MIRROR BROKERS CONFIGURATIONS		
Key	**Description**	**Values**
	BASE CONFIGURATION	
KAFKA_CONNECT_BOOTSTRAP_SERVERS	Enter source Kafka broker	Enter a HOST:PORT
SASLMECHANISM	Enter the SASL mechanism to authenticate with the broker.	You can enter: PLAIN, SCRAM256, SCRAM512
COMPRESSIONTYPE	Enter type of compression algorithm.	You can use: SNAPPY, GZIP, LZ4
CLOUD_USERNAME	Enter the username or Key for the BOOTSTRAP server	
CLOUD_PASSWORD	Enter the password or Secret for the BOOTSTRAP server	
	MIRROR BROKERS CONFIGURATION	
BROKER_USERNAME_PASS_FROM	Enter the SOURCE broker username and password. You can separate multiple username/pass with a comma	See example in viper-generic.env
BROKER_HOSTPORT_FROM	Enter the SOURCE broker host and port.	See example in viper-generic.env
BROKER_USERNAME_PASS_TO	Enter the DESTINATION broker username and password.	See example in viper-generic.env You can separate multiple username/pass with a comma
BROKER_HOSTPORT_TO	Enter the DESTINATION broker host and port.	See example in viper-generic.env
TOPICS_LIST_FROM	You can leave blank and MB will find the topics for you.	For example, topic1,topic2:topic1,topic2 OR if you want to manually enter topics you can add them here and separate with comma for each broker.
ENABLETLS_FROM	This tells MB if the SOURCE broker uses TLS.	Enter 1=TLS, or 0=NOTLS
ENABLETLS_TO	This tells MB if the DESTINATION broker uses TLS.	Enter 1=TLS, or 0=NOTLS
REPLICATIONFACTOR_FROM	Leave blank to let Viper determine replication factor on SOURCE broker.	
REPLICATIONFACTOR_TO	Leave blank to let Viper determine replication factor on DESTINATION broker.	
COMPRESSION_FROM	Enter compression type on SOURCE broker.	You can use SNAPPY, LZ4, GZIP, NONE
COMPRESSION_TO	Enter compression type on DESTINATION broker.	You can use SNAPPY, LZ4, GZIP, or NONE
SASL_FROM	Enter SASL mechanism on SOURCE broker.	Use None, PLAIN, SCRAM512, or SCRAM256
SASL_TO	Enter SASL mechanism on DESTINATION broker.	Use None, PLAIN, SCRAM512, or SCRAM256
PARTITIONS	Leave blank to let MB determine the partitions	
SERVICENAME_FROM	Enter any name for the service on SOURCE broker	
SERVICENAME_TO	Enter any name for the service on DESTINATION broker	
PARTITION_CHANGE_PERC	Enter a percentage between 0-100, this will tell MB to increase or decrease partitions on the DESTINATION broker.	
TOPIC_FILTER	Yon can enter a filter for the Topic to migrate to destination.	Format is: searchstring1,searchstring2,...:0 or 1:0,1,2 # Middle is 0=AND, 1=OR, # Last is: 0=beginswith, 1=Any, 2=Endswith
SYNC_INTERVAL	Enter a number in seconds. This will tell MB to continuously check the SOURCE brokers for any new data.	MB will ensure no data is duplicated on the destination brokers.
FILEAGEMAX	Enter a number in seconds that indicates the ALIVE state of the container.	If FILEAGEMAX=5, and the container file is older than 5 seconds, then a new container can use these brokers for migration. This eliminates any two containers migrating the same brokers and causing duplication issues.

BROKERJSONFILE	This is the JSON array file containing the brokers to migrate. It can contain as many brokers you like i.e. hundreds or thousands. The file must be formatted as follows (the KEYS must be the same, but of course values will change).	{
		"brokers": [{
		"id":1,
		"brokerfrom": "pkc-6ojv2.us-west4.gcp.confluent.cloud:9092",
		"brokerusernamefrom": "iBCRfjSqyHZ4+n",
		"brokerto": "pkc-419q3.us-east4.gcp.confluent.cloud:9092",
		"brokerusernameto": "L:pm5j+BYyRgGl7zf",
		"enabletlsfrom": "1",
		"enabletlsto": "1",
		"compressionfrom": "snappy",
		"compressionto": "snappy",
		"saslfrom": "PLAIN",
		"saslto": "PLAIN" }] }

MAXBROKERSPERCONTAINER	This is a number that indicates how many brokers each container will migrate. A container can migrate multiple brokers at the same time.	If MAXBROKERSPERCONTAINER=2, means a container will migrate two brokers at the same time. ***To determine number of containers (Replicas), divide total brokers to migrate by MAXBROKERSPERCONTAINER***