

**Confluent – Kafka**

**Kafka Installation Guide**

**Prerequisites:**

* Open needed Connections:

|  |  |  |
| --- | --- | --- |
| **Source IP Address** | **Destination IP Address** | **Port number** |
| Your servers | Active Directory | 749,464,389,53,88 |
| Your servers | Proxy | 8080 |
| Local/JUMP | Servers | SSH, and 9021 on C3 |

* Create needed domain-users:

NOTE: you can to check them all from ***\*\**** **server.properties *\*\**** for old .yml script.

* Install Python and Java:
* Yum install python3
* Yum install java-11-openjdk-devel.x86\_64
* Turn off firewall service:

⇨ Systemctl stop firewalld

* Change /etc/hosts:

🡺 If this is a FRESH installation

\* IP’s for all servers in the cluster.  
\* Follow this schema: *IP FDQN HOSTNAME*

🡺 To add a NEW-NODE like (KSQL)

\*Get the current one from brokers then modify the new one.

* Proxy configuration:

🡺Add the below to /etc/environment:

* export http\_proxy=http://<PROXY\_SERVER\_IP>:8080
* export https\_proxy=https:// <PROXY\_SERVER\_IP>:8080
* export no\_proxy=<SATALLITE\_SERVER\_IP>

Then apply the below command:

* Source /etc/environment

🡺Move the proxy certificates to:

* /etc/pki/ca-trust/source/anchors

*NOTE:*

Give 644 authority on the certificates:

* update-ca-trust enable
* update-ca-trust extract
* Configure the host table and send it to DNS owner to add them to AB DNS:

Add all servers’ entries & add AB DNS and Domain IPs to /etc/hosts

Follow this schema: IP FDQN HOSTNAME

* ADD DNS entry as record (By windows team), Open the needed connection on AD and DNS
* Configure the DNS lookups:

Add DNS servers to /etc/resolv.conf  
nameserver IP

* Configure Kerbores for deafult\_realm :

Edit default\_realm = domain

* Domain joining:

realm join <DOMAIN Name // IP> --user-principal=host/$(hostname -f) --computer-ou="OU=servers,OU=prod,OU=clusters,OU=Confluent,OU=Jordan Users,OU=Jordan,OU=LNA-IT HUB,OU=Arab Bank,DC=ArabBank,DC=Plc" --computer-name=$(hostname -s) --user <DOMAIN ADMIN USER>

* SSH the server with ***\*\**** **Hostnames** ***\*\**** in Ansible.
* To insert host fingerprint to know hosts in ansible.
* Add servers name in the no-proxy parameter in .yml script in Ansible.
* Merge as (full chain) cer/int/root
* And move the Root and intermediate to anchors (in servers)

An update-ca-trust enable / extract

Very Important:

* When configuring a new node make sure that public.pem are send to the new node.

**Security pre-requests:**

* Generating private key:

Run the below commands:

* export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-11.0.9.11-0.el8\_2.x86\_64
* mkdir -p /opt/security/cer/pki
* keytool -genkeypair -alias $(hostname -f) -keyalg RSA -keystore /opt/security/cer/pki/$(hostname -f).jks -keysize 2048 -dname "CN=$(hostname -f),OU=ITSO,O=ARAB BANK PLC,L=AMMAN,ST=AMMAN,C=JO" -ext san=dns:$(hostname -f)
* Generate CSR and send it to CA owners to get it signed (With DNS & TLS entries):
* keytool -certreq -alias $(hostname -f) -keystore /opt/security/cer/pki/$(hostname -f).jks -file /opt/security/cer/pki/$(hostname -f).csr -ext san=dns:$(hostname -f) -ext EKU=serverAuth,clientAuth

Place the generated certificate from CA team in /opt/security/cer/pki, after changing the extension to .PEM

* Verify if the certificate is signed correctly:

openssl x509 -in /opt/security/cer/pki/$(hostname -f).pem -noout –text | grep DNS you should find the DNS hostname

openssl x509 -in /opt/security/cer/pki/$(hostname -f).pem -noout –text | grep TLS you should find the following sentence: TLS Web Server Authentication, TLS Web Client Authentication

* Copy the JDK cacerts file to jssecacerts as follows:
* cp /etc/pki/java/cacerts /etc/pki/java/jssecacerts
* Import the root CA certificate into the JDK truststore:
* keytool -importcert -alias rootca -keystore /etc/pki/java/jssecacerts -file /opt/security/cer/jks/Root.pem
* Append the intermediate CA certificate to the signed host certificate:
* cat /opt/security/cer/jks/Int.pem >> /opt/security/cer/pki/$(hostname -f).pem
* Import it into the keystore:
* keytool -importcert -alias $(hostname -f) -file /opt/security/cer/pki/$(hostname -f).pem -keystore /opt/security/cer/pki/$(hostname -f).jks

**Installation (on Ansible server):**

* Install confluent sources:
* git clone [www.github.com/confluentinc/cp-ansible](http://www.github.com/confluentinc/cp-ansible)
* Modify the installation script /root/cp-ansible/hosts.yml with new IPs, certificate path, …etc
* Run the installation:
* ansible-playbook -i hosts\_example.yml all.yml