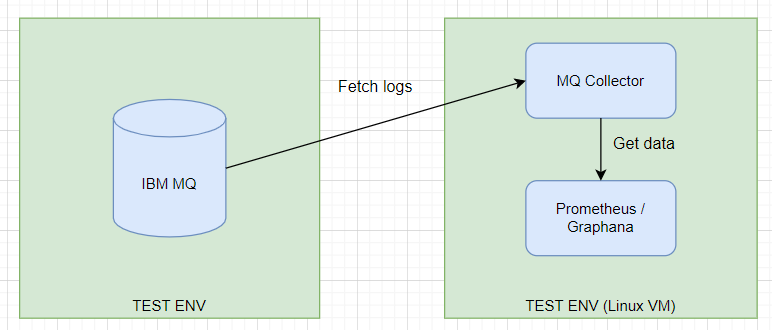
**IBM MQ MONITORING SYSTEM USING PROMETHEUS AND GRAFANA**IBM provides a [sample](https://github.com/ibm-messaging/mq-metric-samples/tree/master/cmd/mq_prometheus) “MQ Exporter for Prometheus monitoring” on [GitHub](https://github.com/ibm-messaging/mq-metric-samples/tree/master/cmd/mq_prometheus). (the original article introducing this sample can be found [here](https://community.ibm.com/community/user/imwuc/viewdocument/using-prometheus-and-grafana-to-mon?CommunityKey=183ec850-4947-49c8-9a2e-8e7c7fc46c64&tab=librarydocuments) on the IBM Community site)

A simple graphical overview illustrates the overall monitoring setup:

  
  
  
The “collector” will retrieve statistics from the IBM MQ queue manager. It will expose them over HTTP to Prometheus. Prometheus scrapes the metrics and stores them in its database. Grafana is used for showing the data on a dashboard (also available in the samples provided by IBM).  
  
The “collector” is written in GO, we only need to build it from the source in order to use it.  
  
**Building the collector for Linux debian based distribution:**  
Few packages will be required to build and run the IBM MQ collector program.  **Required packages –** golang v-1.13 or latest version, ibmmq-client, ibmmq-sdk, java openjdk-8 or 11 and gcc compiler  
  
**golang installation steps:**  
  
$ sudo add-apt-repository ppa:longsleep/golang-backports

$ sudo apt update  
$ sudo apt install golang-go  
$ go version

**Install ibmmq-clinet and ibmmq-sdk packages:**

$ wget https://public.dhe.ibm.com/ibmdl/export/pub/software/websphere/messaging/mqadv/mqadv\_dev921\_ubuntu\_x86-64.tar.gz

$ tar -xvf mqadv\_dev921\_ubuntu\_x86-64.tar.gz

$ cd mqadv\_dev921\_ubuntu\_x86-64 && sudo ./mqlicense.sh -text\_only

$ sudo vi /etc/apt/sources.list.d/ ibmmq-install.list

Inside the file, put this line:

deb [trusted=yes] file:/home/username/Downloads/MQServer ./

$ sudo apt update

$ sudo apt install ibmmq-client ibmmq-sdk

**Install Java openjdk:**  
  
$ sudo apt install default-jdk  
  
**Install IBM MQ collector:**  
  
$ export GOPATH=~/go

$ export GOROOT=/usr/lib/go

$ mkdir -p $GOPATH

$ cd $GOPATH  
$ git clone https://github.com/ibm-messaging/mq-metric-samples.git src/github.com/ibm-messaging/mq-metric-samples  
$ go mod vendor  
$ cd $GOPATH/src/github.com/ibm-messaging/mq-metric-samples

$ export CGO\_LDFLAGS\_ALLOW='-Wl,-rpath.\*'

$ go build -o $GOPATH/bin/mq\_prometheus ./cmd/mq\_prometheus/\*.go

$ cd ~/go/bin && ./ mq\_prometheus –help  
  
  
Install Prometheus and Grafana:  
  
Use below link for installing and configuring Prometheus:  
<https://linuxhint.com/install_prometheus_ubuntu/>  
  
Use below link for installing and configuring Grafana:  
<https://grafana.com/docs/grafana/latest/installation/debian/>  
  
  
**Configurations and setup required in the IBM MQ server side:**  
  
1. Create a server connection channel SYSTEM.ADMIN.SVRCONN if not already exist. It’s required for the remote connection.   
2. Grant read only permission to monitoring user for the IBM MQ queue manager.  
  
$ setmqaut -m QM1 -t QUEUE\_MANAGER -p CONNECTION\_USER\_NAME +connect +inq +dsp

$ setmqaut -m QUEUE\_MANAGER\_NAME -t q -n SYSTEM.DEFAULT.MODEL.QUEUE -p CONNECTION\_USER\_NAME +get +browse +inq

$ setmqaut -m QUEUE\_MANAGER\_NAME -t q -n SYSTEM.ADMIN.COMMAND.QUEUE -p CONNECTION\_USER\_NAME +get +browse +inq +put

$ setmqaut -m QUEUE\_MANAGER\_NAME -t q -n SYSTEM.MQEXPLORER.REPLY.MODEL -p CONNECTION\_USER\_NAME +inq +browse +get +dsp

$ setmqaut -m QUEUE\_MANAGER\_NAME -t channel -n '\*\*' -p CONNECTION\_USER\_NAME +dsp

$ setmqaut -m QUEUE\_MANAGER\_NAME -t q -n '\*\*' -p CONNECTION\_USER\_NAME +dsp +inq +browse

$ setmqaut -m QUEUE\_MANAGER\_NAME -n "\*\*" -t topic -p CONNECTION\_USER\_NAME +dsp +sub  
  
3. Add required auths for the server connection channel  
$ runmqsc QUEUE\_MANAGER\_NAME

$ SET CHLAUTH(SYSTEM.ADMIN.SVRCONN) TYPE(BLOCKUSER) USERLIST('nobody')

$ SET CHLAUTH(SYSTEM.ADMIN.SVRCONN) TYPE(USERMAP) CLNTUSER('CLIENT\_USER\_NAME') USERSRC(MAP) MCAUSER('CONNECTION\_USER\_NAME')  
  
Once the read only user setup is completed for the queue manager, we are good to establish the remote connection between MQ collector and IBM MQ server.   
  
  
**Running IBM MQ collector remotely:**  
  
$ cd ~/go/bin/  
$ ./mq\_prometheus -ibmmq.client=true -ibmmq.connName="ibmmq.server.com(queue\_manager\_port number)" -ibmmq.channel=SYSTEM.ADMIN.SVRCONN -ibmmq.userid='mqtest' -ibmmq.password='mqtest' -ibmmq.queueManager=QM1 -ibmmq.monitoredQueues=q\*,test\* -ibmmq.httpListenPort=9157 -log.level=debug  
  
Ex:-   
  
   
 **Connections parameters required for the IBM MQ collector to establish the remote connection:**  
  
ibmmq.channel - Channel Name

ibmmq.client - Connect as MQ client

ibmmq.connName - Connection Name

ibmmq.httpListenPort - HTTP Listener Port (default "9157")

ibmmq.queueManager - Queue Manager name

ibmmq.monitoredQueues - Patterns of queues to monitor

ibmmq.userid - UserId for MQ connection

ibmmq.password - Password for MQ connection  
  
**Note** - For the connName parameter, IBM MQ server address and Queue Manager tcp listener port is required  
  
  
**Configurations required for the Prometheus:**   
  
The Prometheus server has to know how to contact the MQ monitor. The simplest way is just to add a reference to the monitor in the server's configuration file. For example, by adding this block to /etc/prometheus/prometheus.yml.  
  
 # Adding a reference to an MQ monitor. All we have to do is

# name the host and port on which the monitor is listening.

# Port 9157 is the reserved default port for this monitor.

- job\_name: 'ibmmq'

scrape\_interval: 15s

static\_configs:

- targets: ['hostname.example.com:9157']  
  
  
**Dashboards with Grafana:**The final step is to link Grafana to your Prometheus instance and import a dashboard. The “MQ Prometheus-1541759594068.json” dashboard is a nice start to explore the available metrics, it gives you a global overview of your queue managers.

