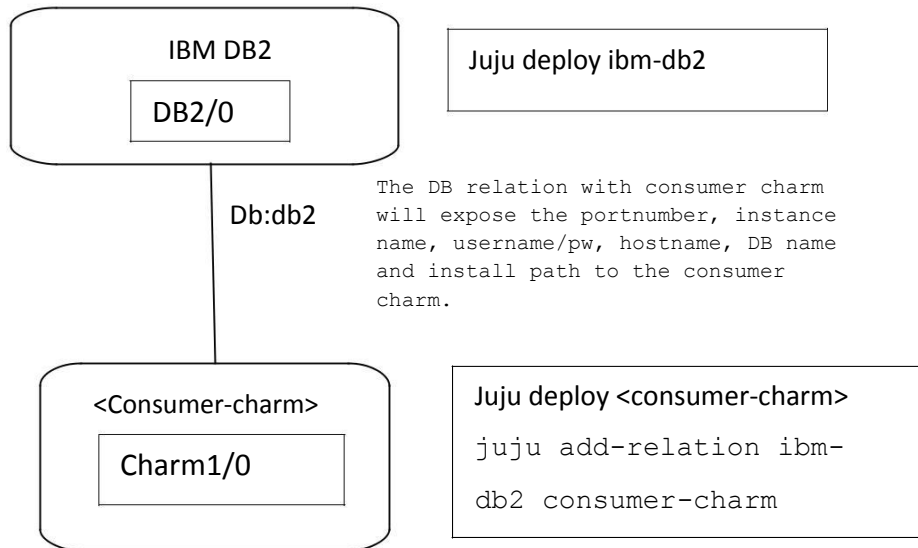


## **IBM DB2 charm design**

This charm is supported for Ubuntu (Trusty) on X86, S390x and Power.



Consumer-charm is any charm which needs to connect to ibm db2 charm.

## Design Considerations

IBM DB2 is a reactive charm which is created from the IBM Base layer and DB2 Interface. The IBM Base layer takes care of common things like downloading the package, checking the checksum after download, accepting the license etc. The DB2 layer takes care of installation, post configuration etc.

The Consumer charm requires a relationship to IBM DB2 charm using the “db2” interface. The IBM DB2 charm implements the db2 interface, which fulfills the db relation. Db2 charm will provide hostname where DB2 is installed, port number, user name and password to connect to DB, instance name and db name to the consumer charm. If the consumer charm provides DB names to be created as a string of DBs separated with comma, (eg : dbnames="APPCNTR,WRKLGHT") DB2 charm will create these Dbs. Otherwise DB2 charm will create a default DB based on the service name.

The basic layer and db2 interface use charms.reactive to coordinate with ibm-db2 layer. This makes it easy to coordinate the state of multiple layers, relations, configuration options, etc. Code for reactive handlers will be under a reactive/ folder in ibm-db2 charm layer. In our case, we'll use reactive/install.sh.

## DB2 Interface

The DB2 interface has both provides and requires part. The consumer charm sends the SSH key value and DB names to DB2 charm, which will set the SSH key in the `authorized_keys` file in the DB2 unit. This will help the consumer charm to connect to DB2 machine and run DB2 commands. The DB2 charm on the other hand will send the hostname, port number, user name and password, instance name and DB name to the consumer charm. These are done in the provides and requires part of the DB2 interface. These sets the states **db.connected** when the relation hook is joined and **db.available** when the relation hook is changed. Both these states are unset when the relation hook is departed.

## Install.sh

The pattern is “reactive” because usage of `@when` and similar decorators to indicate that blocks of code react to certain conditions, such as a relation reaching a specific state, a file changing, certain config values being set, etc. More

importantly, you can react to not just individual conditions, but meaningful combinations of conditions that can span multiple hook invocations, in a natural way.

### ***@hook 'install'***

1. Check whether the arch is x86 or S390x or PPCLE – If none, exit with proper message to the user using juju status message.
2. If X86 or S390x or PPCLE install the required packages for installing db2. Also create users for installing DB2. db2inst1 is the default username created for installing DB2. Once installed db2inst1 will be the default instance name.

### ***@when 'license.accepted'***

### ***@when\_not 'db.installed'***

1. Check if db2 is already installed. If true, do nothing
2. If db2 is not installed , packages needs to be downloaded from the SFTP server using hostname, package dir, username, password which are configured in config.yaml file. If any of these values are missing, user is alerted and it will not go through installation. If no package name is selected by user, based on the architecture a default value is selected for the package name.
3. Check whether user has selected right package name from the package name string. If he selects wrong package name, alert the user.
4. If already db2 is installed with a different package name, and if the user has configured a different name or if its first time installation for db2, download the package. Is an old package file is present, delete it
5. After the download is completed, the package integrity is checked using the provided checksum. The checksum is a configurable value and differs as per the package name. If no checksum value is provided, installation will not proceed further and a message is thrown to the user.
6. DB2 is then installed using the response file provided along with the charm.

7. Some initial configurations to set and bring up DB2 is done
8. Now DB2 is installed and the user can log in using db2inst1 user.
9. Other configurable items like dftd path can be configured by the user.  
By default this path is set to /tmp.

***@when 'db.installed'***

***@when not 'license.accepted'***

If db2\_license\_accepted is false, uninstall DB2 if already installed. Else throw a message that DB2 is not installed.

1. As part of uninstalling DB2, all databases in all DB instances are first removed.
2. Then Db2 is stopped using db2stop command.
3. DAS is dropped followed by removing all active instances.
4. Finally db2 uninstall command is called.

***@when 'db.available'***

1. Receive the SSH key from consumer charm and update the AUTH\_KEY\_FILE .
2. This hook will then create a new db username, password , instance and db for the consumer charm and would expose hostname, port number, db username, password,db2 instance name and db to any charm which connects with DB2. The DB name is created based on the service name if nothing is set by the consumer. If the consumer charm sets the DB name as a string of DBs separated with comma, (eg : dbnames="APPCNTR,WRKLGHT") the DB2 charm will create and expose these DB names.

***@hook 'start'***

This hook will check if the db2 is already started for the db2inst1 user . If no, it is started.

### **@hook 'stop'**

This hook will check whether db2 is running and stop it for db2inst1 instances.

### **@when not 'db.available'**

### **@when not 'db.connected'**

When relation between the consumer charm is broken, db2stop is called for that user.

### **Note:**

*The instance as well as the Dbs are not deleted . If instance is deleted, then db2cfexp command has to be used to create a config file which can later be imported using db2cfimp command. But if its deleted and the same charm tries to connect to DB2 it will cause problem. Hence retaining the instance name and deleting it only when Db2 is uninstalled.*

## **Post Deployment**

1. Once its deployed, the user can login using the <db2 user name> created and start using db2. User can run any of the db2 commands once logged in using this credentials.

*su - <db2usr1>*

*db2*

*db2> list db directory*

2. The same charm can be extended to Db2 Advanced Enterprise Server Edition or DB2 Advanced Workgroup server edition by accepting the appropriate license and doing some configuration settings as mentioned in Readme. This is not added as a configurable item in config.yaml because once it applies the license for BLU acceleration we cannot switch it back.