Ansible Tutorial

How to configure a cluster, install programs like hadoop and spark and run commands remotely on the cluster using Ansible

Setting up machines:

First we create the configuration files:

inventory

[workers]

10.195.6.170 ansible_user=ubuntu
10.195.6.170 ansible_python_interpreter=/usr/bin/python3

[master]

10.195.6.121 ansible_user=ubuntu
10.195.6.121 ansible_python_interpreter=/usr/bin/python3

ansible.cfg

[defaults]

inventory = inventory
remote_user = ubuntu
private_key_file = ~/.ssh/id_rsa

To check correct setup:

$ ansible all -m ping

Adding mapping for /etc/hosts:

setup_hosts.yml

```
- hosts: all
  tags: all
  tasks:
      - name: Add mappings to /etc/hosts file
        become: yes
        become_user: root
        copy:
            dest: /etc/hosts
            content: |
                127.0.0.1 localhost
                192.168.3.113 node1
                192.168.3.130 node2
                # The following lines are desirable for IPv6 capable hosts
                ::1 ip6-localhost ip6-loopback
```

```
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

$ ansible-playbook --ask-become-pass setup_hosts.yml

Setting up ssh connection

connections.yml

```
- name: SSH Connection - Master
  hosts: master
  tags: ssh, master
  vars_files:
        - external_vars.yml
  tasks:
        - name: generate key pair
          tags: run
          shell: ssh-keygen -t rsa -N ""  -f ~/.ssh/id_rsa
          args:
                creates: "~/.ssh/id_rsa"
        - name: Fetch public key of master
          fetch:
                src: "~/.ssh/id_rsa.pub"
                dest: "files/id_rsa.pub"
                flat: yes
        - name: Fetch private key of master
          fetch:
                src: "~/.ssh/id_rsa"
                dest: "files/id_rsa"
                flat: yes
        - name: Update authorized_hosts
          shell: "cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys"
        - name: Add Nodes to known hosts master
          tags: known_hosts_master
          shell: ssh-keyscan -H { { item.ip } } >> ~/.ssh/known_hosts
          with_items:
              - "{ { nodes } }"
- name: SSH Connection - Worker Nodes
  hosts: workers
  tags: ssh, workers
  vars_files:
     - external_vars.yml
  tasks:
     - name: Copy the public key to worker nodes
       copy:
          src: files/id_rsa.pub
          dest: ~/.ssh/id_rsa.pub
```

```
    - name: Copy the private key to worker nodes
      copy:
        src: files/id_rsa
        dest: ~/.ssh/id_rsa
        mode: 0400
    - name: Update authorized_hosts
      shell: "cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys"
    - name: Add Nodes to known hosts worker
      tags: known_hosts, workers
      shell: ssh-keyscan -H { {item.ip} } >> ~/.ssh/known_hosts
      with_items:
         - "{ {nodes} }"
```

where external_vars.yml looks like:

```
- nodes:
     - {hostname: node1, ip: 192.168.3.113}
     - {hostname: node2, ip: 192.168.3.130}
- master_node:
     - {hostname: node1, ip: 192.168.3.113}
- worker_nodes:
     - {hostname: node2, ip: 192.168.3.130}
```

$ ansible-playbook --ask-become-pass connections.yml

Install Hadoop and Spark:

install_programs.yml

```
---
- name: Installation of Java and Hadoop
  hosts: all
  tags: all
  vars_files:
       - external_vars.yml
  tasks:
       - name: Install Java
         become_user: root
         become: yes
         apt:
              name: openjdk-8-jdk
              update_cache: yes
       - name: Add Java to bashrc
         blockinfile:
              path: ~/.bashrc
              block: |
                   export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
                   export PATH=$PATH:$JAVA_HOME/bin
              marker: "# {mark} ANSIBLE MANAGED BLOCK JAVA"
       - name: Install unzip
```

```
      become_user: root
      become: yes
      package:
         name: unzip
    - name: Install Hadoop
      unarchive:
          src: https://archive.apache.org/dist/hadoop/common/hadoop-2.7.3/hadoop-
2.7.3.tar.gz
          dest: /home/ubuntu
          remote_src: yes
    - name: Add Hadoop to bashrc
      blockinfile:
          path: ~/.bashrc
          block: |
               export HADOOP_HOME=/home/ubuntu/hadoop-2.7.3
               export HADOOP_CONF_DIR=$HOME/hadoop-2.7.3/etc/hadoop/
               export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
          marker: "# {mark} ANSIBLE MANAGED BLOCK HADOOP"
    - name: Install Spark
      unarchive:
          src: https://downloads.apache.org/spark/spark-3.1.1/spark-3.1.1-bin-hadoop2.7.tgz
          dest: /home/ubuntu
          remote_src: yes
    - name: Add Spark to bashrc
      blockinfile:
          path: ~/.bashrc
          block: |
               export SPARK_HOME=$HOME/spark-3.1.1-bin-hadoop2.7
               export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin
          marker: "# {mark} ANSIBLE MANAGED BLOCK SPARK"
```

$ ansible-playbook --ask-become-pass install_programs.yml

configure_hadoop.yml

```
- name: For master
  hosts: master
  tags: hadoop_master_configuration
  vars_files:
      - external_vars.yml
  tasks:
      - name: Edit core-site.xml
        blockinfile:
          path: ~/hadoop-2.7.3/etc/hadoop/core-site.xml
          insertafter: <configuration>
          block: |
               <property>
               <name>fs.default.name</name>
               <value>hdfs://node1:9000</value>
               </property>
          marker: ""
```

```
    - name: Copy the template of mapred-site.xml
      shell: cp ~/hadoop-2.7.3/etc/hadoop/mapred-site.xml.template ~/hadoop-
2.7.3/etc/hadoop/mapred-site.xml
    - name: Edit mapred-site.xml
      blockinfile:
          path: ~/hadoop-2.7.3/etc/hadoop/mapred-site.xml
          insertafter: <configuration>
          block: |
              <property>
              <name>mapred.job.tracker</name>
              <value>node1:54311</value>
              <description>The host and port that the MapReduce job tracker runs
              at.  If "local", then jobs are run in-process as a single map
              and reduce task.
              </description>
              </property>
              <property>
              <name>mapred.child.java.opts</name>
              <value>-Xmx1024m</value>
              </property>
          marker: ""
    - name: Edit hdfs-site.xml
      blockinfile:
          path: ~/hadoop-2.7.3/etc/hadoop/hdfs-site.xml
          insertafter: <configuration>
          block: |
              <property>
              <name>dfs.replication</name>
              <value>1</value>
              </property>
              <property>
              <name>dfs.namenode.name.dir</name>
              <value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
              </property>
          marker: ""
    - name: Ansible check directory.
      stat:
          path: /usr/local/hadoop_tmp/hdfs/namenode
      register: namenode_folder
    - name: Configure hdfs directory if not exists
      become_user: root
      become: yes
      shell: |
          mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
          chown ubuntu:ubuntu -R /usr/local/hadoop_tmp/
          chmod 777 /usr/local/hadoop_tmp/hdfs/namenode/
      when: namenode_folder.stat.exists == false
    - name: Edit yarn-site.xml
      blockinfile:
          path: ~/hadoop-2.7.3/etc/hadoop/yarn-site.xml
          insertafter: <configuration>
```

```
        block: |
            <property>
            <name>yarn.nodemanager.aux-services</name>
            <value>mapreduce_shuffle</value>
            </property>
            <property>
            <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
            <value>org.apache.hadoop.mapred.ShuffleHandler</value>
            </property>
            <property>
            <name>yarn.resourcemanager.resource-tracker.address</name>
            <value>node1:8025</value>
            </property>
            <property>
            <name>yarn.resourcemanager.scheduler.address</name>
            <value>node1:8030</value>
            </property>
            <property>
            <name>yarn.resourcemanager.address</name>
            <value>node1:8050</value>
            </property>
            <property>
            <name>yarn.resourcemanager.webapp.address</name>
            <value>node1:8088</value>
            </property>
            <property>
            <name>yarn.app.mapreduce.am.staging-dir</name>
            <value>/tmp</value>
            </property>
        marker: ""
- name: Add to master file
  blockinfile:
        path: ~/hadoop-2.7.3/etc/hadoop/masters
        create: true
        block: |
            Template:Item.hostname
        marker: "#{mark} ANSIBLE MANAGED BLOCK  Template:Item.hostname"
  with_items:
          - "Template:Master node"
- name: Add to slaves files
  blockinfile:
        path: ~/hadoop-2.7.3/etc/hadoop/slaves
        create: true
        block: |
            Template:Item.hostname
        marker: "#{mark} ANSIBLE MANAGED BLOCK  Template:Item.hostname"
  with_items:
          - "Template:Worker nodes"
- name: change Spark-env.config
  copy:
        dest: ~/spark-3.1.1-bin-hadoop2.7/conf/spark-env.sh
```

```
            force: no
            content: |
                export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
                export SPARK_WORKER_CORES=2
      - name: change Spark-env.slaves
        blockinfile:
            path: ~/spark-3.1.1-bin-hadoop2.7/conf/slaves
            create: true
            block: |
                Template:Item.hostname
                <property>
                <name>yarn.resourcemanager.address</name>
                <value>node1:8050</value>
                </property>
                <property>
                <name>yarn.resourcemanager.webapp.address</name>
                <value>node1:8088</value>
                </property>
            marker: ""
      - name: Add to master file
        blockinfile:
            path: ~/hadoop-2.7.3/etc/hadoop/masters
            create: true
            block: |
                Template:Item.hostname
                        marker: "#{mark} ANSIBLE MANAGED BLOCK Template:Item.hostname"
        with_items: "Template:Worker nodes"
      - name: set JAVA_HOME environment variable
        action: lineinfile dest=~/hadoop-2.7.3/etc/hadoop/hadoop-env.sh regexp='export
JAVA_HOME.*' line='export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64'
      - name: Copy configuration files to all workers
        tags: scp
        shell: scp ~/hadoop-2.7.3/etc/hadoop/masters ~/hadoop-2.7.3/etc/hadoop/slaves
~/hadoop-2.7.3/etc/hadoop/core-site.xml  ~/hadoop-2.7.3/etc/hadoop/mapred-site.xml
~/hadoop-2.7.3/etc/hadoop/yarn-site.xml ~/hadoop-2.7.3/etc/hadoop/hadoop-env.sh
ubuntu@Template:Item.ip:~/hadoop-2.7.3/etc/hadoop/.
        with_items: "Template:Worker nodes"
      - name: Copy spark configuration files to all workers
        tags: scp_spark
        shell: scp ~/spark-3.1.1-bin-hadoop2.7/conf/spark-env.sh ~/spark-3.1.1-bin-
hadoop2.7/conf/slaves ubuntu@Template:Item.ip:~/spark-3.1.1-bin-hadoop2.7/conf/.
        with_items: "Template:Worker nodes"
- name: For workers

 hosts: workers
 tags: hadoop_worker_configuration
 vars_files:
    - external_vars.yml
 tasks:
    - name: Edit worker hdfs-site.xml
      blockinfile:
```

```
        path: ~/hadoop-2.7.3/etc/hadoop/hdfs-site.xml
        insertafter: <configuration>
        block: |
            <property>
            <name>dfs.replication</name>
            <value>1</value>
            </property>
            <property>
            <name>dfs.datanode.name.dir</name>
            <value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
            </property>
        marker: ""
    - name: Ansible check datanode directory
      stat:
        path: /usr/local/hadoop_tmp/hdfs/datanode
      register: datanode_folder
    - name: Configure worker hdfs directory if not exists
      become_user: root
      become: yes
      shell: |
          mkdir -p /usr/local/hadoop_tmp/hdfs/datanode
          chown ubuntu:ubuntu -R /usr/local/hadoop_tmp/
          chmod 777 /usr/local/hadoop_tmp/hdfs/datanode/
      when: datanode_folder.stat.exists == false
```

$ ansible-playbook configure_hadoop.yml

Adding a new machine

1. Update inventory
2. Update external_vars.yml
3. Update setup_hosts.yml
4. Run following playbooks:

$ ansible-playbook --ask-become-pass setup_hosts.yml

$ ansible-playbook --ask-become-pass connections.yml

$ ansible-playbook --limit <new machine's IP> install_programs.yml eg: ansible-playbook --limit 10.195.6.167 install_programs.yml

$ ansible-playbook configure_hadoop.yml

Running hadoop

(NOTE: TO BE ONLY DONE ONCE IN THE BEGINNING WHEN CONFIGURING THE CLUSTER)

namenode_hdfs_format.yaml

```
- name: To format hdfs, used on master
  hosts: master
  tags: hadoop_master_configuration
```

```
  tasks:
        - name: Format namenode
          tags: format
          shell: ~/hadoop-2.7.3/bin/hdfs namenode -format
```

$ ansible-playbook namenode_hdfs_format.yaml

start_hadoop_spark.yml

```
- name: start hadoop and spark cluster
  hosts: master
  tags: start_master
  tasks:
        - name: Start dfs
          shell: ~/hadoop-2.7.3/sbin/start-dfs.sh
        - name: Start yarn
          shell: ~/hadoop-2.7.3/sbin/start-yarn.sh
        - name: Start Spark
          shell: ./spark-3.1.1-bin-hadoop2.7/sbin/start-all.sh
        - name: Check if running
          shell: jps
          register: result
        - debug:
            var: result.stdout_lines
```

$ ansible-playbook start_hadoop_spark.yml

Check status of all Hadoop Nodes:

```
- name: Checking status of workers
  hosts: all
  gather_facts: no
  tasks:
        - name: Check if running
          shell: jps
          register: result
        - debug:
              var: result.stdout_lines
```