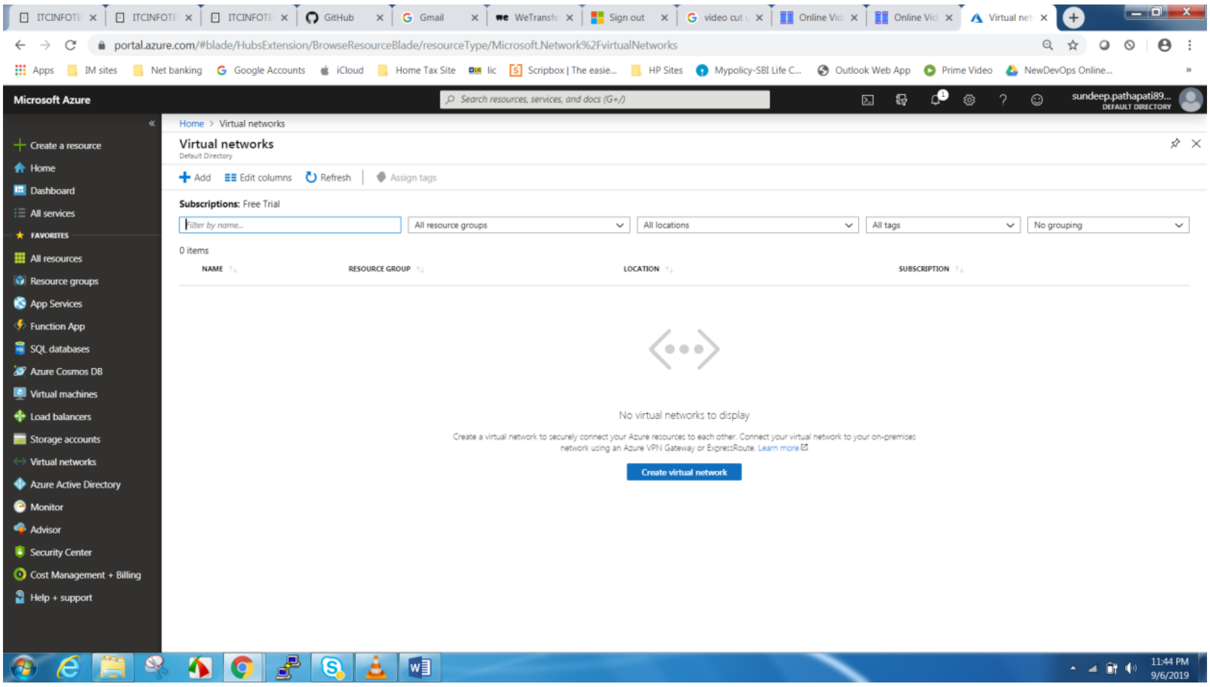
Step 1 : Go to Virtual Networks in Azure.



portal.azure.com/#blade/HubsExtension/BrowseResourceBlade/resourceType/Microsoft.Network%2FvirtualNetworks

Home > Virtual networks

Virtual networks
Default Directory

Subscriptions: Free Trial

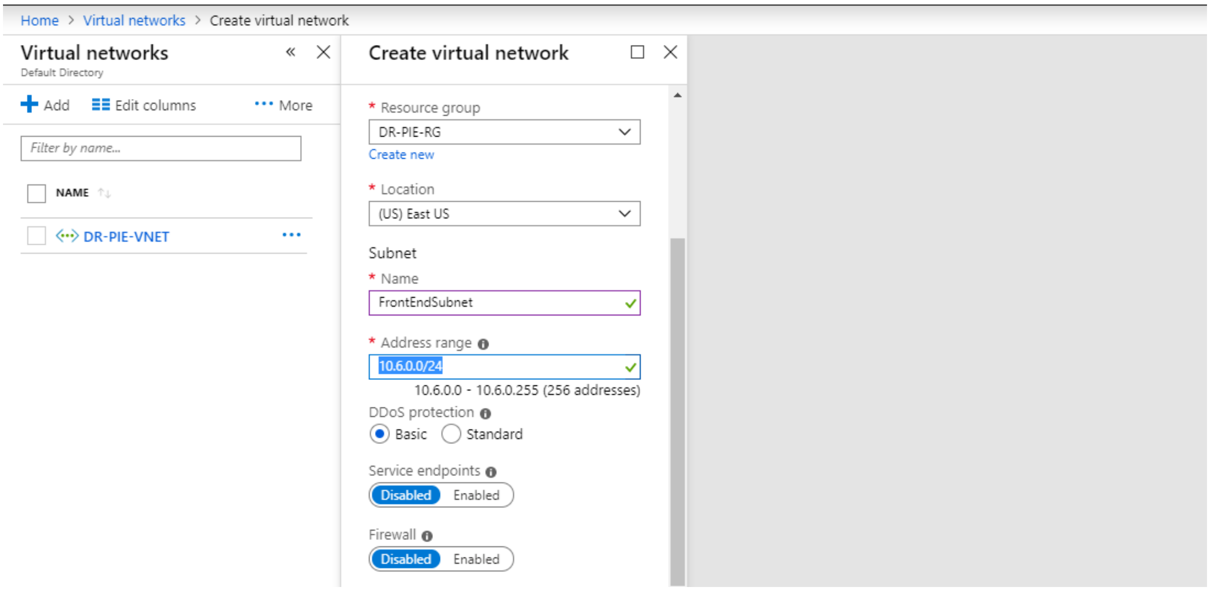
0 items

NAME	RESOURCE GROUP	LOCATION	SUBSCRIPTION

No virtual networks to display

Create a virtual network to securely connect your Azure resources to each other. Connect your virtual network to your on-premises network using an Azure VPN Gateway or ExpressRoute. Learn more

Step 2 : Create a virtual network in **EAST US** with CIDR Address Range(10.6.0.0/16).



Home > Virtual networks > Create virtual network

Virtual networks
Default Directory

DR-PIE-VNET

Create virtual network

* Resource group: DR-PIE-RG

* Location: (US) East US

Subnet

* Name: FrontEndSubnet

* Address range: 10.6.0.0/24

10.6.0.0 - 10.6.0.255 (256 addresses)

DDoS protection: ● Basic ○ Standard

Service endpoints: Disabled

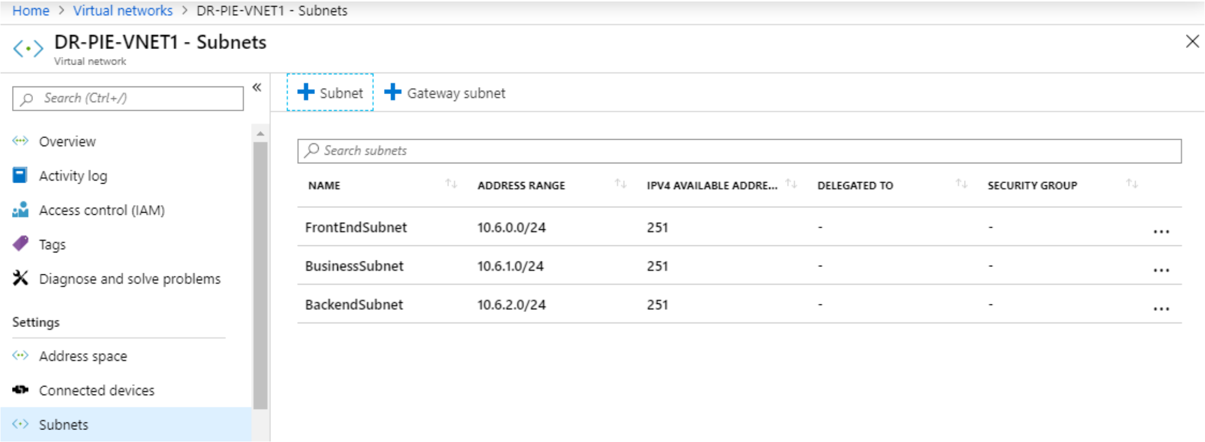
Firewall: Disabled

Step 3: Go to Subnets in the virtual Network and add the Frontend Subnet ,Business Subnet and Backend Subnet with the specified address range.

Frontend Subnet: CIDR Range 10.6.0.0./24

Business Subnet: CIDR Range 10.6.1.0./24

Backend Subnet: CIDR Range 10.6.2.0./24



NAME	ADDRESS RANGE	IPV4 AVAILABLE ADDRE...	DELEGATED TO	SECURITY GROUP	
FrontEndSubnet	10.6.0.0/24	251	-	-	...
BusinessSubnet	10.6.1.0/24	251	-	-	...
BackendSubnet	10.6.2.0/24	251	-	-	...

Step 4: Create a Virtual Machine's in the specified subnet like Apache Web Server in FrontEnd Subnet and Tomcat Server in Business subnet and Mongo server in Backend Subnet.

Apache web server Configuration:

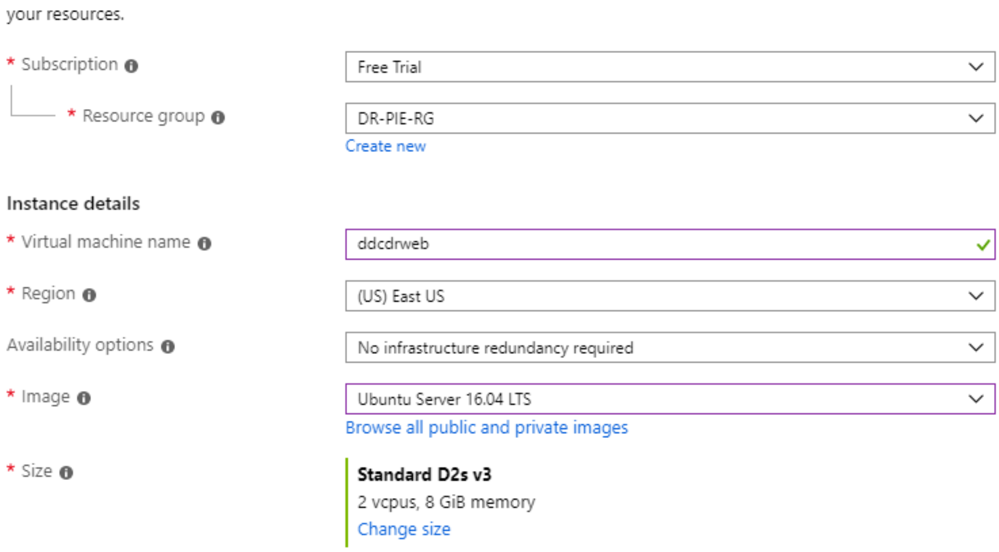
Server name: ddcdrweb

Region: East US

Image: Ubuntu server 16.04 LTS

Size: Standard D2sV3

Network: FrontEnd Subnet (10.6.0.0/24)

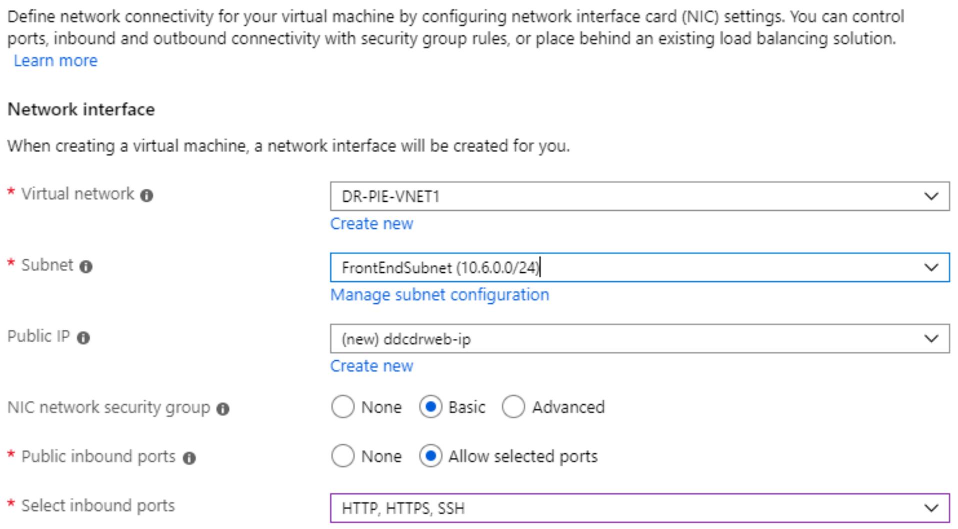


your resources.

* Subscription	Free Trial
* Resource group	DR-PIE-RG Create new

Instance details

* Virtual machine name	ddcdrweb ✓
* Region	(US) East US
Availability options	No infrastructure redundancy required
* Image	Ubuntu Server 16.04 LTS Browse all public and private images
* Size	**Standard D2s v3** 2 vcpus, 8 GiB memory Change size



Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution.
Learn more

Network interface

When creating a virtual machine, a network interface will be created for you.

* Virtual network	DR-PIE-VNET1 Create new
* Subnet	FrontEndSubnet (10.6.0.0/24) Manage subnet configuration
Public IP	(new) ddcdrweb-ip Create new
NIC network security group	○ None ● Basic ○ Advanced
* Public inbound ports	○ None ● Allow selected ports
* Select inbound ports	HTTP, HTTPS, SSH

Follow the same configuration for app server and mongo server.

App Server Configuration:

Server name: **ddcdrapp**

Region: East US

Image: Ubuntu server 16.04 LTS

Size: Standard D2sV3

Network: Business Subnet (10.6.1.0/24)

Mongo Server Configuration:

Server name: **ddcdrmongo**

Region: East US

Image: Ubuntu server 16.04 LTS

Size: Standard D2sV3

Network: Backend Subnet (10.6.2.0/24)

Step 5: After the creation of Virtual machine in each subnet specified. Clone each machine to install required software’s into it.

Installation of Apache Web Server in the VM configured in Front End Subnet of Vnet:

1) Update the Ubuntu Packages
   Command: **sudo apt-get update**
2) Install the Apache Web Server in the system by using the below command
   sudo apt-get install apache2

3) Adjust the Firewall

```
sudo ufw app list

sudo ufw allow 'Apache Full' -> Allow all incoming traffic

sudo ufw status
```

sudo systemctl status apache2

Additional Link:
https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-16-04

Configuration of web server:

Installing and configuring mod_jk

Command**: sudo apt-get install libapache2-mod-jk**

Now we will create our workers.properties file for Apache.

sudo vim /etc/apache2/workers.properties

paste the text

```
# Define 1 real worker using ajp13
worker.list=worker1
# Set properties for worker (ajp13)
worker.worker1.type=ajp13
worker.worker1.host= <Tomcat internal IP>
worker.worker1.port=8009
```

Now to ask Apache to use this worker

sudo vim /etc/apache2/mods-available/jk.conf

change the JkWorkersFile property to

/etc/apache2/workers.properties

Finally to configure the URL Apache should pass through the Tomcat

sudo vim /etc/apache2/sites-enabled/000-default

and add the following line in your configuration

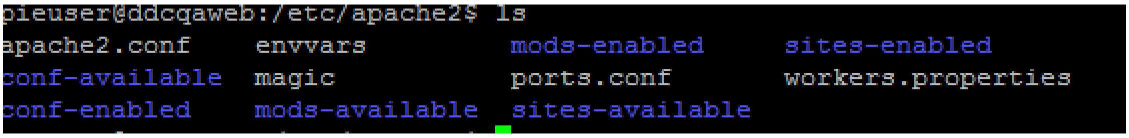
```
<VirtualHost *:80>
  ......................................
  .....................................
  JkMount /incentiveengine* worker1
 JkMount /ruleengine* worker1

</VirtualHost *:80>
```

Now, restart the server

sudo /etc/init.d/apache2 restart

Additional link:
https://medium.com/@arnab.k/installing-tomcat-7-and-apache2-with-mod-jk-on-ubuntu-f2f4d3a9e646

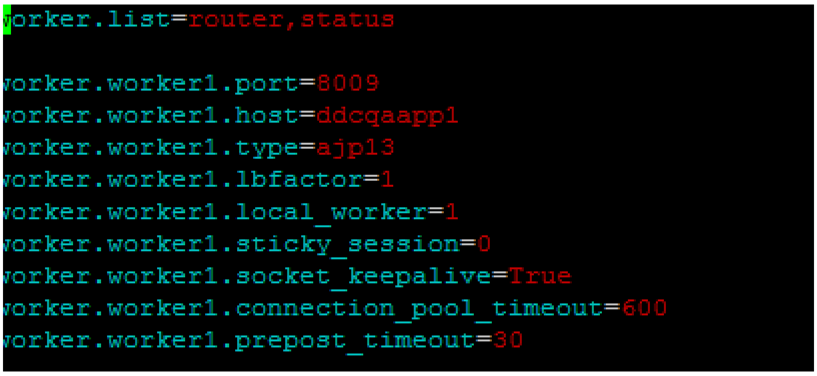
cd /etc/apache2/



```
pieuser@ddcqaweb:/etc/apache2$ ls
apache2.conf    envvars         mods-enabled     sites-enabled
conf-available  magic           ports.conf       workers.properties
conf-enabled    mods-available  sites-available
```

Vi workers.properties

Add the “Tomcat internal IP” in the worker.worker1.host.

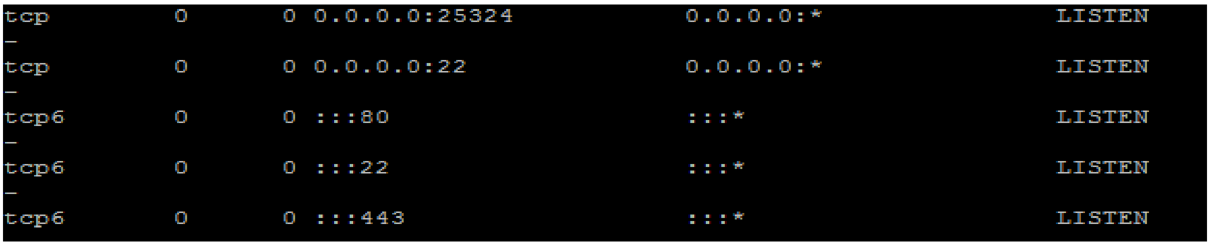


```
worker.list=router,status

worker.worker1.port=8009
worker.worker1.host=ddcqaapp1
worker.worker1.type=ajp13
worker.worker1.lbfactor=1
worker.worker1.local_worker=1
worker.worker1.sticky_session=0
worker.worker1.socket_keepalive=True
worker.worker1.connection_pool_timeout=600
worker.worker1.prepost_timeout=30
```

To check the process is running or not:

Command**: netstat -lntp**

Check the web server port 80 is listening or not.



```
tcp        0      0 0.0.0.0:25324           0.0.0.0:*               LISTEN
-
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
-
tcp6       0      0 :::80                   :::*                    LISTEN
-
tcp6       0      0 :::22                   :::*                    LISTEN
-
tcp6       0      0 :::443                  :::*                    LISTEN
```

Installation of Apache Tomcat Server in the VM configured in Business Subnet of Vnet:

prerequisites:

1) Java Installation
   Command: **sudo apt-get install openjdk-8-jdk**

Additional Link:
https://www.geofis.org/en/install/install-on-linux/install-openjdk-8-on-ubuntu-trusty/

2)Tomcat Installation

Command: **sudo apt-get install tomcat8**

Check the tomcat status: **sudo service tomcat8 status**

Or **systemctl status tomcat8.service**

Enable the service:

systemctl enable tomcat8.service

/lib/systemd/systemd-sysv-install enable tomcat8

Deploy the “**incentiveengine.war**” and “**ruleengine.war**” in tomcat webapps folder

cd /var/lib/tomcat8/webapps/

ls

incentiveengine.war ruleengine.war

Restart the service:

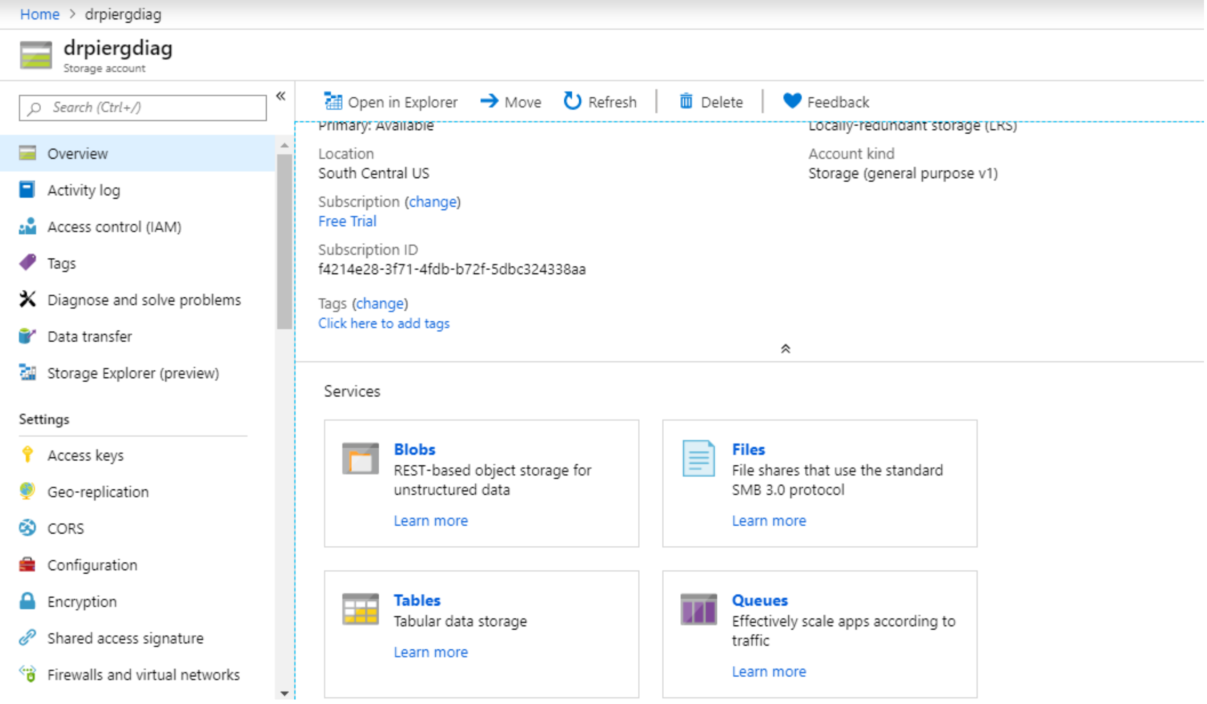
sudo service tomcat8 restart

Additional Link:

http://www.ubuntugeek.com/install-tomcat-8-on-ubuntu-16-04-server.html

Configuration:

Create a storage account



Primary: Available

Locally-redundant storage (LRS)

Location
South Central US

Account kind
Storage (general purpose v1)

Subscription (change)
Free Trial

Subscription ID
f4214e28-3f71-4fdb-b72f-5dbc324338aa

Tags (change)
Click here to add tags

Services

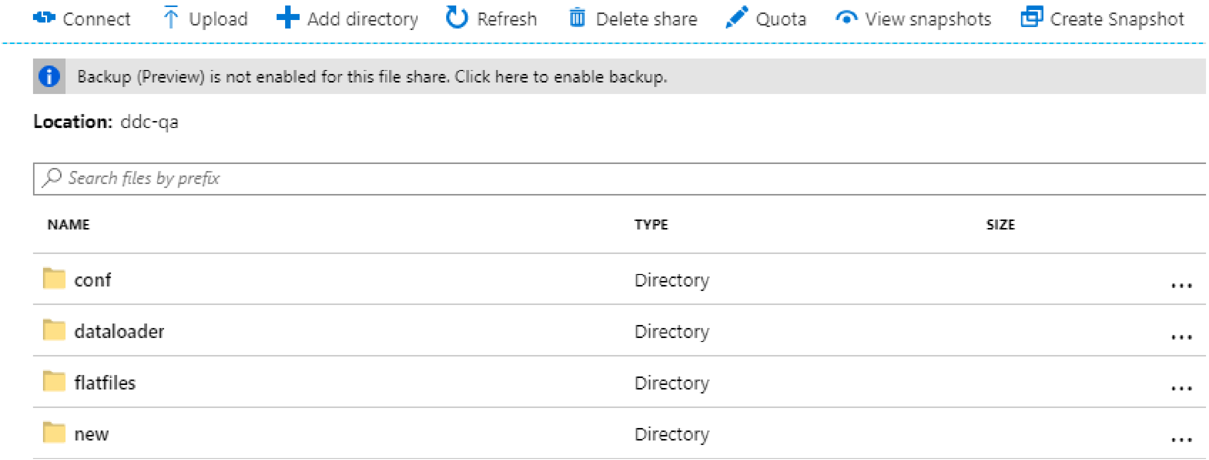
Blobs
REST-based object storage for unstructured data
Learn more

Files
File shares that use the standard SMB 3.0 protocol
Learn more

Tables
Tabular data storage
Learn more

Queues
Effectively scale apps according to traffic
Learn more

In the files section create a file share and add the below mentioned files in the created container.



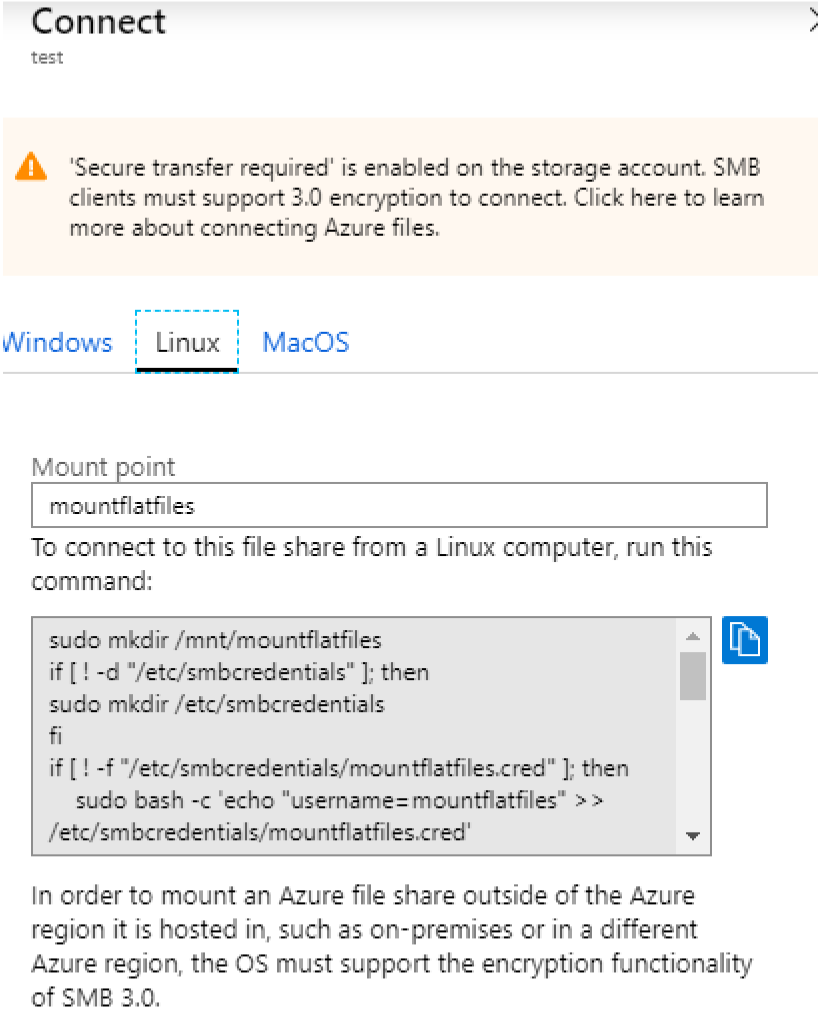
Backup (Preview) is not enabled for this file share. Click here to enable backup.

Location: ddc-qa

NAME	TYPE	SIZE
conf	Directory	...
dataloader	Directory	...
flatfiles	Directory	...
new	Directory	...

Mount the above files in the /opt/pie location of the app server.

To connect use the following command, which we got from the file share connect option.



'Secure transfer required' is enabled on the storage account. SMB clients must support 3.0 encryption to connect. Click here to learn more about connecting Azure files.

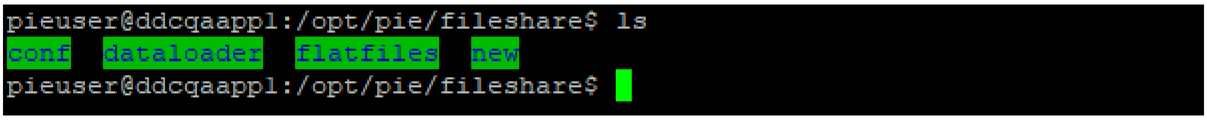
Mount point: mountflatfiles

To connect to this file share from a Linux computer, run this command:

```
sudo mkdir /mnt/mountflatfiles
if [ ! -d "/etc/smbcredentials" ]; then
sudo mkdir /etc/smbcredentials
fi
if [ ! -f "/etc/smbcredentials/mountflatfiles.cred" ]; then
    sudo bash -c 'echo "username=mountflatfiles" >>
/etc/smbcredentials/mountflatfiles.cred'
```

In order to mount an Azure file share outside of the Azure region it is hosted in, such as on-premises or in a different Azure region, the OS must support the encryption functionality of SMB 3.0.

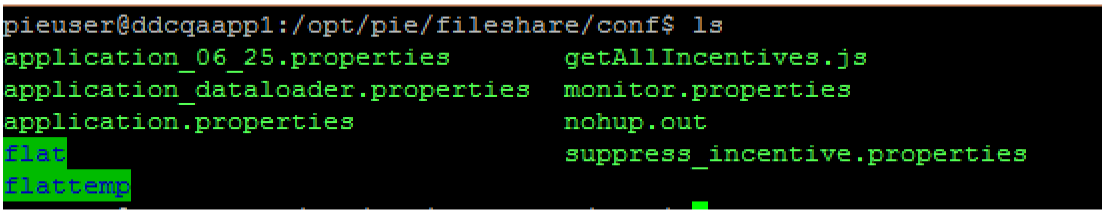
After mount, we will see the list of the files.



```
pieuser@ddcqaappl:/opt/pie/fileshare$ ls
conf  dataloader  flatfiles  new
pieuser@ddcqaappl:/opt/pie/fileshare$
```

Go to the location:

cd /opt/pie/fileshare/conf/



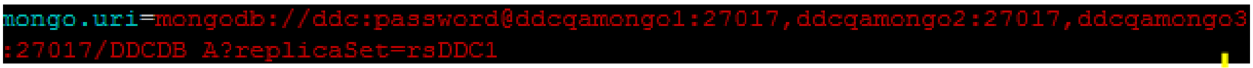
```
pieuser@ddcqaappl:/opt/pie/fileshare/conf$ ls
application_06_25.properties        getAllIncentives.js
application_dataloader.properties   monitor.properties
application.properties              nohup.out
flat                                suppress_incentive.properties
flattemp
```

vi application.properties

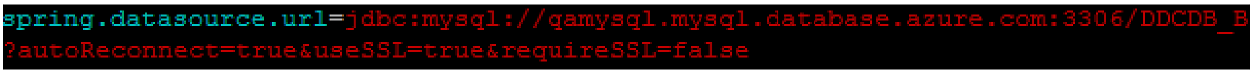
Add the **mongo url** and **mysql url** in the “**application.properties**” and “**application_dataloader.properties**”

mongo.uri=mongodb://dbuser:Welcome123@mongoserverdb:27017

spring.datasource.url=jdbc:mysql://ddcmysqldr.mysql.database.azure.com:3306/DDCDB_A?autoReconnect=true&useSSL=true&requireSSL=false



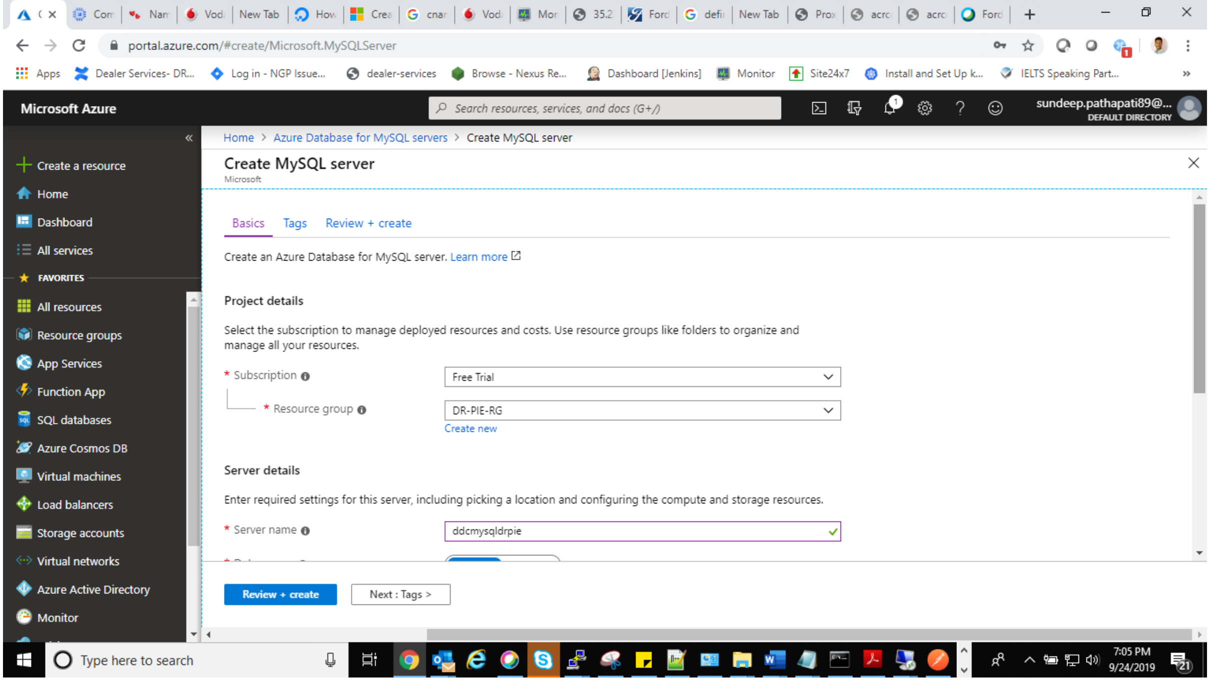
```
mongo.uri=mongodb://ddc:password@ddcqamongo1:27017,ddcqamongo2:27017,ddcqamongo3
:27017/DDCDB A?replicaSet=rsDDC1
```



```
spring.datasource.url=jdbc:mysql://qamysql.mysql.database.azure.com:3306/DDCDB_B
?autoReconnect=true&useSSL=true&requireSSL=false
```

Mysql Service in Azure:

Create mysql service in azure



portal.azure.com/#create/Microsoft.MySQLServer

Home > Azure Database for MySQL servers > Create MySQL server

Create MySQL server

Microsoft

Basics | Tags | Review + create

Create an Azure Database for MySQL server. Learn more

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

* Subscription: Free Trial

* Resource group: DR-PIE-RG

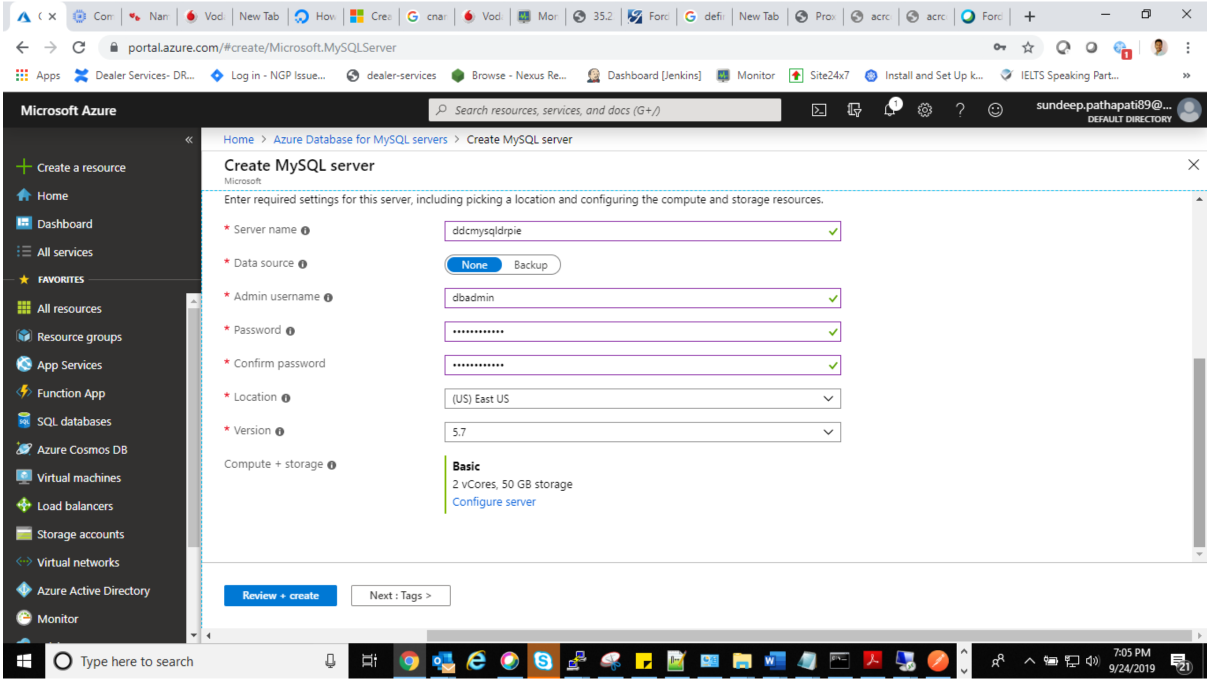
Create new

Server details

Enter required settings for this server, including picking a location and configuring the compute and storage resources.

* Server name: ddcmysqldrpie ✓

Review + create | Next : Tags >



portal.azure.com/#create/Microsoft.MySQLServer

Home > Azure Database for MySQL servers > Create MySQL server

Create MySQL server

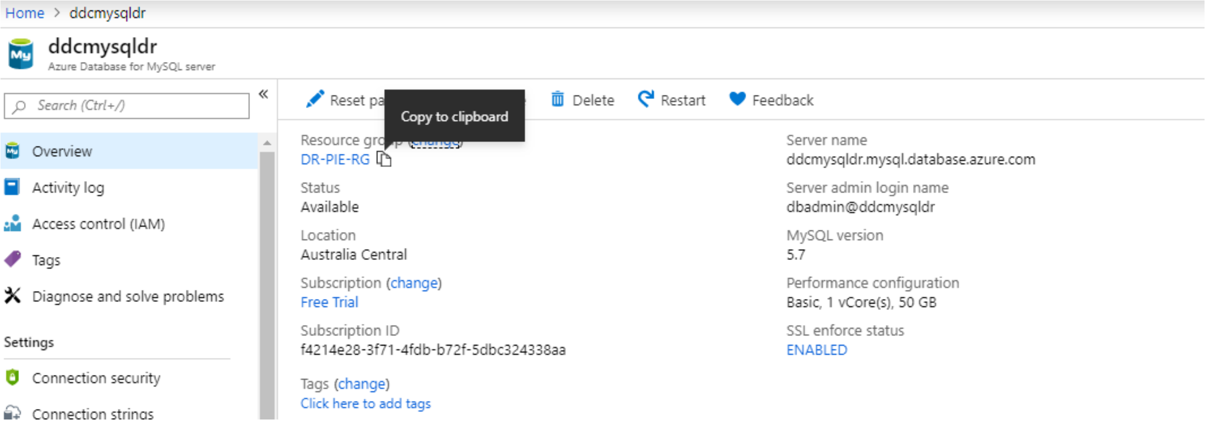
Microsoft

Enter required settings for this server, including picking a location and configuring the compute and storage resources.

* Server name	ddcmysqldrpie
* Data source	[✓ None] Backup
* Admin username	dbadmin
* Password	•••••••••••••
* Confirm password	•••••••••••••
* Location	(US) East US
* Version	5.7
Compute + storage	**Basic** 2 vCores, 50 GB storage Configure server

Review + create | Next : Tags >

After review and create, check the status of the server.



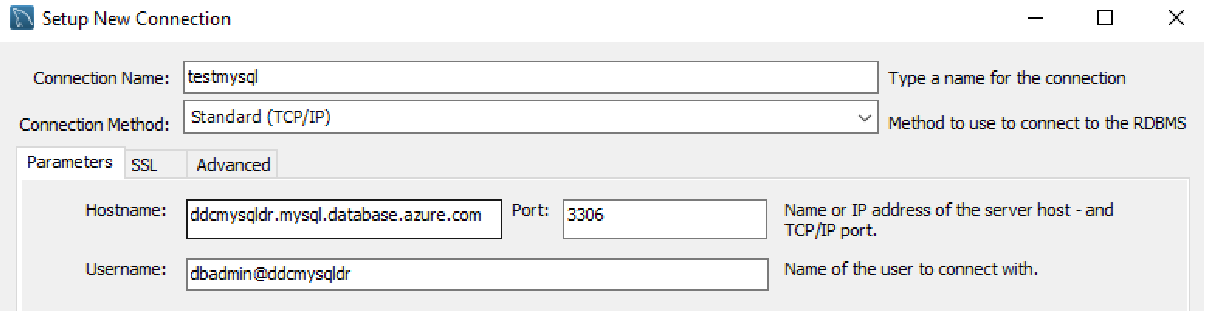
Home > ddcmysqldr

ddcmysqldr

Azure Database for MySQL server

Resource group (change) DR-PIE-RG	Server name ddcmysqldr.mysql.database.azure.com
Status Available	Server admin login name dbadmin@ddcmysqldr
Location Australia Central	MySQL version 5.7
Subscription (change) Free Trial	Performance configuration Basic, 1 vCore(s), 50 GB
Subscription ID f4214e28-3f71-4fdb-b72f-5dbc324338aa	SSL enforce status ENABLED
Tags (change) Click here to add tags	

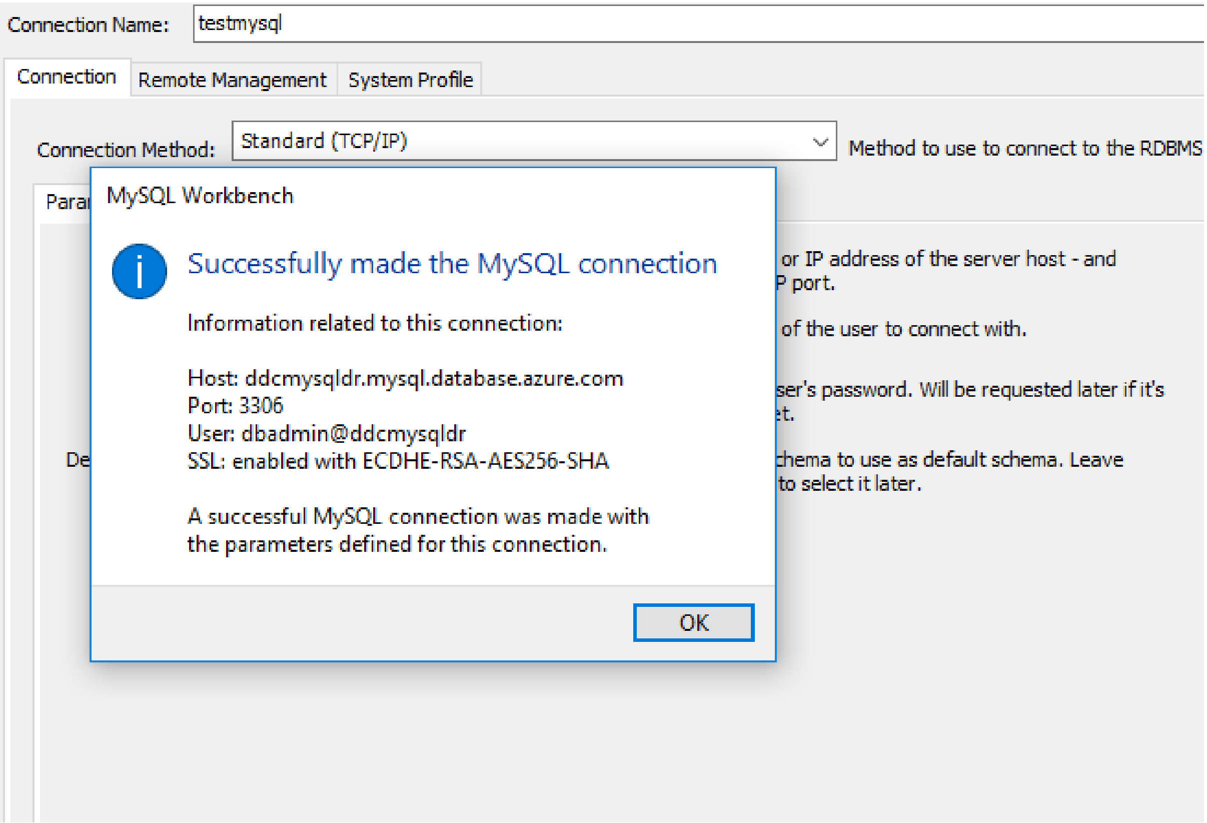
Establish the connection from MySQL Workbench



Setup New Connection

Connection Name:	testmysql	Type a name for the connection
Connection Method:	Standard (TCP/IP)	Method to use to connect to the RDBMS
Hostname:	ddcmysqldr.mysql.database.azure.com Port: 3306	Name or IP address of the server host - and TCP/IP port.
Username:	dbadmin@ddcmysqldr	Name of the user to connect with.

Ensure that connection is successful.



Connection Name: testmysql

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

MySQL Workbench

Successfully made the MySQL connection

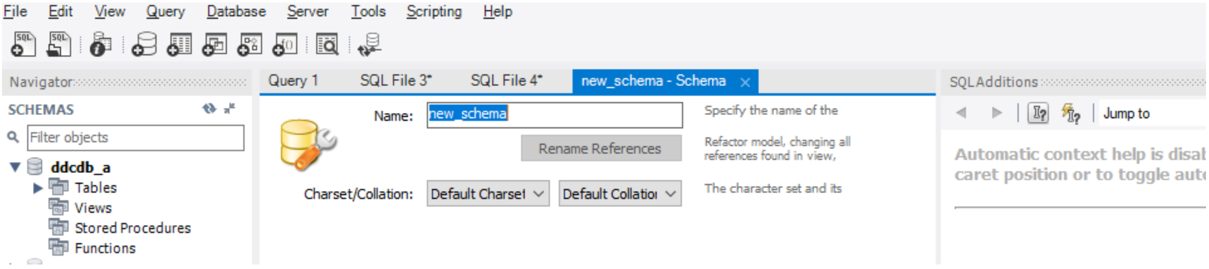
Information related to this connection:

Host: ddcmysqldr.mysql.database.azure.com
Port: 3306
User: dbadmin@ddcmysqldr
SSL: enabled with ECDHE-RSA-AES256-SHA

A successful MySQL connection was made with the parameters defined for this connection.

OK
Para
De
or IP address of the server host - and
P port.
of the user to connect with.
ser's password. Will be requested later if it's
et.
chema to use as default schema. Leave
to select it later.

Create a new database ddcdc_a

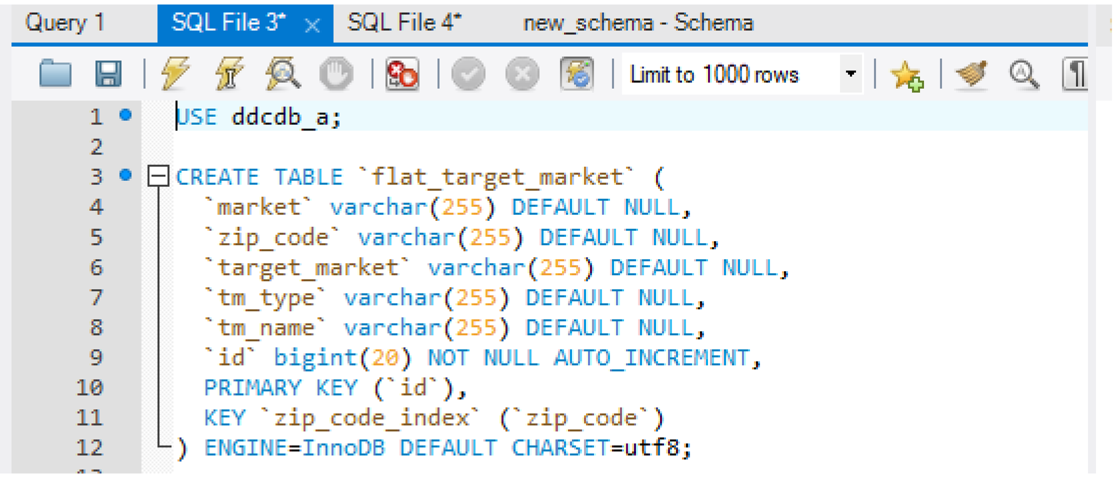


File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
ddcdb_a
Tables
Views
Stored Procedures
Functions
Query 1
SQL File 3*
SQL File 4*
new_schema - Schema
Rename References
Specify the name of the
Refactor model, changing all references found in view.
The character set and its
SQLAdditions
Jump to
Automatic context help is disal
caret position or to toggle auto

Name: new_schema

Charset/Collation: Default Charset · Default Collation

Execute the create_flat_tables.sql file

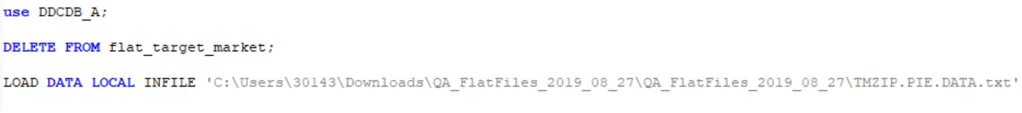


```
USE ddcdb_a;

CREATE TABLE `flat_target_market` (
  `market` varchar(255) DEFAULT NULL,
  `zip_code` varchar(255) DEFAULT NULL,
  `target_market` varchar(255) DEFAULT NULL,
  `tm_type` varchar(255) DEFAULT NULL,
  `tm_name` varchar(255) DEFAULT NULL,
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`id`),
  KEY `zip_code_index` (`zip_code`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Download the Flat Files from the wiki page

https://wiki.ngptools.com/confluence/pages/viewpage.action?spaceKey=PIE&title=Flat+files

change the location of the flatfiles in the “load_flat_csv_to_mysql.sql” and execute the file.



```
use DDCDB_A;

DELETE FROM flat_target_market;

LOAD DATA LOCAL INFILE 'C:\Users\30143\Downloads\QA_FlatFiles_2019_08_27\QA_FlatFiles_2019_08_27\TMZIP.PIE.DATA.txt'
```

Step 3: Installation of mongo server in Backend subnet:

Command: **sudo apt-get install -y mongodb-org**

Command to start mongo service:

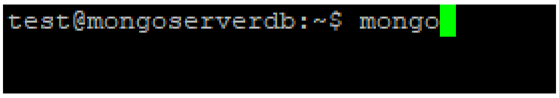
sudo service mongod start

Enable the mongo Service:

systemctl enable mongod

Commands to hit in mongo:

mongo



```
test@mongoserverdb:~$ mongo
```

use admin

db.createUser({user:"admin", pwd:"admin123", roles:[{role:"root", db:"admin"}]})

use DDCDB_A;

use the below command to create mongo user.

```
db.createUser(
 {
   user: "dbmongouser",
   pwd: "Welcome123",
   roles: [ { role: "dbOwner", db: "DDCDB_A" } ]
 }
)
```

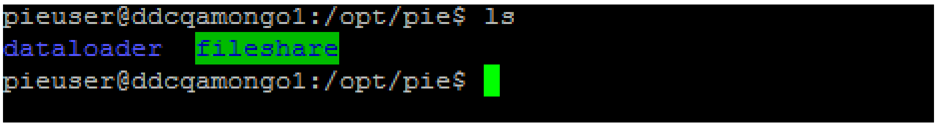
db.auth("dbmongouser", "Welcome123")

Additional Link:

https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/

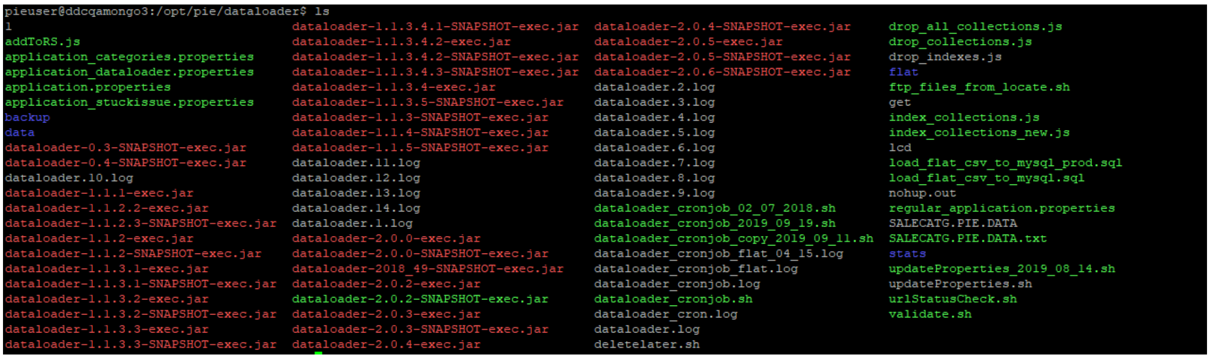
Configuration:

1)Mount the fileshare to /opt/pie/ location in the mongo server.



```
pieuser@ddcqamongo1:/opt/pie$ ls
dataloader  fileshare
pieuser@ddcqamongo1:/opt/pie$
```

2)copy the “application.properties”, “application_dataloader.properties” and the latest jar file of dataloader (dataloader-2.0.3-exec.jar)



```
pieuser@ddcqamongo3:/opt/pie/dataloader$ ls
1                                    dataloader-1.1.3.4.1-SNAPSHOT-exec.jar dataloader-2.0.4-SNAPSHOT-exec.jar     drop_all_collections.js
addToRS.js                           dataloader-1.1.3.4.2-exec.jar          dataloader-2.0.5-exec.jar              drop_collections.js
application_categories.properties    dataloader-1.1.3.4.2-SNAPSHOT-exec.jar dataloader-2.0.5-SNAPSHOT-exec.jar     drop_indexes.js
application_dataloader.properties    dataloader-1.1.3.4.3-SNAPSHOT-exec.jar dataloader-2.0.6-SNAPSHOT-exec.jar     flat
application.properties               dataloader-1.1.3.4-exec.jar            dataloader.2.log                       ftp_files_from_locate.sh
application_stuckissue.properties    dataloader-1.1.3.5-SNAPSHOT-exec.jar   dataloader.3.log                       get
backup                               dataloader-1.1.3-SNAPSHOT-exec.jar     dataloader.4.log                       index_collections.js
data                                 dataloader-1.1.4-SNAPSHOT-exec.jar     dataloader.5.log                       index_collections_new.js
dataloader-0.3-SNAPSHOT-exec.jar     dataloader-1.1.5-SNAPSHOT-exec.jar     dataloader.6.log                       lcd
dataloader-0.4-SNAPSHOT-exec.jar     dataloader.11.log                      dataloader.7.log                       load_flat_csv_to_mysql_prod.sql
dataloader.10.log                    dataloader.12.log                      dataloader.8.log                       load_flat_csv_to_mysql.sql
dataloader-1.1.1-exec.jar            dataloader.13.log                      dataloader.9.log                       nohup.out
dataloader-1.1.2.2-exec.jar          dataloader.14.log                      dataloader_cronjob_02_07_2018.sh       regular_application.properties
dataloader-1.1.2.3-SNAPSHOT-exec.jar dataloader.1.log                       dataloader_cronjob_2019_09_19.sh       SALECATG.PIE.DATA
dataloader-1.1.2-exec.jar            dataloader-2.0.0-exec.jar              dataloader_cronjob_copy_2019_09_11.sh  SALECATG.PIE.DATA.txt
dataloader-1.1.2-SNAPSHOT-exec.jar   dataloader-2.0.0-SNAPSHOT-exec.jar     dataloader_cronjob_flat_04_15.log      stats
dataloader-1.1.3.1-exec.jar          dataloader-2018_49-SNAPSHOT-exec.jar   dataloader_cronjob_flat.log            updateProperties_2019_08_14.sh
dataloader-1.1.3.1-SNAPSHOT-exec.jar dataloader-2.0.2-exec.jar              dataloader_cronjob.log                 updateProperties.sh
dataloader-1.1.3.2-exec.jar          dataloader-2.0.2-SNAPSHOT-exec.jar     dataloader_cronjob.sh                  urlStatusCheck.sh
dataloader-1.1.3.2-SNAPSHOT-exec.jar dataloader-2.0.3-exec.jar              dataloader_cron.log                    validate.sh
dataloader-1.1.3.3-exec.jar          dataloader-2.0.3-SNAPSHOT-exec.jar     dataloader.log
dataloader-1.1.3.3-SNAPSHOT-exec.jar dataloader-2.0.4-exec.jar              deletelater.sh
```

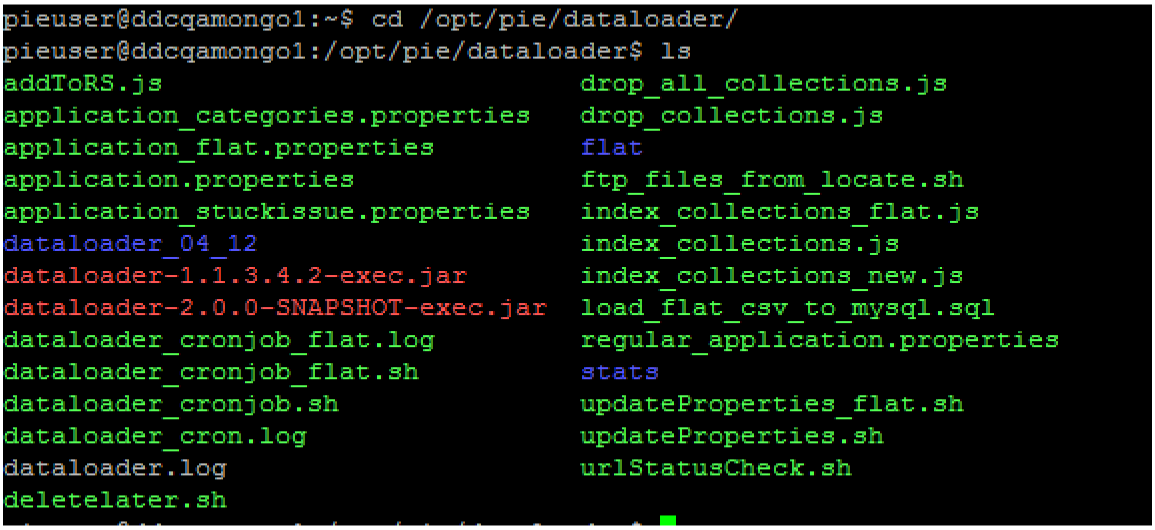
Ensure the **mongo url** and **MySQL url** in the **application.Properties** and **application_dataloader.properties**.

mongo.uri=mongodb://dbuser:Welcome123@mongoserverdb:27017

spring.datasource.url=jdbc:mysql://ddcmysqldr.mysql.database.azure.com:3306/DDCDB_A?autoReconnect=true&useSSL=true&requireSSL=false

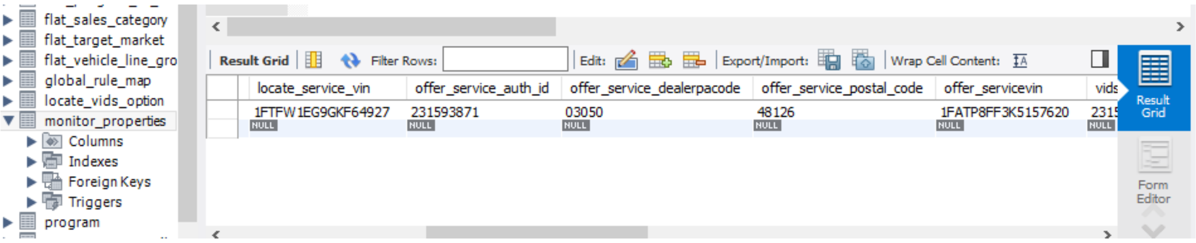
Go to /opt/pie/dataloader/ and Run the latest jar file file by using the command:

Java -jar dataloader-2.0.5-exec.jar



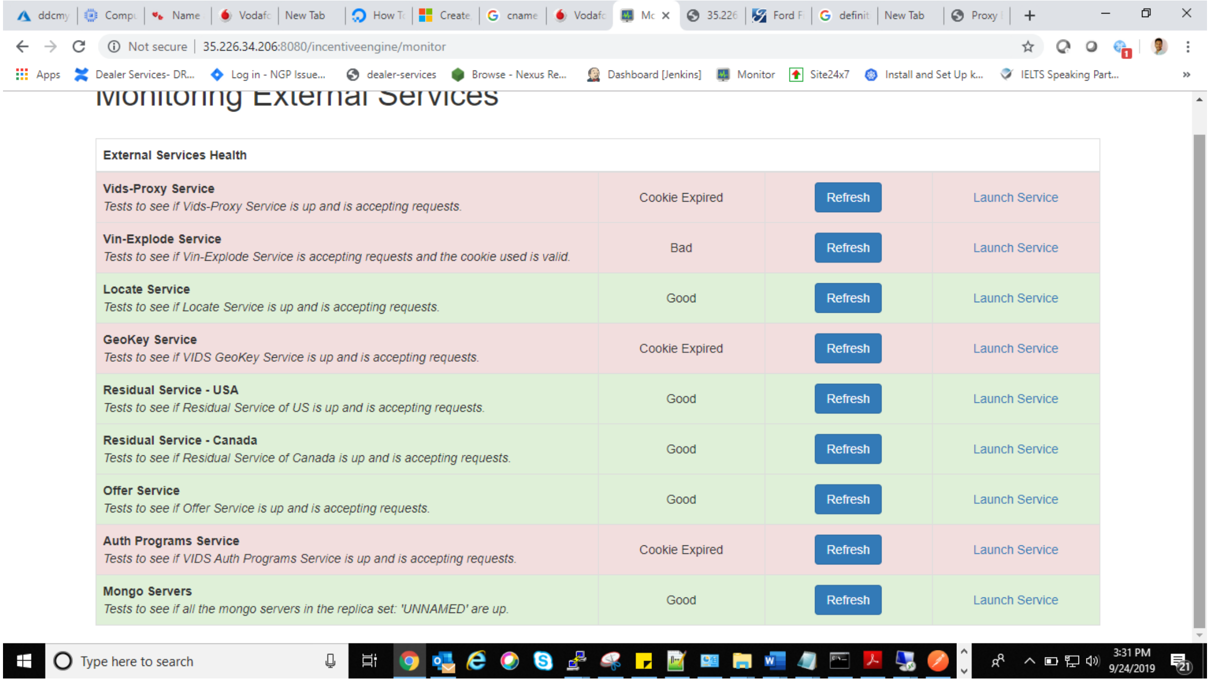
```
pieuser@ddcqamongo1:~$ cd /opt/pie/dataloader/
pieuser@ddcqamongo1:/opt/pie/dataloader$ ls
addToRS.js                              drop_all_collections.js
application_categories.properties       drop_collections.js
application_flat.properties             flat
application.properties                  ftp_files_from_locate.sh
application_stuckissue.properties       index_collections_flat.js
dataloader_04_12                        index_collections.js
dataloader-1.1.3.4.2-exec.jar           index_collections_new.js
dataloader-2.0.0-SNAPSHOT-exec.jar      load_flat_csv_to_mysql.sql
dataloader_cronjob_flat.log             regular_application.properties
dataloader_cronjob_flat.sh              stats
dataloader_cronjob.sh                   updateProperties_flat.sh
dataloader_cron.log                     updateProperties.sh
dataloader.log                          urlStatusCheck.sh
deletelater.sh
```

If offer service is bad then need to update “monitor.properties” with the valid VIN.



locate_service_vin	offer_service_auth_id	offer_service_dealerpacode	offer_service_postal_code	offer_servicevin	vids
1FTFW1EG9GKF64927	231593871	03050	48126	1FATP8FF3K5157620	2315
NULL	NULL	NULL	NULL	NULL	NULL

Monitor Page:



Monitoring External Services

External Services Health			
Vids-Proxy Service *Tests to see if Vids-Proxy Service is up and is accepting requests.*	Cookie Expired	Refresh	Launch Service
Vin-Explode Service *Tests to see if Vin-Explode Service is accepting requests and the cookie used is valid.*	Bad	Refresh	Launch Service
Locate Service *Tests to see if Locate Service is up and is accepting requests.*	Good	Refresh	Launch Service
GeoKey Service *Tests to see if VIDS GeoKey Service is up and is accepting requests.*	Cookie Expired	Refresh	Launch Service
Residual Service - USA *Tests to see if Residual Service of US is up and is accepting requests.*	Good	Refresh	Launch Service
Residual Service - Canada *Tests to see if Residual Service of Canada is up and is accepting requests.*	Good	Refresh	Launch Service
Offer Service *Tests to see if Offer Service is up and is accepting requests.*	Good	Refresh	Launch Service
Auth Programs Service *Tests to see if VIDS Auth Programs Service is up and is accepting requests.*	Cookie Expired	Refresh	Launch Service
Mongo Servers *Tests to see if all the mongo servers in the replica set: 'UNNAMED' are up.*	Good	Refresh	Launch Service