Using HP Vertica Distributed R with Amazon Web Services

Version 0.5

HP Vertica Distributed R is compatible with Amazon Web Services (AWS). The installation process, however, is slightly different from the default Distributed R installation.

Getting Started

Perform the following steps to install Distributed R on AWS:

1. Choose or create an AWS instance.
2. Provide your private key to each AWS instance.
3. Install the Distributed R packages.

Choose or Create an AWS Instance

Your AWS instance must meet the following requirements:

- Your Amazon Machine Image (AMI) must be based on a supported operating system. For a list of supported operating systems, refer to the Distributed R installation guide.
- Your Security Group must have open ports corresponding to the ports specified in your cluster_conf.xml file and in your Linux kernel firewall (iptables).

For optimal performance on multiple nodes, Hewlett-Packard recommends that you specify a Network type of VPC.

Provide your Private Key to AWS

Note: These steps are necessary only if you are running Distributed R across multiple AWS instances.

When running Distributed R across multiple instances, those instances require the ability to log in to each other without user interaction. You can enable this interaction by placing a copy of your AWS private key on each instance. You can download a copy of this key from AWS. These steps replace the section **Password-less login** in the **Distributed R Installation Guide**.

1. Rename your private key .pem file to id_rsa.
2. Connect to your AWS instance.

3. On the AWS instance, copy id_rsa to the directory /.ssh. The instance accepts the private key and can now connect to other nodes in the cluster without a password.

Install the Distributed Packages

Once you have configured your private key, you can install Distributed R as you would normally. For detailed installation instructions, refer to the Distributed R Installation Guide.

Distributed R Performance on a Single Instance of Amazon Web Services

The following graphs show the performance of Distributed R versus the performance of Standard R on a single instance of Amazon Web Services. Distributed R parallel algorithms leverage multiple cores of a single instance and provide improved performance over standard R, particularly as data size increases.