Kafka Concepts

Cluster, un insieme di brokers.  
Ogni Broker rappresenta un kafka server.

I brokers di un Cluster, possono essere geolocalizzati in posti differenti. Esempio Il cluster che gestisce il nostro applicativo chiamato Amazoom che ha 4 brokers, 2 di essi possono trovarsi in America, 1 in Italia e 1 in India.

Un kafka server può avere uno o più topic.  
Ogni topic può avere più partizioni. Esempio topic “frutta” partizioni 0,1,2,3,4. (Le partizioni sono 0 based)

Il producer, pubblica messaggi sul topic ma la decisione su quale partizione verrà allocato il messaggio sarà preso in carico da Zookeeper.

Zookeeper è un software che gestisce i dati in/out da kafka.

Se Il producer manda 3 messaggi su un topic con 2 partizioni, può capitare che ll messaggio 1 vada sulla partizione 1, il messaggio 2 sulla partizione 1 e il messaggio 3 sulla partizione 2. E’ zookeeper che decide dove pubblicare i messaggi, noi possiamo solo scegliere il topic.

Il consumer è colui che “consuma i messaggi”, tuttavia per utilizzare un consumer, esso deve essere appartenente ad un consumer group.  
Un consumer group è un gruppo di 1 o più consumer iscritti allo stesso topic. Questo perché il consumer group deve iscriversi ad un topic per poter “consumare” i messaggi presenti in esso. Un consumer non iscritto ad un consumer group non ha senso. (Sebbene kafka gli assegnerà un consumer group randomico).

Offset, è il numero dei messaggi pubblicati in un topic.  
ConsumerOffset è il numero di messaggi che un consumer ha letto da uno specifico topic.

Un consumer che legge da un topic, è in grado di leggere da tutte le sue partizioni, il metodo di lettura è round robin style, ovvero un messaggio per partizione alla volta. Esempio di un topic con 1 consumer e 3 partizioni, il consumer procederà in questo modo: primo messaggio partizione 1, primo messaggio partizione 2, primo messaggio partizione 3, secondo messaggio partizione 1, secondo messaggio partizione 2 ecc.

Nel caso ci fosse una sola partizione all’interno di un topic e due consumer (appartenenti allo stesso consumer group), solo uno dei 2 potrà leggere da quella partizione, pertanto, la regola dice i consumer appartenenti allo stesso consumer group iscritti allo stesso topic non possono leggere dalla stessa partizione. Esempio: Topic “TvShow” partizioni 0,1,2, Consumer group “Televisori” fatto da 2 Consumer chiamati c1,c2. Prendendo questo esempio, a c1 potrà essere assegnata la partizione 0 e 2, mentre a c2 la partizione 1, c2 non avrà accesso alle partizioni 0 e 2 cosi come c1 non avrà accesso alla partizione 1.

2 Consumers ognuno appartenente ad un proprio consumer group possono leggere dallo stesso topic e dalla stessa partizione. Ad esempio c1 appartenente al consumer group A e c2 appartenente al consumer group B, potranno entrambi leggere dalla partizione 0 del topic frutta.

Kafka replication system

In kafka esiste il fenomeno di replicazione delle partizioni. Cosa significa? Significa che data una partizione di un topic (esempio partizione 0), essa può essere replicata in altre partizioni identiche, ove in caso di fault di quella partizione, queste istanze copia entreranno in gioco. La partizione dove vengono pubblicati inizialmente i dati (e vengono letti) viene chiamata partizione “Leader” mentre le istanze copie vengono chiamate “Follower”. Il processo di replicazione delle partizioni viene fatto da zookeeper, esso avviene che mentre stiamo pubblicando i dati nelle partizioni del nostro topic, zookeeper replica nello stesso momento copie identiche di tali partizioni. Se il replicator factor è = 1 significa che esiste solo la partizione leader.

Kafka memorizza e trasporta le informazioni sottoforma di array di bytes, questo avviene anche in caso di replicazione delle partizioni, perciò, il producer sarà deputato alla serializzazione dei messaggi, cosi come il consumer sarà deputato alla de serializzazione.