NODE-RED 101

IN A NUTSHELL

// AGENDA

> Motivation und Geschichte
> Installation
> Hello World
> Grundlagen
> Demo

WHOAMI

CHRISTIAN STANKOWIC

> Trainer und Techncial Leader Linux
  > RHEL, Satellite, Ansible
  > SLES, SUSE Manager, SaltStack
  > Git(Lab), Vagrant, Terraform,...

> Blogger (cstan.io) und Podcaster (FOCUS ON: Linux)

// MOTIVATION

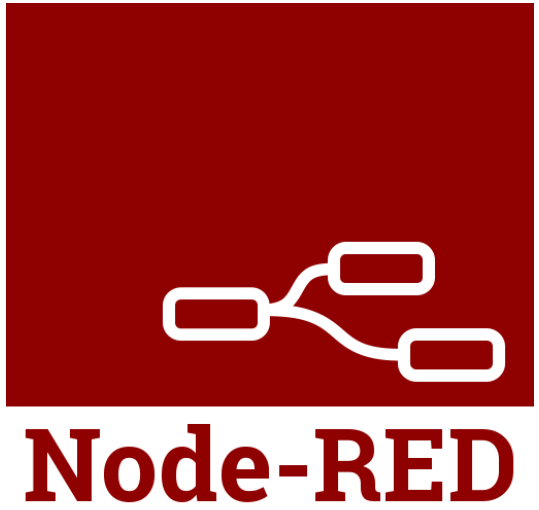
WAS IST NODE-RED?

> Grafisches **low-code** Entwicklungswerkzeug

> Entwickeln per Drag&Drop und JavaScript

> Ursprünglich für IoT-Zwecke entwickelt

> Ideal für schnelles **Prototyping**



GESCHICHTE UND PARADIGMA

> **2013** von *IBM Emerging Technologies* entwickelt

> **2016** von IBM an die JavaScript-Foundation übergeben*

> **2019** erscheint stabile Version 1.0

> **2021** erscheint Version 2.0, die nun Node.js 12.17+ nutzt

GESCHICHTE UND PARADIGMA

- **2013** von *IBM Emerging Technologies* entwickelt
- **2016** von IBM an die JavaScript-Foundation übergeben*
- **2019** erscheint stabile Version 1.0
- **2021** erscheint Version 2.0, die nun Node.js 12.17+ nutzt
- **Flow-basierte** Programmierung
  - 1969 von J. Paul Morrison bei IBM entwickeltes Paradigma
  - Überlegung: Anwendung ist ein Netz aus "*Black Box*"-Prozessen
  - Austausch zwischen Prozessen erfolgt über **Nachrichten**
  - Blöcke können so einfach und beliebig wiederverwendet werden

FEATURES UND BEGRIFFLICHKEITEN

> Integration in bestehende Systeme via **Standard-Protokolle**
  > tcp/udp, http/https, WebSocket, MQTT,...

> Definition/Anpassung von Nachrichten via **JavaScript**

> modular, große **Bibliothek** mit Modulen und vorgefertigten Inhalten*

FEATURES UND BEGRIFFLICHKEITEN

> Integration in bestehende Systeme via **Standard-Protokolle**
  > tcp/udp, http/https, WebSocket, MQTT,...

> Definition/Anpassung von Nachrichten via **JavaScript**

> modular, große **Bibliothek** mit Modulen und vorgefertigten Inhalten*

> In Node-RED werden Programmabläufe auch **Flows** genannt

> diese werden grafisch dargestellt und per Drag&Drop angepasst

> Funktionalitäten werden als Baustein (**Node**) definiert

* derzeit ~6.500 verschiedene Inhalte

// INSTALLATION

INSTALLATION

Node-RED kann auf mehrere Wege installiert werden:

> manuell via `npm`

> als Paket via Snapcraft

> als Container via Docker oder Podman

> Binärpakete für Raspberry Pi OS (Buster)

INSTALLATION

Node-RED kann auf mehrere Wege installiert werden:

> manuell via `npm`

> als Paket via Snapcraft

> als Container via Docker oder Podman

> Binärpakete für Raspberry Pi OS (Buster)

> Klare Empfehlung: **Container-Image** (`nodered/node-red`) nutzen

INSTALLATION

Installation via `npm`:

```
# npm install -g --unsafe-perm node-red
...
+ node-red@1.1.0
added 332 packages from 341 contributors in 18.494s
found 0 vulnerabilities
```

Installation via Snapcraft:

```
# snap install node-red
```

Achtung: Snap-Installation hat **keinen Zugriff** auf Git-Projekte

INSTALLATION

Beispielhaftes `docker-compose.yml` (`docker-compose up -d`):

```
version: "3"

services:
  node-red:
    container_name: nodered
    image: nodered/node-red:2.2.2
    ports:
      - "1880:1880/tcp"
    environment:
      TZ: 'Europe/Berlin'
    volumes:
      - data:/data
    restart: unless-stopped

volumes:
  data:
```

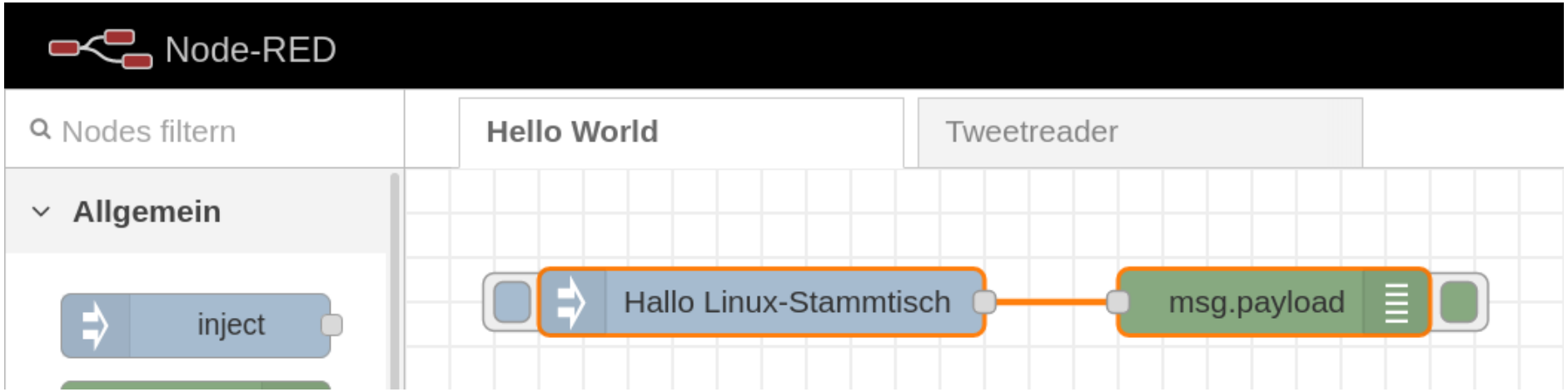
// HELLO WORLD

HELLO WORLD

- Ziel:
  - Definition einer Nachricht (**inject**)
  - Ausgabe als Debug-Nachricht

HELLO WORLD

- Ziel:
  - Definition einer Nachricht (**inject**)
  - Ausgabe als Debug-Nachricht
- Anforderungen:
  - 1x **inject**-Node
  - 1x **debug**-Node



// GRUNDLAGEN

GRUNDLAGEN

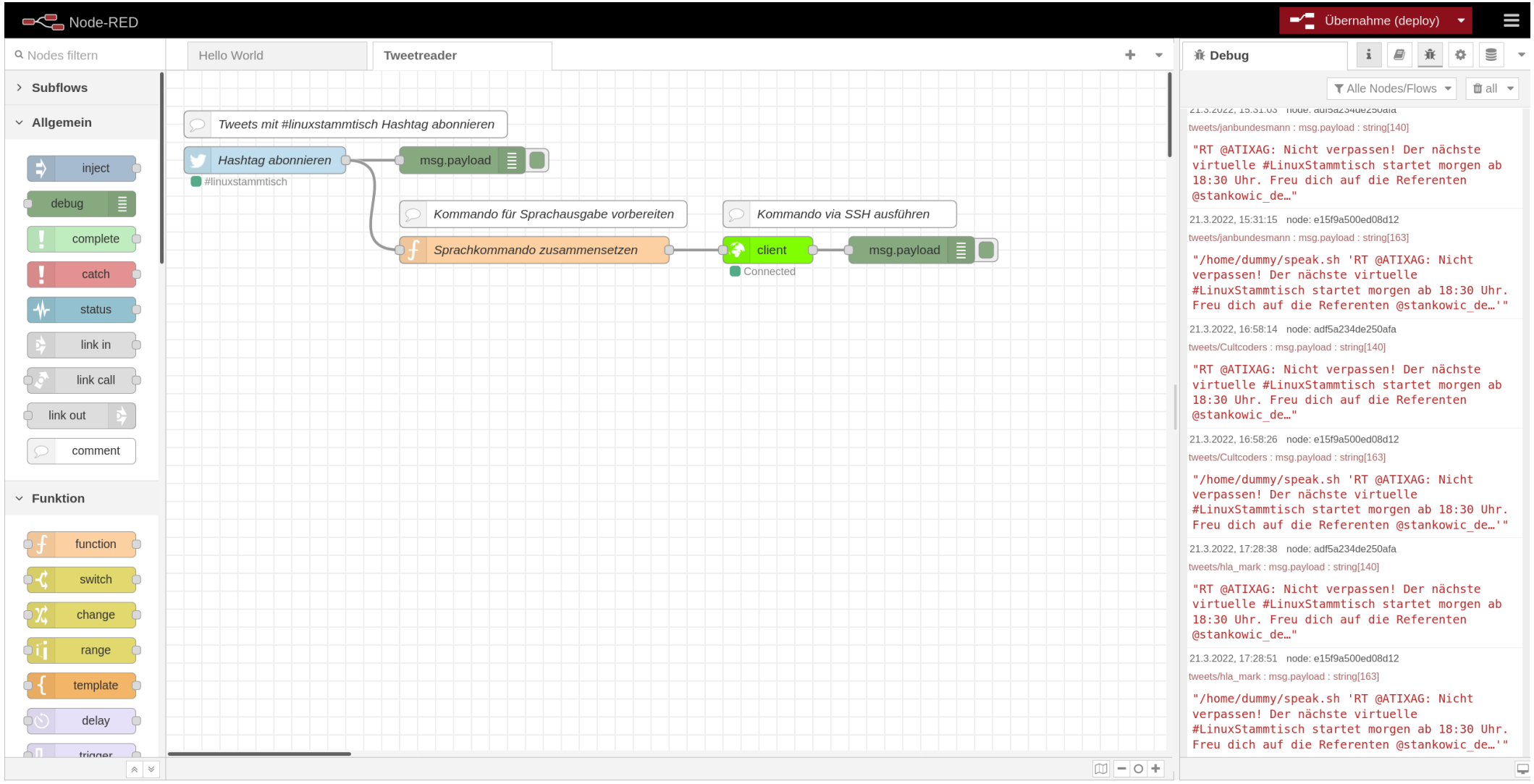
Die Node-RED GUI besteht im Wesentlichen aus:

> Palette mit Nodes (**links**)

> Seitenleiste mit Hilfe-, Debug- und Einstellungsbereich (**rechts**)

> Entwicklungsbereich (**mitte**)

Rechts oben im Hamburger-Menü finden sich weitere Einstellungsoptionen.


Node-RED
Übernahme (deploy)
Nodes filtern
Hello World
Tweetreader
Subflows
Allgemein
inject
debug
complete
catch
status
link in
link call
link out
comment
Funktion
function
switch
change
range
template
delay
Tweets mit #linuxstammtisch Hashtag abonnieren
Hashtag abonnieren
#linuxstammtisch
msg.payload
Kommando für Sprachausgabe vorbereiten
Sprachkommando zusammensetzen
Kommando via SSH ausführen
client
Connected
msg.payload
Debug
Alle Nodes/Flows
all
tweets/janbundesmann : msg.payload : string[140]
"RT @ATIXAG: Nicht verpassen! Der nächste virtuelle #LinuxStammtisch startet morgen ab 18:30 Uhr. Freu dich auf die Referenten @stankowic_de…"
21.3.2022, 15:31:15 node: e15f9a500ed08d12
tweets/janbundesmann : msg.payload : string[163]
"/home/dummy/speak.sh 'RT @ATIXAG: Nicht verpassen! Der nächste virtuelle #LinuxStammtisch startet morgen ab 18:30 Uhr. Freu dich auf die Referenten @stankowic_de…'"
21.3.2022, 16:58:14 node: adf5a234de250afa
tweets/Cultcoders : msg.payload : string[140]
"RT @ATIXAG: Nicht verpassen! Der nächste virtuelle #LinuxStammtisch startet morgen ab 18:30 Uhr. Freu dich auf die Referenten @stankowic_de…"
21.3.2022, 16:58:26 node: e15f9a500ed08d12
tweets/Cultcoders : msg.payload : string[163]
"/home/dummy/speak.sh 'RT @ATIXAG: Nicht verpassen! Der nächste virtuelle #LinuxStammtisch startet morgen ab 18:30 Uhr. Freu dich auf die Referenten @stankowic_de…'"
21.3.2022, 17:28:38 node: adf5a234de250afa
tweets/hla_mark : msg.payload : string[140]
"RT @ATIXAG: Nicht verpassen! Der nächste virtuelle #LinuxStammtisch startet morgen ab 18:30 Uhr. Freu dich auf die Referenten @stankowic_de…"
21.3.2022, 17:28:51 node: e15f9a500ed08d12
tweets/hla_mark : msg.payload : string[163]
"/home/dummy/speak.sh 'RT @ATIXAG: Nicht verpassen! Der nächste virtuelle #LinuxStammtisch startet morgen ab 18:30 Uhr. Freu dich auf die Referenten @stankowic_de…'"

FLOWS

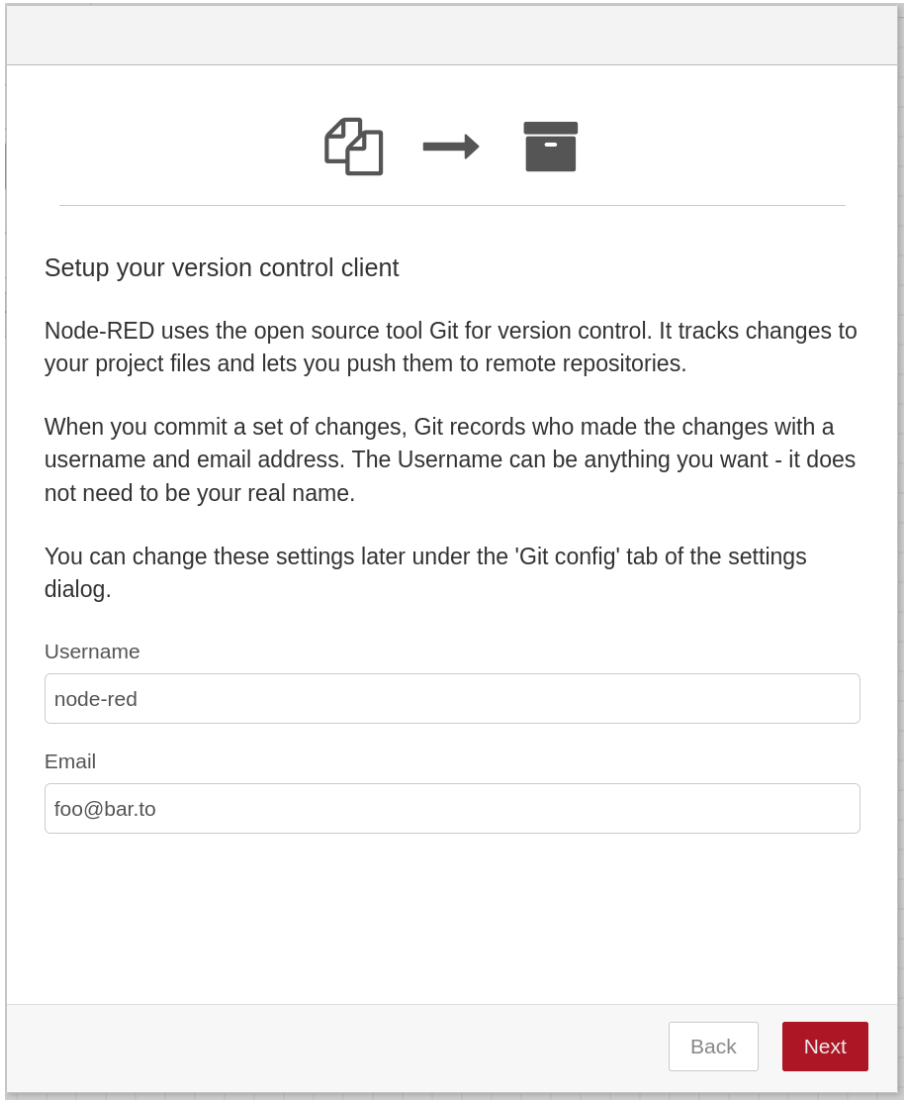
> werden in Form von Registerkarten in der GUI angezeigt

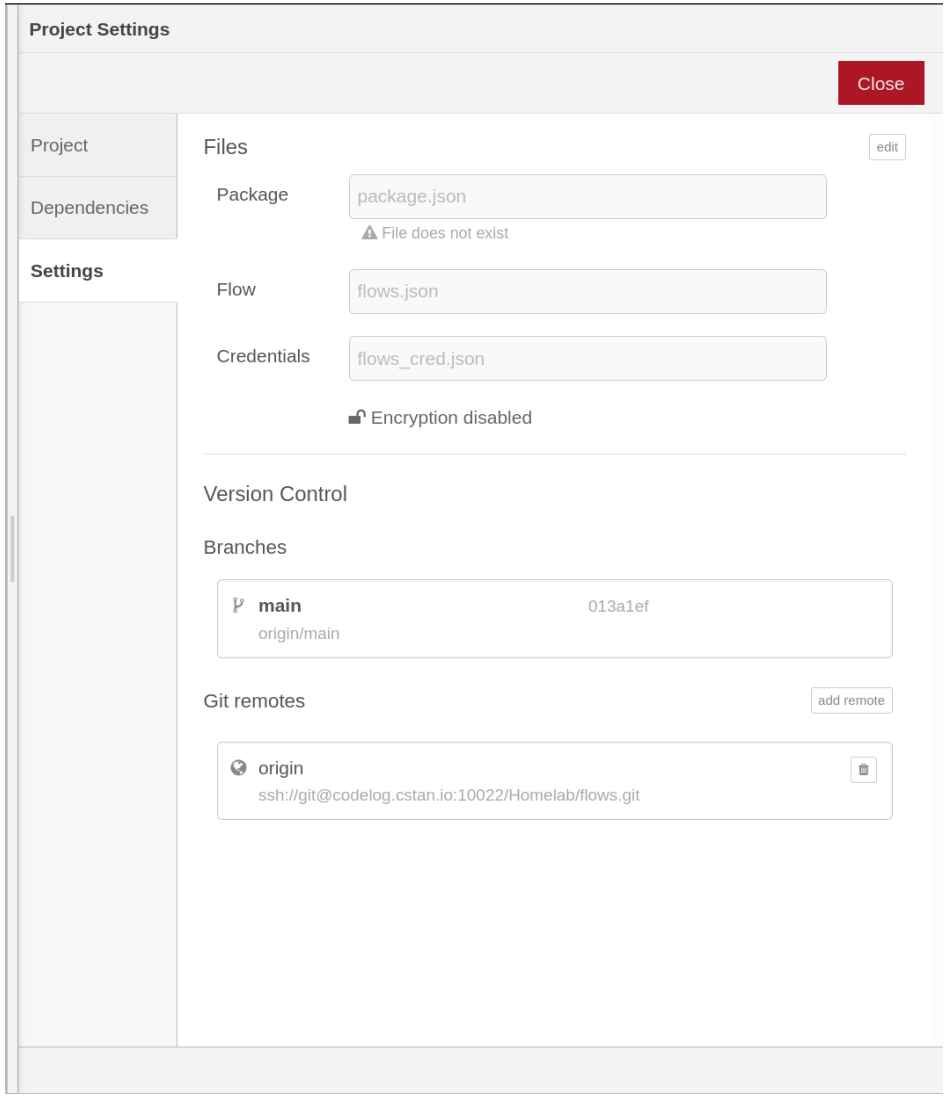
> können komfortabel als **JSON** exportiert/importiert werden

> komplexe Anwendungen können in **Subflows** aufgeteilt werden

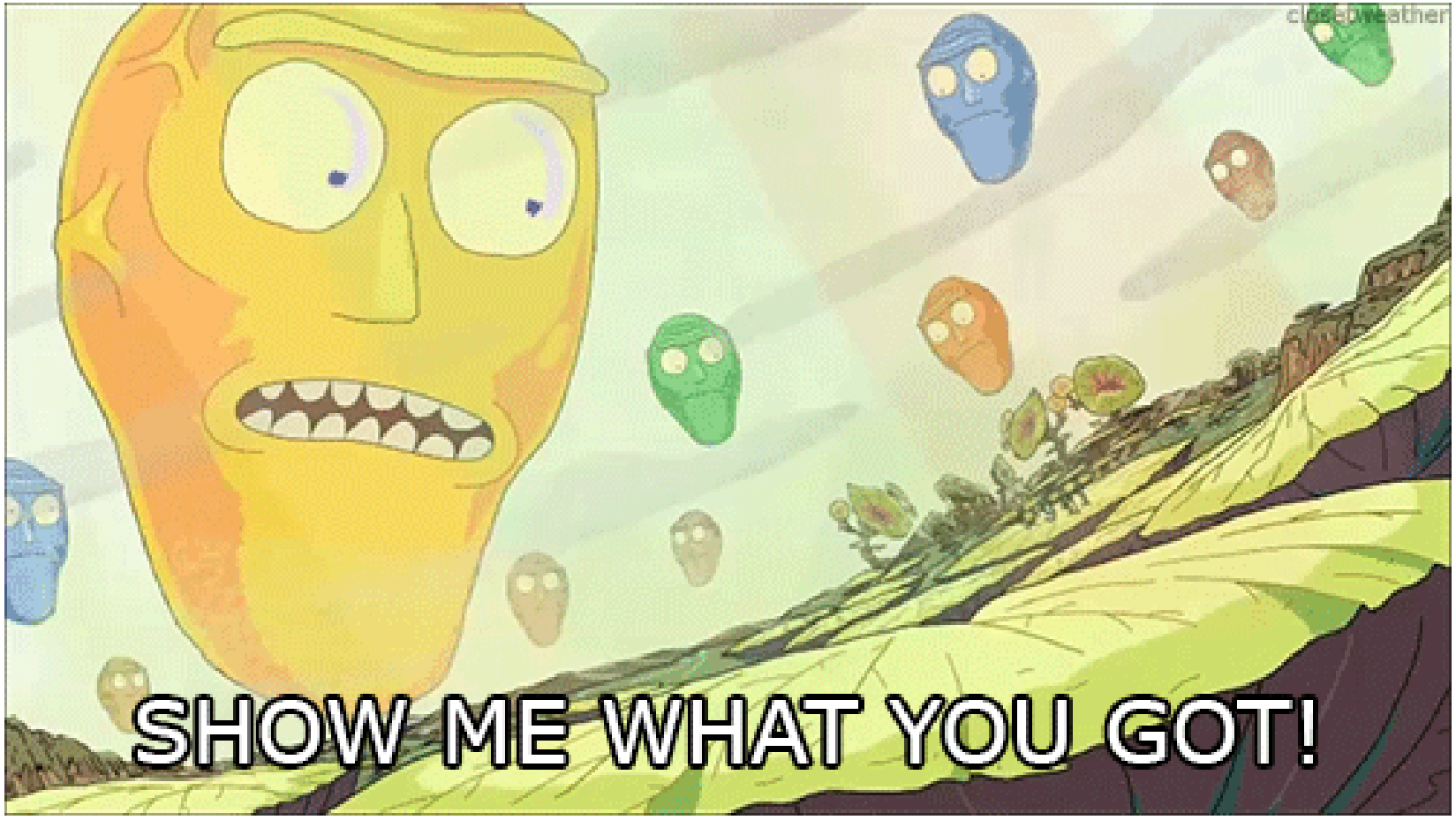
BACKUP & RESTORE

- Flows können via JSON exportiert/importiert werden
- Über Projekte ist auch eine Git-Integration möglich
  - muss über `settings.js`-Datei erst aktiviert werden
- Für ältere Versionen gibt es ein Modul, das Git-Support nachrüstet


Setup your version control client
Node-RED uses the open source tool Git for version control. It tracks changes to your project files and lets you push them to remote repositories.
When you commit a set of changes, Git records who made the changes with a username and email address. The Username can be anything you want - it does not need to be your real name.
You can change these settings later under the 'Git config' tab of the settings dialog.
Username
node-red
Email
foo@bar.to
Back
Next


Project Settings
Close
Project
Dependencies
Settings
Files
edit
Package
package.json
File does not exist
Flow
flows.json
Credentials
flows_cred.json
Encryption disabled
Version Control
Branches
main
013a1ef
origin/main
Git remotes
add remote
origin
ssh://git@codelog.cstan.io:10022/Homelab/flows.git

// DEMO



// LINKLISTE

> Node-RED-Webseite: [klick!]

> Node-RED-Dokumentation: [klick!]

> Node-RED Projects (*Git-Integration*): [klick!]

> Node-RED Library: [klick!]

> Node-RED YouTube-Kanal: [klick!]

> Interview mit *flow-based programming*-Erfinder J. Paul Morrson: [klick!]

> Eigenbau-Sprachassistent mit Node-RED und Rhasspy, Artikelserie: [klick!]

> FrOSCon 2020: Eigenbau-Sprachassistenten: [klick!] (PDF)

DANKE FÜR DIE AUFMERKSAMKEIT

(JETZT BITTE WIEDER AUFWACHEN)