CGI

Course Outline Template

Please furnish the details in the below format and submit the form to the TD SPOC for review The expectation is that the course should include 30% theory and 70% hands-on exercises	
Course Name	Apache Kafka
Course Duration	5 Days (4 Hours Per Day)
Course Timings	2:00 PM to 6:00 PM
Course Start Date	10-Feb-20
Course End Date	14-Feb-20
Mode of Training	Webinar
Webinar Link	**CGI_ Apache Kafka_10-Feb-2020 - 14-Feb-2020_ 2:00 PM to 6:00 PM_ IST** **Please join my meeting from your computer, tablet or smartphone.** https://global.gotomeeting.com/join/231741685 **You can also dial in using your phone.** India (Toll Free): 18002669254 United States: +1 (646) 749-3129 **Access Code: 231-741-685** New to GoToMeeting? Get the app now and be ready when your first meeting starts: https://global.gotomeeting.com/install/231741685

Hardware Requirement required for the participant	Hard disk capacity:	500 GB	
	RAM	8 GB	
	CPU	I5/i7	
Software Requirement required for the participant	Operating system	Ubuntu with Internet Connection VGA/HDMI Connector	
	Software	Confluent Kafka download 4.1 or 5.2.1 Apache Zookeeper tarball Version 3.4.10 Apache kafka tarball Version 2.2.1 JDK 1.8 required Eclipse MySQL Apache Cassandra tarball Version 3.11 Python 2.7 version required and not 3.x	

	Internet access required	Yes(Mandatory).	
	Admin access	sudo to root should be allowed	
	Browser	Any (Chrome preferred)	
	to download		
	Link1	http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html http://mirrors.estointernet.in/apache/zookeeper/ https://www.apache.org/dyn/closer.cgi?path=/kafka/2.2.1/kafka_2.11-2.2.1.tgz http://cassandra.apache.org/download/ https://academy.datastax.com/planet-cassandra/cassandra https://docs.datastax.com/en/developer/devcenter/doc/devcenter/dcInstallation.html https://www.python.org/downloads/ Confluent Kafka: https://www.confluent.io/download/	

Faculty Name	Karthick
Faculty Profile	Attached
Course Pre-requisites	• knowledge of any messaging system, basic knowledge of Java or any programming language. • Some knowledge of Linux or Unix-based systems is desired
Course Objectives	• Describe the architecture of Kafka and explain business use cases thereof • Install & configure Kafka cluster in either single-node or multi-node setup • Implement Kafka Producer and Consumer using real time streaming data • Appreciate and leverage reliability, replication, and persistence features of the Kafka system • Apply their learning to use powerful features like partitions & the messaging system • Use Kafka interfaces as per their organization's requirement
Agenda - Day 1	• Introduction to Apache Kafka • Importance for messaging queue • Need for distributed messaging queue • Conventional solutions and associated problems • What is Kafka? • Need for Kafka • Real time Analytics with Kafka

	• Core Concepts of Kafka • Kafka Architecture • Kafka Use-cases
Agenda - Day 2	• Kafka Cluster • Understanding Kafka Cluster • Installing and Configuring Kafka Cluster • Kafka Producer • Kafka Consumer • Producer and Consumer in Action • Reading Data from Kafka • Lab: Implement Kafka Producer,Consumer using real time streaming data
Agenda - Day 3	Kafka Installation • Single Node and MultiNode Setup • Type of Messaging System • Kafka Data Model • Topics • Partitions • Partition Distribution • Producer • Consumer • Kafka Reliability • Replication in Kafka • Persistence in Kafka • Creating a topic • Modifying a Topic • Creating a Message • Reading a Message

	• Java Interface to Kafka • Producer Side API • Consumer Side API • Lab: Working with Kafka API
Agenda - Day 4	**Advanced Kafka Producers** • Using batching (time/size) • Using compression • Async producers and sync producers • Commit and async commit • Default partitioning (round robin no key, partition on key if key) • Controlling which partition records are written to (custom partitioning) • Message routing to a particular partition (use cases for this) • Advanced Producer configuration Advanced Kafka Consumers • Adjusting poll read size • Implementing at most once message semantics using Java API • Implementing at least once message semantics using Java API • Implementing as close as we can get to exactly once Java API • Re-consume messages that are already consumed
Agenda - Day 5	**Kafka REST Proxy** • Using the REST API to write a Producer • Using the REST API to write a Consumer • Kafka Connect Basics • Modes of Working: Standalone and Distributed • Configuring Connectors • Tracking Kafka Connector Offsets