The New Tools and Toys of Informix

Thomas Beebe

tom@advancedatatools.com

New Technology For Informix

- New Key Features of 12
- NoSQL
- Mongo Support
- REST Listener
- Node-Red
- Spark Accessibility
- IoT Data Visualization

New Features of V12.10

- New PSM storage manager
- Multitenancy
- Improved Timeseries Features
- SPL includes 'case' statement
- Grid Queries
- Bluemix Integration

NoSQL

- Unstructured data
- Often stored in JSON/BSON formats
  - **Previously XML was the preferred method**
- Key:Value pairs of data
- Supports many different types of data including arrays even nested ones
- It is a generic term so there are several types

NoSQL Types

- Key:Value pair databases
  - **Simple**
  - **Just stores lists of key/values for each element**
- Graph Stores
  - **Store related data for graphing networks**
- Wide-Columnar Databases
  - **Stores nested 'super' elements**
  - **Each can have multiple rows per super element**
- Document Databases
  - **Stores full documents, can be key/value pairs, arrays, nested arrays, documents, other data**

Document Database	Graph Databases
Couchbase MarkLogic mongoDB	Neo4j InfiniteGraph The Distributed Graph Database
Wide Column Stores	**Key-Value Databases**
redis amazon DynamoDB AEROSPIKE riak	accumulo HYPERTABLE INC Cassandra APACHE HBASE Amazon SimpleDB

@cloudtxt http://www.aryannava.com

JSON – JavaScript Object Notation

```
{
    "id": "0001",
    "type": "donut",
    "name": "Cake",
    "ppu": 0.55,
    "batters":
        {
            "batter":
                [
                    { "id": "1001", "type": "Regular" },
                    { "id": "1002", "type": "Chocolate" },
                    { "id": "1003", "type": "Blueberry" },
                    { "id": "1004", "type": "Devil's Food" }
                ]
        },
    "topping":
        [
            { "id": "5001", "type": "None" },
            { "id": "5002", "type": "Glazed" },
            { "id": "5005", "type": "Sugar" },
            { "id": "5007", "type": "Powdered Sugar" },
            { "id": "5006", "type": "Chocolate with Sprinkles" },
            { "id": "5003", "type": "Chocolate" },
            { "id": "5004", "type": "Maple" }
        ]
}
```

NoSQL Advantages

- Fast application development
  - **No waiting on schema changes**
- Easy to scale horizontally
- Easy to handle multiple versions
- Handle very different data elements in one place
- Fast inserts
- Can handle delayed inserts if using a middle layer
- Direct inserts without having to go through a database layer

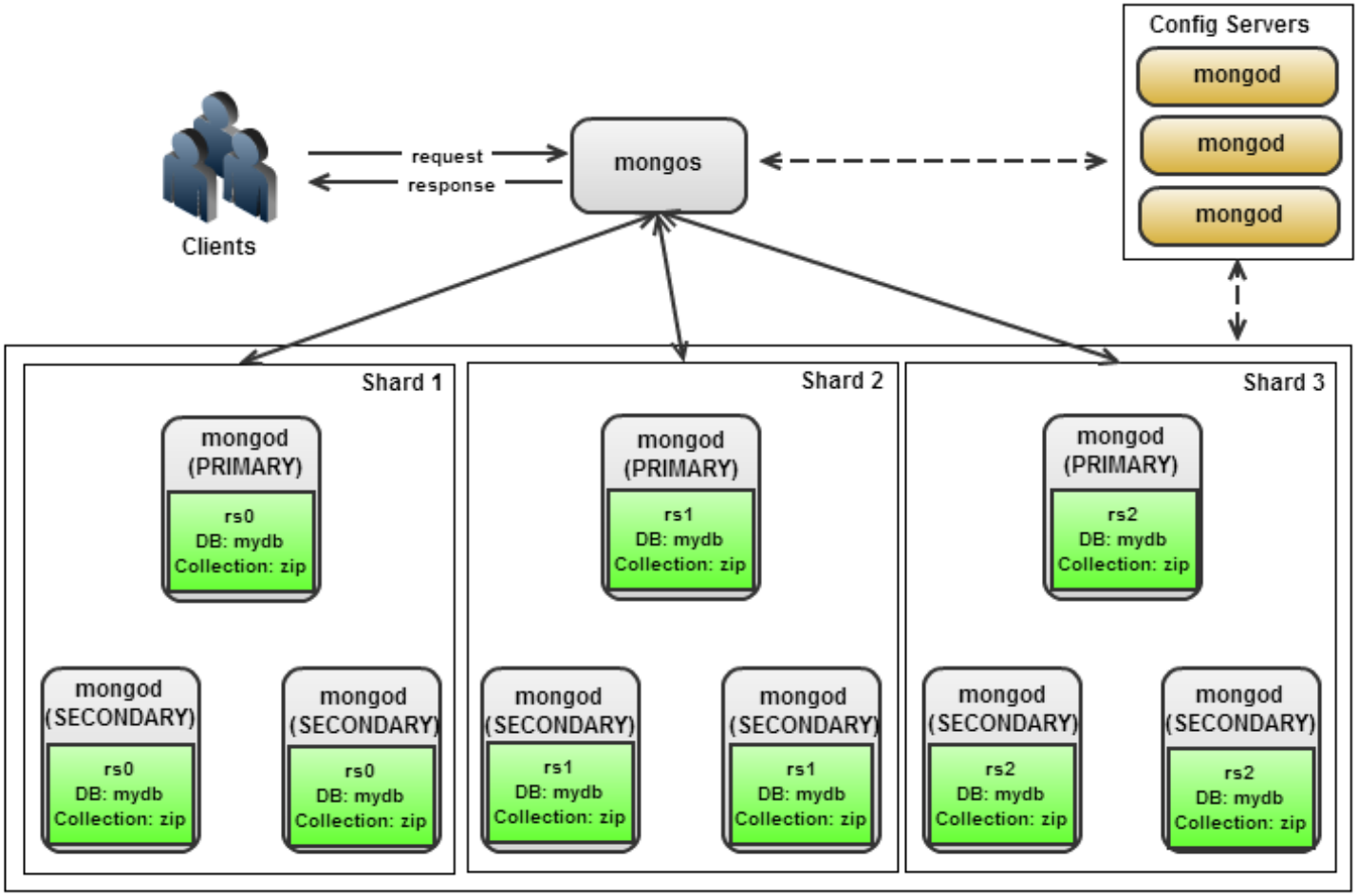
NoSQL Disadvantages

- Can be slower to update
- Lack of constraints
- Easier to get bad data in a database
- Hard to connect to linking data
- Elements such as default values need to be added by the application
- Need to trust your developers to get it right

Sharding

- Horizontal scaling
- Splitting data on to different nodes
- Sharded queries read the associated nodes in parallel
- Allows for very high performance and distributed queries without large boxes

Sharding



Neil Lunn - http://dba.stackexchange.com/questions/82551/data-distribution-in-mongos-with-shards-or-replica-sets

MongoDB

- Document Store style NoSQL database
- Open source
- Indexes (limited)
- Horizontal Replication
- Load Balancing
- Extensive querying syntax options

MongoDB Downsides

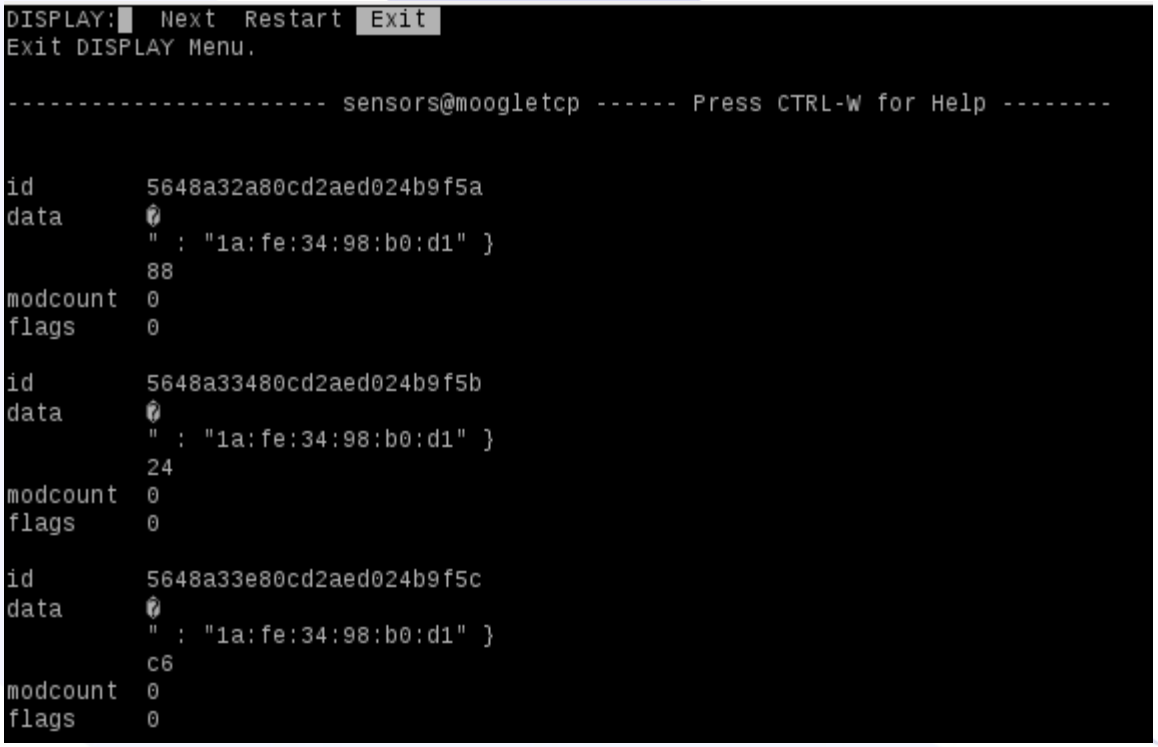
- No traditional constraints
- No transactions
- Limited types of indexes
- Did not scale vertically well
- Stability issues
- Reliability issues

Informix + Mongo

- In 12.10xc2 support for the Mongo API was introduced
- New additions to Informix:
  - **full support for the mongo command set**
  - **Added in the json and bson data types**
  - **Added the mongo wire listener**
  - **Data sharding**

Selecting Data

- Select id, data, modcount, flags from sensor_json



Selecting Data – Casing To JSON

- select id, data::json, modcount, flags from sensor_json

```
DISPLAY:  Next  Restart  Exit
Display next page of results.

---------------------- sensors@moogletcp ------ Press CTRL-W for Help --------


id              5648a32a80cd2aed024b9f5a
(expression)    {"topic":"sensors/temp","payload":"{ "data" : "32" , "sensor_id"
                : "1a:fe:34:98:b0:d1" }","qos":0,"retain":false,"_msgid":"8952797
                8.76ad88","_id":ObjectId("5648a32a80cd2aed024b9f5a")}
modcount        0
flags           0

id              5648a33480cd2aed024b9f5b
(expression)    {"topic":"sensors/temp","payload":"{ "data" : "20" , "sensor_id"
                : "1a:fe:34:98:b0:d1" }","qos":0,"retain":false,"_msgid":"5aacdb3
                8.a55324","_id":ObjectId("5648a33480cd2aed024b9f5b")}
modcount        0
flags           0

id              5648a33e80cd2aed024b9f5c
(expression)    {"topic":"sensors/temp","payload":"{ "data" : "25" , "sensor_id"
                : "1a:fe:34:98:b0:d1" }","qos":0,"retain":false,"_msgid":"34243a0
                1.cbdbc6","_id":ObjectId("5648a33e80cd2aed024b9f5c")}
modcount        0
flags           0
```

Setting Up

- Set up the Mongo listener
  - **Configure jsonlistener.properties**
  - **Start the listener**
    - Command Line
    - SQL
- Start storing NoSQL data

jsonlistener.properties

- Sample in $INFORMIXDIR/etc/ jsonlistener-example.properties
- Content:

```
listener.type=mongo              #Mongo or REST
listener.hostName=localhost      #Use hostname if you want remote connections
listener.port=26351              #Port for mongo listener to run on
security.sql.passthrough=false   # allow embedded SQL in the mongo commands
url=jdbc:informix-sqli://moogle:10098/sysmaster:INFORMIXSERVER=mooglenosql;USER=nosql;PASSWORD=n0sql!
```

- Sqlhosts:
  - **mooglenosql onsoctcp cubietruck 10098**

Starting Mongo Listener

```
CLASSPATH=${INFORMIXDIR}/lib/ifxjdbc.jar
  /usr/bin/java \
      -jar ${INFORMIXDIR}/bin/jsonListener.jar \
      -config ${INFORMIXDIR}/etc/jsonListener.properties \
      -logfile ${INFORMIXDIR}/tmp/jsonListener.log \
      -start &
```

Or:

```
EXECUTE FUNCTION task("start json listener",
    "/opt/informix/etc/jsonListener.properties");
```

Connecting To Informix

- Install Mongo Client pacakge
- mongo <server>:<port>/<database>
- mongo localhost:26351/sensors
  - **Will open the mongo shell**
  - **If it returns an error check the log from the startup script**

Using Mongo

- db.<table>.find()
  - **Equilivent to select * from <table>**
- db.sensor_json.find({_msgid:"d72c79f8.28d388"})
  - **{ "topic" : "sensors/temp", "_msgid" : "d72c79f8.28d388", "_id" : ObjectId("5648a37a80cd2aed024b9f62") }**
- Inserts automatically create the table as needed

Selecting Using Mongo

```
informix@cubie:~/nosql$ mongo localhost:26351/sensors
MongoDB shell version: 2.4.10
connecting to: localhost:26351/sensors
> db.sensor_json.find()
{ "topic" : "sensors/temp", "payload" : "{ \"data\" : \"32\" , \"sensor_id\" : \"1a:fe:34:98:b0:d1\" }", "qo
{ "topic" : "sensors/temp", "payload" : "{ \"data\" : \"20\" , \"sensor_id\" : \"1a:fe:34:98:b0:d1\" }", "qo
{ "topic" : "sensors/temp", "payload" : "{ \"data\" : \"25\" , \"sensor_id\" : \"1a:fe:34:98:b0:d1\" }", "qo
{ "topic" : "sensors/temp", "payload" : "{ \"data\" : \"32\" , \"sensor_id\" : \"1a:fe:34:98:b0:d1\" }", "qo
{ "topic" : "sensors/temp", "payload" : "{ \"data\" : \"21\" , \"sensor_id\" : \"1a:fe:34:98:b0:d1\" }", "qo
{ "topic" : "sensors/temp", "payload" : "{ \"data\" : \"35\" , \"sensor_id\" : \"1a:fe:34:98:b0:d1\" }", "qo
{ "topic" : "sensors/temp", "payload" : "{ \"data\" : \"33\" , \"sensor_id\" : \"1a:fe:34:98:b0:d1\" }", "qo
{ "topic" : "sensors/temp", "payload" : "{ \"data\" : \"21\" , \"sensor_id\" : \"1a:fe:34:98:b0:d1\" }", "qo
{ "topic" : "sensors/temp", "payload" : "{ \"data\" : \"38\" , \"sensor_id\" : \"1a:fe:34:98:b0:d1\" }", "qo
{ "topic" : "sensors/temp", "payload" : "{ \"data\" : \"34\" , \"sensor_id\" : \"1a:fe:34:98:b0:d1\" }", "qo
{ "topic" : "sensors/temp", "payload" : "{ \"data\" : \"21\" , \"sensor_id\" : \"1a:fe:34:98:b0:d1\" }", "qo
{ "topic" : "sensors/temp", "payload" : "{ \"data\" : \"35\" , \"sensor_id\" : \"1a:fe:34:98:b0:d1\" }", "qo
{ "topic" : "sensors/temp", "payload" : "{ \"data\" : \"25\" , \"sensor_id\" : \"1a:fe:34:98:b0:d1\" }", "qo
{ "topic" : "sensors/temp", "payload" : "{ \"data\" : \"25\" , \"sensor_id\" : \"1a:fe:34:98:b0:d1\" }", "qo
{ "topic" : "sensors/temp", "payload" : "{ \"data\" : \"28\" , \"sensor_id\" : \"1a:fe:34:98:b0:d1\" }", "qo
{ "topic" : "sensors/temp", "payload" : "{ \"data\" : \"33\" , \"sensor_id\" : \"1a:fe:34:98:b0:d1\" }", "qo
{ "topic" : "sensors/temp", "payload" : "{ \"data\" : \"25\" , \"sensor_id\" : \"1a:fe:34:98:b0:d1\" }", "qo
{ "topic" : "sensors/temp", "payload" : "{ \"data\" : \"22\" , \"sensor_id\" : \"1a:fe:34:98:b0:d1\" }", "qo
{ "topic" : "sensors/temp", "payload" : "{ \"data\" : \"21\" , \"sensor_id\" : \"1a:fe:34:98:b0:d1\" }", "qo
{ "topic" : "sensors/temp", "payload" : "{ \"data\" : \"21\" , \"sensor_id\" : \"1a:fe:34:98:b0:d1\" }", "qo
Type "it" for more
>
```

Mongo Commands

Mongo Command	Informix Statement
db.customer.insert({ name: “John", age: 65 })	INSERT INTO customer (name, age) VALUES (“John”,65)
db.customer.find()	SELECT * FROM customer
db.customer.find({age: { $gt:65 } })	SELECT * FROM customer WHERE age > 65
db.customer.drop()	DROP TABLE customer
db.customer.ensureIndex({ name : 1, age : -1 })	CREATE INDEX idx_1 on customer(name , age DESC)
db.customer.remove({age: { $gt:65 } })	DELETE FROM customer where age > 65
db.customer.update({ age: { $gt: 64 } }, { $set: { status: “Retire" } }, { multi: true })	UPDATE customer SET status = “Retire" WHERE age > 64

REST Listener

- Set up and started the same way as the Mongo listener
- Change listener.type to rest
- Sets up a listener that handles REST POST/GET commands such as from a web server or even from javascript
- Uses JSON to pass/receive data
- REST Syntax is used for Bluemix data transfer
- No drivers needed

REST Methods

- Get – Query
- Post – Insert
- Delete – Delete
- Put - Update

REST Examples

- GET /mydb/people?sort={age:1}&fields={_id:0,lastName:0}
- [{"firstName":"Sherry","age":31},

{"firstName":"John","age":31},

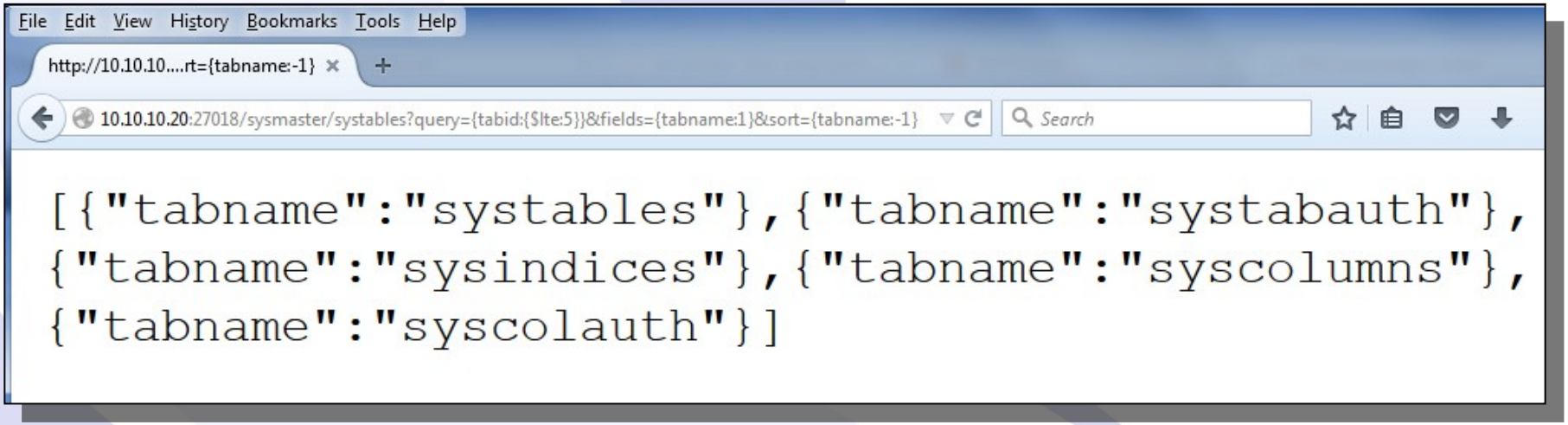
{"firstName":"Bob","age":47},

{"firstName":"Larry","age":49}]

REST

Example: Connect from a browser to an Informix database

http://10.10.10.20:27018/sysmaster/systables?query={tabid:{$lte:5}}&fields={tabname:1}&sort={tabname:-1}

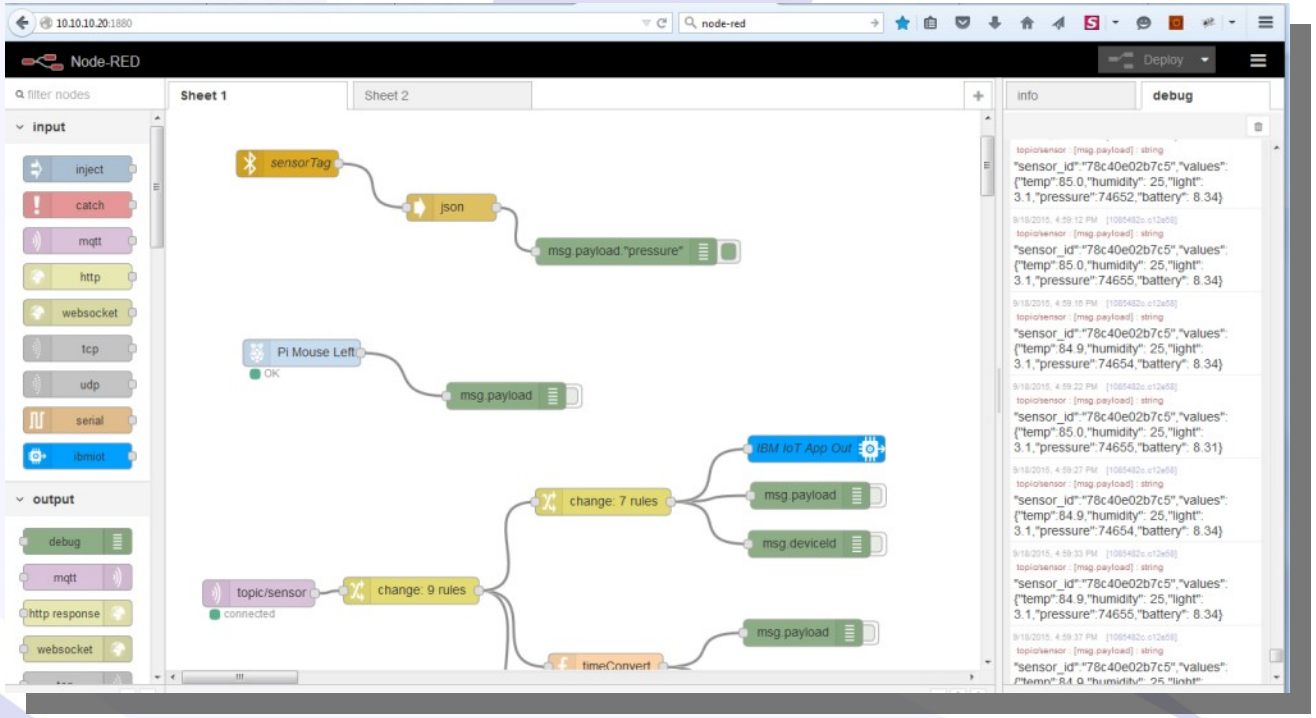


Node RED

- Free opensource tool based on node.js
- Visual tool, little to no programming needed
- Drag and drop elements to set up data flows
- Can handle many types of data through freely available modules
- Interfaces extensively with IBM services
  - **Streams**
  - **Watson**
  - **Bluemix**

Node-RED

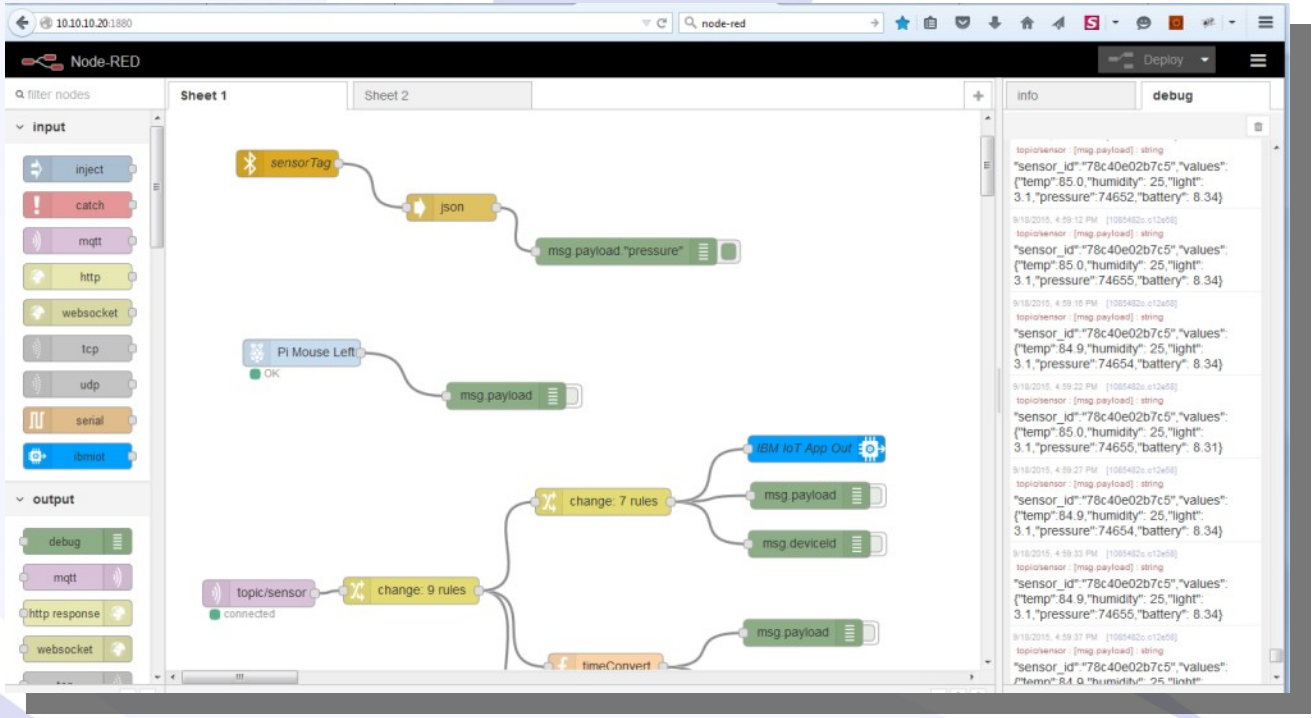
- Graphical tool – run in a browser
- Use to put together data flows and apply operations
- Additional “nodes” can be added to increase the functionality
- Custom “nodes” can be written



Advanced DataTools

Node-RED

- Graphical tool – run in a browser
- Use to put together data flows and apply operations
- Additional “nodes” can be added to increase the functionality
- Custom “nodes” can be written



Advanced DataTools

Node-RED Install

- sudo npm install -g node-red
- node-red
- No step 3

Node-RED Modules

- cd ~/.node-red
- npm install node-red-node-{filename}

- That is it.
- Another node.js utility called pm2 can be useful to control the service

MQTT

- MQ Telemetry Transport
- Publisher nodes connect and send data
- Subscribers see the data and make use of it
- Many available servers
  - **Including Websphere**

Links

- https://www.mongodb.com/nosql-explained
- https://adobe.github.io/Spry/samples/data_region/JSONDataSetSample.html
- **https://hub.docker.com/r/ibmcom/iotvisualization**
- **http://ibm.co/1N0XcKU**
  - **setting up the wire listener**
- **http://www-01.ibm.com/support/docview.wss?uid=swg27041825**
  - **Mongo and Informix command comparison**
- **http://ibm.co/1MSiSWz**
  - **REST Api Information**
- **http://nodered.org/docs/getting-started/running.html**
- **http://flows.nodered.org/**

Questions?

Send follow-up questions to
tom@advancedatatools.com

More Resources

- Webcasts covering Informix, ARM, Internet of Things:

  http://advancedatatools.com/Informix/Webcasts.html

- Compare Informix versions:

  http://www.ibm.com/developerworks/data/library/techarticle/dm-0801doe

- Docker Site:

  https://hub.docker.com/r/ibmcom/informix-innovator-c/

- Advanced DataToolsTraining:

  http://advancedatatools.com/Training/InformixTraining.html

Informix Support and Training from the Informix Champions!

Advanced DataTools is an Advanced Level IBM Informix Data Management Partner, and has been an authorized Informix partner since 1993. We have a long-term relationship with IBM, we have priority access to high-level support staff, technical information, and Beta programs. Our team has been working with Informix since its inception, and includes 8 Senior Informix Database Consultants, 4 IBM Champions, 2 IIUG Director's Award winners, and an IBM Gold Consultant.

- ***Informix Training***
- ***Informix Consulting***
- ***Informix Development***
- ***Informix Remote DBA Support Monitoring***
- ***Informix Performance Tuning***

Free Informix Performance Tuning Webcast replays at:
http://advancedatatools.com/Informix/Webcasts.html

Call: (800) 807-6732 x101 or Email: info@advancedatatools.com
Web: http://www.advancedatatools.com

Advanced DataTools

IBM Business Partner

Thank You

Thomas Beebe

Advanced DataTools Corporation

tom@advancedatatools.com

For more information:

http://www.advancedatatools.com