# MS R Server bzw. MS Machine Learning Server

Machine Learning Server 9.2.1 expands upon Microsoft R Server 9.1 with new Python libaries for integrating machine learning and data science into analytical solutions in the enterprise. Language-specific development is available when you add Python or R support (or both) during setup.

Microsoft continues its commitment and development in R -- not only in the latest Machine Learning Server release, but also in the newest [Microsoft R Client](https://docs.microsoft.com/en-us/machine-learning-server/r-client/what-is-microsoft-r-client) and [Microsoft R Open](https://mran.microsoft.com) releases.

<https://docs.microsoft.com/en-us/machine-learning-server/what-is-machine-learning-server>

<https://docs.microsoft.com/en-us/machine-learning-server/rebranding-microsoft-r-server>

**Download prior R Server releases**

The following table provides links for downloading older versions of Microsoft R Server.

| **Site for R Server** | **Edition** | **Details** |
| --- | --- | --- |
| [Visual Studio Dev Essentials](http://go.microsoft.com/fwlink/?LinkId=717968&clcid=0x409) | Developer (free) | This option provides a zipped file, free when you sign up for Visual Studio Dev Essentials. Developer edition has the same features as Enterprise, except it is licensed for development scenarios.   1. Click **Join or Access Now** and enter your account information. 2. Make sure you're in the right place: my.visualstudio.com. 3. Click **Downloads**, and then search for Microsoft R. |
| [Volume Licensing Service Center (VLSC)](http://go.microsoft.com/fwlink/?LinkId=717966&clcid=0x409) | Enterprise | Sign in, search for R Server. Choose the right version for your OS. |
| [MSDN subscription downloads](https://msdn.microsoft.com/subscriptions/downloads/hh442898.aspx) | Developer or Enterprise | Subscribers can download software at given subscription levels. Depending on your subscription, you can get either edition. |

**MS R Client**

[Microsoft R Client](https://docs.microsoft.com/en-us/machine-learning-server/r-client/what-is-microsoft-r-client) continues on as an R-only client for Machine Learning Server and R Server.

Microsoft R Client is a free, [community-supported](https://social.msdn.microsoft.com/Forums/en-US/home?forum=MicrosoftR), data science tool for high performance analytics. R Client is built on top of [Microsoft R Open](https://mran.microsoft.com/open/) so you can use any open source R package to build your analytics. Additionally, R Client introduces the [powerful ScaleR technology](https://docs.microsoft.com/en-us/machine-learning-server/r/tutorial-revoscaler-data-import-transform) and its proprietary functions to benefit from parallelization and remote computing.

R Client allows you to work with production data locally using the full set of ScaleR functions, but there are some constraints. On its own, the data to be processed must fit in local memory, and processing is limited up to two threads for RevoScaleR functions. To benefit from disk scalability, performance and speed, you can push the compute context to a production instance of Machine Learning Server (or R Server) such as [SQL Server Machine Learning Services](https://msdn.microsoft.com/en-us/library/mt604845.aspx) and Machine Learning Server for Hadoop. [Learn more about its compatibility](https://docs.microsoft.com/en-us/machine-learning-server/r-client/compatibility-with-server).

You can offload heavy processing to Machine Learning Server or test your analytics during their development. You by running your code remotely using [remoteLogin() or remoteLoginAAD()](https://docs.microsoft.com/en-us/machine-learning-server/r/how-to-execute-code-remotely) from the mrsdeploy package.

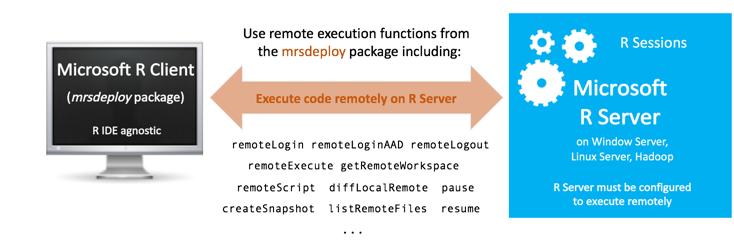
# **Execute on a remote server using the mrsdeploy package**

<https://docs.microsoft.com/en-us/machine-learning-server/r/how-to-execute-code-remotely>

Remote execution is the ability to issue R commands from either Machine Learning Server (or R Server) or R Client to a remote session running on another Machine Learning Server instance.

Remote execution is supported in several ways:

* From the command line in console applications
* In R scripts that call [functions from the mrsdeploy package](https://docs.microsoft.com/en-us/machine-learning-server/r-reference/mrsdeploy/mrsdeploy-package)
* From code that calls the APIs.



## Publishing web services in a remote session

After you understand the mechanics of remote execution, consider incorporating web service capabilities. You can publish an R web service composed of arbitrary R code block that runs on the remote R Server. For more information on publishing services, begin with the [Working with web services in R](https://docs.microsoft.com/en-us/machine-learning-server/operationalize/how-to-deploy-web-service-publish-manage-in-r#publishService) guide.

To publish a web service after you create a remote session (argument session = TRUE with remoteLogin() or remoteLoginAAD()), you have two approaches:

* Publish from your local session: At the REMOTE> prompt, use pause() to return the R command line in your local session. Then, publish your service. Use resume() from your local prompt to return to the command line in the remote R session.
* Authenticate again from within the remote session to enable connections from that remote session to the web node API. At the REMOTE> prompt, authenticate with remoteLogin() or remoteLoginAAD(). However, explicitly set the argument session = FALSE this time so that a second remote session is NOT created and provide your username and password directly in the function. When attempting to log in from a remote session, you are not prompted for user credentials. Instead, pass valid values for username and password to this function. Then, you are authenticated and able to publish from the REMOTE> prompt.

# **How to publish and manage R web services in Machine Learning Server with mrsdeploy**

<https://docs.microsoft.com/en-us/machine-learning-server/operationalize/how-to-deploy-web-service-publish-manage-in-r>

This article details how you can publish and manage your [analytic web services](https://docs.microsoft.com/en-us/machine-learning-server/operationalize/concept-what-are-web-services) directly in R. You can deploy your R models, scripts, and code as web services using the functions in the [mrsdeploy R package](https://docs.microsoft.com/en-us/machine-learning-server/r-reference/mrsdeploy/mrsdeploy-package). The mrsdeploy R package containing these functions is installed with both Machine Learning Server (and Microsoft R Server) and Microsoft R Client.

After you've authenticated, use the publishService() function in the mrsdeploy package to publish a web service.

# **APIs for operationalizing your models and analytics with Machine Learning Server**

<https://docs.microsoft.com/en-us/machine-learning-server/operationalize/concept-api>

The Machine Learning Server (and Microsoft R Server) [REST APIs](https://microsoft.github.io/deployr-api-docs/) are exposed by the operationalization server, a standards-based server technology [capable of scaling to meet the needs of enterprise-grade deployments](https://docs.microsoft.com/en-us/machine-learning-server/operationalize/configure-machine-learning-server-enterprise). With the Machine Learning Server configured to operationalize, the full statistics, analytics, and visualization capabilities of R and Python can now be directly leveraged inside Web, desktop and mobile applications.

The APIs available with Machine Learning Server can be categorized into two groups: Core APIs and the Service Consumption APIs.

## Core APIs for Operationalization

These core REST APIs expose the R or Python platform as a service allowing the integration of Python models and R statistics, analytics, and visualizations inside Web, desktop and mobile applications. These APIs enable you to publish Machine Learning Server-hosted **R analytics web services**, making the full capabilities of R available to application developers on a simple yet powerful REST API. These core operationalization APIs can be grouped into several categories as shown in this table.

## Service Consumption APIs

The service consumption REST APIs expose a wide range of Python and R analytics services to client application developers. After Python or R code is published and exposed by the server as a web service, an application can make API calls to pass inputs to the service, execute the service, and retrieve outputs from the service.

# SQL Server Machine Learning Services

<https://docs.microsoft.com/en-us/sql/advanced-analytics/r/sql-server-r-services>

SQL Server 2017 Machine Learning Services (previously SQL Server 2016 R Services) provides a platform for developing and deploying intelligent applications that uncover new insights. You can use the rich and powerful R language and the many packages from the community to create models and generate predictions using your SQL Server data.