Node.js für Admins

Hands-On: jetzt wird's spannend!

Oliver Busse

We4IT GmbH

aveedo

http://oliverbusse.com

@zeromancer1972

IBM CHAMPION

Agenda

- Transportable Umgebung dank Docker
- Admin-Module (npm)
- Admin-Scripte (node)
- Domino pre v10
- Automation mit Node-RED
- Extra: Smart mit Homebridge

Transportable Umgebung dank Docker

- Vorteile
- Herausforderungen
- Dockerfile
- docker-compose

Vorteile

- schlanke Runtime gegenüber einer Vollinstallation des OS
- skalierbar
- einfaches Deployment
  - lokal vorbereiten, remote einrichten

Herausforderungen

- gewisse Lernkurve des Container-Prinzips
- neue Konfigurationselemente (Dockerfile, yml Files)
- Netzwerkzugriff (exposing)

Dockerfile

- Textdatei im proprietären Format, ähnlich Batchfile
- enthält Angaben zu:
  - Basis OS
  - Workdirs
  - Network
  - Commands
  - Comments
- wird von der docker Binary ausgeführt

Dockerfile - Beispiel


```
FROM node

RUN mkdir /src

RUN npm install nodemon -g

WORKDIR /src
ADD app/package.json /src/package.json
RUN npm install

ADD app/nodemon.json /src/nodemon.json

EXPOSE 3000

CMD npm start
```

Dockerfile - Ausführung

```
$ docker build .
```

Es gibt aber fast unendlich viele Parameter... Vertippen ist vorprogrammiert ;-)

Docker File Reference

docker-compose

- eigenes Modul in Docker
- nutzt docker-compose.yml als Konfiguration
- Textdatei im YAML Format
- besser lesbar
- schwieriger zu erstellen (Syntax, Linebreaks, Indents)
  - alles hat eine spezielle Aufgabe
- einfachere Bedienung als docker CLI
- nutzt allerdings auch das Dockerfile

docker-compose - Beispiel

```
web:
  build: .
  volumes:
    - "./app:/src/app"
  ports:
    - "3030:3000"
  net: "bridge"
```

Der Pfad . bezieht sich auf das aktuelle Verzeichnis, in dem dann das Dockerfile erwartet wird. Entscheidend ist hier die einfache Netzwerk-Konfiguration und das Port-Mapping für die Node.js App.

Admin-Module

- child_process / sys
- shelljs
- commander

child_process / sys

- erlaubt das Ausführen von Consolen-Befehlen (Linux, Windows) in einer Node.js app
- sys vereinfacht Ausgaben unter Linux
- VORSICHT!

child_process / sys - Beispiel


```
// http://nodejs.org/api.html#_child_processes
var sys = require('sys')
var exec = require('child_process').exec;
var child;
// executes `pwd`
child = exec("pwd", function (error, stdout, stderr) {
  sys.print('stdout: ' + stdout);
  sys.print('stderr: ' + stderr);
  if (error !== null) {
    console.log('exec error: ' + error);
  }
});
// or more concisely
var sys = require('sys')
var exec = require('child_process').exec;
function puts(error, stdout, stderr) { sys.puts(stdout) }
exec("ls -la", puts);
```

Quelle: https://dzone.com/articles/execute-unix-command-nodejs

shelljs

- Suchen und Ersetzen in Dateien
- Zugriff aufs Filesystem
- Extrahieren von Passagen aus Textdateien
- Bringt viele bekannte Unix-Funktionen mit
  - which
  - sed
  - grep
  - cp
  - mv
  - ...

shelljs - Beispiel (Seite 1)

```
var shell = require('shelljs');

if (!shell.which('git')) {
  shell.echo('Sorry, this script requires git');
  shell.exit(1);
}

// Copy files to release dir
shell.rm('-rf', 'out/Release');
shell.cp('-R', 'stuff/', 'out/Release');
```

... to be continued

shelljs - Beispiel (Seite 2)


```
// Replace macros in each .js file
shell.cd('lib');
shell.ls('*.js').forEach(function (file) {
  shell.sed('-i', 'BUILD_VERSION', 'v0.1.2', file);
  shell.sed('-i', /^.*REMOVE_THIS_LINE.*$/, '', file);
  shell.sed('-i', /.*REPLACE_LINE_WITH_MACRO.*\n/, shell.cat('macro.js
});
shell.cd('..');

// Run external tool synchronously
if (shell.exec('git commit -am "Auto-commit"').code !== 0) {
  shell.echo('Error: Git commit failed');
  shell.exit(1);
}
```

Quelle: https://github.com/shelljs/shelljs

commander

- Erlaubt die Übergabe von CLI Parameter an Node.js Scripte
- inkl. Hilfe

commander - Beispiel


```
// pizza script
var program = require('commander');

program
  .version('0.1.0')
  .option('-p, --peppers', 'Add peppers')
  .option('-P, --pineapple', 'Add pineapple')
  .option('-b, --bbq-sauce', 'Add bbq sauce')
  .option('-c, --cheese [type]',
        'Add the specified type of cheese [marble]', 'marble')
  .parse(process.argv);

console.log('you ordered a pizza with:');
if (program.peppers) console.log('  - peppers');
if (program.pineapple) console.log('  - pineapple');
if (program.bbqSauce) console.log('  - bbq');
console.log('  - %s cheese', program.cheese);
```

Quelle: https://www.npmjs.com/package/commander

commander - Ausführung

Angenommen, das Node.js Script heißt `pizza.js`

```
node pizza --cheese Gouda
```

Admin-Scripte

- Domino starten und stoppen
- Consolenbefehle ausführen

Domino starten und stoppen

```
// start
<dominoProgram>/server

// stop
<dominoProgram>/server -q
```

Domino starten und stoppen mit Node.js (1)

```
var shell = require('shelljs');
var express = require('express');
var app = express();

app.get('/', (request, response) => {
    response.send('Available commands: /start /stop');
});

app.get('/start', (request, response) => {
    shell.cd('/local/notesdata');
    shell.exec('/opt/ibm/domino/bin/server');
    shell.echo('Server started');
    response.send('Done.');
});
// continued on next page
```

Domino starten und stoppen mit Node.js (2)

```
// here we go
app.get('/stop', (request, response) => {
    shell.cd('/local/notesdata');
    shell.exec('/opt/ibm/domino/bin/server -q');
    shell.echo('Server stopped');
    response.send('Done.');
});

app.listen(3000, () => {
    console.log('server is listening on 3000');
});
```

Consolenbefehle ausführen

```
var shell = require('shelljs');

shell.cd('/local/notesdata');
shell.exec('/opt/ibm/domino/bin/server -c \"load compact -B\"');
```

Domino pre V10?

- Domino Apps mit Node.js nutzen
  - Windows only
  - experimental
  - https://github.com/nthjelme/nodejs-domino
- Domino Access Services (DAS)

Domino V10

- die Beta enthält noch kein Node.js :-(
- GA soll Node Stack enthalten
  - Domino als Node server
  - natives Node.js Modul zum Zugriff auf Domino API
- Domino Query Language (DQF) wird direkt unterstützt

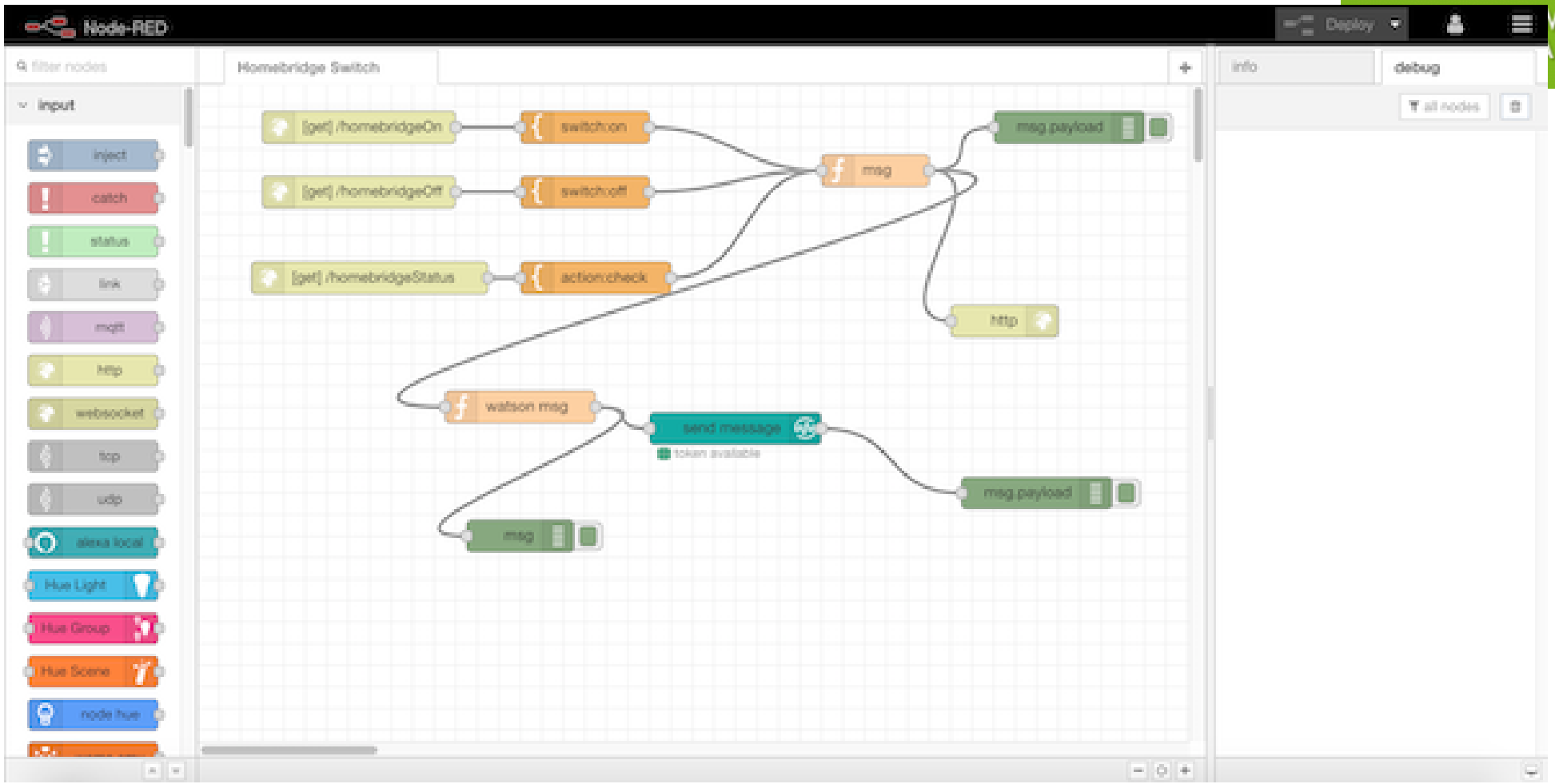
Automation mit Node-RED

- Scripte zeitgesteuert ausführen
- IoT Services nutzen

Was ist Node-RED?

- visueller, Low-Code Editor und Runtime
- Webserver, Gateway
- Programme sind Flows, keine Scripte
- unzählige Plugins für IoT, HTTP, MQTT, Smarthome, Alexa, Watson, ...
- eine Node.js Anwendung :-)

Was ist Node-RED?




```
npm install node-red -g --unsafe-perm
```

Empfehlung: mittels pm2 beim Booten starten

Extra: Smart mit Homebridge

Oder: wie starte ich den Domino Server mittels iOS neu?

- iOS ist im Business das beliebtere Mobile OS (im Gegensatz zu Android)
- Homekit ist Apples Smart Home Zentrale
- Homebridge ist die Schnittstelle zu Homekit
- Basiert auf Node.js
- Rezepte & Zutaten

Domino Maintenance - aber smart!

- Raspberry Pi mit
  - Node.js
  - Homebridge
  - Node-RED
  - Plugins
- Apple TV oder iPad als Schaltzentrale
- iOS Device (iPhone, iPad)

Dockerizing a Node.js web app

Writing Command Line Apps in Node.js

Sources