Image Name: R-Studio Server with Spark 2.1.0

Location	https://s3.amazonaws.com/bluedata-catalog/solutions/bins/bdcatalog-centos-bluedata-rstudio136sp210-3.0.bin
DistroId	bluedata/rstudio136sp210
Version	3.0
Category (Cluster Type)	DataScience
Software Included	R version 3.3.2 R libraries pre-installed on all nodes - sparklyr, devtools, knitr, tidyr, ggplot2, shiny R Hadoop client for accessing HDFS from R - rHadoopClient_0.2.tar.gz spark-2.1.0-bin-hadoop2.6
R-Studio access	R-Studio Server - Create a OS user for each user who needs access on cluster controller node. 'sudo useradd test' 'sudo passwd test' -> provide password Login with test/test
Systemv Service names and commands	sudo service rstudioserver status (start, stop) sudo service spark-master status (start, stop) sudo service spak-slave status (start, stop)
OS	Centos. Works with both Bluedata Centos and RHEL hosts

Sample Code for Testing

Base-R testing

```
data(iris)  # Load the dataset iris
str(iris)  # Structure of the dataset
mean(iris$Sepal.Length)
str(iris$Sepal.Length)
tapply(iris$Sepal.Length, iris$Species, mean)
```

Sparklyr testing

```
 >if (nchar(Sys.getenv("SPARK_HOME")) < 1) {
               Sys.setenv(SPARK_HOME = "/usr/lib/spark/spark-2.1.0-bin-hadoop2.6")
             }
> library(sparklyr)
> sc <- spark_connect(master = "spark://bluedata-266.bdlocal:7077")  (*Replace with your master)


# Simple Test
> data(iris)  # Load the dataset iris
str(iris)  # Structure of the dataset
mean(iris$Sepal.Length)
str(iris$Sepal.Length)
tapply(iris$Sepal.Length, iris$Species, mean)

#MLLib usage test

> library(dplyr)

# copy mtcars into spark
> mtcars_tbl <- copy_to(sc, mtcars)
 #  ** May show an error regarding problem with database. Seems to work OK after that
>src_tbls(sc)

# transform our data set, and then partition into 'training', 'test'
> partitions <- mtcars_tbl %>%
  filter(hp >= 100) %>%
  mutate(cyl8 = cyl == 8) %>%
  sdf_partition(training = 0.5, test = 0.5, seed = 1099)

# fit a linear model to the training dataset
> fit <- partitions$training %>%
  ml_linear_regression(response = "mpg", features = c("wt", "cyl"))

> summary(fit)
```

Wrapper functions

Reading data from dtap

```
> count_lines <- function(sc, file) {
    spark_context(sc) %>%
       invoke("textFile", file, 1L) %>%
       invoke("count")
 }
> count_lines(sc, "dtap://TenantStorage/data/samples/bank-full.csv")
```