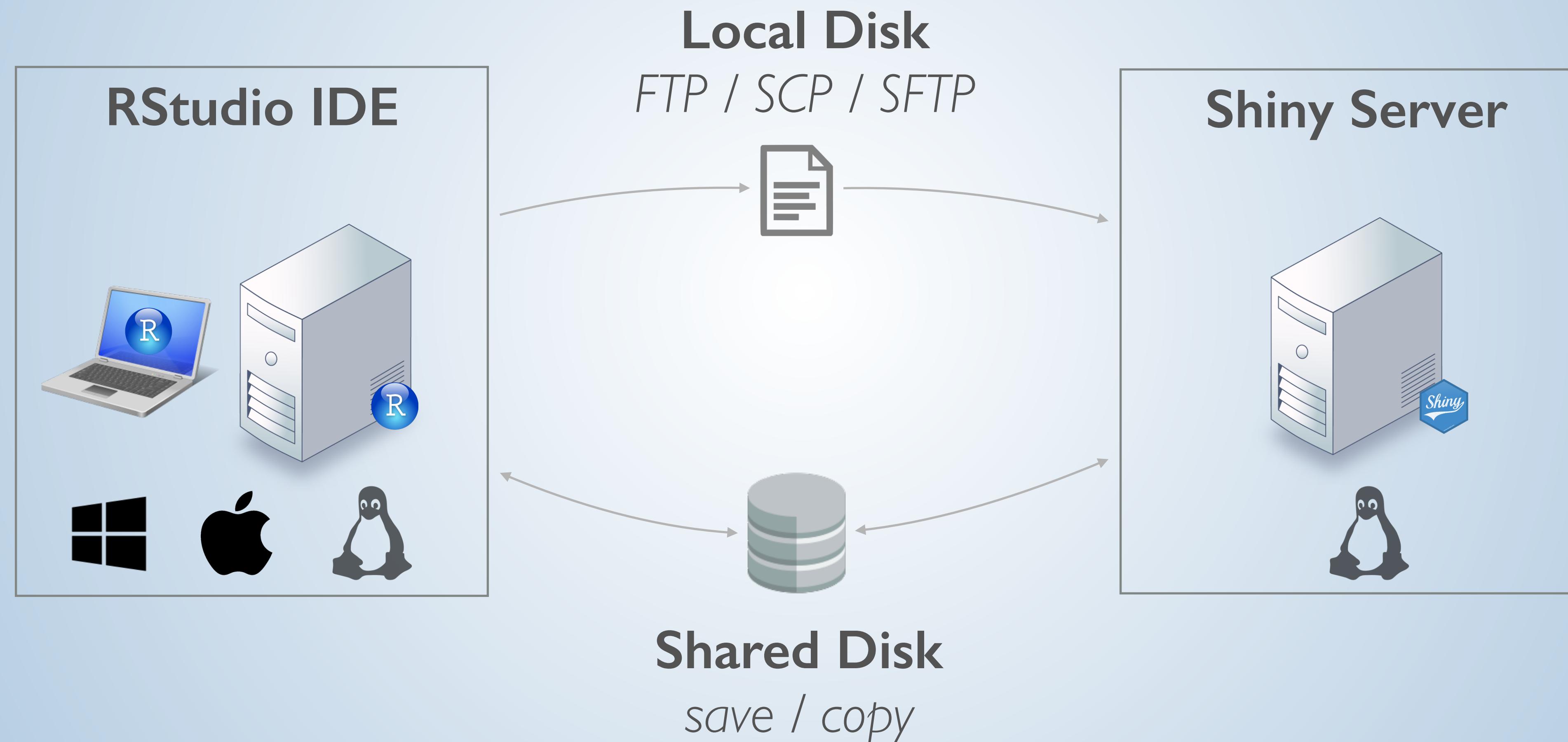
It is important to note that when a user logs in to Shiny Server Pro, that authentication will be applied server-wide. This means that a visitor who authenticates as a particular user on one application will be identified as that user to the other applications on this server to which they connect. There is currently no notion in Shiny Server Pro of logging in exclusively to only one particular application.



# DEPLOY APPS

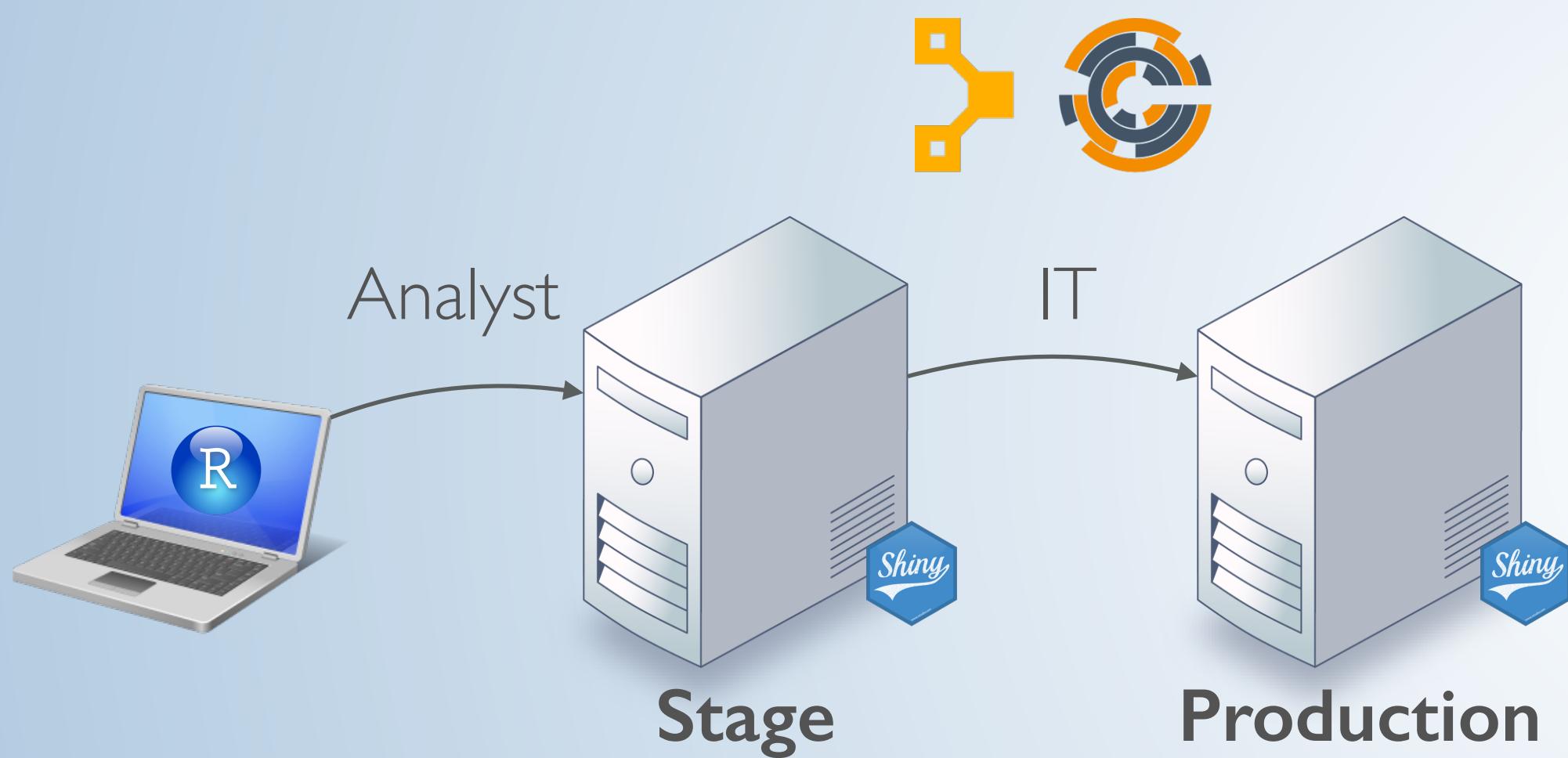


# How do I deploy apps?



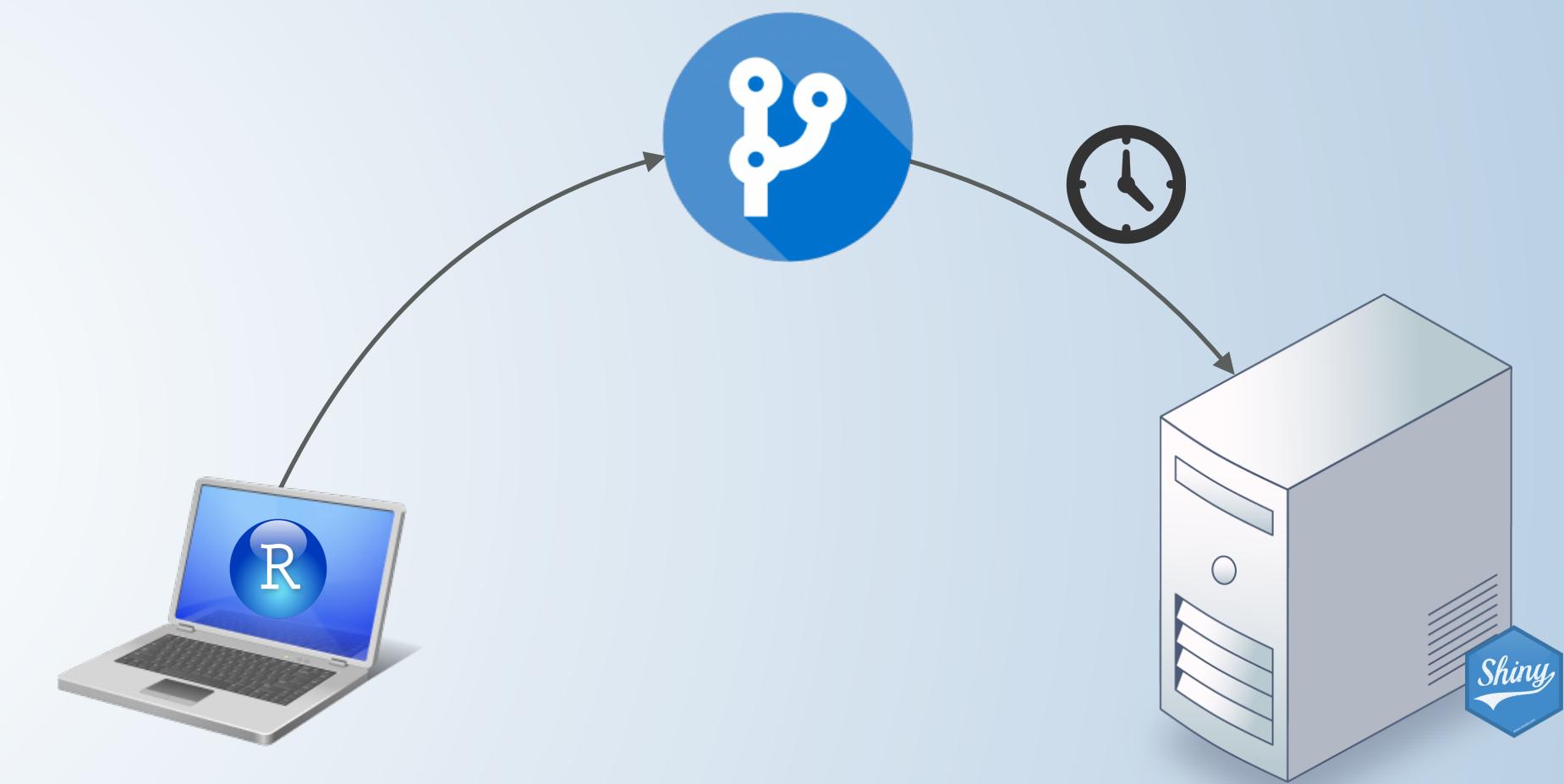
# How do I deploy apps in production?

## Handoff



Analyst stage app  
IT deploys to production

## Version Control



Push to version control and  
automate clones via a scheduler

*Develop and deploy on the same machine*

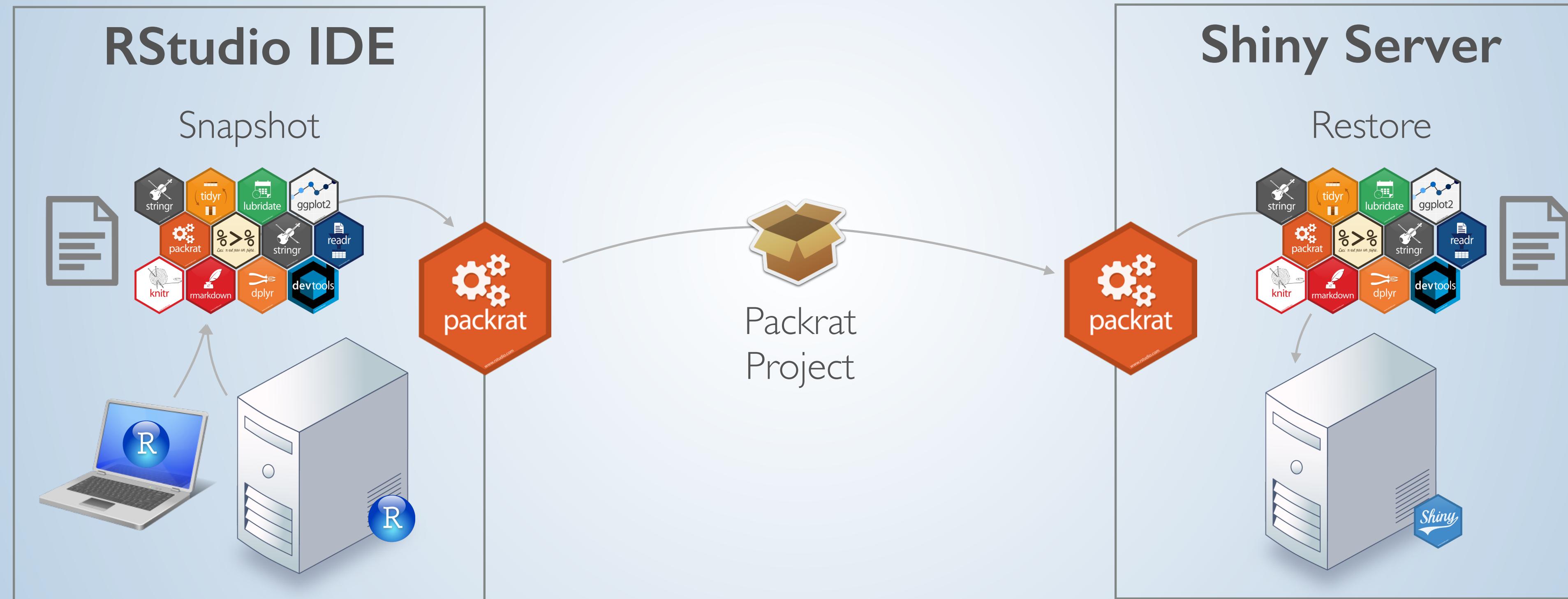
## RStudio Server Pro Shiny Server Pro



# Deploy Apps, R Markdown, and HTML



# Manage R Dependencies with Packrat



# CONTROL YOUR ENVIRONMENT



# Install Shiny Server

## Install (RedHat)

```
$ sudo yum install R  
$ sudo yum install --nogpgcheck \  
shiny-server-1.4.0.rpm
```

## Install directory

/opt/shiny-server/

## Executable

/opt/shiny-server/bin/shiny-server

## Server Log

/var/log/shiny-server.log

## New User

shiny

<http://<myip>:3838/>

**Welcome to Shiny Server!**

If you're seeing this page, that means Shiny Server is installed and running. **Congratulations!**

**What's Next?**

Now you're ready to setup Shiny — if you haven't already — and start deploying your Shiny applications.

If you see a Shiny application running on the right side of this page, then Shiny is configured properly on your server and already running an example. Bravo! You can see this application on your server at </sample-apps/hello>.

If you see a gray box or an error message, then there's a bit more work to do to get Shiny running fully. You can continue with the [installation instructions](#) or use the [Admin Guide](#) for more information. If you're seeing an error message in the panel to the right, you can use it to help diagnose what may be wrong. If you think Shiny is installed and setup properly and things still aren't working, you can look in the Shiny Server log which may have more information about what's wrong. By default, the log is stored in </var/log/shiny-server.log>.

**It's Alive!**

Number of bins:

**Histogram of x**

x	Frequency
50	5
55	10
60	15
65	10
70	10
75	15
80	25
85	15
90	10

# Configure Shiny Server Pro

## /etc/shiny-server/shiny-server.conf

### Server

Defines an HTTP server which will listen on a port and host.

### Location

Defines how a particular URL path should be served.

### Auth\_passwd\_file / Auth\_pam / etc.

Defines the authentication used.

### Admin

Enables the admin interface.

```
# Define the user we should use when spawning R Shiny processes
run_as shiny;

# Define a top-level server which will listen on a port
server {
    # Instruct this server to listen on port 3838
    listen 3838;

    # Define the location available at the base URL
    location / {
        ##### PRO ONLY #####
        # Only up to 20 connections per Shiny process and at most 3 Shiny processes
        # per application. Proactively spawn a new process when our processes reach
        # 90% capacity.
        utilization_scheduler 20 .9 3;
        ##### END PRO ONLY #####
        # Run this location in 'site_dir' mode, which hosts the entire directory
        # tree at '/srv/shiny-server'
        site_dir /srv/shiny-server;

        # Define where we should put the log files for this location
        log_dir /var/log/shiny-server;

        # Should we list the contents of a (non-Shiny-App) directory when the user
        # visits the corresponding URL?
        directory_index on;
    }
}
```

```
# Setup a flat-file authentication system. {.pro}
auth_passwd_file /etc/shiny-server/passwd;

# Define a default admin interface to be run on port 4151. {.pro}
admin 4151 {
    # Only permit the user named `admin` to access the admin interface.
    required_user admin;
}
```

# *Host apps and app directories*

## **Directory**

Serve an entire directory tree.

```
location / {  
    site_dir /srv/shiny-server/  
}
```

## **Single application**

Serve a single application

```
location /myApp {  
    app_dir /srv/shiny-server/myApp;  
}
```

```
run_as :HOME_USER:;  
location /users {  
    user_dirs;  
}
```

## **Per user application directories**

Serve applications in users home

# Specify multiple versions of R

Multiple versions of R can be used to distinguish which applications should use which version of R.

```
run_as shiny;

# In general, use the R 3.0 binary provided below. {.pro}
r_path "/opt/R-3.0/bin/R";

server {
  listen 3838;

  location / {
    site_dir /srv/shiny-server;
    log_dir /var/log/shiny-server;

    location /oldApp {
      # For this particular location, use an older version of R.
      r_path "/opt/R-2.14/bin/R";
    }
  }
}
```

# *Run apps under various users*

## **Home User**

Run process as the user in whose home directory the app exists.

```
location / {  
  run_as :HOME_USER: :AUTH_USER: shiny;  
}
```

## **Auth User**

Run process as the user the visitor logged in as.

## **Shiny**

Run process as “shiny”. Ensure shiny has permissions to access all necessary files.

# Restrict App Access by User or Group

## **required\_user**

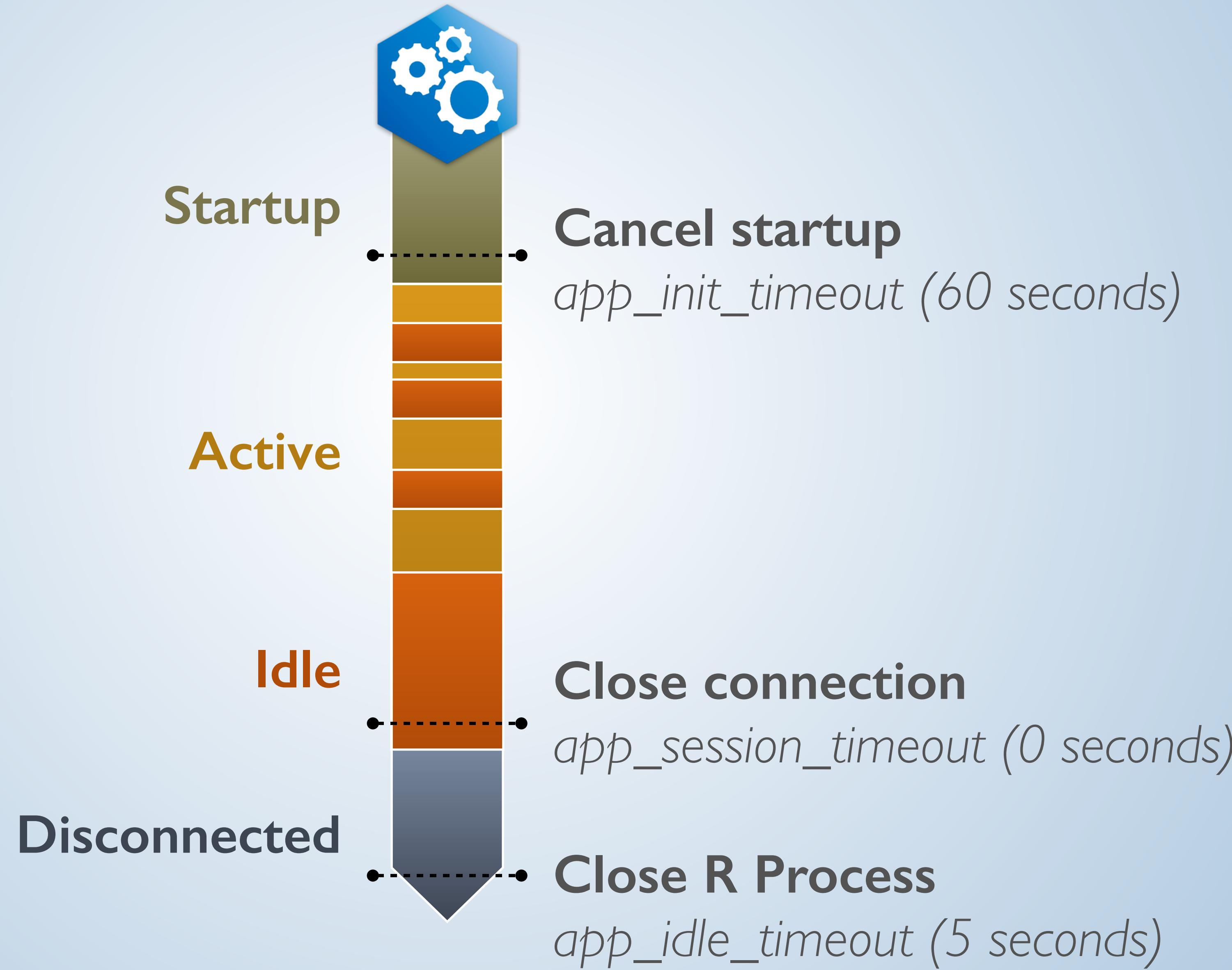
Controls which users should be able to access a particular server, location, or admin.

## **required\_group**

Control access based on the groups of which users are a member.

```
location /app1 {  
    required_user kim tom;  
}  
  
location /app2 {  
    required_group shinyUsers admins;  
}
```

# Set timeouts



**Tip for Large Apps:**  
Applications which require a substantial amount of data to be loaded on startup may merit a longer `app_init_timeout` to give the data time to load, and a longer `app_idle_timeout` as the task of spawning a new process is more expensive and should be minimized.

# Add Google Analytics



## Add globally to server or location

Automatically inserts JavaScript to enable Google Analytics tracking for a particular server or location.

## Add locally to app

[shiny.rstudio.com/articles/google-analytics.html](https://shiny.rstudio.com/articles/google-analytics.html)

```
# Add to server
server {
  google_analytics_id gUA-12345-1;
}

# Add to location
location / {
  google_analytics_id gUA-12345-1;
}
```

# Example Shiny App Directory Contents

Files and Directories	Status	Content
ui.R	Required*	Define the app user interface
server.R	Required*	Define the app functions
app.R	Required*	Define both the app user interface and functions
DESCRIPTION	Optional	Insert metadata before sharing the app
Readme.md	Optional	Describe the app to developers
restart.txt	Optional	Restart an application by altering the modified time
.shiny_app.conf	Optional	Override server settings
.git/	Optional	Use version control with this directory
packrat/	Optional	Bundle all R package dependencies into this directory
www/	Optional	Put javascript and css into this directory
data/	Optional	Put data files into this directory

\* You must define either `server.R` or `app.R`

# Shiny Server Root Requirements



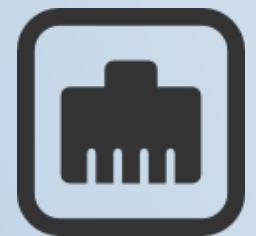
## 1. Running as other users

Running apps as multiple users or as home users (e.g. user\_dirs setting)



## 2. Authentication

Using PAM or auth\_google authentication



## 3. Privileged Ports

Using a privileged port (1-1024)



## 4. Tracking

Tracking and collecting historic metrics using metrics\_user



# ADMIN DASHBOARD



# Administrative Dashboards

Shiny Server Dashboard Applications Processes Connections admin ▾

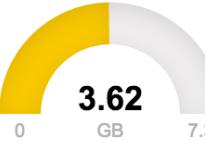
**Evaluation.** Your trial license is set to expire in 6848 days.

**Load**



**0.90**

**Memory**



**3.62** GB

**Active Users**

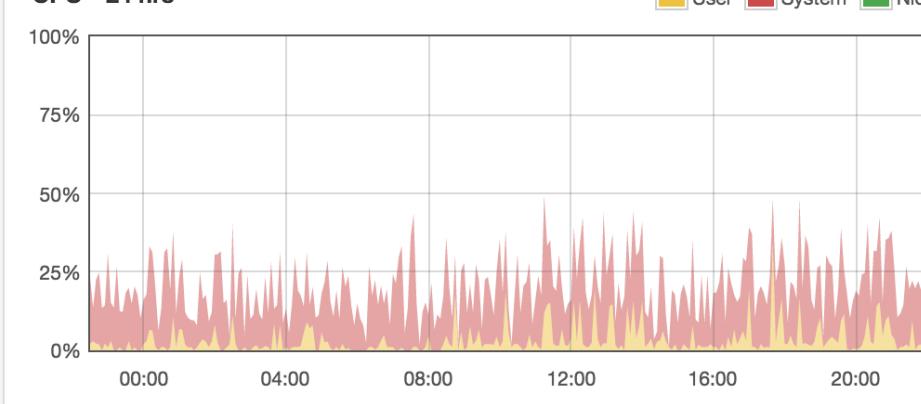
**2**

**Open Connections**

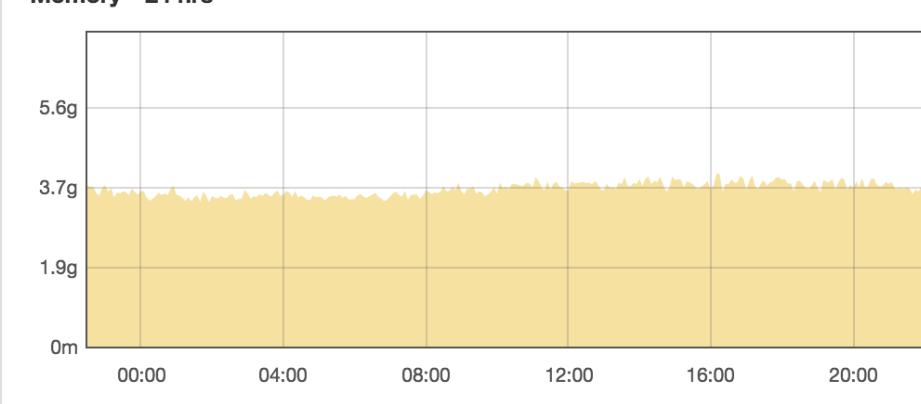
**10**

**System History**

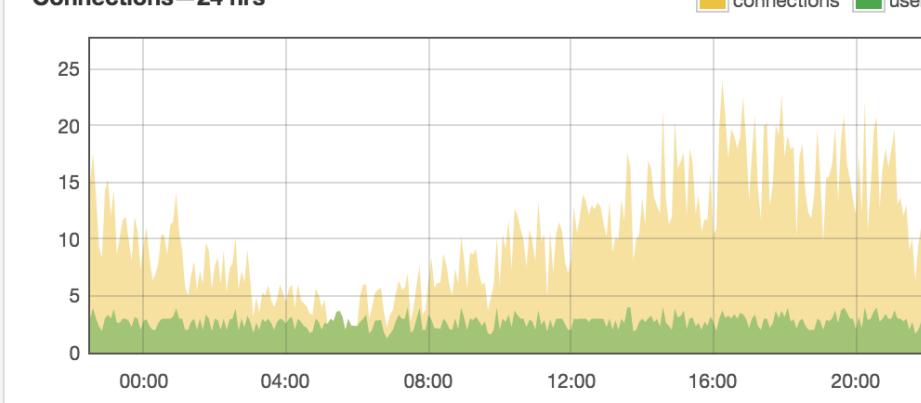
**CPU—24 hrs**



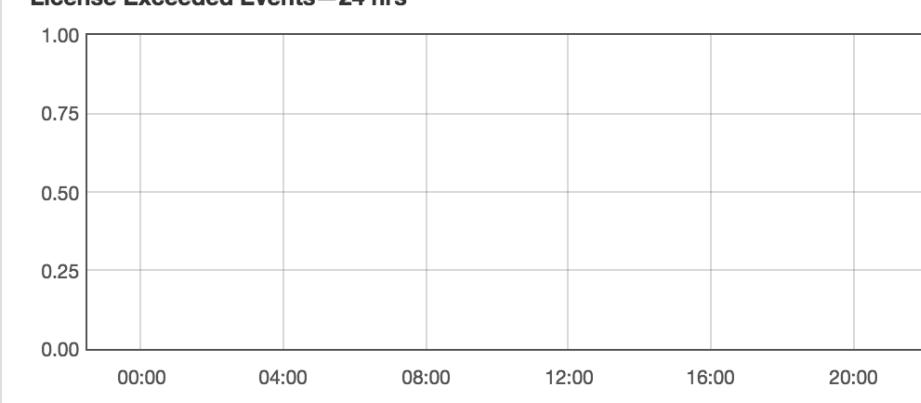
**Memory—24 hrs**



**Connections—24 hrs**



**License Exceeded Events—24 hrs**



**System Information**

Status	Authentication
License Status	Evaluation will expire in 6848 days.
Licensed Concurrent Users	Unlimited
Licensed Concurrent Connections	Unlimited
Running Shiny Applications	4
Running Shiny Processes	4

Shiny Server Dashboard Applications Processes Connections admin ▾

**Evaluation.** Your trial license is set to expire in 6848 days.

**/srv/shiny-server/ssp-demo/07\_widgets**

**Current Status**

CPU Usage	0.00 of 1
Memory Usage	49MB
Current Processes	1
Busy Sessions	0
Active HTTP Requests	0
Recent Quota Exceeded Events	0

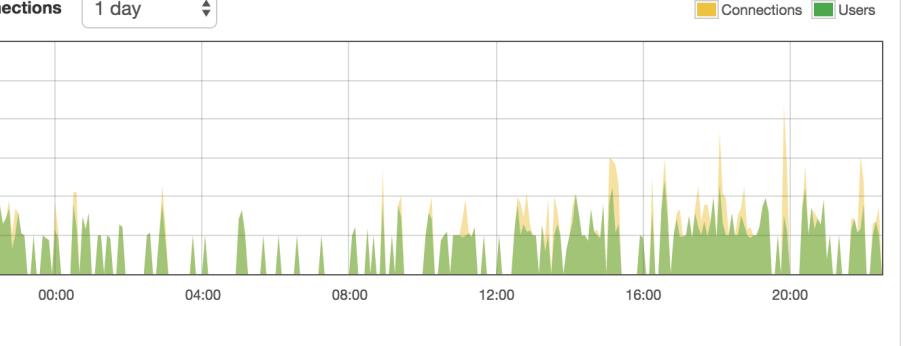
**Open Connections**

**4**

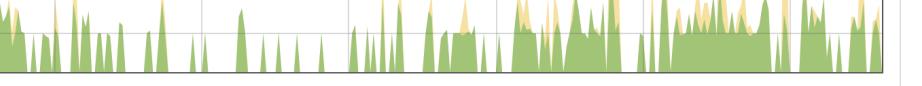
**Active Users**

**2**

**Connections** 1 day

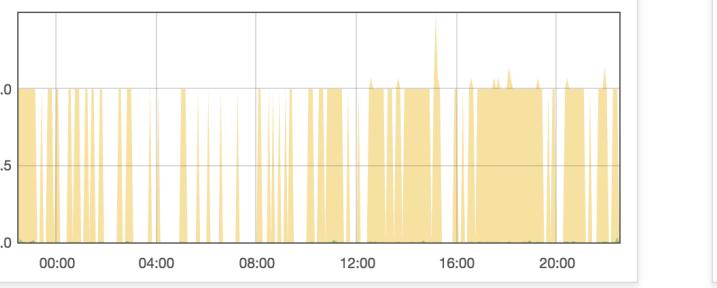


**Average Latency**

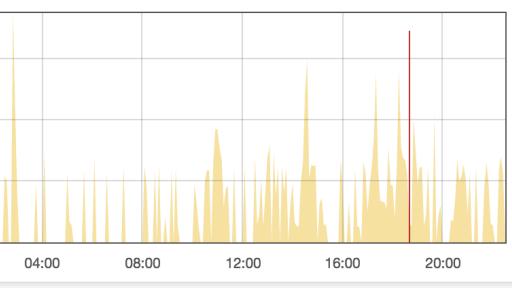


**28** ms

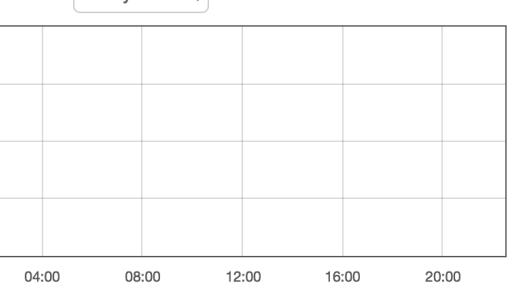
**Processes** 1 day



**Latency (ms)** 1 day



**Declined Requests/min** 1 day



**Active Processes**

Process	Application	Conn's	Latency	CPU	RAM
fd6be8b...	/srv/shiny-server/ssp-demo/07_widgets	4	28ms	0%	49MB

**Active Connections**

User	Initialized	Status
xhr-p	172.17.42.1	4 minutes ago
xhr-p	172.17.42.1	4 minutes ago
xhr-p	172.17.42.1	2 minutes ago
xhr-p	172.17.42.1	2 minutes ago

# Killing Process and Connections

From the admin dashboards

### Active Processes

Process	Application	Conn's	Latency	CPU	RAM
a577164...	/home/nathan/ShinyApps/shiny-examples/001-hello	3	102ms	0%	61MB <button>Kill</button>
0a43ec2...	/home/nathan/ShinyApps/shiny-examples/001-hello	1	166ms	0%	58MB <button>Kill</button>

10 25 50 100

### Active Connections

User	Initialized	Status
nathan (68.134.36.58)	less than a minute ago	Idle <button>Kill</button>
nathan (68.134.36.58)	less than a minute ago	Idle <button>Kill</button>
nathan (68.134.36.58)	less than a minute ago	Idle <button>Kill</button>
nathan (68.134.36.58)	less than a minute ago	Idle <button>Kill</button>

10 25 50 100

# *Health Check Endpoint*

[http://colorado.rstudio.com:3838/\\_health-check](http://colorado.rstudio.com:3838/_health-check)

## **Customizable**

Specify whatever format you like

## **Multiple URL's**

Supports multiple endpoints

## **Pluggable**

Use with other systems

**http://<IP:Port>/\_health-check**

```
server-version: 1.2.3.4
active-connections: 8
active-apps: 2
active-processes: 3
cpu-percent: 13
memory-percent: 49
swap-percent: 39.1
load-average: 1.01953125
```

# Graphite

Output RRD metrics to Graphite or any other engine compatible with the Carbon protocol. This can be enabled in addition to, or in place of, local RRD file monitoring.

```
# Enable graphite metrics
graphite_enabled 127.0.0.1 2003;

# Disable RRD metrics
rrd_disabled;

# Run admin dashboard on port 4151
admin 4151 {
    required_user admin;
}
```

# SCALE YOUR ENVIRONMENT



# *Scaling Concurrent User Capacity*

## **Shiny Server Pro**

Included

Standard User Pack

Large User Pack

## **Concurrent Users**

**20**

**+ 20**

**+ 150**



# Methods for Scaling

## Scaling Up

Simplicity



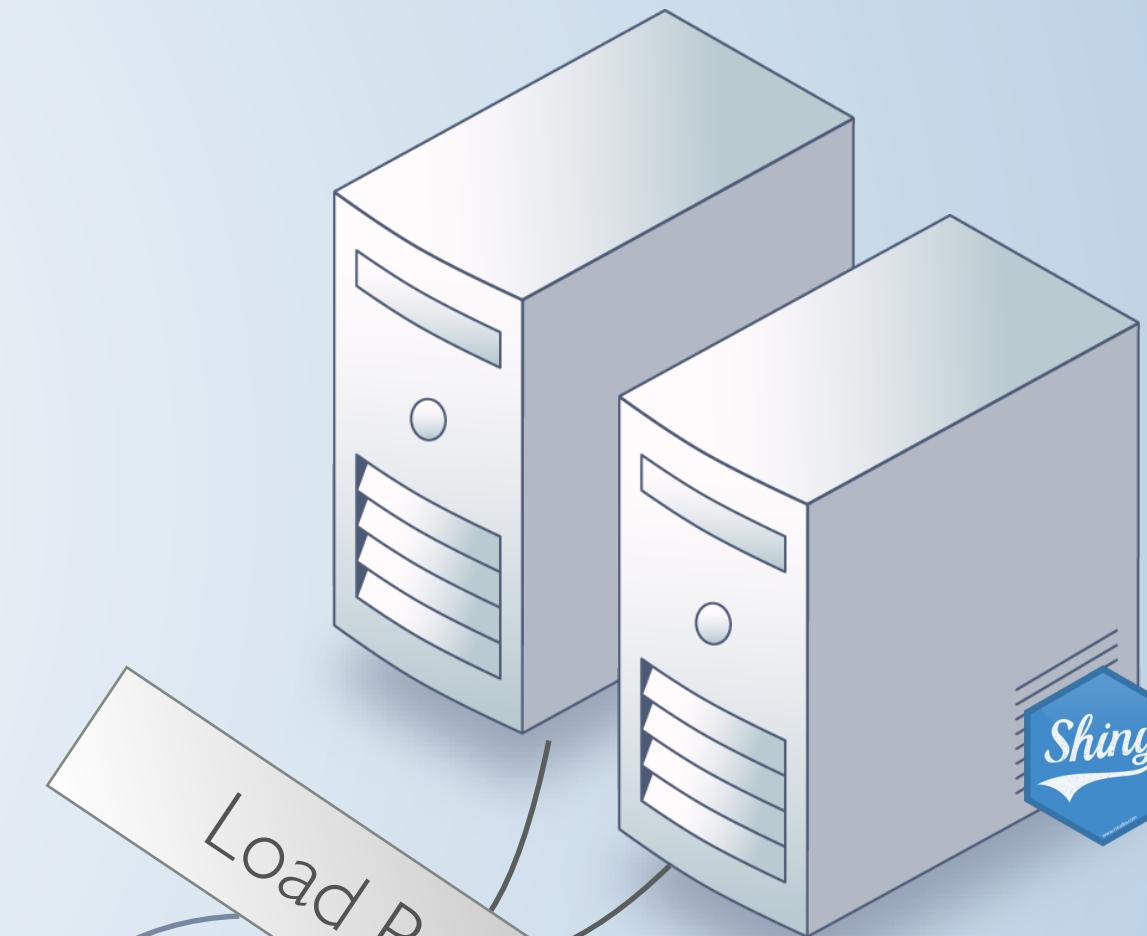
**20+20**



**vs.**

## Scaling Out

High Availability



**20**



**20**



# Concurrent User vs Connection



## Concurrent User

One Human

One Active Browser

One Shiny Server Pro



## Connection

One concurrent user

Four connections

# *Scaling and Tuning Shiny Server Pro*



Applications

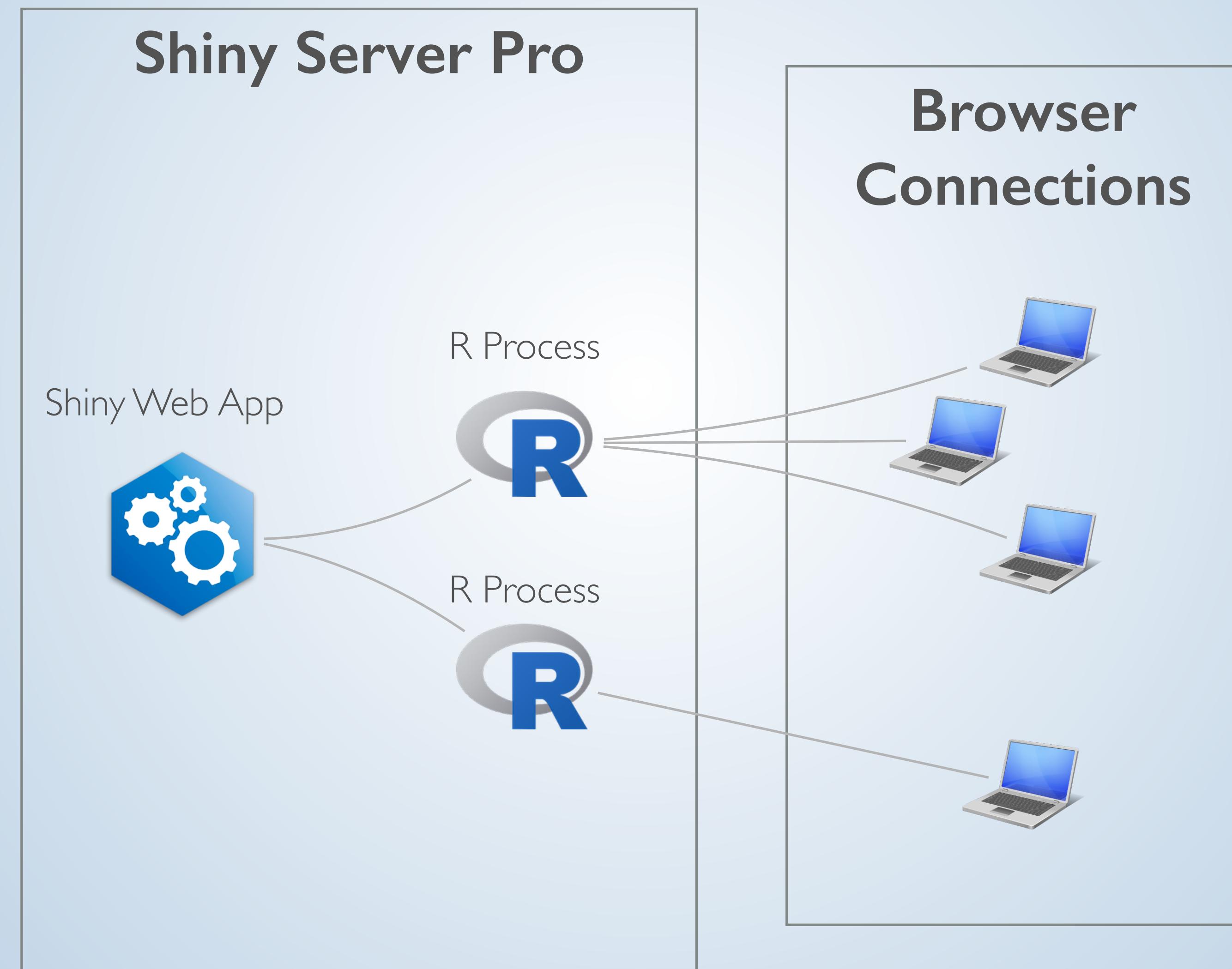


R Processes



Connections

# Scaling Via Multiple R Processes



# Utilization Scheduler

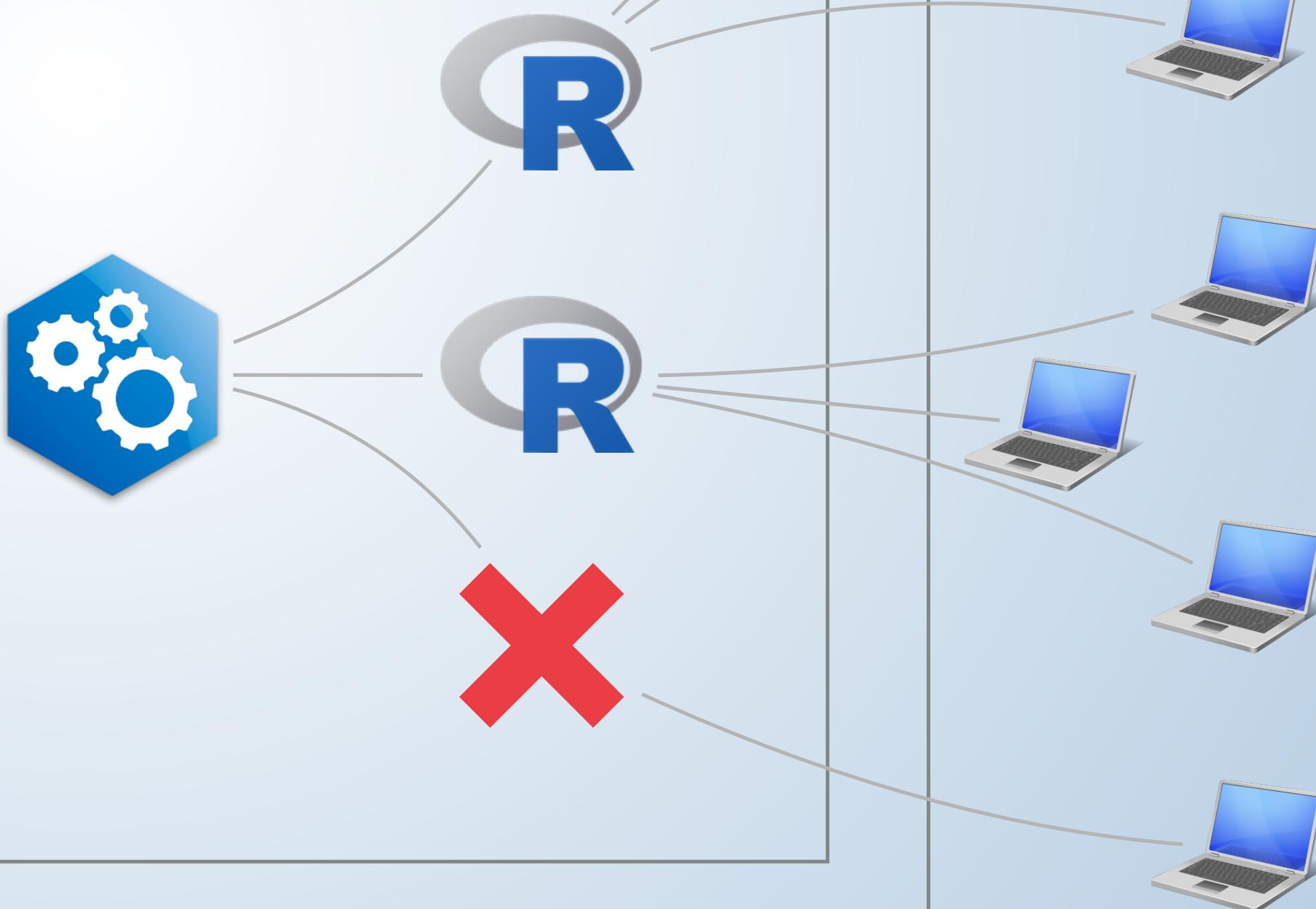
```
location /shinyApp1 {  
  # Define the scheduler to use for this application  
  utilization_scheduler 5 0.5 3;  
  
  ...  
}
```

**maxRequestsPerProc:** The number of concurrent connections which a single R process can support

**loadFactor:** The "trigger percentage" (on a scale from 0 to 1) at which point a new R process should be spawned.

**maxProc:** The maximum number of processes which should exist concurrently associated with this Shiny application

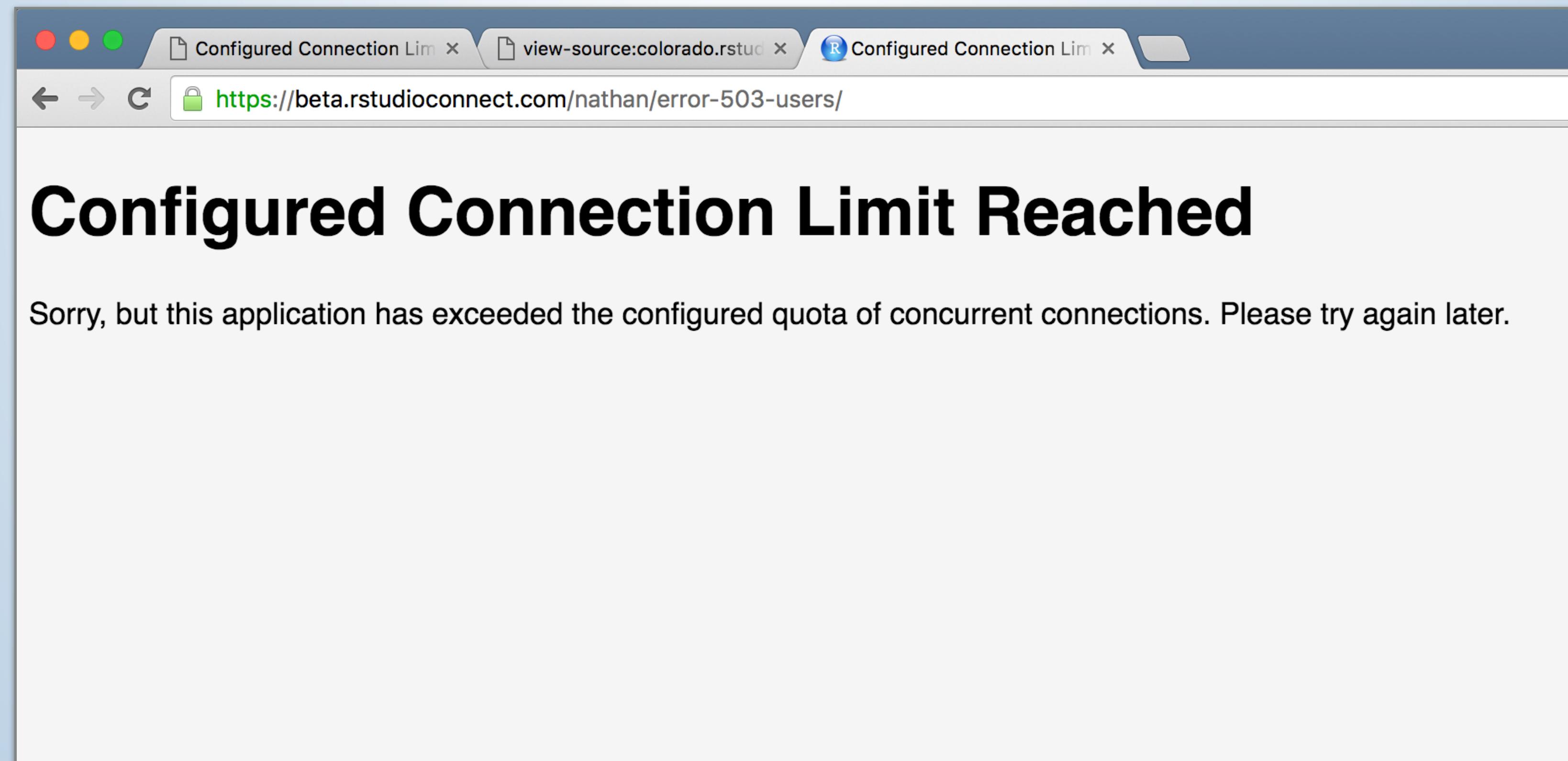
**Example**  
Utilization\_scheduler  
3 | 2



## Connections

# *Connection Limit Exceeded*

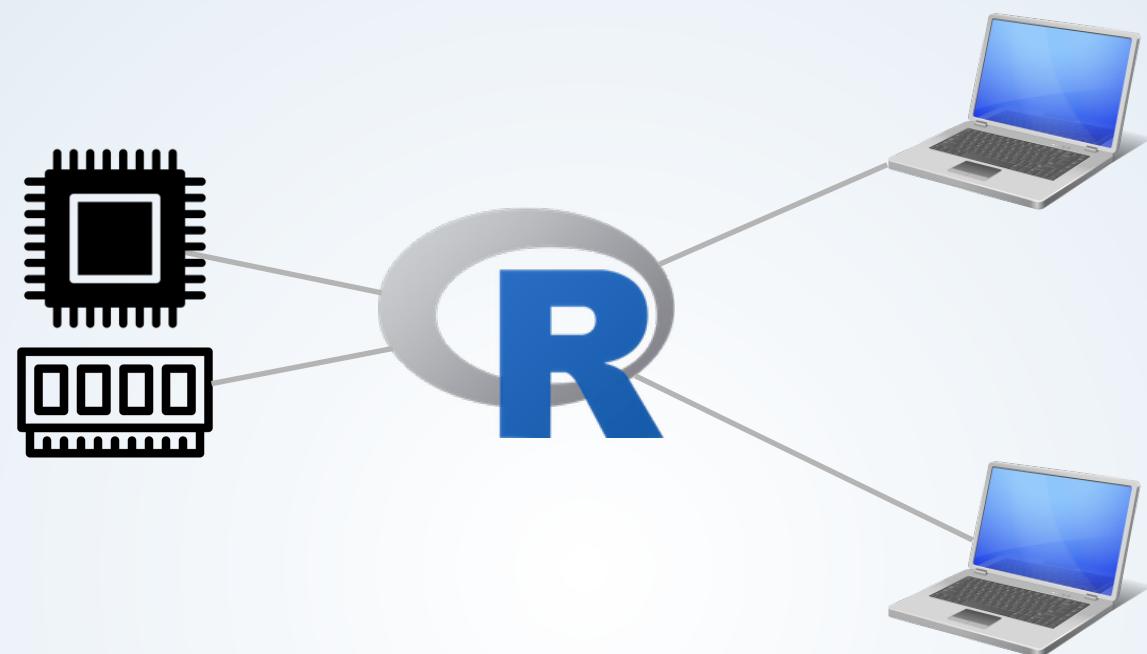
error-503-users: The application has exceeded its maximum # of users



# Which Is Better?

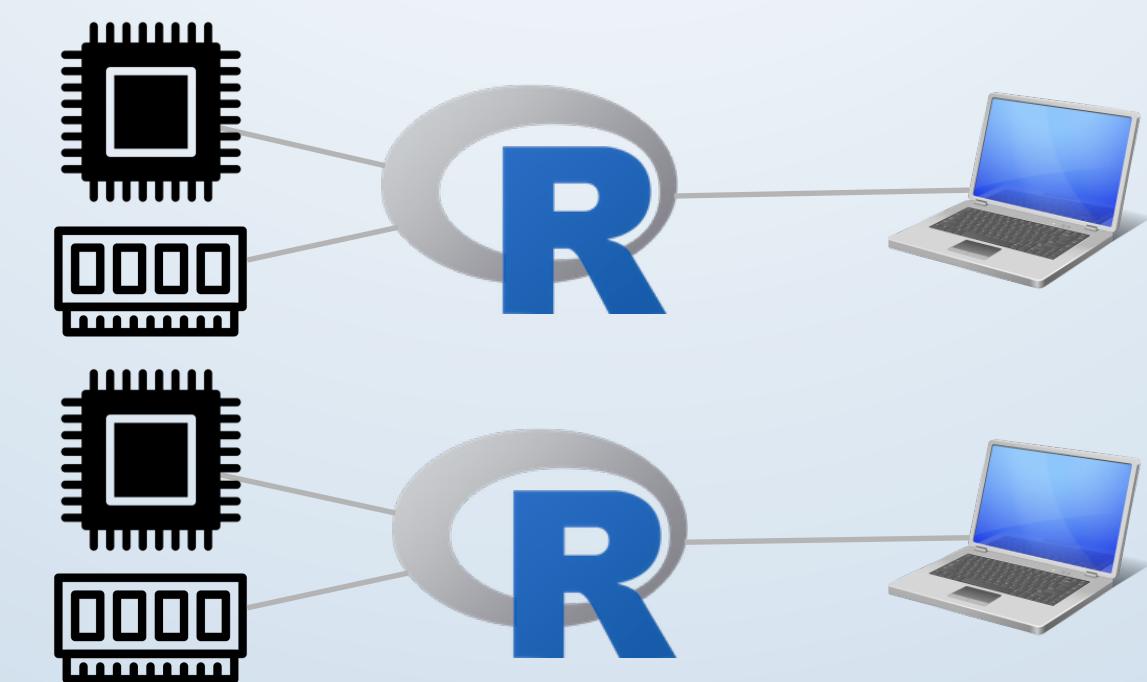
## Users Share R Processes

Faster start up; Use less resources



## Users get their own R process

No contention (potentially) for resources



# Admin Dashboard

<http://shiny-docker-gallery.us-east-1.rstudiосervices.com:4151/#/dashboard#%23%23>

Applications



Shiny Server    Dashboard    Applications    Processes    Connections    nathan ▾

/home/nathan/ShinyApps/shiny-examples/001-hello

Current Status

CPU Usage	0.00 of 2
Memory Usage	117MB
Current Processes	2
Busy Sessions	0

Open Connections    4

Active Users    1

Average Latency

Process Details

Statistics

Memory Usage	60MB
HTTP Requests	0
Latency	98ms
PID	26449

Connections    3

Users    1

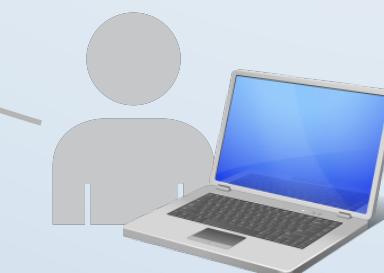
Processes    1

Time Initialized

R Processes



Connections



Shiny Server    Dashboard    Applications    Processes    Connections    nathan ▾

Active Connections

App Directory	User	Initialized	Status
/home/nathan/ShinyApps/shiny-examples/001-hello	nathan (68.134.36.58)	2 minutes ago	Idle
/home/nathan/ShinyApps/shiny-examples/001-hello	nathan (68.134.36.58)	2 minutes ago	Idle
/home/nathan/ShinyApps/shiny-examples/001-hello	nathan (68.134.36.58)	2 minutes ago	Idle
/home/nathan/ShinyApps/shiny-examples/001-hello	nathan (68.134.36.58)	2 minutes ago	Idle

Filter by App

10 25 50 100

# Shiny Server Pro Server Pro Features



## Security & Authentication

Password file authentication

LDAP and Active Directory authentication

Group based authorization

SSL support



## Tuning & Scaling

Scale applications across multiple processes

View and manage active sessions

Allocate resources on a per application basis

Define application concurrency limits



## Server Monitoring

System performance and resource metrics

Per-application: Usage, performance, and resource metrics

Health check end point



# SHINY SERVER PRO

