**CS 4287/5287: Principles of Cloud Computing**

Programming Assignment #2

Handed out 9/26/2024; Due 10/18/2024 (11:59 pm CDT in Brightspace)

# **Theme: IaC-driven Deployment and Orchestration of a Cloud-based IoT Data Analytics Pipeline using Virtual Machines and Docker**

**Objective**

We will continue to deploy and test our data analytics pipeline as in PA1. Instead of the manual approach in PA1, PA2 will automate the process using Infrastructure-as-Code (IaC) frameworks, in our case Ansible.

**Changes for PA2**

First, terminate all the 4 VMs from PA1. There are four key changes from assignment #1 as follows:

1. **Ansible playbooks:** Convert all the manual steps that you have documented in PA1 into Ansible plays. This will involve creating the four VMs, installing all the necessary packages (including Docker for this assignment), setting firewall rules, downloading and installing Kafka distribution all accomplished using Ansible plays. A skeleton master playbook has been provided. You will have to fill up the logic for all the child playbooks that are invoked by the master playbook. Make changes as deemed necessary.
2. **Producer code changes:** Since our producer is currently sending traffic in one-way direction as it behaves as a strict publisher, we will need to enhance the producer code such that it runs an additional thread that behaves as a consumer. We need this capability to collect the end-to-end response times for ML inferencing for every sample that we send in the pipeline. The producer’s consumer thread will now read the “inference result” topic from the Kafka broker but only the sample that corresponds to the original image sent by this producer. The reason we need to match this is because we are going to run multiple producers as described below. Recall that the ML inference server already produces this topic to the Kafka broker for every sample it is inferencing, and you are already updating the database with the result.
3. **Execution of the data pipeline:** Manually execute Kafka, ZooKeeper and DB as processes just like we did in PA1 on VMs distributing them evenly across the VMs. However, we will execute the producer, consumer and ML server inside Docker containers. Distributed these containers also across the four VMs. Let the producer send at least 1,000 samples at some frequency like one sample per second. Collect all the end-to-end latency results and save them to a file for later graph plotting.
4. **Workload variation:** Once you have the pipeline working, we now increase the workload by introducing multiple producers running in their own containers. We test our solution varying the number of producers from 1 to 4. Basically, we run one producer inside its container on each VM.

**VM and Docker Configuration and Number of VMs per Team**

As before, we will be creating 4 VMs per team (via Ansible) each of m1.medium flavor. Use your team’s private key to create each VM.

**Data collection**

As you are doing this assignment, although I have provided scaffolding code and slides to help you get started, there will still be enough learning curve, and trial/error. Please keep track of the manual effort expended in all the steps as part of this assignment. In a later assignment, we want to compare how much effort was saved through automation.

**Scaffolding:**

Please look under the Scaffolding subfolder in this directory. It comprises a skeleton master playbook, an inventory file and a variables.yaml file to get you started and to provide a structure for the assignment. There is an additional README.docx in this folder providing more details on Ansible and some hints at Ansible packages that you will need.

**Milestones (and hence the deliverables and expectations):**

Note that the deadlines in Brightspace are on Fridays and not Thursdays (our lecture schedule shows Thursdays but that is because we have the Tue/Thu columns)

***Milestone 1: Due in one week***

* All VMs should be created, firewall rules should be set, and all packages installed all using Ansible plays
* Upload short status report in Brightspace.

***Milestone 2: Due end of second week***

* Enhance the producer logic as specified
* Dockerize your producer, consumer and ML inference server code by creating appropriate docker files for each of these artifacts, build the corresponding images on all 4 VMs so that you can start them on any of the VMs, and now run these elements as containers that communicate with the process-based Broker, ZooKeeper and DB
* Upload short status report

***Milestone 3: Final part, due end of third week***

* Vary the number of producers from 1 to 4. For each workload, collect the results and plot the CDF curves one each for when num of producers = 1, 2, 3 and 4
* Observe the impact on 90th and 95th percentile response times for each case
* Submit the code, docker files and plots for the four cases
* Demo the entire assignment to the TA either via Zoom (make appointment) or create a Video showing the entire assignment and its working and submit the entire package to Brightspace. Only one submission per team.

**Rubrics**

* Correctness (program works): 40%
* Documentation of Effort expended (learning curve): 10%
* Code and other artifacts developed for this PA2: 20%
* Documentation of work split among team: 10%
* Zoom-demo or self-explanatory video to TA: 20%