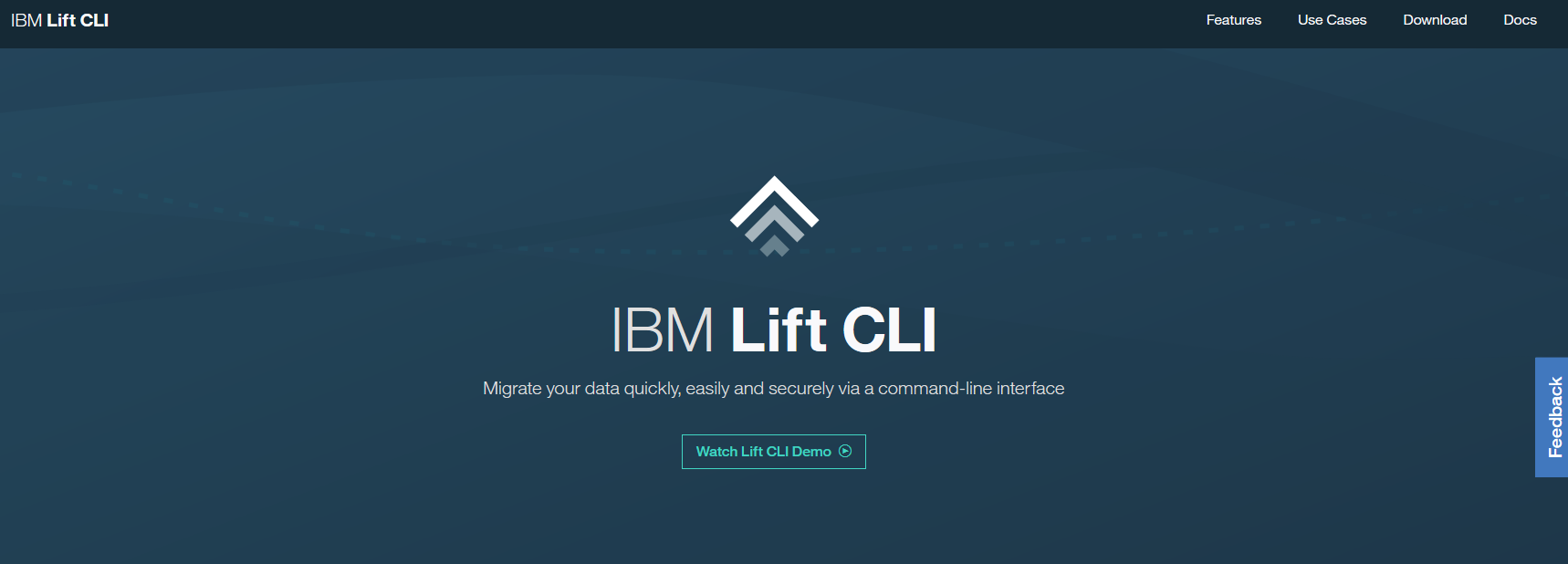
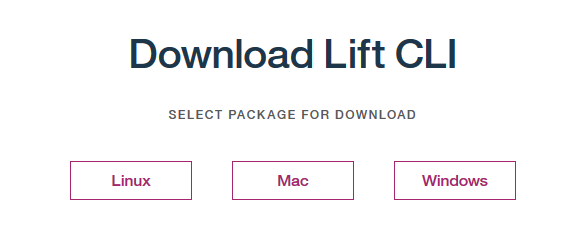
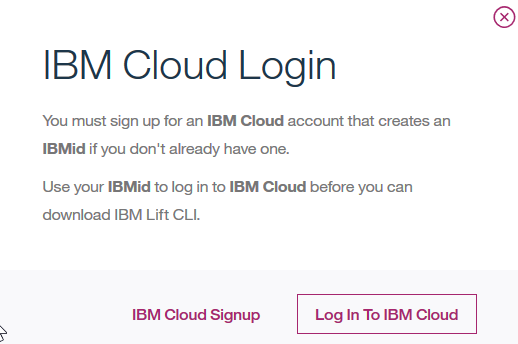
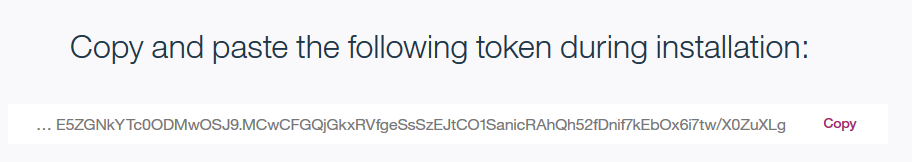
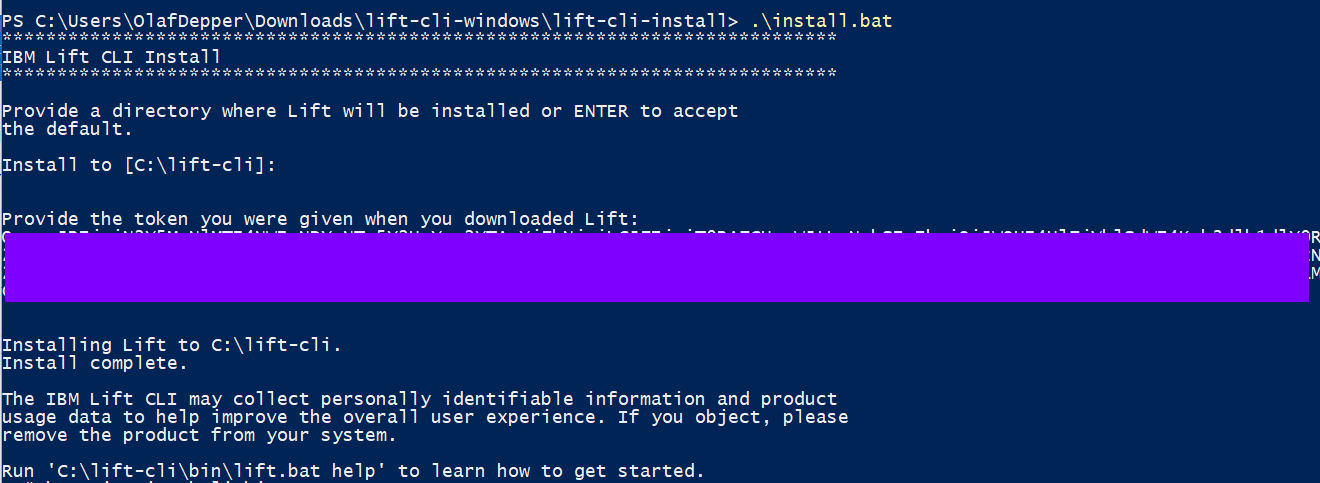
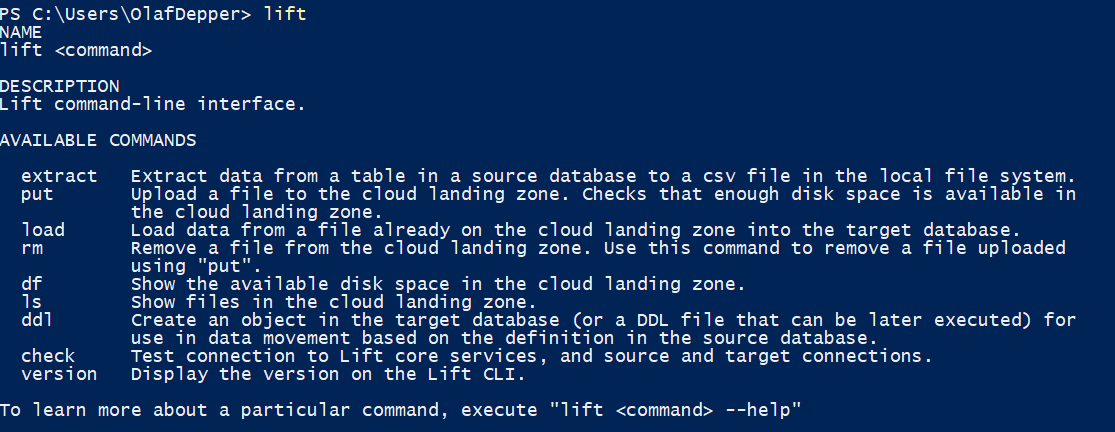
* Learn how to install IBM Lift CLI
* Configure IBM Lift CLI to connect to a cloud database
* Use IBM Lift CLI to migrate data from on-premises to the cloud

### Download and install IBM Lift CLI

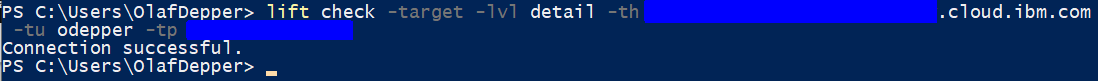
This task will guide you through the download and installation process of IBM Lift CLI. We will install the tool on a Windows 10 system. If you are installing Lift CLI on another platform, the process will be very similar.

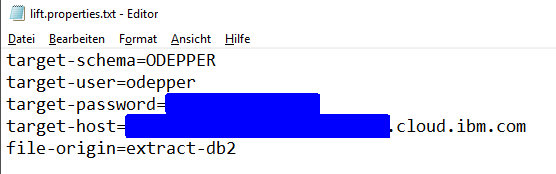
This tutorial assumes that you are familiar with using the command line/ terminal of your operating system. We will be using Windows PowerShell for our examples.

1. Navigate your browser to <https://www.lift-cli.cloud.ibm.com/> and click **Download** in the top-right corner of the page.  
   
2. Select the package for the operating system where you want to install IBM Lift CLI: **Linux, Windows, or Mac**  
   
3. You will need an IBM Cloud account to be able to download IBM Lift CLI. If you already have an account, click **Log In To IBM Cloud** and accept to the terms of use.  
   
4. Once the download has started, you will be directed to a page containing the activation token for your IBM Lift CLI environment. Make sure to **copy**this code. You will need to enter it during the IBM Lift CLI installation.  
   
5. Locate the downloaded file on your machine. Open the **terminal**application of your choice (we will use Windows PowerShell in the following examples). Unzip the downloaded file with **unzip <file>**. This will extract the content of the archive into a new subdirectory <extract-directory>.
6. Start the installation of IBM Lift CLI by running **<extract-directory>/install[.bat] <target-directory>**. You have to enter the token you received during the download:  
   
7. You should now be able to execute **lift**. The command is located in the bin subdirectory of the installation directory. You can also add the path for the lift executable to the PATH environment variable of your user, if you'd like to the ability to execute lift from any directory.  
   

### Test the Lift CLI connection to Db2 Warehouse on Cloud

In this task, we test the connection between the local Lift CLI installation and a Db2 Warehouse on Cloud system. We will use the check command option of Lift for this purpose.

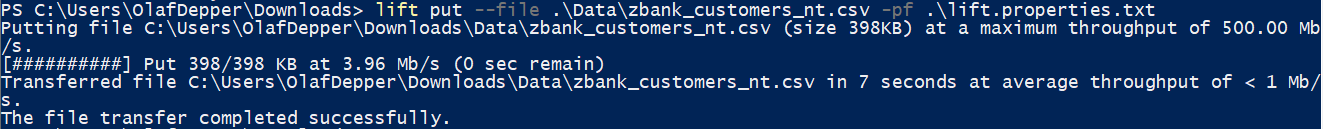
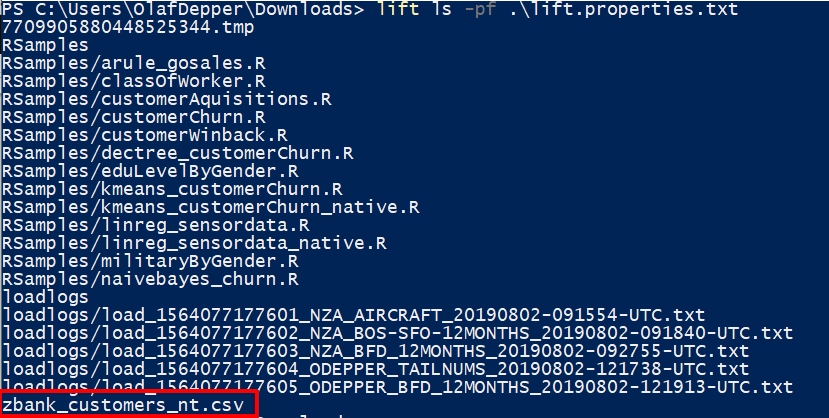
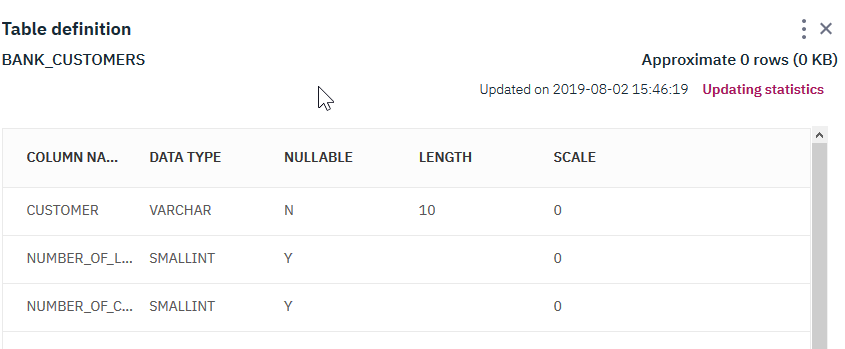
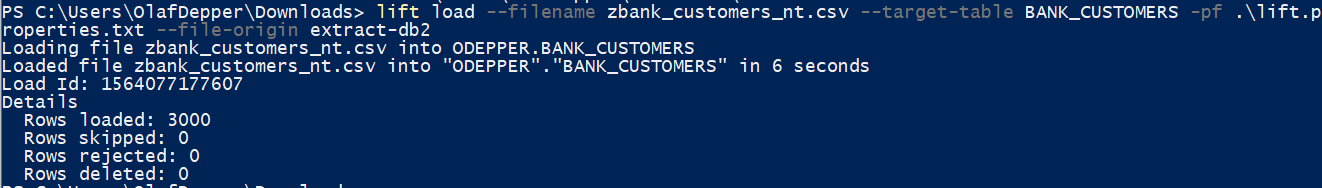
1. If you followed the steps from the prior task, you should now be able to execute some connection tests. In the first step, execute **lift check -core** to see whether IBM Lift CLI is able to connect to the IBM Lift CLI services (to check, for example: for Lift CLI upgrades).  
   image-20190802165415-4
2. In the next step, we will try to connect to our Db2 Warehouse on Cloud system. To provide parameters for IBM Lift CLI there are two options: You can either enter the necessary information every time you execute Lift CLI. Or you put information into a properties file and reference this file as part of the Lift command options. Let's first start with providing the connection information as part of the Lift command.  
   Execute the following command: **lift check -target -lvl detail -th <target host> -tu <Db2 Warehouse on Cloud-User> -tp <password>**  
     
   The command should return successful as in the above example.
3. As you have seen, we provided information like the target host as part of the Lift command. However, it will be much easier to provide information that you will need repeatedly into a properties file. You will then point the Lift command to this file. Alternatively, you could set environment variables on Linux and Mac with the required information.  
   Create a new properties file by using notepad (or any other editor). **notepad lift.properties.txt**. In this example, we will enter the following information into the file
4. target-schema=<IN CAPITAL LETTERS: Database schema in Db2 Warehouse on Cloud>
5. target-user=<User in Db2 Warehouse on Cloud>
6. target-password=<password>
7. target-host=<Database host of your Db2 Warehouse on Cloud system>

For example, the file we will use for the next steps:  


1. You can now - instead of providing all the command options - point to the properties file: **lift check -target [-lvl detail] --properties-file <file>**  
   image-20190802172300-8  
   As you can see in this example, Lift was able to connect to the target, using the information from the provided properties file. Instead of using --properties-file you can also use the shortened option -pf.

### Load data into IBM Db2 Warehouse on Cloud

In this task you will use IBM Lift CLI to upload a CSV file to your IBM Db2 Warehouse on Cloud System. You will then create an empty table in the database. With Lift CLI, you will then call Db2 LOAD utility to load the contents of the CSV file into the database table.

1. You will now load a CSV file into an IBM Db2 Warehouse on Cloud system. Download the sample **zbank\_customers.csv** file [here](https://ibm.box.com/s/0ktm6ghdfo2cizc2dri3strjkl9v1pit) (the file is stored on an external site, hosted by Box) and save it to your local computer.
2. Once the download has completed successfully, use Lift CLI to copy it to the Cloud landing area of your Db2 Warehouse on Cloud system. Again, we will be using the properties files rather than providing target information as part of the command itself: **lift put --file <path/file> -pf <properties-file>**  
   
3. You can use the ls command option of Lift CLI to show the files that are available in the cloud landing zone:**lift ls -pf <properties-file>**.  
     
   In this example, you will see a couple of files are already stored in the landing zone of the Db2 Warehouse on Cloud system. The CSV file is listed at the bottom.  
   You can also use the df command option to list the available space in the landing zone: **lift df -pf .\lift.properties.txt**  
   image-20190805120122-1
4. You can now load the data into a database table. Since we currently do not have a BANK\_CUSTOMERS table in the database, we will first have to create an empty table. Log on to your Db2 Warehouse on Cloud system. Go to the '**RUN SQL**' screen and copy the following statements into the SQL editor. Then click **Run all**to create the table with a primary key constraint.
5. CREATE TABLE BANK\_CUSTOMERS (
6. CUSTOMER VARCHAR(10) NOT NULL,
7. NUMBER\_OF\_LATE\_PAYMENTS SMALLINT,
8. NUMBER\_OF\_CREDIT\_APPLICATIONS SMALLINT,
9. DECLARED\_BANKRUPTCY\_IN\_PAST\_7\_YRS VARCHAR(3),
10. AVERAGE\_MONTHLY\_FEE SMALLINT,
11. CREDIT\_SCORE SMALLINT,
12. SATISFACTION SMALLINT,
13. DEFAULT\_DATA SMALLINT,
14. AVERAGE\_CASH\_BALANCE VARCHAR(3),
15. AGE SMALLINT,
16. AGE\_RANGE VARCHAR(5),
17. GENDER VARCHAR(6),
18. NUMBER\_OF\_PRODUCTS SMALLINT,
19. CUST\_ACQUISITION\_YEAR SMALLINT,
20. NO\_OF\_UNIQUE\_PRODUCTS SMALLINT,
21. NO\_OF\_UNIQUE\_PRODUCTS\_GROUPED VARCHAR(9),
22. EQUITIES\_BALANCE INTEGER,
23. INITIAL\_MORTGAGE\_AMT VARCHAR(14),
24. ADDRESS\_CHANGES SMALLINT,
25. HOUSEHOLD SMALLINT,
26. CLIENTS\_IN\_HOUSEHOLD SMALLINT,
27. AVERAGE\_CREDIT\_CARD\_BALANCE INTEGER,
28. CUSTOMER\_TYPE VARCHAR(14),
29. HOME\_BRANCH\_CITY VARCHAR(27),
30. HOME\_BRANCH\_STATE VARCHAR(14),
31. SALESPERSON VARCHAR(4),
32. CHURN VARCHAR(3),
33. COUNT SMALLINT,
34. BANKID VARCHAR(1)
35. )
36. ;
37. ALTER TABLE BANK\_CUSTOMERS
38. ADD CONSTRAINT CUSTOMER\_PK
39. PRIMARY KEY (CUSTOMER)
40. ;
41. You can double-check in the console by calling **Explore -> Tables** that the table has been created and does not contain any rows.  
    
42. Go back to your command line/ shell. You will now use Lift CLI to load the CSV file from the landing zone into the database table BANK\_CUSTOMERS. Run the following command:  
    **lift load --filename zbank\_customers\_nt.csv --target-table BANK\_CUSTOMERS -pf .\lift.properties.txt --file-origin extract-db2**. The parameter --file-origin has to be provided. We will use extract-db2 in our example for the origin. For the target table we provide the table we just created, using the --target-table option.  
    
43. Once the Lift CLI command has finished loading data, you can go back to the console of your Db2 Warehouse on Cloud system. In **Explore -> Tables** you will see that the BANK\_CUSTOMERS table now contains 3000 rows (you might need to click **Refresh**).  
    