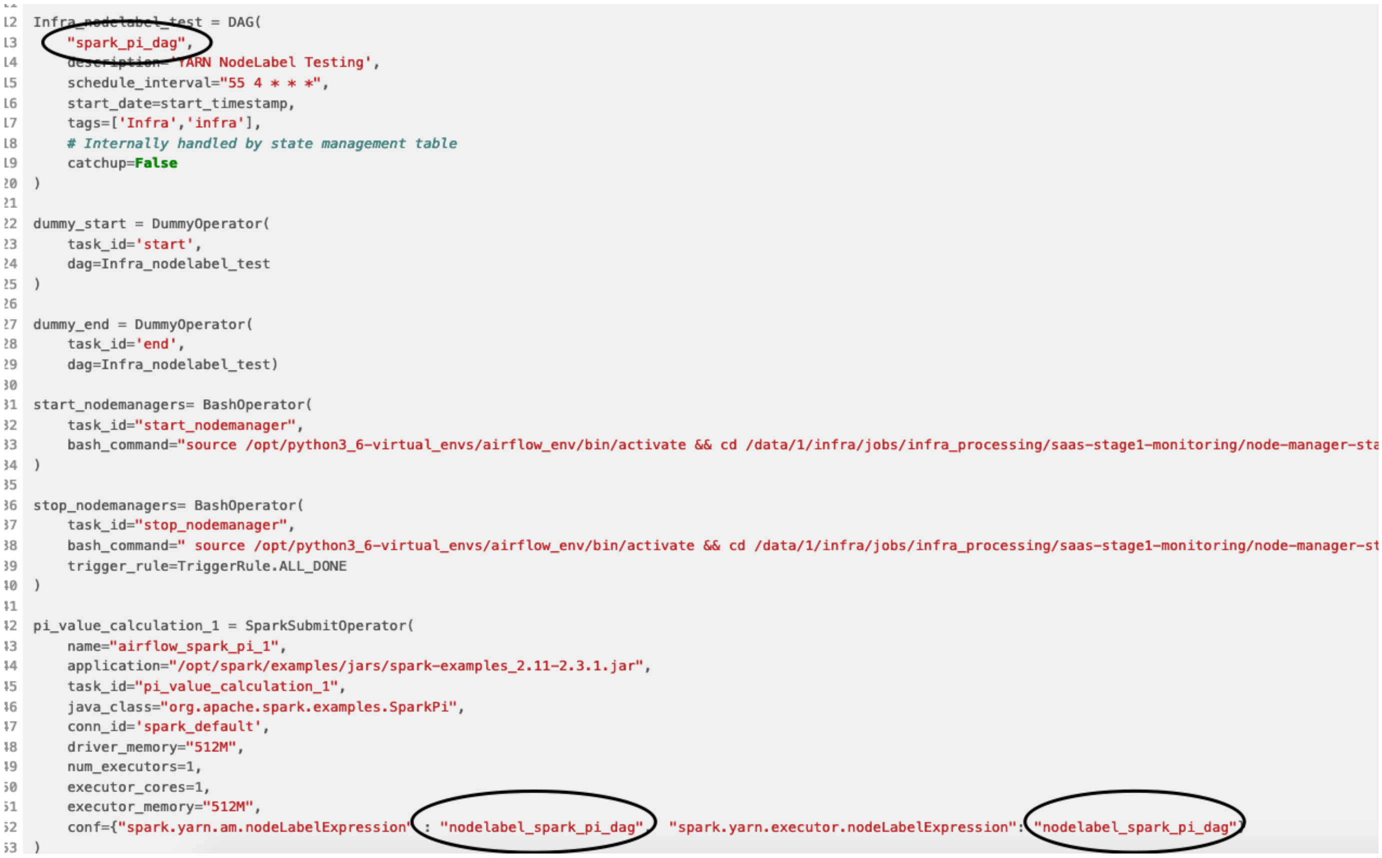
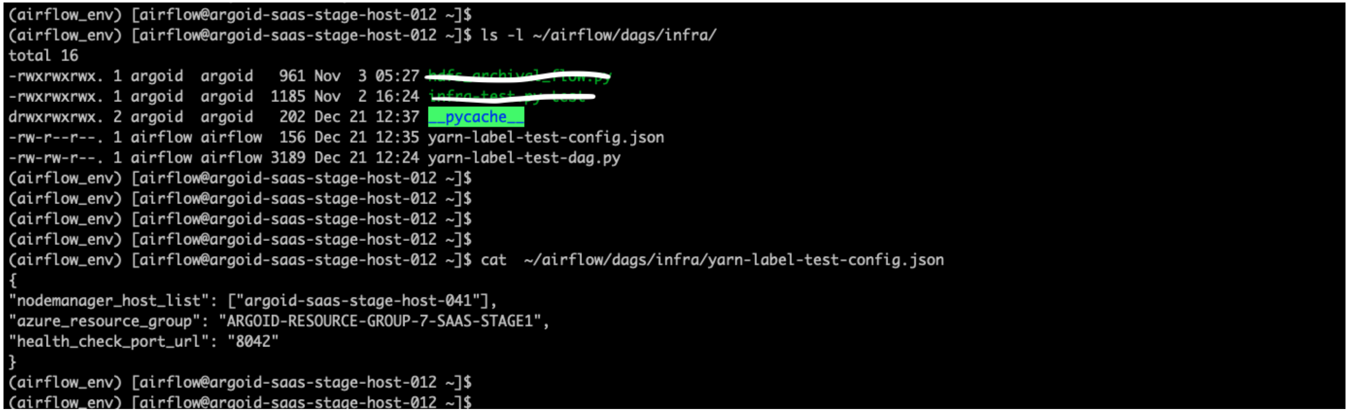
Setting YARN NodeManager Labels in Airflow DAGs

- Triggering spark job with nodelabels option ensures your job to land on particular YARN nodemanager
- Set nodelabels only if your DAG consists atleast one or more `SparkSubmitOperator`
- For each new DAG creation , notify Infra team to create new YARN nodemanager VM and to map nodelabels to YARN nodemanager Service
- Nodelabel would be same as DAG name but with suffix `nodelabels_` .
  For Example: If DAG name is `spark_pi_dag` then nodelabel would be `nodelabel_spark_pi_dag`



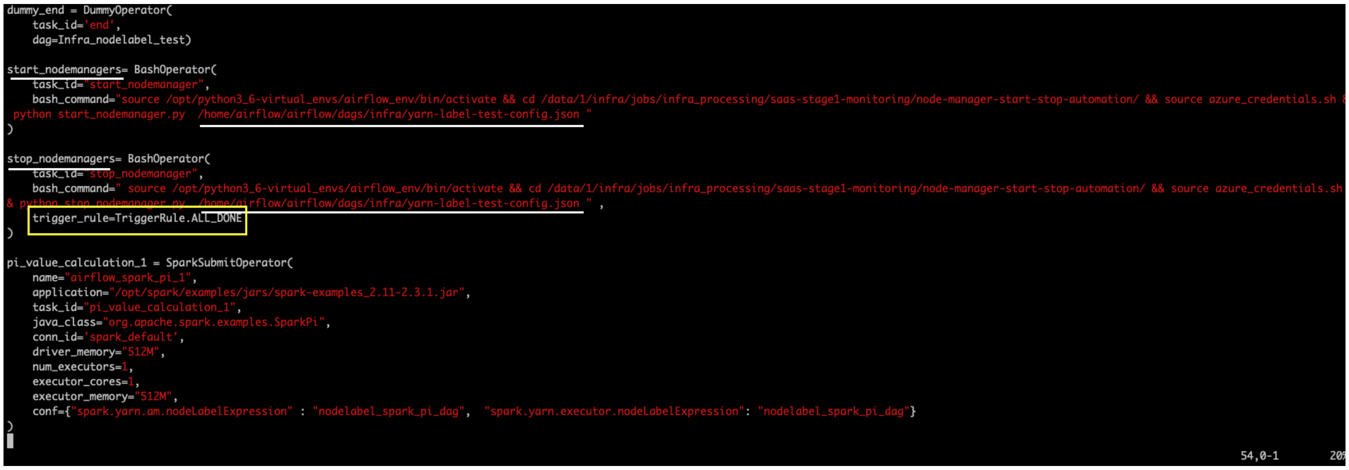
```
Infra_nodelabel_test = DAG(
    "spark_pi_dag",
    description='YARN NodeLabel Testing',
    schedule_interval="55 4 * * *",
    start_date=start_timestamp,
    tags=['Infra','infra'],
    # Internally handled by state management table
    catchup=False
)

dummy_start = DummyOperator(
    task_id='start',
    dag=Infra_nodelabel_test
)

dummy_end = DummyOperator(
    task_id='end',
    dag=Infra_nodelabel_test)

start_nodemanagers= BashOperator(
    task_id="start_nodemanager",
    bash_command="source /opt/python3_6-virtual_envs/airflow_env/bin/activate && cd /data/1/infra/jobs/infra_processing/saas-stage1-monitoring/node-manager-sta
)

stop_nodemanagers= BashOperator(
    task_id="stop_nodemanager",
    bash_command=" source /opt/python3_6-virtual_envs/airflow_env/bin/activate && cd /data/1/infra/jobs/infra_processing/saas-stage1-monitoring/node-manager-st
    trigger_rule=TriggerRule.ALL_DONE
)

pi_value_calculation_1 = SparkSubmitOperator(
    name="airflow_spark_pi_1",
    application="/opt/spark/examples/jars/spark-examples_2.11-2.3.1.jar",
    task_id="pi_value_calculation_1",
    java_class="org.apache.spark.examples.SparkPi",
    conn_id='spark_default',
    driver_memory="512M",
    num_executors=1,
    executor_cores=1,
    executor_memory="512M",
    conf={"spark.yarn.am.nodeLabelExpression" : "nodelabel_spark_pi_dag"  "spark.yarn.executor.nodeLabelExpression":"nodelabel_spark_pi_dag"
)
```

- Notify Infra team to create a config-file where YARN nodemanager VM details will be placed.
  The config file to be placed in DAG folder path.



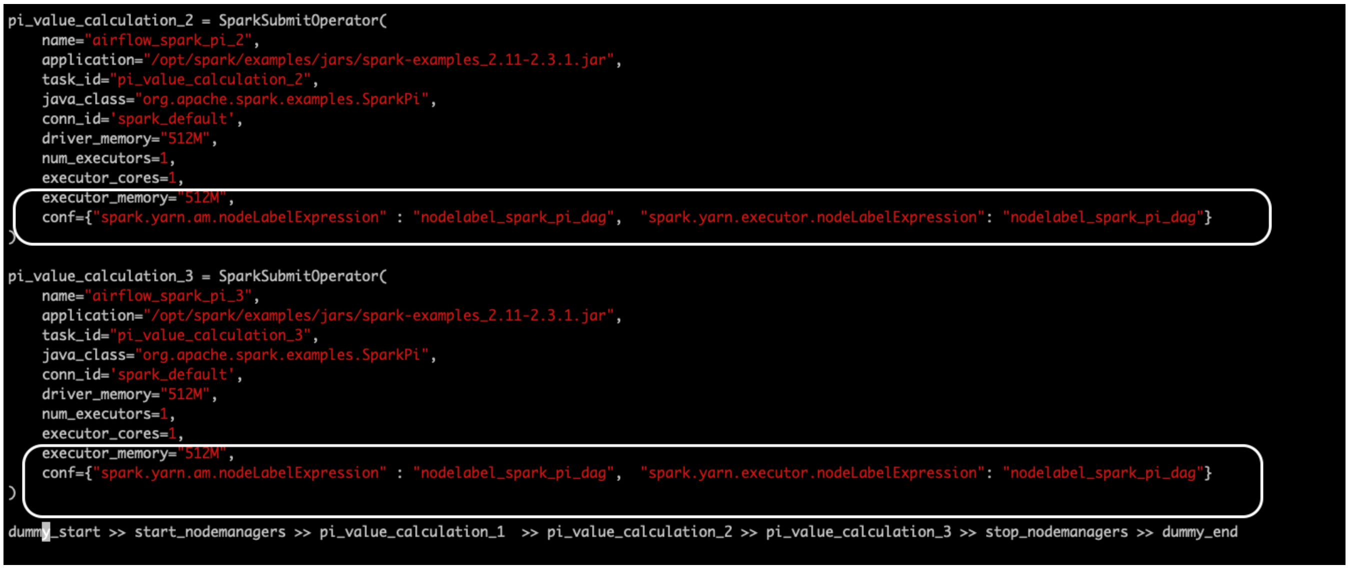
```
(airflow_env) [airflow@argoid-saas-stage-host-012 ~]$
(airflow_env) [airflow@argoid-saas-stage-host-012 ~]$ ls -l ~/airflow/dags/infra/
total 16
-rwxrwxrwx. 1 argoid  argoid   961 Nov  3 05:27 [redacted]
-rwxrwxrwx. 1 argoid  argoid  1185 Nov  2 16:24 [redacted]
drwxrwxrwx. 2 argoid  argoid   202 Dec 21 12:37 __pycache__
-rw-r--r--. 1 airflow airflow  156 Dec 21 12:35 yarn-label-test-config.json
-rw-rw-r--. 1 airflow airflow 3189 Dec 21 12:24 yarn-label-test-dag.py
(airflow_env) [airflow@argoid-saas-stage-host-012 ~]$
(airflow_env) [airflow@argoid-saas-stage-host-012 ~]$
(airflow_env) [airflow@argoid-saas-stage-host-012 ~]$
(airflow_env) [airflow@argoid-saas-stage-host-012 ~]$
(airflow_env) [airflow@argoid-saas-stage-host-012 ~]$ cat  ~/airflow/dags/infra/yarn-label-test-config.json
{
"nodemanager_host_list": ["argoid-saas-stage-host-041"],
"azure_resource_group": "ARGOID-RESOURCE-GROUP-7-SAAS-STAGE1",
"health_check_port_url": "8042"
}
(airflow_env) [airflow@argoid-saas-stage-host-012 ~]$
(airflow_env) [airflow@argoid-saas-stage-host-012 ~]$
```

- Add `BashOperator` tasks in your DAG to start and stop Nodemanager VMs
  also , `trigger_rule` also to be set in `stop_nodemanagers` part.



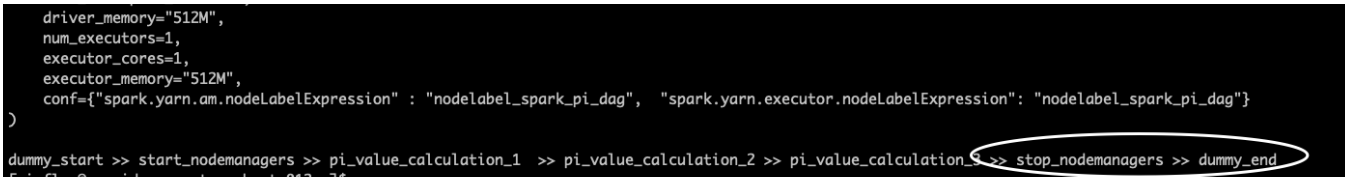
```
dummy_end = DummyOperator(
    task_id='end',
    dag=Infra_nodelabel_test)

start_nodemanagers= BashOperator(
    task_id="start_nodemanager",
    bash_command="source /opt/python3_6-virtual_envs/airflow_env/bin/activate && cd /data/1/infra/jobs/infra_processing/saas-stage1-monitoring/node-manager-start-stop-automation/ && source azure_credentials.sh &
 python start_nodemanager.py  /home/airflow/airflow/dags/infra/yarn-label-test-config.json "
)

stop_nodemanagers= BashOperator(
    task_id="stop_nodemanager",
    bash_command=" source /opt/python3_6-virtual_envs/airflow_env/bin/activate && cd /data/1/infra/jobs/infra_processing/saas-stage1-monitoring/node-manager-start-stop-automation/ && source azure_credentials.sh
& python stop_nodemanager.py  /home/airflow/airflow/dags/infra/yarn-label-test-config.json " ,
    trigger_rule=TriggerRule.ALL_DONE
)

pi_value_calculation_1 = SparkSubmitOperator(
    name="airflow_spark_pi_1",
    application="/opt/spark/examples/jars/spark-examples_2.11-2.3.1.jar",
    task_id="pi_value_calculation_1",
    java_class="org.apache.spark.examples.SparkPi",
    conn_id='spark_default',
    driver_memory="512M",
    num_executors=1,
    executor_cores=1,
    executor_memory="512M",
    conf={"spark.yarn.am.nodeLabelExpression" : "nodelabel_spark_pi_dag",  "spark.yarn.executor.nodeLabelExpression": "nodelabel_spark_pi_dag"}
)
                                                                                              54,0-1        20
```

- Set nodelabels for `spark.yarn.am.nodeLabelExpression` `spark.yarn.executor.nodeLabelExpression` properties in each `SparkSubmitOperator` tasks



```
pi_value_calculation_2 = SparkSubmitOperator(
    name="airflow_spark_pi_2",
    application="/opt/spark/examples/jars/spark-examples_2.11-2.3.1.jar",
    task_id="pi_value_calculation_2",
    java_class="org.apache.spark.examples.SparkPi",
    conn_id='spark_default',
    driver_memory="512M",
    num_executors=1,
    executor_cores=1,
    executor_memory="512M",
    conf={"spark.yarn.am.nodeLabelExpression" : "nodelabel_spark_pi_dag",  "spark.yarn.executor.nodeLabelExpression": "nodelabel_spark_pi_dag"}
)

pi_value_calculation_3 = SparkSubmitOperator(
    name="airflow_spark_pi_3",
    application="/opt/spark/examples/jars/spark-examples_2.11-2.3.1.jar",
    task_id="pi_value_calculation_3",
    java_class="org.apache.spark.examples.SparkPi",
    conn_id='spark_default',
    driver_memory="512M",
    num_executors=1,
    executor_cores=1,
    executor_memory="512M",
    conf={"spark.yarn.am.nodeLabelExpression" : "nodelabel_spark_pi_dag",  "spark.yarn.executor.nodeLabelExpression": "nodelabel_spark_pi_dag"}
)

dummy_start >> start_nodemanagers >> pi_value_calculation_1  >> pi_value_calculation_2 >> pi_value_calculation_3 >> stop_nodemanagers >> dummy_end
```

- Keep `stop_nodemanagers` task to run just before the `dummy_end` task



```
    driver_memory="512M",
    num_executors=1,
    executor_cores=1,
    executor_memory="512M",
    conf={"spark.yarn.am.nodeLabelExpression" : "nodelabel_spark_pi_dag",  "spark.yarn.executor.nodeLabelExpression": "nodelabel_spark_pi_dag"}
)

dummy_start >> start_nodemanagers >> pi_value_calculation_1  >> pi_value_calculation_2 >> pi_value_calculation_3 >> stop_nodemanagers >> dummy_end
```