# **Running Spark Queries in Intellij**

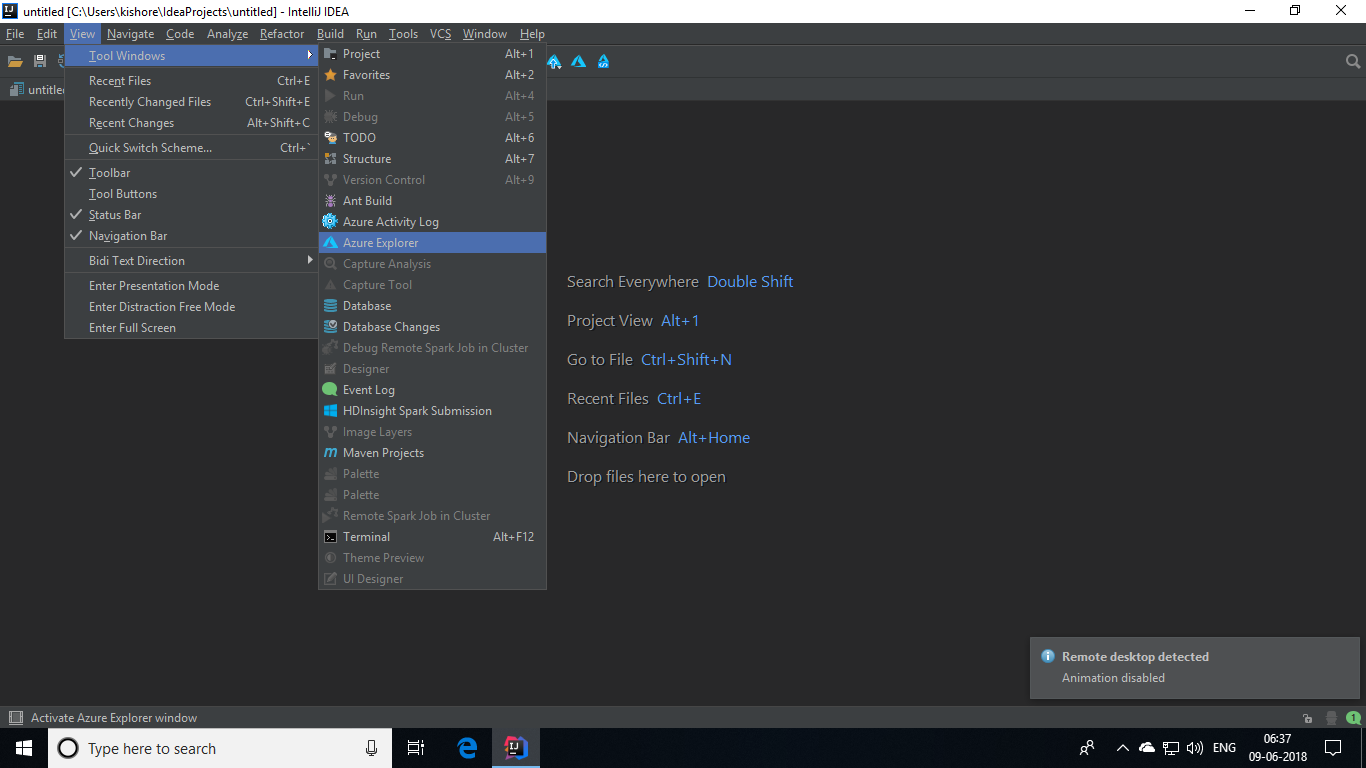
This demo is about executing spark queries using the Intellij toolkit for Azure. The prerequisites for this demo are,

1. IntelliJ IDEA. This article uses version 2017.1. You can install it from the [JetBrains website](https://www.jetbrains.com/idea/download/).
2. Oracle Java Development Kit. You can install it from the [Oracle website](http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html).
3. Create one HDInsight spark cluster.

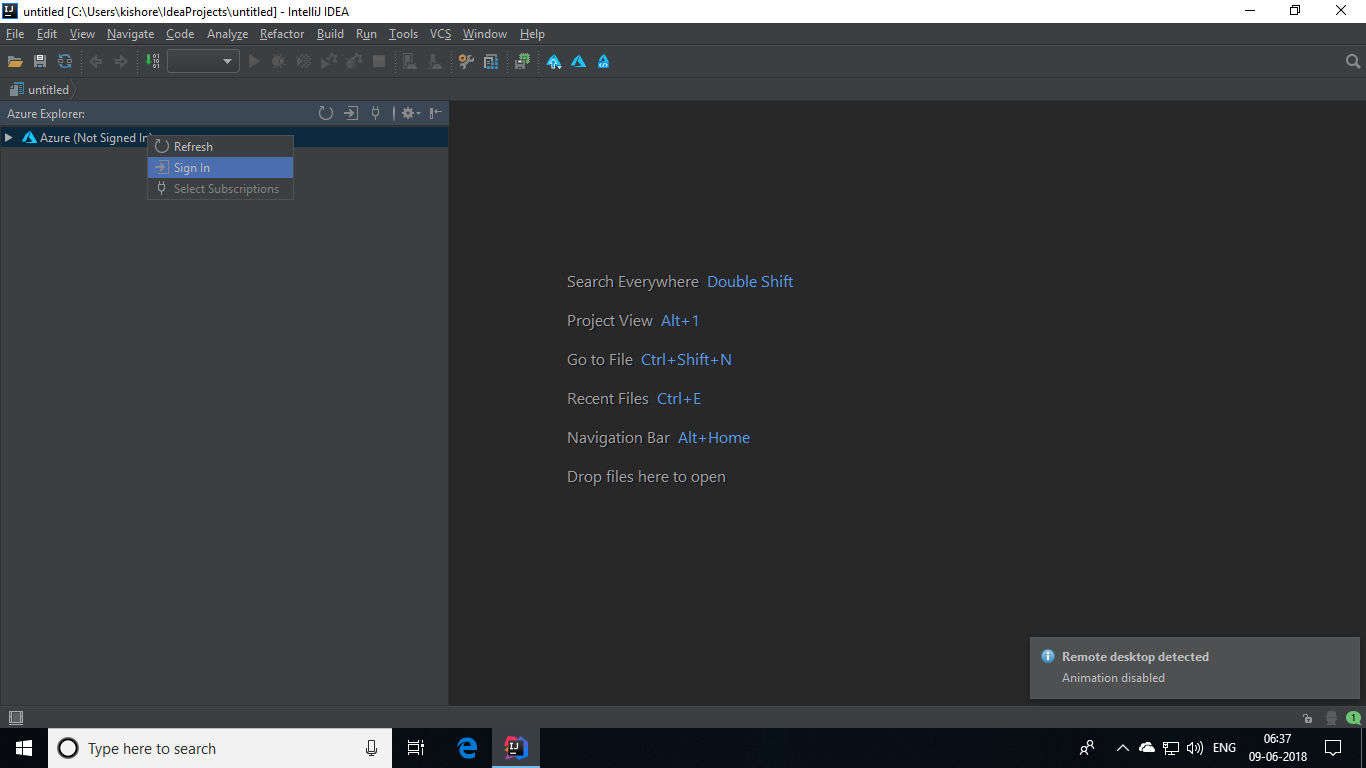
Also,we need Azure Toolkit added to Intellij. To add this, follow the article in the link - [Install Azure Toolkit for IntelliJ](https://docs.microsoft.com/azure/azure-toolkit-for-intellij-installation). Open a new project initially to access the rest of menus in Intellij.

**Signing into Azure Subscription:**

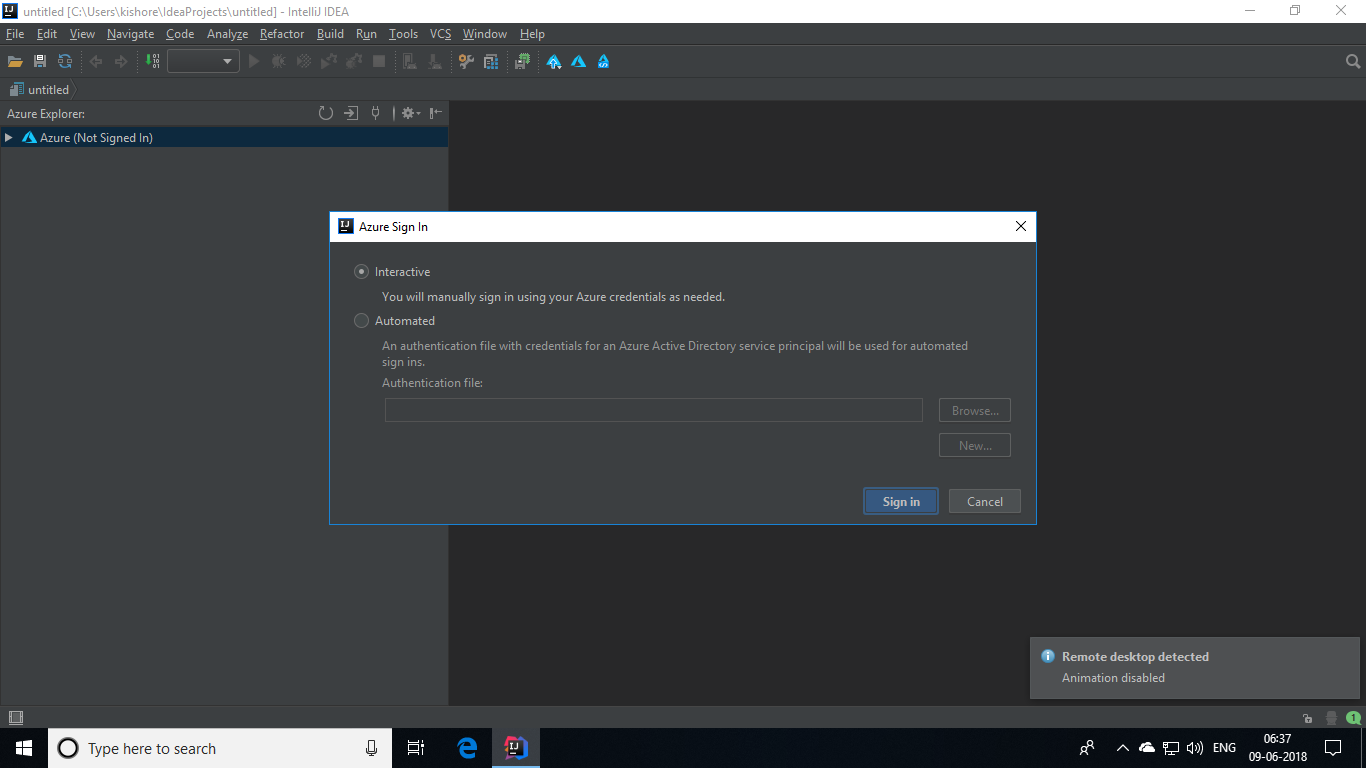
Once after all the toolkits are added, in the Intellij, go to **View -> Azure Explorer**.



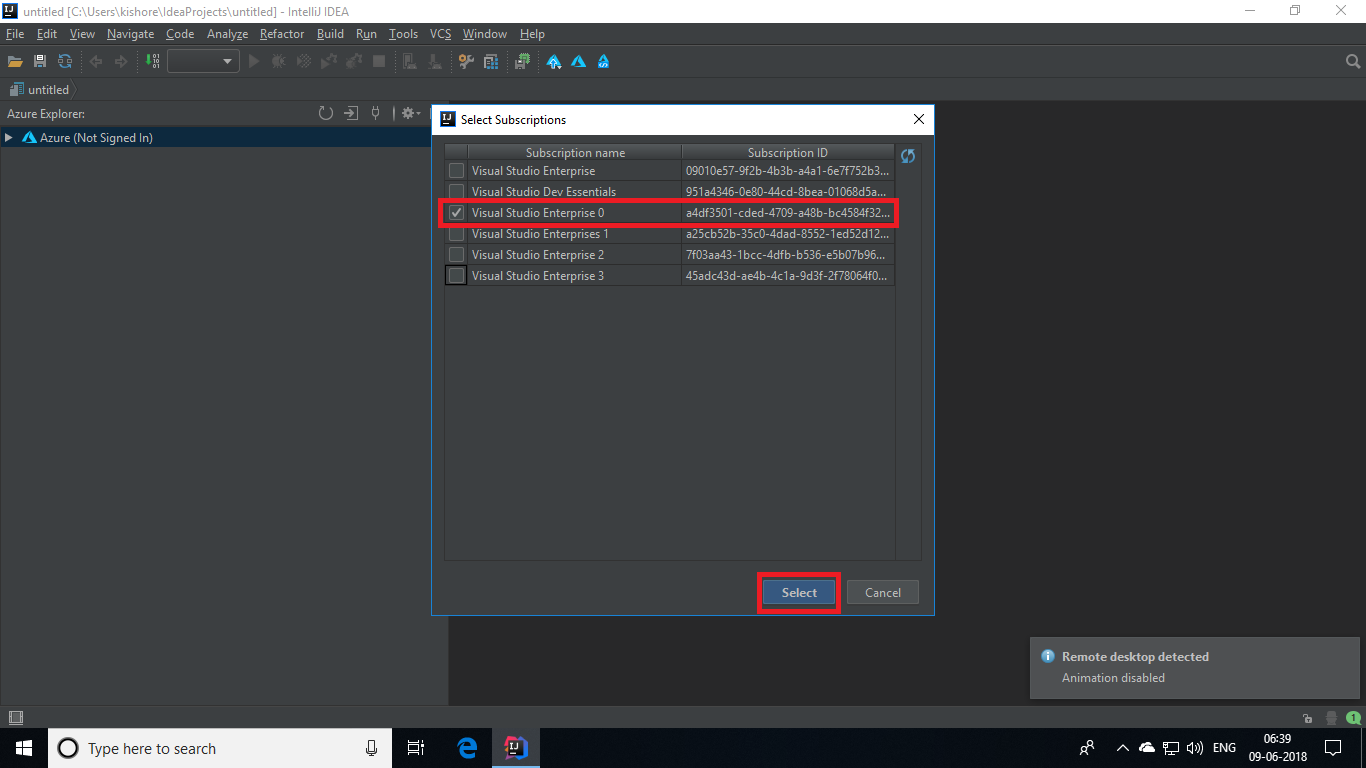
This will show you the explorer pane. In there, right click on **Azure** and select **Sign In**.



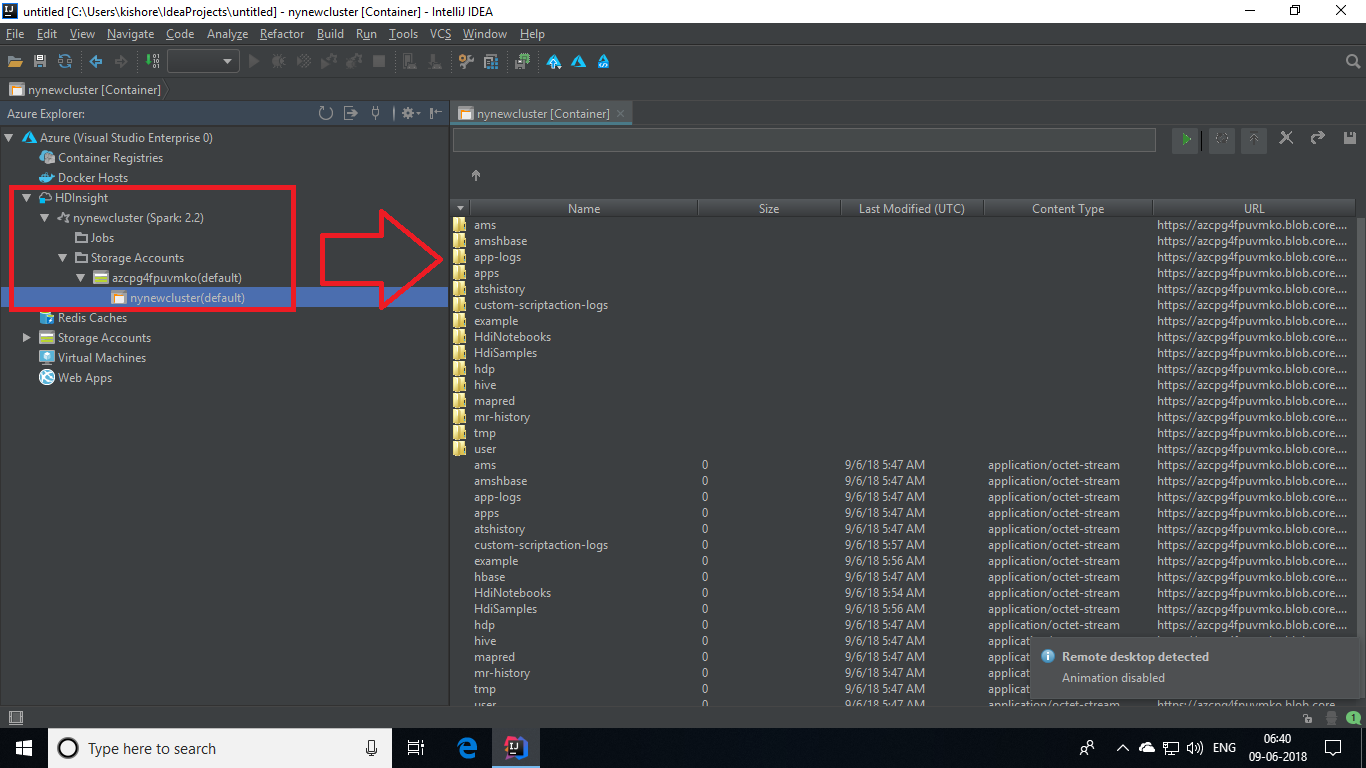
Choose the sign in option as **Interactive** and sign in buy giving your azure credentials where you have your HDInsight cluster.



Then select the subscription of the cluster.

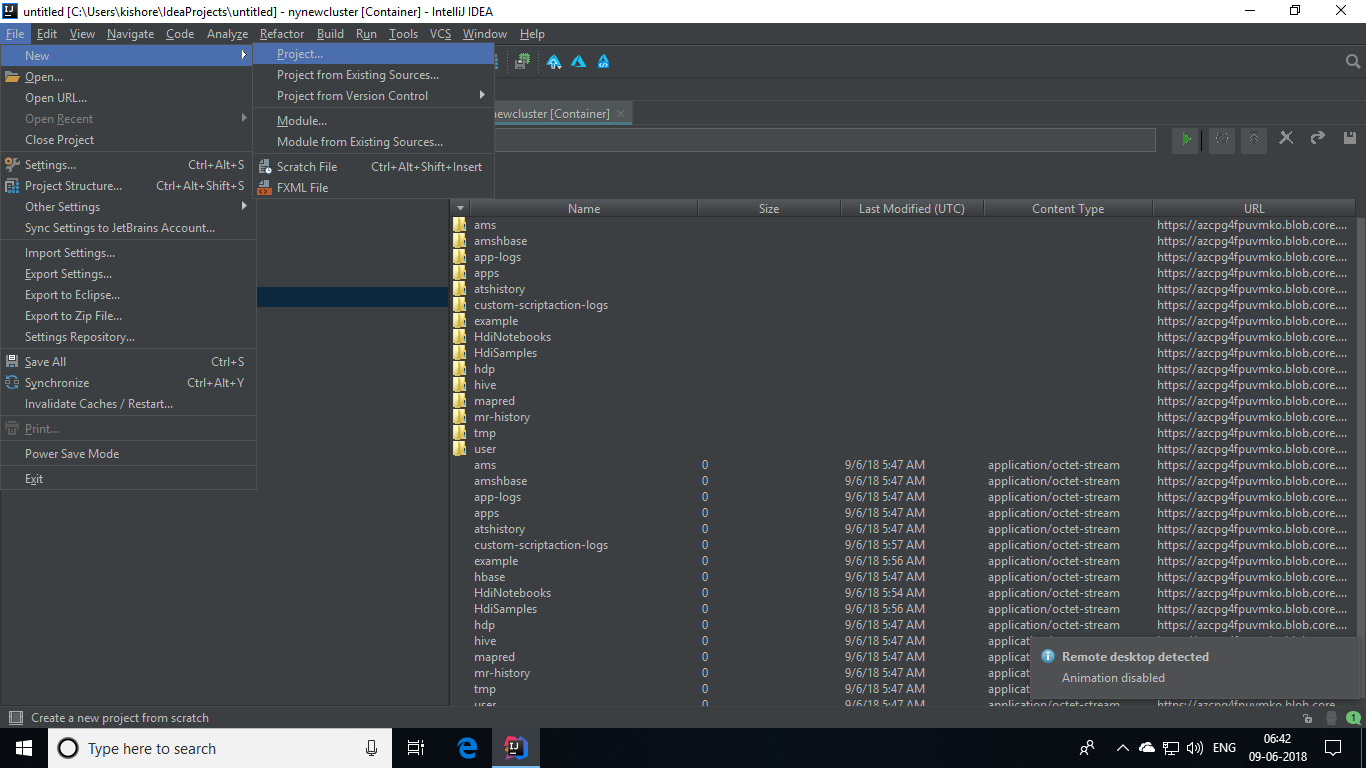


After choosing the subscription, you will be able to see the data in the Hive table by expanding into your cluster and its storage account.

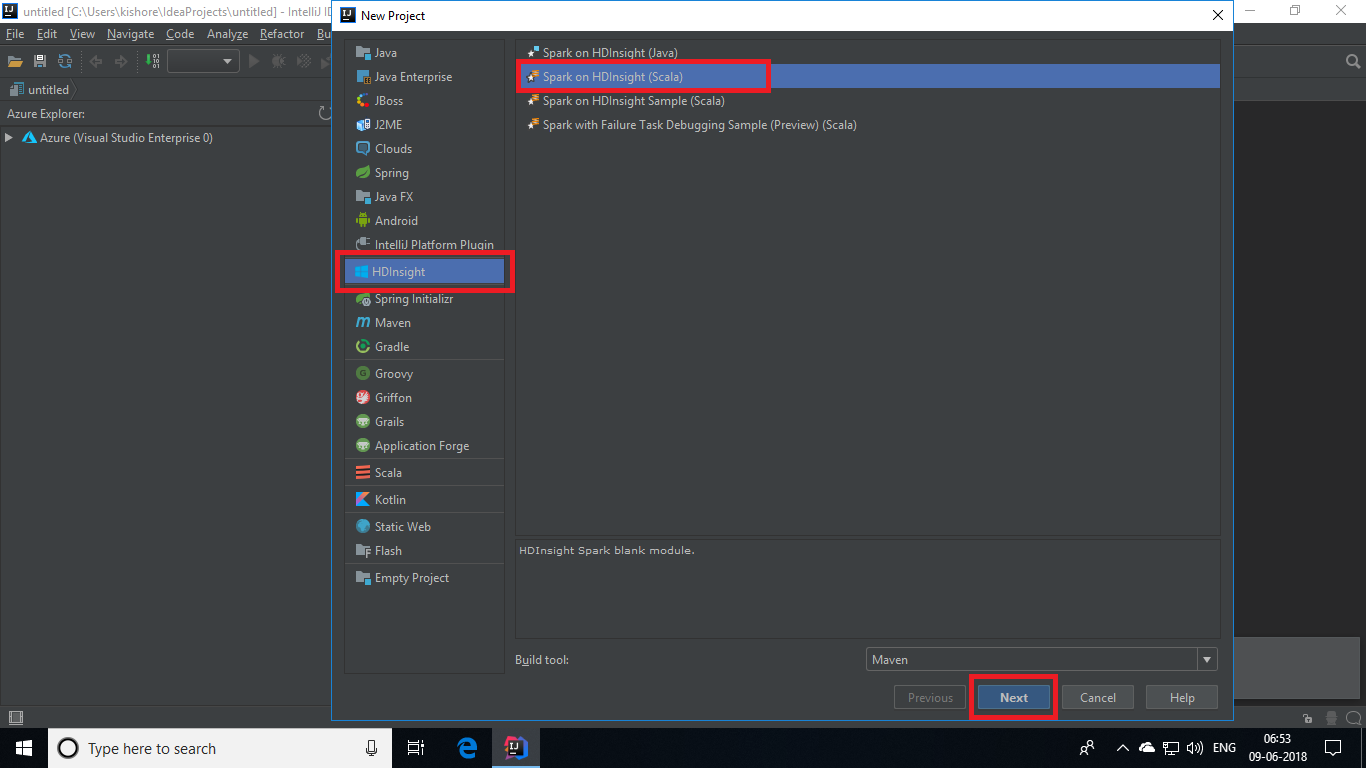


**Running a Spark Scala Application on HDInsight Spark Cluster:**

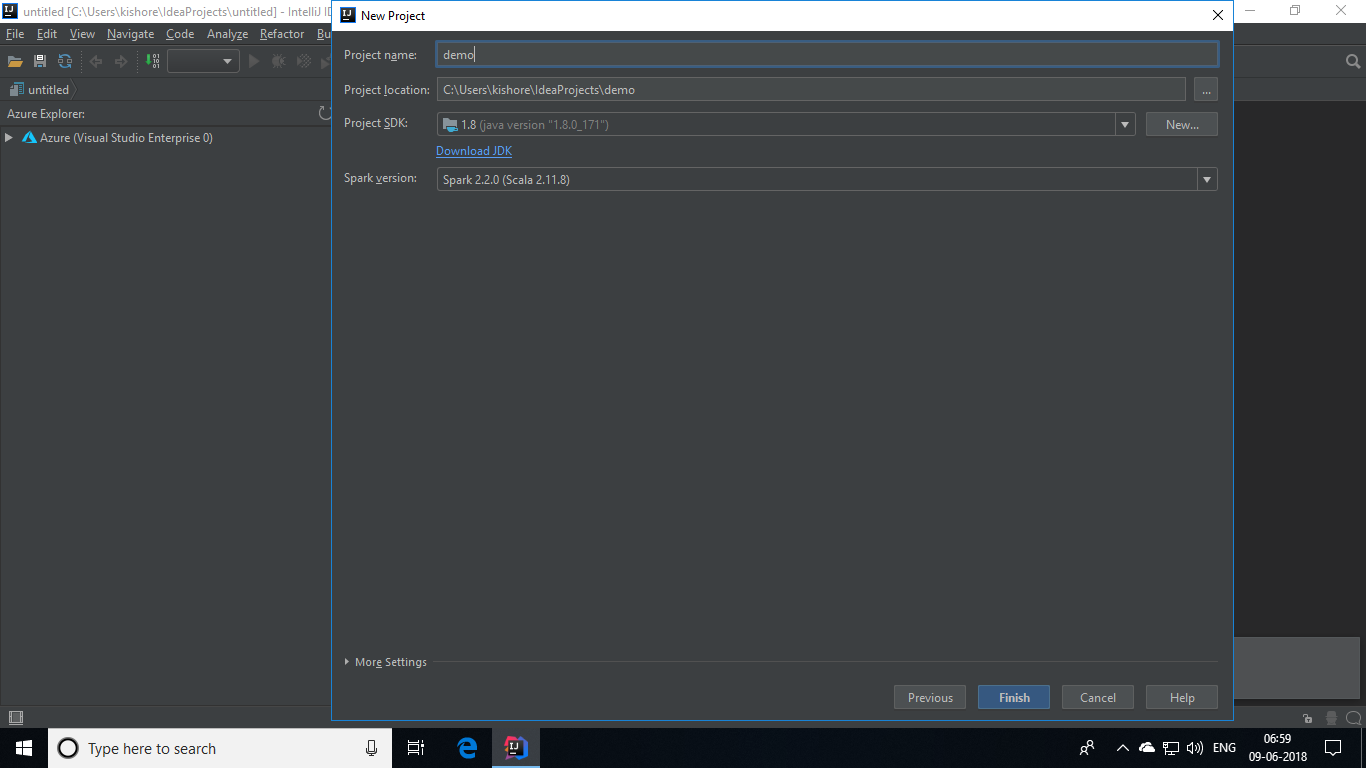
In the Intellij IDE, go to **File -> Project.**

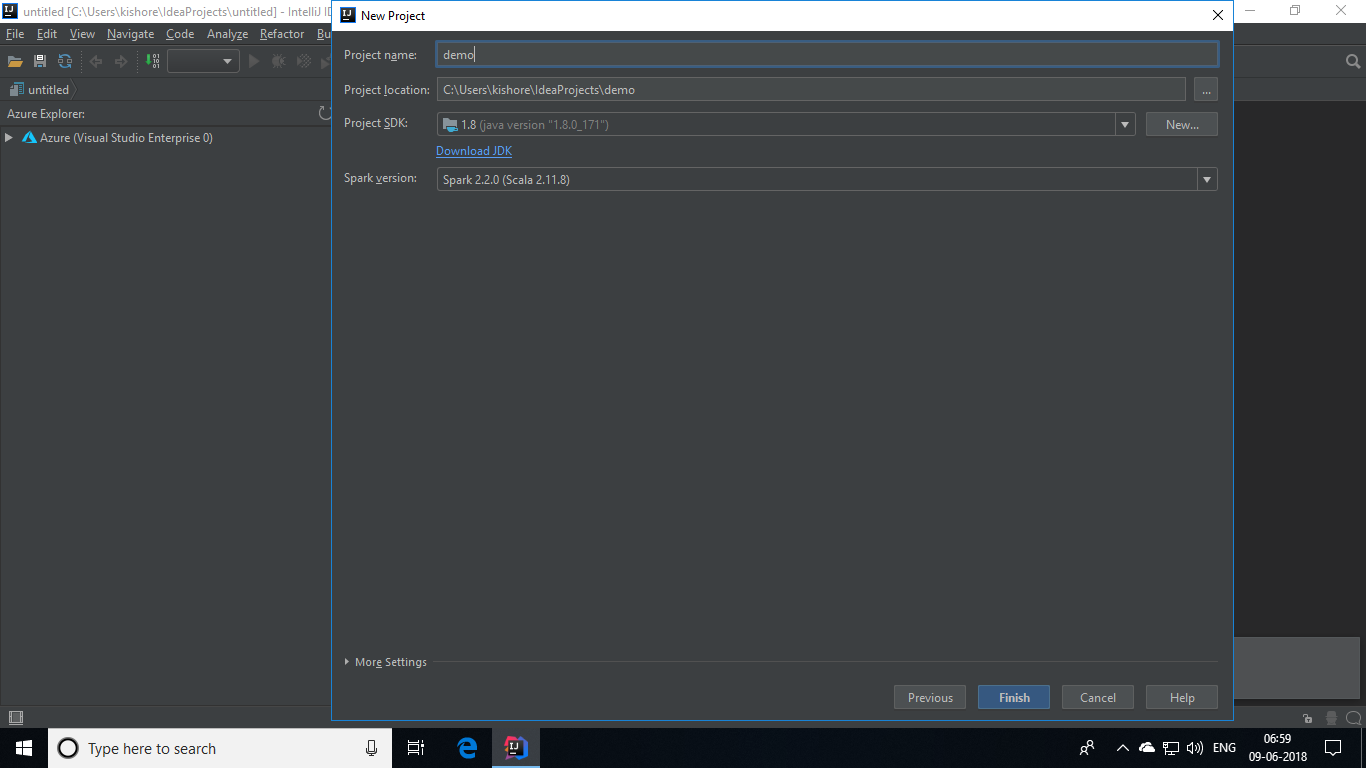


Then select **HDInsight** and **choose Spark on HDInsight (Scala)** and click on **Next** button**.**

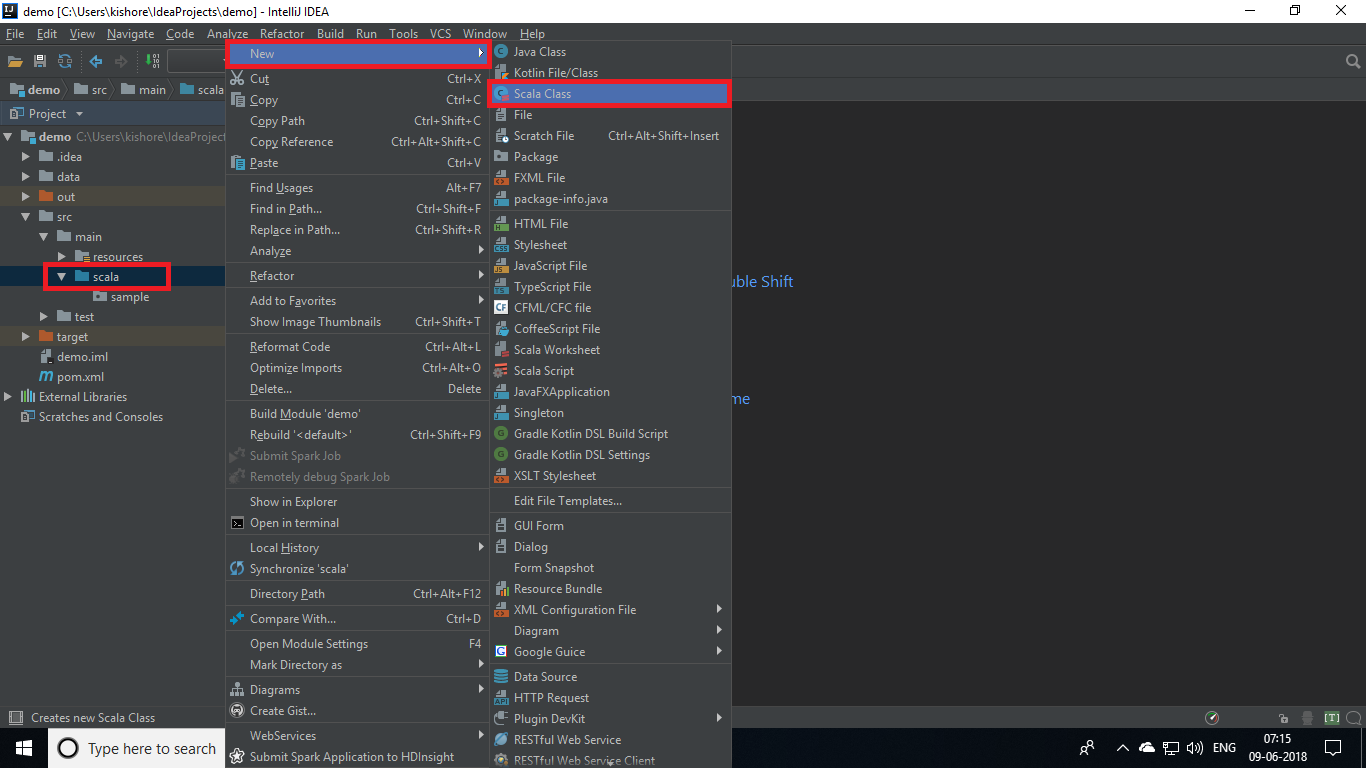


Then, give your project a name and choose a Spark version and hit on **Finish.**

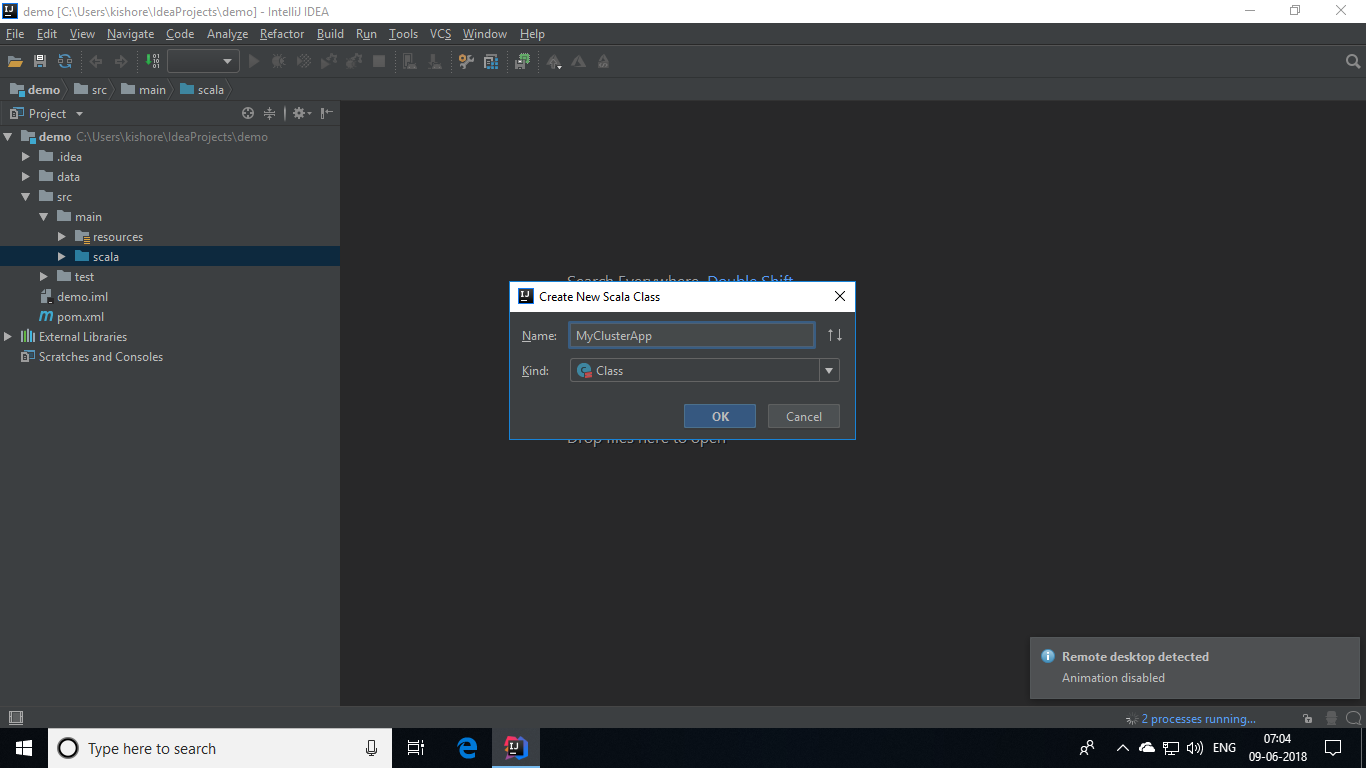




After the solution gets deployed, go to **src -> main -> scala** in theproject structure pane. There, right click the **scala** folder and go to **New -> Scala Class.**



Now, you will be asked to give a name for the class. Name the class as **MyClusterApp** and click on **OK** button.



After the class gets created, copy and paste the below given code in the **MyClusterApp.scala** page.

**MyClusterApp.scala:**

import org.apache.spark.SparkConf

import org.apache.spark.SparkContext

object MyClusterApp{

def main (arg: Array[String]): Unit = {

val conf = new SparkConf().setAppName("MyClusterApp")

val sc = new SparkContext(conf)

val rdd = sc.textFile("wasb:///HdiSamples/HdiSamples/SensorSampleData/hvac/HVAC.csv")

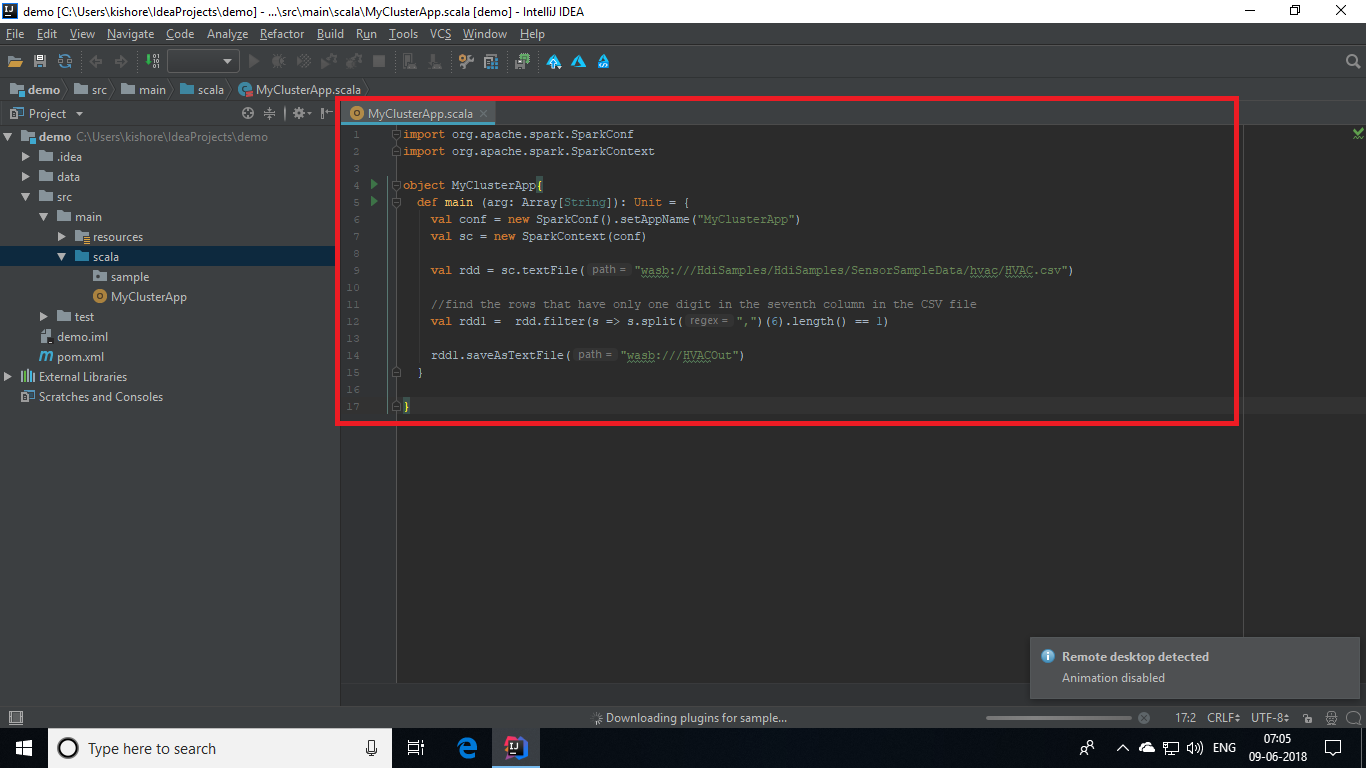
//find the rows that have only one digit in the seventh column in the CSV file

val rdd1 = rdd.filter(s => s.split(",")(6).length() == 1)

rdd1.saveAsTextFile("wasb:///HVACOut")

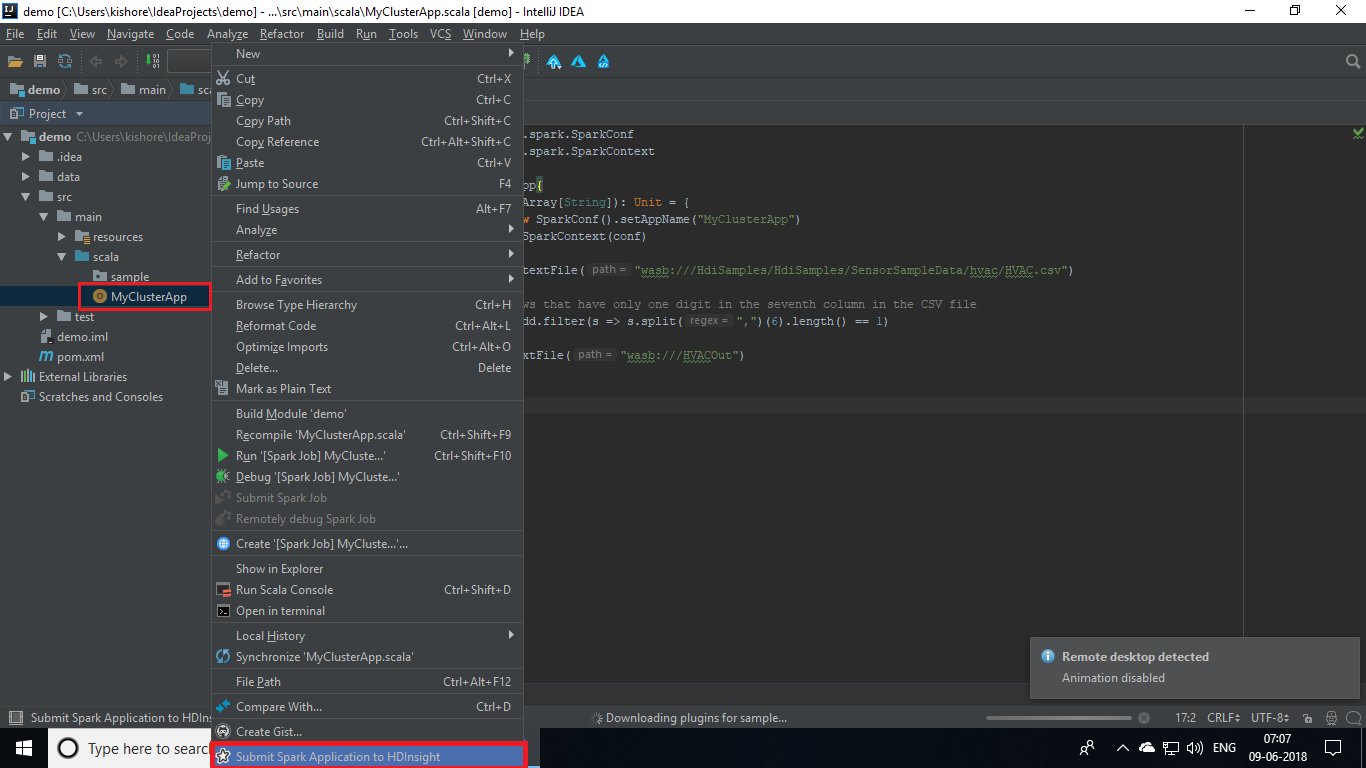
}

}

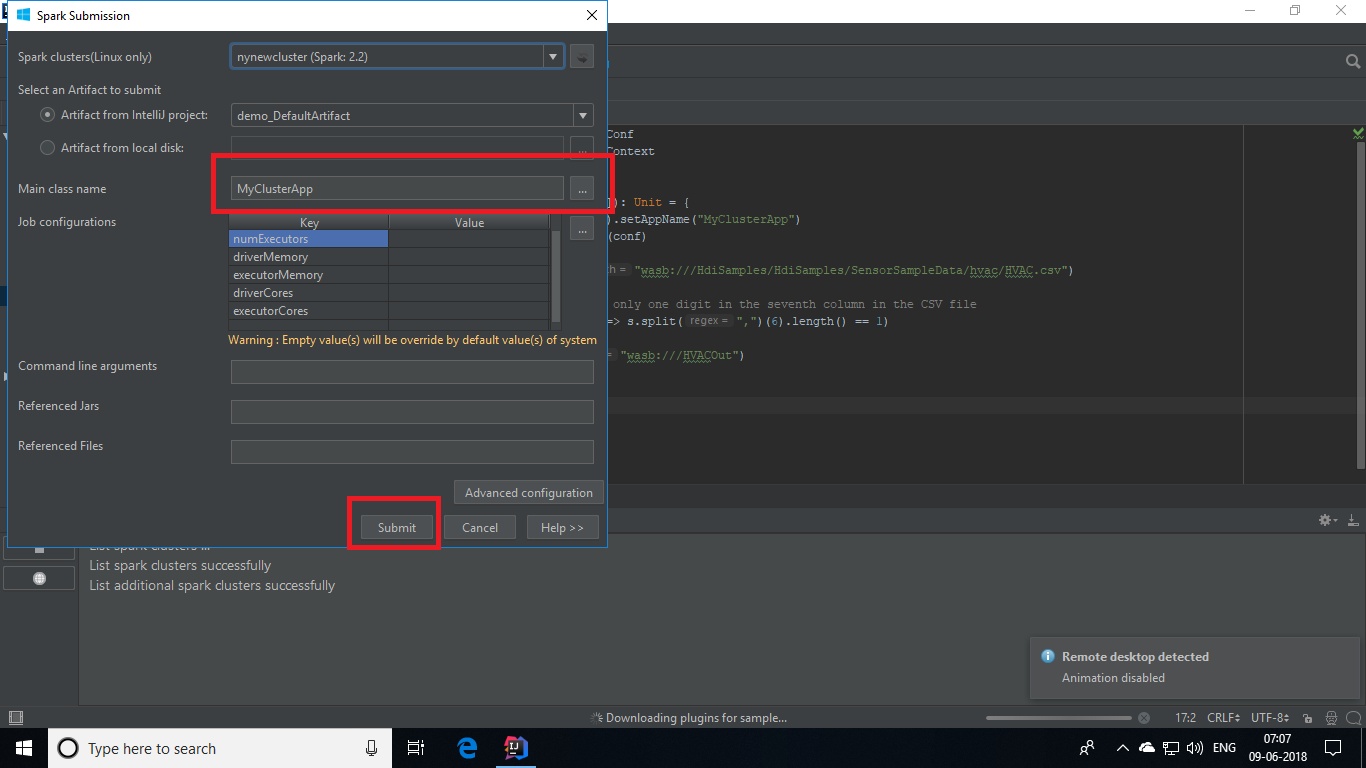


**Submitting Spark Application to HDInsight:**

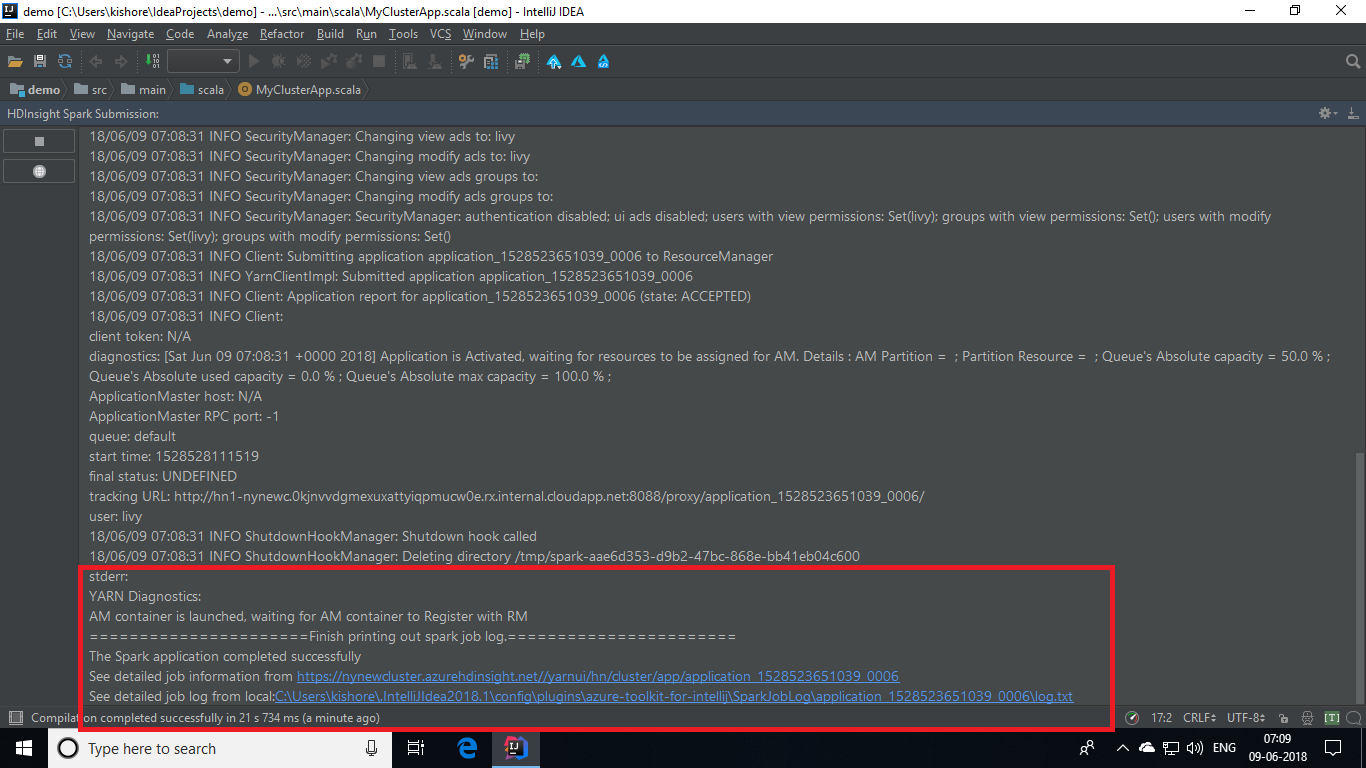
After completing the code, in the project structure pane right click on the **MyClusterApp** and click on **Submit Spark Application to HDInsight.**



In the next window, choose the **MyClusterApp** and click on **Submit.**



This will be deploying the solution in the spark cluster. You will be getting the following output when the submission is successful.



**Checking the Job:**

In the **Azure** **Explorer** find **Job** under cluster. This will show you the job that got executed and a diagrammatic representation of the Job that was committed to the cluster.

