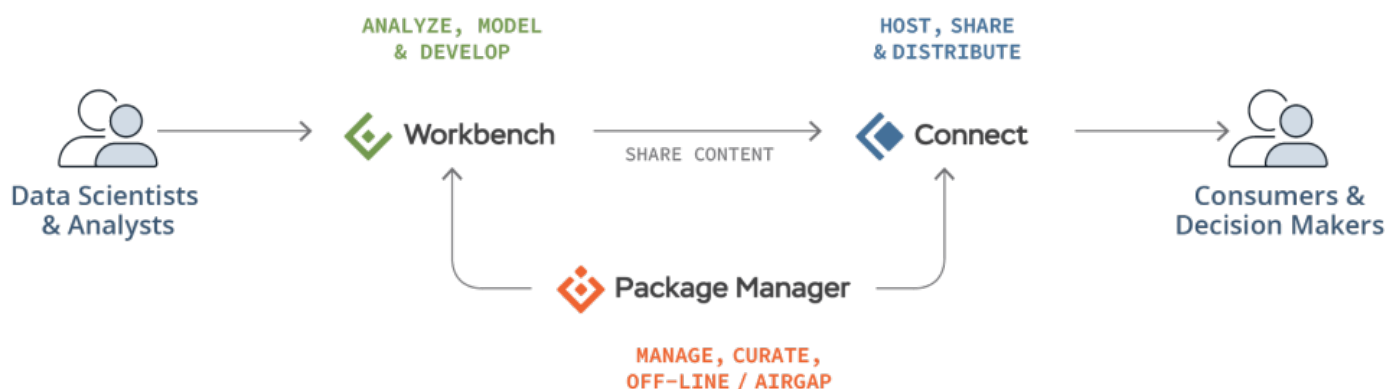


Posit Team Bundle: product tier descriptions

Posit Team Bundle

The complete self-hosted solution for every data science team using Python or R.

All the enterprise-friendly features you expect: centralized management, secure access to data, scalable workflows, and commercial support. Reduce the cost and uncertainty of building your own deployment solutions.



Posit Workbench

The enterprise-ready home for open source, code-first data science teams to collaboratively analyze data and build models, applications, and reproducible reports at scale. It supports both R and Python, helping make data scientists more productive with secure access to all the development environments they love, including RStudio, Jupyter and VS Code.

Posit Connect

Data scientists use Posit Connect to automate time-consuming tasks with code, distribute custom-built tools and solutions across teams, and securely share insights with decision makers. Connect supports the work you create in both R and Python. Publish interactive applications, documents, notebooks, and dashboards. Deploy models as HTTP APIs, and configure reports to run on a custom schedule.

Posit Package Manager

The repository management server to organize and centralize R and Python packages across your organization. Give your users access to the Data Science packages they want — public or private — and limit access to those you don't want them to use.

Posit Workbench

Feature	BSC	ENH	ADV	Description
Server activations				
Activations	1+1	2+2	∞	Production + Staging
Minimum user count	5	10	10	
Maximum user count	10	100	∞	Your contract and license determines the actual named user count allowed on your system.
Disaster recovery activations	1	2	∞	Stand up a passive disaster recovery environment that matches your production installation. Refer to the Software License Description .
Essentials				
Access Workbench through browser	✓	✓	✓	Access Workbench securely from any web browser and prevent exposing data directly to desktop.
Multiple Python / R versions	✓	✓	✓	Provides consistent versions of R/Python for all users across team, making collaboration easier.
Concurrent sessions	✓	✓	✓	Users can run multiple sessions of JupyterLab, Positron Pro, RStudio Pro and VS Code at the same time.
Advanced Python / R session management	✓	✓	✓	Users can manage sessions in Jupyter Notebook, JupyterLab, VS Code, Positron Pro, and RStudio Pro via a single platform. RStudio Pro and Positron Pro users can dynamically change the version of R (or Python in Positron) via built-in UI.
JupyterLab, Positron Pro, RStudio Pro and VS Code	✓	✓	✓	Users have access to all of the best open-source IDEs, including JupyterLab, Positron Pro, RStudio Pro and VS Code all in a enterprise-supported platform.
Feature Limits	✓	✓	✓	For compliance or increased security, administrators can turn off features such as file upload/download, access to the terminal, package installation, and more.
Supported ODBC drivers	✓	✓	✓	Workbench supports Posit's Professional Open Database Connectivity (ODBC) drivers, allowing Workbench to easily 'pull in' data from database management systems.

Server-grade compute	✓	✓	✓	Server grade compute provides shared and efficient resource allocation for user populations of varying size.
Administrative Dashboard	✓	✓	✓	Admin UI for managing users, sessions, and a simple dashboard for viewing server utilization.
Posit Assistant	✓	✓	✓	A platform-agnostic data analysis assistant for Positron, RStudio, and beyond. Connects to your R and Python runtimes. Inspects dataframes, understands schemas, and generates code that works with your actual data.

Execution

Configurable computational resources	✓	✓	✓	Depending on configuration, admins are able to limit available resources per user or user group within guardrails, meaning that certain users may have access to small resources while power users have access to more resources for computationally intensive work.
Local Sessions	✓	✓	✓	The Posit Workbench application and all sessions execute on the same Linux host. Load-balanced server and cluster or container-based configurations on Slurm or Kubernetes are supported where applicable.
Load Balancing and/or High-Availability		✓	✓	Workbench automatically distributes incoming network traffic across backend servers to enable peak performance and minimize downtime. High-availability can be used to avoid overall Workbench downtime in the case of a node failure.
Off-Host Execution on Kubernetes			✓	Users can access Workbench backed by native integration with Kubernetes clusters on-premise or in Cloud. This also adds support for Container-backed sessions for more distinct isolation of environments including Python / R or system dependencies.
Off-Host Execution on SLURM			✓	Users can access Workbench backed by native integration with SLURM, a high performance computing (HPC) framework. Also includes Apptainer/Singularity -- a variant of Slurm with support for container-backed sessions.

Governance & Security

SSO / Authentication	✓	✓	✓	Posit Workbench access is gated behind enterprise-grade authentication, including SSO via SAML, OIDC, Okta, AzureAD/EntraID. PAM/Kerberos or LDAP are also supported.
----------------------	---	---	---	---

Monitoring	✓	✓	Publish Workbench session, job, HTTP, system, and licensing metrics to a Prometheus-compatible /metrics endpoint, with support for load-balanced multi-node aggregation and a ready-to-use Grafana dashboard for visualization.
Session Auditing	✓	✓	Capture historical audit records of every Workbench session and launcher job, including user, IDE type, R version, project, timestamps, status, and resource configuration, in a queryable audit database accessible via the Workbench API.
Console Auditing	✓	✓	Console auditing includes the ability to capture and audit user-run code in RStudio and Positron.
Workbench managed credentials for AWS, Azure, Snowflake, and Databricks	✓	✓	Workbench can manage and refresh short-lived OAuth tokens for seamlessly accessing Cloud-based data and services without the need for PATs. Current support for AWS, Azure, Snowflake and Databricks credentials for RStudio Pro; AWS, Snowflake, and Databricks credentials for Positron Pro and VS Code.
Custom OAuth integrations (Preview)	✓	✓	Workbench can provide user-specific OAuth credentials for sessions that integrate with any OAuth 2.0-compliant service that supports the authorization code flow. These integrations allow users to securely access third-party APIs and services programmatically from Workbench sessions without managing credentials manually.
Package Manager Custom Metadata in RStudio Pro Packages Pane	✓	✓	✓ Posit Workbench can communicate with Posit Package Manager to display custom metadata associated with specific packages, including risk or CVE scores.
Positron Enforced Configuration	✓	✓	Admins can pre-set or enforce any setting in Positron's settings.json

Managed services

Managed services	✓	For teams that want to simplify the management of Posit Workbench, they can offload administration to our Cloud partners. This includes Snowflake Native Apps, AWS SageMaker (RStudio Pro only), AzureML, and Google Cloud Workstations.
Available in Posit Team Dedicated (PTD)	✓	Refer to PTD terms of service for details

Productivity

Execute long-running tasks as Workbench Jobs in RStudio Pro, Positron Pro, and VS Code	✓	Workbench Jobs provide a way to offload long-running tasks to a separate session. When Workbench is backed by Kubernetes or Slurm, users also have support for Workbench Jobs that scale up in that attached Cluster.
Execute and audit outputs of critical scripts as Audited Jobs in RStudio pro	✓	You can configure Posit Workbench to track auditing information alongside Workbench job output. This includes digital signatures, environment information, and user-defined ad hoc information. Audited Workbench jobs are available in the RStudio IDE.
Databricks Cluster integration	✓	✓ The Databricks Cluster pane in RStudio Pro provides a way to seamlessly manage, connect to, and access existing Clusters in Databricks.

Posit Connect

Feature	BSC	ENH	ADV	Description
Server activations				
Activations	1+1	2+2	∞	Production + Staging
Minimum user count	20	100	100	
Maximum user count	50	500	∞	Your contract and license determines the actual named user count allowed on your system.
Disaster recovery activations	1	2	∞	Stand up a passive disaster recovery environment that matches your production installation. Refer to the Software License Description .
Essentials				
Publishing Tools	✓	✓	✓	Easy-to-use publishing options designed to fit your deployment workflows: Push-button, Git-backed, or CLI.
Runtime Settings	✓	✓	✓	Content-scoped controls for scaling and tuning Python or R code execution on the server.
Environment Variables	✓	✓	✓	Content-scoped custom environment variables that are encrypted on-disk and in-memory.
Content Logs	✓	✓	✓	An interface that displays execution details for each process associated with a piece of content. Standard output and standard error are captured and streaming log entries appear in real time.
Content Management	✓	✓	✓	Options to delete or deactivate content items.
Content Dashboard	✓	✓	✓	A content listing interface that provides basic search functionality and tag-based organization.
Connect Server API	✓	✓	✓	A developer-style API for publishers and administrators who want to deploy and manage content programmatically, automate tasks, retrieve server information, and perform various administrative functions using HTTP-based requests.
API Keys	✓	✓	✓	Role-scoped API key generation and management interface.
Hosting				

Static and Rendered Content	✓	✓	✓	Support for content types that have no server component: Plots, Pins, HTML, Jupyter Notebooks (not Voila), Quarto (not Shiny), R Markdown (not Shiny), and R or Python scripts.
Interactive Applications and Documents	✓	✓	✓	Support for content types that are seen as interactive and have a hosted server component: Shiny for R, Shiny for Python, Shiny Quarto, Shiny R Markdown, Jupyter Voila, Dash, Streamlit, Bokeh, Panel, and Gradio.
API-type Content		✓	✓	Support for content types that are seen as request-based and have a hosted server component: Plumber, WSGI API types (Flask, Bottle, Falcon, pycnic), and ASGI API types (FastAPI, Quart, Sanic, Falcon-ASGI, Starlette).
TensorFlow Models		✓	✓	Support for hosting TensorFlow models using TensorFlow Serving.
Node.js Content			✓	Support for hosting Node.js applications is available with limitations as an Early Access feature.

Security

SSO / Authentication	✓	✓	✓	Integrate with a number of 3rd party products and authentication providers for user authentication. Connect only supports one active authentication provider at a time.
Browser Security	✓	✓	✓	Secure publishing and interactions with content using SSL/TLS.
Environment Management Controls	✓	✓	✓	Configuration options for package repository integrations, environment management, and process execution.

Automation

Content Scheduling	✓	✓	✓	Re-execute rendered content published with source code on a set (customizable) schedule and view past renderings (using the History tool).
Email Distribution	✓	✓	✓	Schedule an email to be sent (or conditionally suppressed) every time rendered content is re-executed.
Report Parameterization	✓	✓	✓	An interface for users to change inputs on a parameterized R Markdown document and create variations on a single report.

Authorization

Content Collaborators	✓	✓	✓	Content owners can allow other Publishers or Administrators to make updates to the content item by giving them collaborator access.
Authenticated Access to Content	✓	✓	✓	Content owners can share content access with everyone on the server, or specific sets of authenticated users or groups.
Unauthenticated Access to Static Content (with Powered by Connect message)	✓			Users who are not authenticated with Connect can view a content item when the content owner chooses this access level. A "Powered by Posit Connect" tag is applied to this content.
Unauthenticated Access to Static Content		✓	✓	Users who are not authenticated with Connect can view a content item when the content owner chooses this access level.
Public Access to Dynamic Content		✓	✓	Users who are not authenticated with Connect can only view interactive or request-based API-type content if it is open to the public internet. Requires a Public Access license.
Execution				
Local Content Execution	✓	✓	✓	The Posit Connect application service and all published content execute on the same host. Load-balanced server and container-based configurations on Kubernetes are supported where applicable.
Off-Host Content Execution			✓	Published content is executed in an off-host container that is isolated from the Posit Connect service. Requires Kubernetes. Does not support "Current User Execution".
Custom Default Execution User	✓	✓	✓	Administrators can change the execution user for a piece of content by changing the default "RunAs" user.
Current (Visiting) User Content Execution		✓	✓	Connect can use a local Unix account associated with the currently logged-in user when executing applications. Not available for Off-Host Content Execution.
Data Governance				
OAuth Integrations (Viewer)		✓	✓	Interactive content can support OAuth integrations for viewer identity delegation, ensuring a viewer's data access in Connect is consistent with data-level access control lists defined in the data governance system.

OAuth Integrations (Service Account)

✓ ✓

Interactive and rendered content can support OAuth integrations for the client credentials grant type. This integration provides an OAuth access token which represents a service account identity and can be used to access external resources from a content item.

Server Management

Event Auditing

✓ ✓ ✓

A record of changes triggered either by a user or the system which can be accessed by Connect Administrators for auditing purposes.

OpenTelemetry

✓ ✓ ✓

Connect exports standard OpenTelemetry signal types: traces, metrics, and logs, to give administrators visibility into server behavior using standard observability tooling.

Server Status Reports

✓ ✓ ✓

A tool available to Connect Administrators that performs tests on a running Connect server and its process execution environments.

Usage Tracking

✓ ✓ ✓

Content usage data is available in the Connect dashboard via the System tab and in the content Settings. Instrumentation endpoints in the Connect Server API for custom reporting needs.

Custom Branding

✓ ✓

Configuration settings for customizing the default product branding and user experience.

Posit Package Manager

Feature	BSC	ENH	ADV	Description
Servers and repositories				
Activations	1+1	2+2	∞	Production + Staging
Number of repositories	3	10	∞	A repository is a set of packages that you want to make available to a set of users. For example, one repository might be a full-mirror of CRAN for development work, and a second one a separate curated repository with just a set of regulatory-approved packages for use in production. Can also be used to provide different sets of packages for use by different teams and/or projects.
Server activations				
Disaster recovery activations	1	2	∞	Stand up a passive disaster recovery environment that matches your production installation. Refer to the Software License Description .
Essentials				
Mirrors of CRAN, Bioconductor, and PyPI	✓	✓	✓	Provide full mirrors of public repositories internally within your own network/firewall, eliminating the need for end users to access the public repositories directly.
Date-based repository snapshots	✓	✓	✓	Access older versions of packages to help easily reproduce projects or preserve future compatibility.
Posit-built binary CRAN packages	✓	✓	✓	Posit-built binary versions of CRAN packages for Windows, macOS, and many versions of Linux, providing faster installation without the need to build the package from source.
Internal (local) packages	✓	✓	✓	Support for adding your own packages (in source and/or binary format) and make those available for installation using the same tools used for installing from public repositories (e.g. <code>pip</code> or <code>install.packages()</code>).
Package usage dashboard	✓	✓	✓	View download statistics by package or license type to monitor server utilization and most frequently installed packages.
AI Governance				

MCP Server	✓	✓	Integration with AI agents to provide context from your own package environment, enabling better recommendations about curated/approved packages and documentation of locally-developed packages.
Scalability			
Multiple server activations for High Availability	✓	✓	Package Manager can be deployed across multiple servers for high availability and redundancy.
Governance & Security			
VSCode extensions / Open VSX mirrors	✓	✓	Provide internal repositories of locally-developed Visual Studio Code extensions as well as mirrors of the Open VSX Registry, inside your own network. Supports installation from all VSCode-based IDEs, including Posit Workbench, Positron, and many other AI agentic IDEs.
Curated repositories	✓	✓	Create repositories containing only a subset of packages from a public repository, and optionally adding in your own internal packages to create a unified "one-stop shop" repository of approved packages for users.
Custom package metadata	✓	✓	Augment built-in package data with your own custom metadata values to be displayed on the Packages page. Include internal package metrics, links to supplemental documentation, or any other information useful to those browsing available packages. Metadata can also be imported via API (Advanced tier only).
Authenticated repositories	✓	✓	Require token-based authentication for viewing and installing packages from specified repositories. Provide access to protected internal packages only to approved systems or teams.
Security vulnerability blocking for public repositories	✓	✓	Ability to block all package versions with a known vulnerability reported on the OSV.dev vulnerability database.
Fine-grained Vulnerability Blocking		✓	Create custom override exceptions to allow installation of a package with a known vulnerability, or set a threshold to only block vulnerabilities above a desired CVSS score.

SSO Authentication	✓	Full Single Sign-on (SSO) integration with OIDC-compliant Identity Providers (IdPs) such as Okta and Microsoft Entra ID. Manage groups in the IdP and map those groups to permission scopes in Package Manager for both repository access and administration functions.
Offline access in "air-gapped" environments	✓	Support for environments where the Package Manager server does not have outbound Internet access. Administrators must use the Offline Downloader on an Internet-connected machine to download public packages to disk and then copy those files to the air-gapped Package Manager server every time they wish to update the available packages. Note: All product tiers permit using Package Manager as an internal, online repository mirror, as long as a firewall can be opened to allow the Package Manager server to access the Posit Package Service. Individual users do not require Internet access, only access to the internal Package Manager server.
Custom package blocking rules and exceptions	✓	Fully-customizable package blocking rules to allow blocking of individual packages by name, version, or partial name matching; or to provide override exceptions to allow packages that are blocked by other rules. Also provides for remote API integration with other sources of package blocking lists.
Restrict packages by open-source license type	✓	Allows creation of rules for blocking packages with specific open-source licenses (e.g. AGPL), or allowing only packages with approved license types (e.g. MIT).

Integration

Git builders	✓	Monitor a public or internal Git repository containing R or Python package source and automatically build a new version of the source package and add it to Package Manager whenever the Git repo is updated. Can be configured to rebuild the package on every commit to a specific branch, or whenever a new tag is added.
--------------	---	--

Integration with CI/CD pipelines
for remote publishing



Enables remotely publishing new packages via a secure API using our remote tools. Designed for teams that have their own CI/CD systems (Jenkins, TravisCI, GitHub Actions, etc.) for building Python or R packages and want to be able to automatically publish the built packages to Package Manager as part of that pipeline. Also can be used for remote publishing directly by trusted package developers.

Full Remote Administration



Allows server admins to generate an API Token permitting remote administration of Package Manager, without requiring shell access to the server. This can be used to grant non-IT users access to perform administration commands, or to build custom integrations with other tools or applications to perform any Package Manager functions.

Integration with JFrog
Artifactory / Sonatype Nexus



Allows integration with Artifactory or Nexus as the upstream source of public packages. Typically used in environments where all packages must come through Artifactory/Nexus to be scanned/approved, but users still want the benefits of using Package Manager.