Consuming cloud services

After you complete this unit, you should understand:

- How to use cloud services
- How to auto-wire cloud services
- How to consume cloud services such as:
  - SQL Database
  - ElasticMQ
  - Cloudant NoSQL Database
  - Data Cache
  - MongoDB
  - Operational services:
    - Session Cache
    - Auto-Scaling
    - Monitoring and Analytics
    - Log Analysis

Services are easy to consume

"Java EE standard" resources

SQL Database

BLU Data Warehouse

Elastic MQ

Modern resources

Cloudant

Data Cache

Mongo DB

Operational services

Session Cache

Auto Scaling

Monitoring and Analytics

Log Analysis

How to use cloud services

- Create and bind a service
  - Using Command line
    - *cf marketplace* to see available services
    - *cf create-service* to create a service instance
    - *cf bind-service* to bind the service instance to your application
    - *cf restart*, or *cf push* to restage your application
  - Using web portal (bluemix.net)
- Read service connection and credentials
  - Use VCAP_SERVICES environment variable


```
{
   "SQLDB-1.0": [
      {
         "name": "SQLDB-myDB",
         "credentials": {
            "hostname": "75.126.155.139",
            "host": "75.126.155.139",
            "port": 50000,
            "username": "u123456",
            "password": "CasDQ5v72u",
            "db": "I_012345",
            "jdbcurl": "jdbc:db2://75.126.155.139:50000/I_012345",
            "uri": "db2://u123456:CasDQ5v72u@75.126.155.139:50000/I_012345"
         }
      }
   ]
}
```

How to auto-wire cloud services

```
Import javax.sql.DataSource;

public class MyServlet extends HTTPServlet
{
    @Resource(name="jdbc/somedb")
    private DataSource myDataSource;
    ... or ....
    InitialContext ctx = new InitialContext();
    DataSource ds = (DataSource) ctx.lookup("jdbc/somedb");
}
```

1

1. Developer uses standard methods to access Data Source (JNDI lookup or injection).
2. Developer provides standard JEE resource references to declare dependency on external resources.
3. Developer binds database services to the application.

web.xml

```
...
<resource-ref>
   <res-ref-name>jdbc/somedb</res-ref-name>
   <res-type>javax.sql.DataSource</res-type>
   <res-auth>Container</res-auth>
   <res-sharing-scope>Unshareable</res-sharing-scope>
</resource-ref>
...
```

2

Ibm-web-bnd.xml

```
...
<resource-ref name=jdbc/somedb bindingname=jdbc/mydb>
```

- If a single database service is bound, then any/all bindings are bound to it.
- If multiple database services are bound, JNDI

How auto-wiring works

- Looks for a resource using the JNDI name, for example, java:comp/env/jdbc/myDB.
- Binds the service with the name `myDB`, which is mapped to the JNDI name `jdbc/myDB`.
- The bound service is returned as a result of the lookup.

How to consume cloud services

- Use the VCAP_SERVICES variable
- Use Java EE

Connect to the SQL database: Use VCAP_SERVICES

```
import com.ibm.nosql.json.api.*;
import com.ibm.nosql.json.util.*;

...
   String VCAP_SERVICES = System.getenv("VCAP_SERVICES");
   if (VCAP_SERVICES != null) {
      BasicDBObject obj =
                    (BasicDBObject) JSON.parse(VCAP_SERVICES);
      String thekey = null;
      Set<String> keys = obj.keySet();
      for (String eachkey : keys)
         if (eachkey.contains("SQLDB"))
            thekey = eachkey;
      BasicDBList list = (BasicDBList) obj.get(thekey);
      obj = (BasicDBObject) list.get("0");
      obj = (BasicDBObject) obj.get("credentials");
      databaseHost = (String) obj.get("host");
      databaseName = (String) obj.get("db");
      port = (String) obj.get("port").toString();
      user = (String) obj.get("username");
      password = (String) obj.get("password");
      url = (String) obj.get("jdbcurl");
      // Use the jdbcurl or construct your own
      databaseUrl = "jdbc:db2://" + databaseHost + ":" +
                                    port + "/" + databaseName;
```

Connect to the SQL database: Use Java EE

```
public  class TestServlet extends HttpServlet
{
    @Resource (name = "jdbc/mydb")
    private DataSource db;
    ...
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        // Alternatively use InitialContext lookup
        DataSource lookup = (DataSource) new InitialContext().lookup("jdbc/mydb");
        ...
}
```

"mydb" is the name of service instance you create in Bluemix.

That's it! All familiar code, no changes are required to make it work in the cloud!

- No need for a server.xml file
- No need to read VCAP_SERVICES

Connect to ElasticMQ service

Develop responsive, scalable applications with a fully managed messaging provider in the cloud.

```
public  class TestServlet extends HttpServlet
{
    @Resource (name = "jms/emq")
    private ConnectionFactory cf;

    ...
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        // Alternatively use InitialContext lookup
        ConnectionFactory lookup =
              (ConnectionFactory) new InitialContext().lookup(“jms/myemq");
        ...
}
```

More services accessible the same way

Cloudant

The Cloudant distributed database as a service (DBaaS) allows developers who build fast-growing web and mobile apps to focus on building and improving their products, instead of worrying about scaling and managing databases on their own.

DataCache

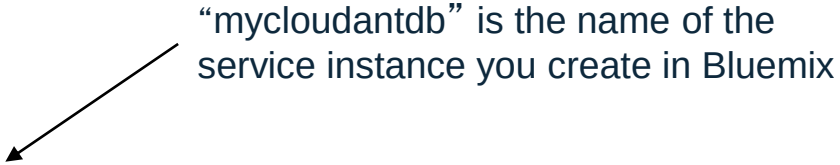
Improve the performance and user experience of web apps by retrieving information from fast, managed, in-memory caches, instead of relying entirely on slower disk-based databases.

mongodb

MongoDB is an open source document database and the leading NoSQL database that is owned by MongoDB, Inc.

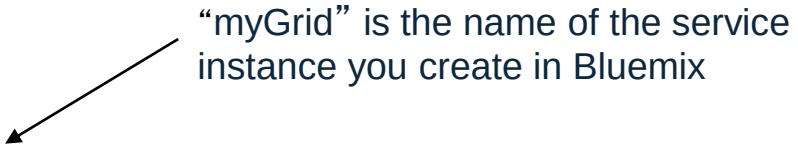
11

Connect to Cloudant



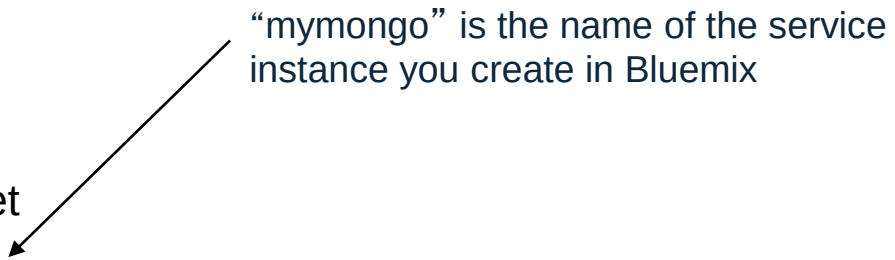

```
public  class TestServlet extends HttpServlet
{
    @Resource (name = “cloudant/mycloudantdb")
    private CouchDbInstasnce db;
    ...
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        // Alternatively use InitialContext lookup
        CouchDbInstance db = (CouchDbInstance) new
        InitialContext().lookup("java:comp/env/cloudant/mycloudantdb");

        CouchDbConnector dbc = _db.createConnector(DATABASE, true);
        CouchDocument dbentry = new CouchDocument();
        dbentry.setContent("testEntry");
        dbc.create(dbentry);
}
```

Connect to Data Cache




```
public  class TestServlet extends HttpServlet
{
    @Resource (name = “wxs/myGrid")
    private ObjectGrid og;
    ...
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        // Alternatively use InitialContext lookup
        ObjectGrid og = (ObjectGrid) new
        InitialContext().lookup(“wxs/myGrid");
        ...
}
```

Connect to MongoDB




```
import com.mongodb.DB;

public  class TestServlet extends HttpServlet
{
    @Resource (name = “mongo/mymongo")
    private DB db;
    ...
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        // Alternatively use InitialContext lookup
        db = (DB) new
        InitialContext().lookup("java:comp/env/mongo/mymongo");
        ...
}
```

Recap

- In this session, you learned how to use cloud services and how to auto-wire cloud services
- You also learned how to consume cloud services such as:
  - SQL Database
  - ElasticMQ
  - Cloudant NoSQL Database
  - Data Cache
  - MongoDB
  - Operational services such as Session Cache and Auto-Scaling

Related links

- IBM Cloud
  - http://www.ibm.com/cloud-computing/us/en/
- IBM Cloud: Infrastructure
  - http://www.ibm.com/cloud-computing/us/en/iaas.html
- IBM Cloud: Platform
  - http://www.ibm.com/cloud-computing/us/en/paas.html
- IBM Cloud: Built on Cloud
  - http://www.ibm.com/cloud-computing/us/en/saas.html
- IBM's open cloud architecture
  - http://www.ibm.com/developerworks/cloud/library/cl-open-architecture/
- IBM Bluemix
  - https://www.bluemix.net

Related links, continued

- IBM Bluemix: Virtual Machines
  - https://www.ng.bluemix.net/docs/virtualmachines/vm_index.html
- IBM Containers
  - https://www.ng.bluemix.net/docs/containers/container_index.html
- Docker
  - https://www.docker.com/
- Cloud Foundry
  - https://www.cloudfoundry.org/