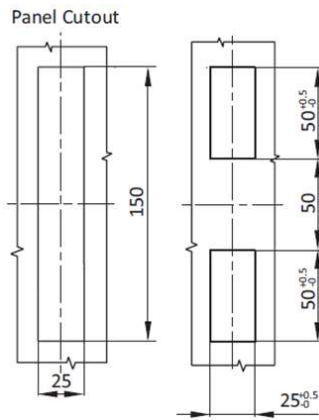


# TANlock 3 — Electronic POE Swing Handle



**Authentication Modules**

RFID+PIN Pad, Touch Pad, Fingerprint, Hand Vein Scan



**Installation:** Standard 150x25mm cutout or custom adapter plates designed to customer requirements.  
**Power:** CAT5 10/100Mbps POE link, USB-C power backup.  
**Interfaces:** Web API, https, SNMP, syslog, LDAP for integration to existing management & monitoring systems.  
**Industry standards:** Full AAA, Two factor authentication. Supports feature requirements of PCI-DSS, SOX, HIPPA.

## RFID Module

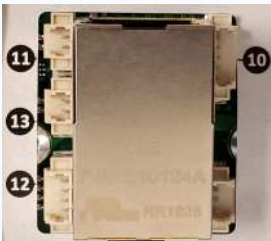


**LEGIC SM6300** Best in class global RFID/Bluetooth chip solution, EAL5+ certified. Supports all globally relevant smartcard technologies such as LEGIC advent and prime, MIFARE and HID iCLASS, 13.56Mhz

**RFID / NFC** ISO 14443 A+B Inside Secure  
 ISO 15693 Song Felica

**Bluetooth** V5.0 Bluetooth Low Energy Central to peripheral device  
 Communication to apps with LEGIC Mobile SDK Client or server role

## Tanlock 3 ports



**POE** Standard POE 10/100Mbps network connection with direct support for DHCP or static IP address, HTTPs, SNMP, Syslog, LDAP.

**10 Future expansion** 5 pin JST 1.25mm connector allows future use for integration of third party peripherals via UART interface. Not currently used by the firmware.

**11 Door Sensor** Two ports, 2 pin JST 1.25mm connectors for reed type door sensors

**13 Door Sensor** SNMP trap event generated on sensor trigger, easy to alert multiple applications from a central SNMP monitoring server.

**12 CANBUS** 4 pin JST 1.25mm connector for CANBUS type peripherals for attaching Temperature, humidity, vibration sensors.

**Relay port** 3 pin JST 1.25mm two circuit relay, Max. current load on each circuit 100mA. Can be triggered on web API commands to open/close relay and turn on/off external devices.



Mechanical Design  
[www.fath24.com](http://www.fath24.com)

**Integrated Solutions**

Electronic Design  
[www.tanlock.com](http://www.tanlock.com)



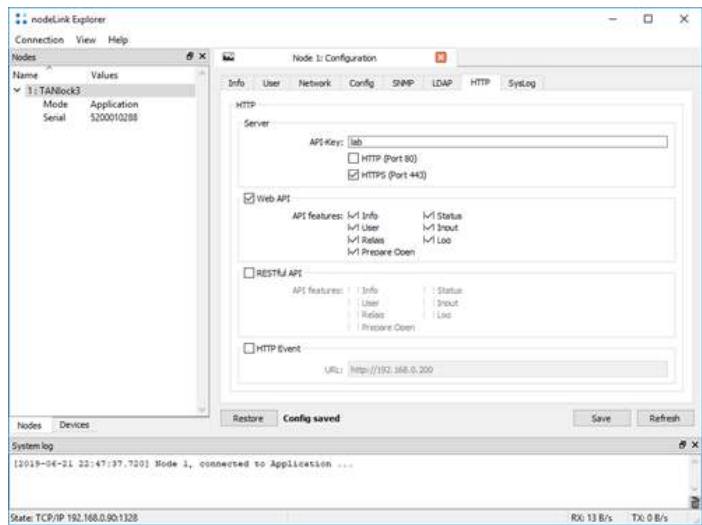
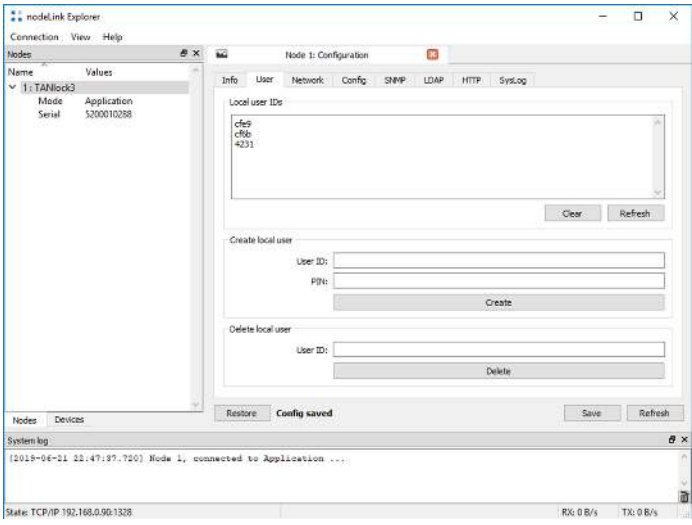


### Secure cabinet access

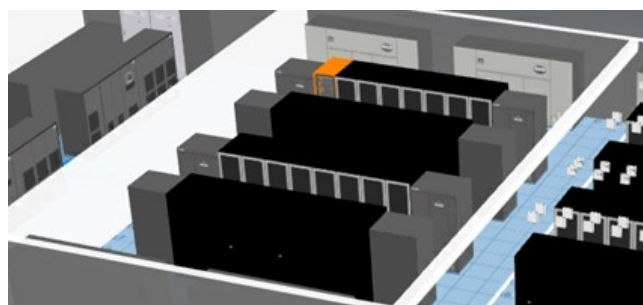
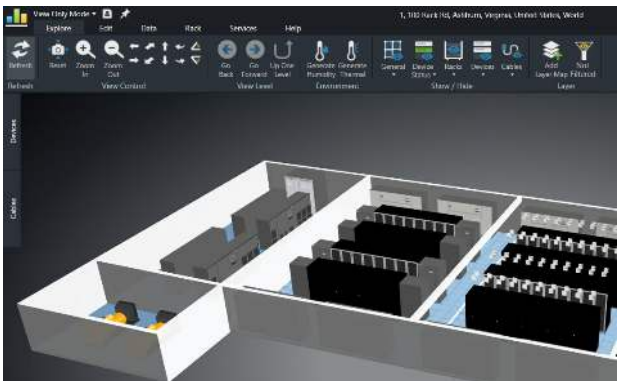
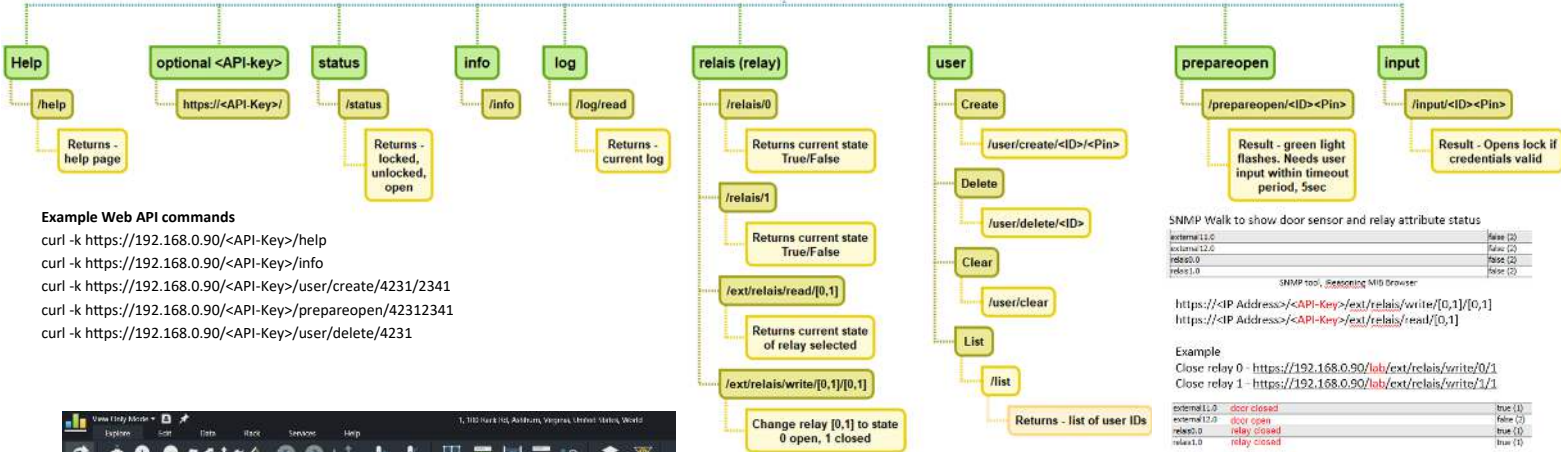
**Authentication:** PIN, RFID Card, Biometric, two factor authentication  
**Authorization:** users, groups, access control, allow/deny controls protected resources.  
**Accounting:** Audit trail, user auth method, date, time stamp, snmp, syslog, lock status.

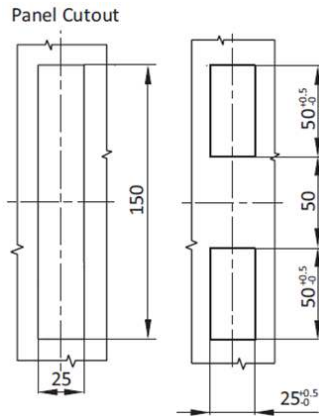
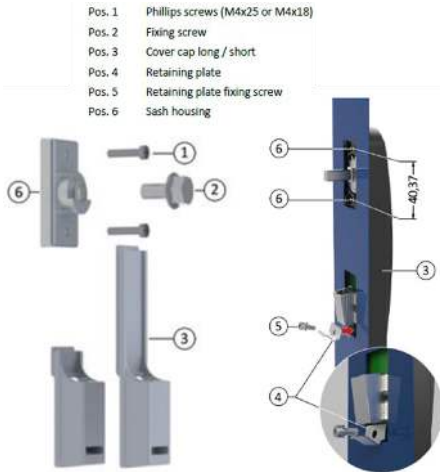
### Lock management

**TANlockExplorer & HTTPS interface:** Simple Windows application to configure basic setup of the TANlock and for user credential management.  
**Web API:** API commands allow easy integration to 3rd party software or direct command line configuration of a suite of locks from a designated host.  
**DCIM:** Full integration to Data Center Infrastructure Management software, Optimum Path Visual Data Center, Eaton VPM/VCOM, Sunbird, Vertiv Intelligence.



### TANLock Web API





The TANlock 3 is powered by using a 48v POE port. This can either be from a switch or a POE injector device. It can also be powered via the USB-C port if the POE source fails.

The default IP address and subnet mask is 192.168.0.90/255.255.255.0 Initial setup is completed using the TANlockExplorer tool. It is a Windows based config tool to set the basic lock details and can be directly run from a USB disk. No specific install or admin permissions are required unless using the USBC-to-Serial interface on the TANlock which requires a device driver to be installed. TANlockExplorer is being replaced with https.

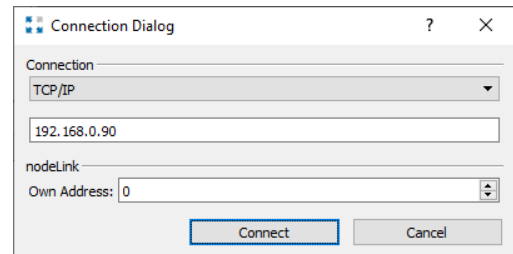
A standard cutout can be used unless using a non standard rack in which case an adapter plate may be required. Parts for assembling the TANlock to the door are supplied in the box. The CAM and any rods are not supplied as part of the TANlock.

**TANlockExplorer:** <http://tanlocks.info/tanlockfiles/TANlockExplorer.zip> (MD5: 644F0249C13270A798A4C26C96881B02)  
**Serial Device Driver:** <http://tanlocks.info/tanlockfiles/USB-TAN3-Driver.zip> (MD5: 52C549DD00BACE53E484ACA995AA834)  
**Curl:** Windows 10 v1803 onwards ships with curl installed. Try 'curl -help' to check it is available. <https://curl.haxx.se> for other OS.

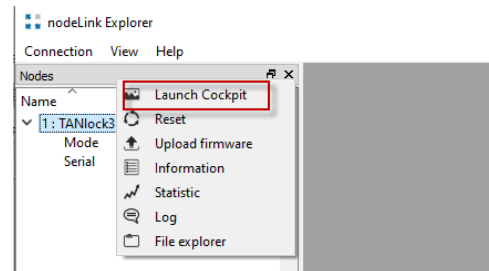
### Typical basic setup steps

1. Set IP address
2. Set Subnet Mask
3. Set Gateway IP address
4. Set User ID & PIN Length (Min. 4 when using RFID cards)
5. Set new 'Cockpit Config' password
6. Set SNMP server IP address
7. Set the <API-Key>
8. Turn off http access (https only)
9. Set syslog server IP address
10. Save config
11. Reset TANlock 3
12. Check connectivity 'curl -k https://<IP>/<API-Key>/info'
13. Document the lock details
14. Add users and RFID Cards

Enter the IP address or change the connection type to Serial.

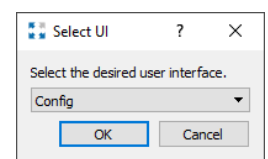
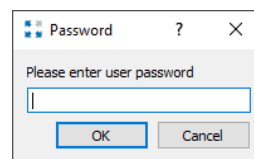
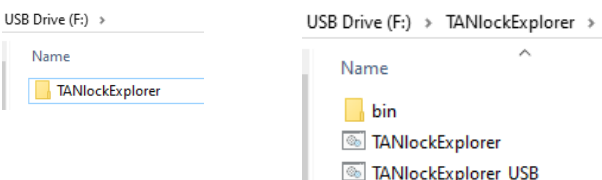


Launch the 'Cockpit' to start the setup.



Enter the default password: 91174. Change this during setup.

Start 'TANlockExplorer' using the batch file. If you require the USB to serial interface use the \_USB batch file.

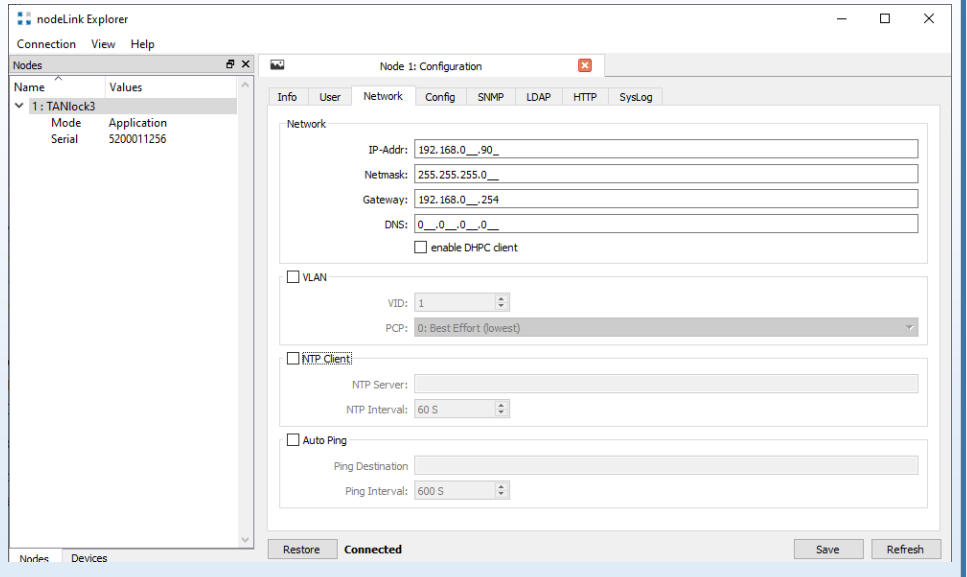


1. Set IP address.
2. Set Subnet Mask.
3. Set Gateway IP address.

DHCP can be used as an alternative to static IP addresses. The lock will be allocated an IP address when it reboots.

VLAN is not required.

NTP client can be set to the next hop gateway if it can act as a time server.



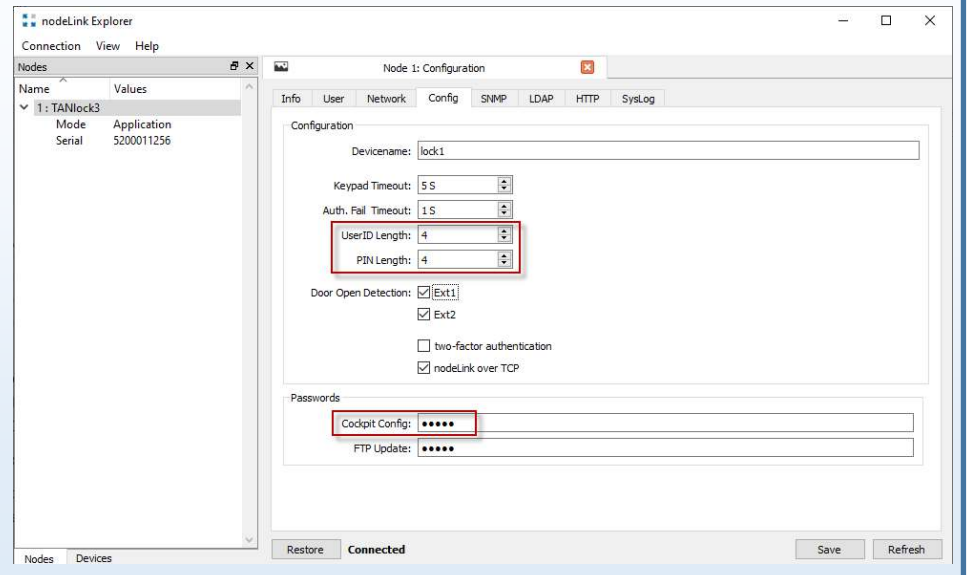
4. Set User ID & PIN Length (default 3).  
Min. 4 if using RFID cards, duplicate card IDs may occur if less than 4 digits are used.

Devicename: hostname (FQDN)

Keypad timeout, time to enter a PIN before selecting OK button.

Auth Fail timeout, time to wait between authentication failures.

Two factor auth, enable to require two authentication events. (lock reset required)

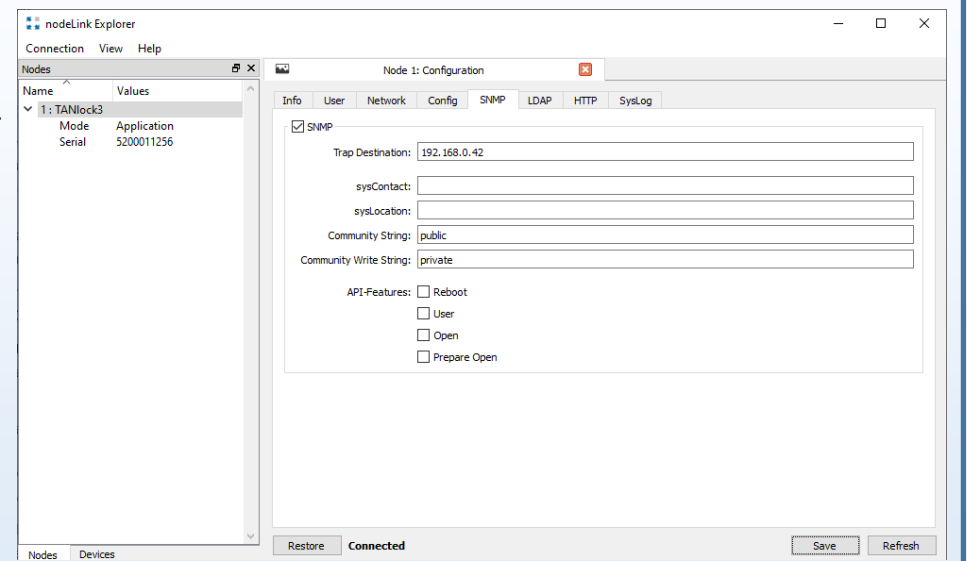


6. Set SNMP server IP address.

SNMP traps for unlock/lock change, door sensor status will be sent to this IP address.

Default community strings are 'public' & 'private'. Change to the local network environment settings. The network administrator will know these.

API-features, Not recommended to use as only SNMP v1 & v2 are currently supported, no authentication or encryption.



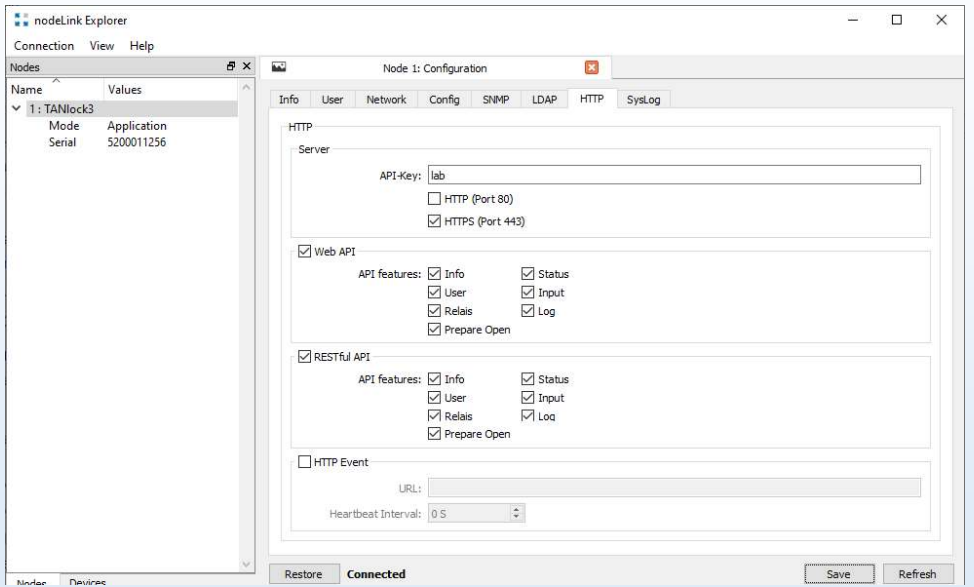
## 7. Set the <API-Key>. (Default 'lab')

This key can be unique per lock or per site and is required in any Web API command if the value is not blank.

## 8. Turn off http (https only).

To ensure client to lock encrypted connections disable http for Web API commands.

Unselect any Web API commands you want to disable. For some sites the 'Input' command may be disabled as this allows the lock to be remotely opened without user interaction at the physical lock location.



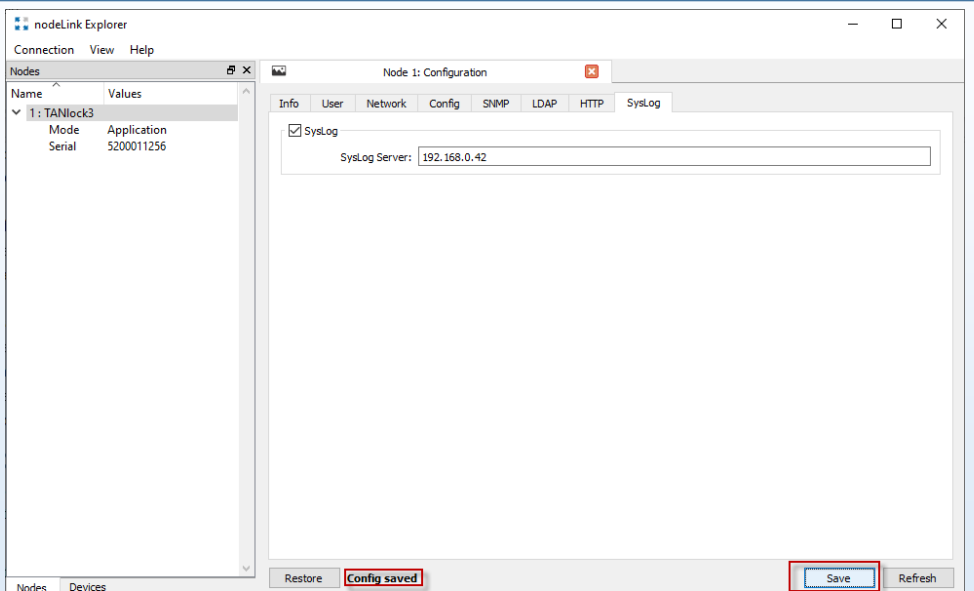
## 9. Set syslog server IP address.

Log events will be sent to this server if defined.

Log events are also stored directly on the lock and can be retrieved using the Web API command.

```
curl -k https://<IP>/<API-Key>/log/read
```

## 10. Save config.



## 11. Reset TANlock.

Any changes to the configuration requires a lock reset/reboot for the changes to take affect.

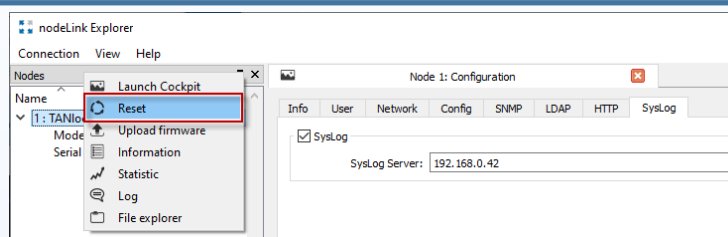
Adding users does not require a reset/reboot.

## 12. Check connectivity to the new IP.

```
curl -k https://<IP>/<API-Key>/info
```

## 13. Document the lock details.

MAC address if tracking those for switch port/cable management documentation.



```
C:\Users\user1>curl -k https://192.168.0.90/lab/info
{
  "software": "07x2",
  "hardware": "04",
  "serialno": "5200011256",
  "macaddr": "70:B3:D5:DC:32:54",
  "time": "Sat Jan 1 03:19:29 2000",
  "user": "",
  "sensor": {
    "lock": true,
    "handle": false,
    "motor": true,
    "temperature": 0
  },
  "external": {
    "ext_11": false,
    "ext_12": false,
    "relais_0": false,
    "relais_1": false
  }
}
```

## 14. Add users and RFID Cards.

Enter User ID + PIN. Example

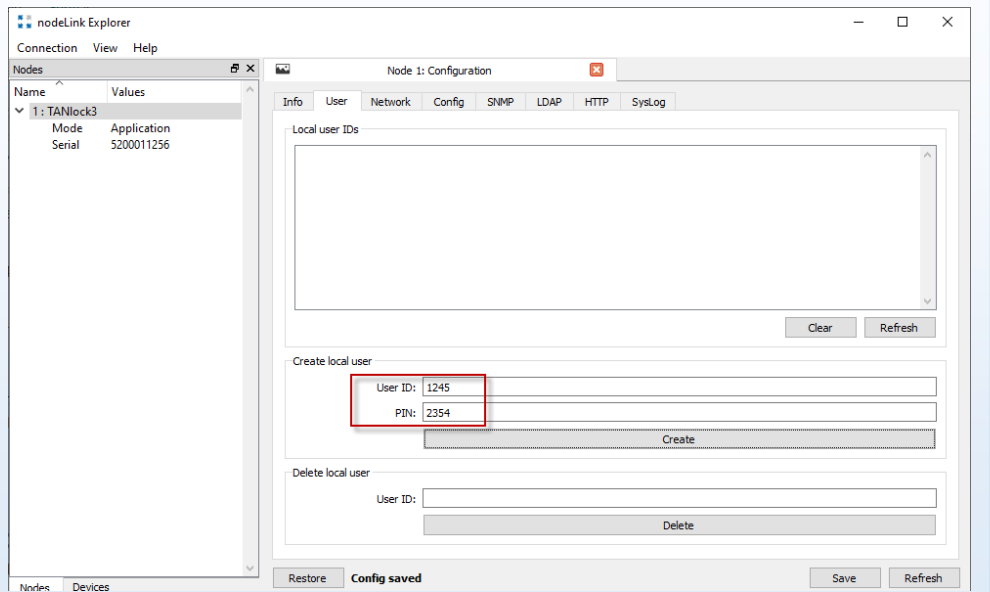
User ID: 1245

PIN: 2354

For this example a user enters 8 digits as the PIN they are assigned to open the lock. 12452354

Web API, user create

```
curl -k https://<IP>/<API-Key>/user/create/<ID>/<PIN>
```



## Adding RFID cards

Start the log event viewer

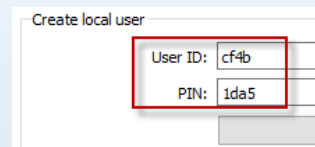
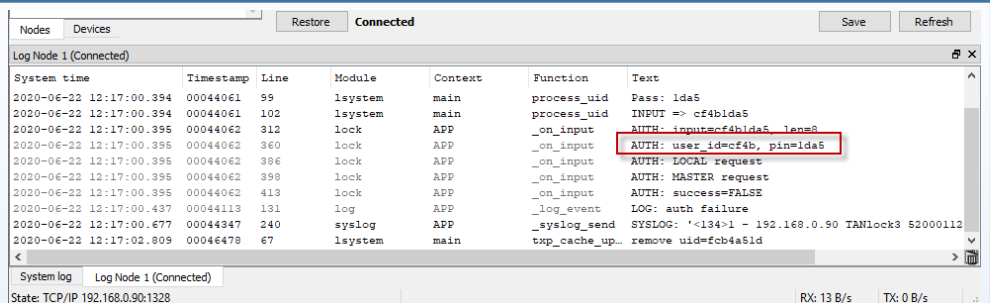


Scan an RFID card.

Identify the RFID credentials in the log events.

Add the User ID + PIN (RFID details)

Repeat for each RFID card.



## Add RFID cards to multiple locks

Create a batch file to run multiple Web API commands to deploy the same RFID cards to multiple locks (IP addresses).

Using the Web API commands RFID cards or user PINs can be enabled on multiple TANlocks in minutes.

There is no special relationship between a source IP address sending the Web API commands and the TANlock 3.

Access controls must be implemented using standard network devices, like firewalls and switch/router ACLs.

```
C:\Users\user1>notepad add-rfid-cards.bat
```

```
curl -k https://192.168.0.91/lab/user/create/cf4b/1da5
curl -k https://192.168.0.92/lab/user/create/cf4b/1da5
curl -k https://192.168.0.93/lab/user/create/cf4b/1da5
curl -k https://192.168.0.94/lab/user/create/cf4b/1da5
REM Next RFID Card
curl -k https://192.168.0.91/lab/user/create/cfe9/cd61
curl -k https://192.168.0.92/lab/user/create/cfe9/cd61
curl -k https://192.168.0.93/lab/user/create/cfe9/cd61
curl -k https://192.168.0.94/lab/user/create/cfe9/cd61
```

```
C:\Users\user1> add-rfid-cards
```

```
c:\Users\user1>curl -k https://192.168.0.90/lab/user/create/cfe9/cd61
{
    "result": "OK"
}
...
...
...
c:\Users\user1>
```

## Web API commands

Batch files can be generated and executed using 'curl'. API commands return JSON objects.

The locks can be managed directly from the command line using the Web API or from a GUI front end. TANlock can use existing syslog and snmp servers.

When using a GUI front end the JSON objects returned will need to be parsed, processed, and displayed back into the GUI. The snmp attributes can be polled for the status of the lock and snmp traps processed for changes in state.

## Web API command format & examples using curl

Format for using 'curl'

```
curl -k https://<IP address or Hostname>/<API-Key>/<cmd>
```

-k suppresses error messages for self-signed certificates

Either HTTP or HTTPS can be used. Recommended only HTTPS be used.

The <IP address or hostname> is the target lock.

<API-Key> is optional, set in TANlockExplorer in the HTTP tab.

<cmd> API command to apply.

## Status

```
C:\marvin>curl -k https://192.168.0.90/lab/status
```

```
{
  "state": "locked"
}
```

```
C:\marvin>curl -k https://192.168.0.90/lab/status
```

```
{
  "state": "unlocked"
}
```

```
C:\marvin>curl -k https://192.168.0.90/lab/status
```

```
{
  "state": "open"
}
```

## Info

```
C:\marvin>curl -k https://192.168.0.90/lab/info
```

```
{
  "software": "07x2",
  "hardware": "04",
  "serialno": "5200011256",
  "macaddr": "70:B3:D5:DC:32:54",
  "time": "Tue Jun 23 14:26:20 2020",
  "user": "",
  "sensor": {
    "lock": true,
    "handle": false,
    "motor": true,
    "temperature": 0
  },
  "external": {
    "ext_11": false,
    "ext_12": false,
    "relais_0": false,
    "relais_1": false
  }
}
```

## Log

'timestamp' is Unix time taken from start date of 01/01/2000. The lock should have an ntp server set to sync to the correct time.

```
C:\marvin>curl -k https://192.168.0.90/lab/log/read
```

```
{
  "timestamp": 1592915127,
  "message": "locked"
},
```

```
{
  "timestamp": 1588339254,
  "message": "auth success via local user uid=cf4b"
},
{
  "timestamp": 1588339254,
  "message": "unlocked"
},
{
  "timestamp": 1588339256,
  "message": "handle position open"
},
{
  "timestamp": 1588339256,
  "message": "handle position close"
},
{
  "timestamp": 1588339257,
  "message": "locked"
}
}
```

## User commands

```
C:\marvin>curl -k https://192.168.0.90/lab/user/list
```

```
[
  {
    "user_id": "cf4b"
  },
  {
    "user_id": "cfe9"
  },
  {
    "user_id": "1234"
  }
]
```

```
C:\marvin>curl -k https://192.168.0.90/lab/user/create/1111/2222
```

```
{
  "result": "OK"
}
```

```
C:\marvin>curl -k https://192.168.0.90/lab/user/delete/1111
```

```
{
  "result": "OK"
}
```

```
C:\marvin>curl -k https://192.168.0.90/lab/user/clear
```

```
{
  "result": "OK"
}
```

Clear all users from the lock database. This should be used with care, deletes all users defined on the lock. There is no undo for this and users will need to be added back or recovered from the backup user file.

Example, add back all the users, 2 x RFID cards, 1 x PIN

```
curl -k https://192.168.0.90/lab/user/create/cf4b/1da5
curl -k https://192.168.0.90/lab/user/create/cfe9/cd61
curl -k https://192.168.0.90/lab/user/create/1234/1234
```

## Prepareopen

This allows a remote open command to be sent to the TANlock but not immediately open the lock. An LED on the lock will flash for the user input period (default 5s). The user must hit the OK button within the timeout period for the lock to open.

```
C:\marvin>curl -k https://192.168.0.90/lab/prepareopen/12341234
```

```
{
  "result": "OK"
}
```

```
C:\marvin>curl -k https://192.168.0.90/lab/prepareopen/cf4b1da5
```

```
{  
  "result": "OK"  
}
```

#### Input

```
C:\marvin>curl -k https://192.168.0.90/lab/input/12341234
```

```
{  
  "result": "OK"  
}
```

```
C:\marvin>curl -k https://192.168.0.90/lab/input/cf4b1da5
```

```
{  
  "result": "OK"  
}
```

#### Relais

Display the current status of the Relay 0 & 1

```
C:\marvin>curl -k https://192.168.0.90/lab/ext/relais/read/0
```

```
{  
  "result": "false"  
}
```

```
C:\marvin>curl -k https://192.168.0.90/lab/ext/relais/read/1
```

```
{  
  "result": "false"  
}
```

```
C:\marvin>
```

Change the state of relay 0 to On

```
C:\marvin>curl -k https://192.168.0.90/lab/ext/relais/write/0/1
```

```
{  
  "result": "OK"  
}
```

Change the status of replay 1 to on

```
C:\marvin>curl -k https://192.168.0.90/lab/ext/relais/write/1/1
```

```
{  
  "result": "OK"  
}
```

Check the status of relay 0 & 1

```
C:\marvin>curl -k https://192.168.0.90/lab/ext/relais/read/0
```

```
{  
  "result": "true"  
}
```

```
C:\marvin>curl -k https://192.168.0.90/lab/ext/relais/read/1
```

```
{  
  "result": "true"  
}
```

Change the status of relay 0 to Off

```
C:\marvin>curl -k https://192.168.0.90/lab/ext/relais/write/0/0
```

```
{  
  "result": "OK"  
}
```

Change the status of relay 1 to Off

```
C:\marvin>curl -k https://192.168.0.90/lab/ext/relais/write/1/0
```

```
{  
  "result": "OK"  
}
```

#### Help

Displays examples usage of the Web API commands. Each release of firmware may introduce new API commands. Check the technical documentation for the latest information.

```
C:\marvin>curl -k https://192.168.0.90/help
```

The <API-Key> is not required for the help text to be displayed.

#### Summary of API commands

```
curl -k https://<IP address or Hostname>/<API-Key>/<cmd>
```

API Key: lab

API Key needs to be configured in TANlockExplorer

/help	Display help text
/lab/status	Lock status: locked, unlocked, open
/lab/info	Display information about the lock
/lab/log/read	Display the log file contents
/lab/user/list	Display list of users defined on the lock
/lab/user/create/1111/2222	Create a user 1111, password 2222
/lab/user/delete/1111	Delete user 1111
/lab/user/clear	Clear all users from the lock.
/lab/prepareopen/12341234	Remote open with user input required at the lock.
/lab/input/12341234	Remote open without user input required at the lock.
/lab/ext/relais/read/0	Check status of relay 0, true/false
/lab/ext/relais/read/1	Check status of replay 1, true/false
lab/ext/relais/write/0/1	Change status of relay 0 to On/true
/lab/ext/relais/write/1/1	Change status of relay 1 to On/true
/lab/ext/relais/write/0/0	Change status of relay 0 to Off/false
/lab/ext/relais/write/1/0	Change status of relay 1 to Off/false