

Overview

Posit Team bundles Posit's premier data science tools, including:

posit™ Workbench

Development environment for creating insights

posit™ Connect

Hosting environment for accessing and sharing insights

posit™ Package Manager

R and Python package repository management

1 Open New Session in Posit Workbench

Open a new development session by clicking **+ New Session**

Available IDEs

Cluster Options
May vary depending how Posit Workbench is configured in your environment

2 Configure Package Repositories

Check Current Repositories

```
options("repos")
```

```
pip config list
```

Example Posit Package Manager Repository URL

<https://pkg.demo.posit.team/pypi/latest/simple>

Tip: Talk to your administrator about configuring your repositories to ensure they persist across sessions

Configure Repositories with Posit Package Manager

```
options(repos = c(CRAN = "https://your-package-manager-url.com"))
```

```
pip config set global.index-url https://your-package-manager-url.com
```

Obtain your repository URL by clicking **SETUP** on Posit Package Manager

3 Access Your Data

Access your data within Posit Team no matter where it lives and what format it's in:

Local or Shared Directory

Examples: data.csv, data.txt, data.parquet

APIs

Interact with APIs using packages such as `{requests}` in Python or `{httr2}` and `{jsonlite}` in R

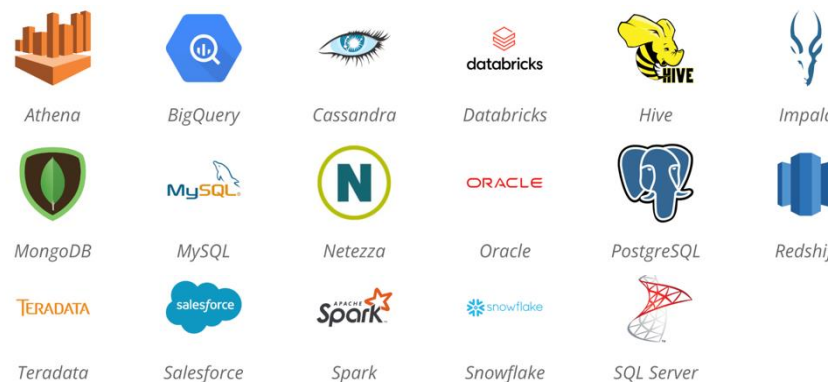
Pins

Store and retrieve Python/R objects including data and models

Databases

Connect to a variety of database types using DBI/DB-API/ODBC

Posit's Professional ODBC Drivers



Tip: Never hard code credentials! Use environment variables or talk to your administrator about setting up a DSN

4 Run your Code

Support for multiple Python and R versions

Leverage virtual environments and project-oriented workflows

RStudio and VS Code: Submit long-running Python/R jobs to Posit Workbench server to run in independent sessions

DB-API
Python Database API

Example
`con = sqlite3.connect('example.db')`

DBI
Database Interface

Example
`con <- DBI::dbConnect(RPostgres::Postgres(), hostname = "db.company.com", port = 5432)`

ODBC
Open Database Connectivity

Examples

```
con = pyodbc.connect(
  driver = 'PostgreSQL',
  database = 'test_db',
  server = 'localhost',
  port = 5432,
  uid = os.getenv('DB_USER'),
  pwd = os.getenv('DB_PASSWORD'))
```

```
con <- DBI::dbConnect(odbc::odbc(),
  driver = "PostgreSQL Driver",
  database = "test_db",
  UID = Sys.getenv("DB_USER"),
  PWD = Sys.getenv("DB_PASSWORD"),
  host = "localhost",
  port = 5432)
```

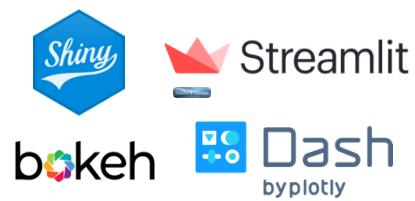
5 Publish your Content to Posit Connect

Supported Content Types on Posit Connect

Documents



Applications



APIs



Pins



Used from Tableau workbooks to make real-time requests from Tableau to your Python and R code.

Methods for Publishing to Posit Connect



Publish content directly from R using the rsconnect R package



Publish content using the CLI within the rsconnect-python package



Publish content with the click of a button using push button deployment



VS Code Posit Publisher Extension



Publish directly from a git repository

6 Share and Control your Content on Posit Connect

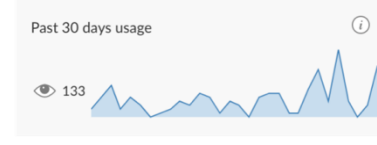
Content controls are accessed by clicking the icon

Info and Content Metadata

- Change content title
- Content information
- Add thumbnail
- Usage data



Example usage data for a Shiny application



Example usage data for an R Markdown report

More usage info here:
pos.it/cookbook

Runtime Settings

Tune and scale **applications** and **APIs** by modifying the number of processes (e.g., Python and R sessions) and connections per process

Min processes:

Max processes:

Max connections per process:

In this above configuration, the content can accommodate up to 100 concurrent connections (20 x 5). There will also be at least one process always running, meaning the content never goes to sleep

Other Runtime Settings

Easily modify how your content runs by modifying any of the below settings. **Note:** some options may not be available depending on how Posit Connect is configured in your environment

PROCESS EXECUTION

Who runs this content on the server:

EXECUTION ENVIRONMENT

Last time:

Next time:

GPU SETTINGS

AMD GPUs Requested:

Nvidia GPUs Requested:

CPU & RAM SETTINGS

Initial Number of CPUs:

Max Number of CPUs:

Initial RAM Requested (GiB):

Max RAM (GiB):

PYTHON ENVIRONMENT MANAGEMENT

Last time:

Next time:

R ENVIRONMENT MANAGEMENT

Last time:

Next time:

Schedule and Emailing

Documents can be configured to execute on a schedule

Timezone:

Start date & time:

Schedule type:

Run every 1 day.

Run every weekday (Monday to Friday).

Publish output after it is generated

Send email after update

Schedule type options

- By Minute
- Hourly
- Daily
- Weekly
- Semimonthly
- Monthly
- Yearly

Email configuration options

Owners are always notified unless they opt-out

Send to all collaborators

Send to all viewers

Additional Recipients:

Tip: Learn how to send custom and conditional emails:
pos.it/rmd_email & pos.it/qmd_email

Tags

Tags make content organization, discovery, and filtering easier across the Connect dashboard

Technology

- Deep Learning
- Output Types
- Email
- BackEnds
- Dashboards
- Scheduled

Projects and Presentations

- Calendars
- QuickStart
- Production Webinar

Environment Variables

Securely pass configuration options to your content as environment variables including **API Keys and Passwords**

Environment Variables

Note: Do not wrap your text in quotation marks; all symbols become part of the value available to your code.

Name:

Value:

Need Help? Want to learn More?

Questions about sales and licensing → sales@posit.co
 Technical Issues → support@posit.co
 Other questions? → info@posit.co



Posit Team docs → <https://docs.posit.co>
 Release notes → <https://docs.posit.co/release-notes.html>



Open forum for any open source or Posit Team question:
<https://forum.posit.co>



Join our monthly Posit Team Demos on the last Wednesday of every month:
pos.it/team-demo