

The background features abstract, overlapping green geometric shapes, primarily triangles and polygons, in various shades of green, creating a modern and dynamic visual effect.

# 第十九讲

## Node-RED (III)

### Lecture 19

## Node-RED (III)

明玉瑞 Yurui Ming  
yrming@gmail.com

# 声明

## Disclaimer

- 本讲义在准备过程中由于时间所限，所用材料来源并未规范标示引用来源。所引材料仅用于教学所用，作者无意侵犯原著者之知识产权，所引材料之知识产权均归原著者所有；若原著者介意之，请联系作者更正及删除。

The time limit for the preparation of these slides incurs the situation that not all the sources of the used materials (texts or images) are properly referenced or clearly manifested. However, all materials in these slides are solely for teaching and the author is with no intention to infringe the copyright bestowed on the original authors or manufacturers. All credits go to corresponding IP holders. Please address the author for remedy including deletion have you had any concern.

# 背景

## Background

- ▶ 虽然Node-RED本身并不是专为独立的工业级MQTT服务器而设计的，但将Node-RED与专用的MQTT代理，如Mosquitto，结合使用允许开发人员构建功能强大且可靠的应用程序。此时，Node-RED通常被用作可视化编程工具，用于创建与MQTT启用的设备或服务进行交互的自动化流程。这种组合通常用于工业自动化、物联网应用程序以及其他需要MQTT作为首选通信协议的场景。它提供了灵活性和开发的便捷性，同时保持了工业用例所需的稳健性。
- ▶ While Node-RED itself is not originally designed as a standalone industrial-strength MQTT server, by combining Node-RED with a dedicated MQTT broker like Mosquitto, it allows developers to build powerful and reliable applications. In this situation, Node-RED is often used as a visual programming tool to create automation flows that interact with MQTT-enabled devices or services. This combination is commonly used for industrial automation, IoT applications, and other scenarios where MQTT is a preferred communication protocol. It provides flexibility and ease of development while retains the necessary robustness for industrial use cases.

# 仪表盘

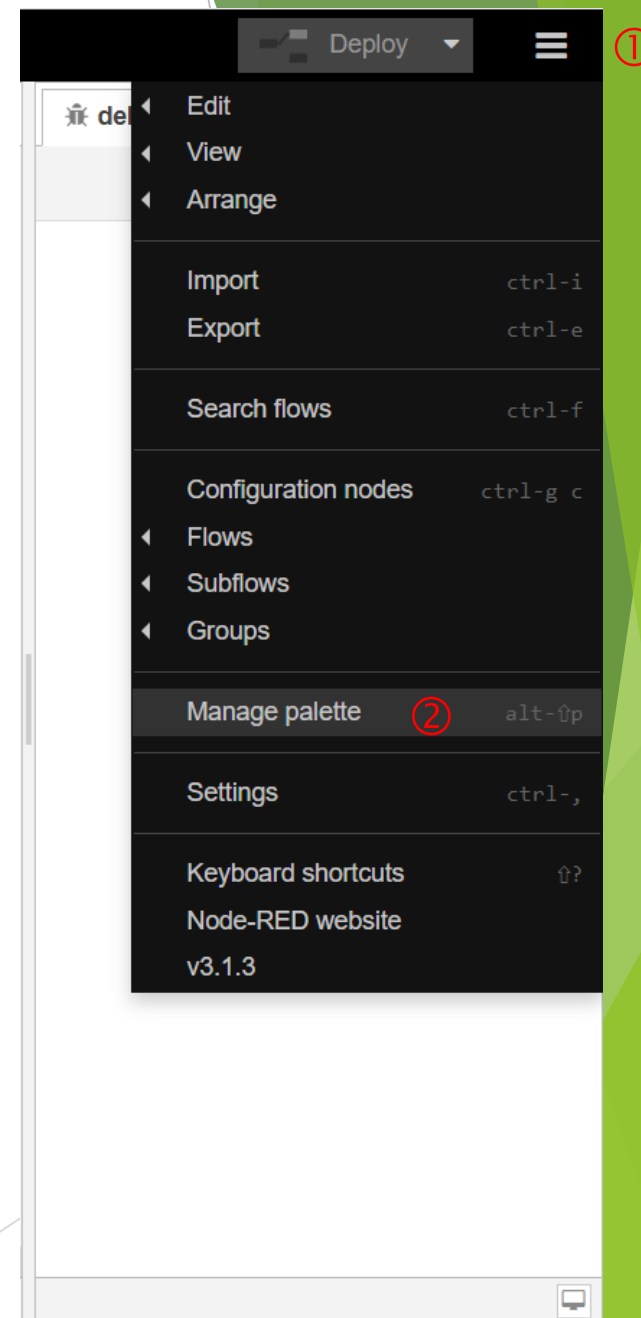
## Dashboard

- ▶ 前面我们已经介绍了Node-RED通过流处理业务逻辑的知识，但作为工业应用，一项重要的功能是进行数据可视化。我们接下来来讲Node-RED的仪表盘相关知识。首先要做的是安装相关节点。

We've covered the basics about how to process business logic in Node-RED. However, as a tool targeting industrial applications, one import functionality is for data visualization. We will discuss the dashboard of Node-RED in the following, but firstly we need to install the relevant nodes.

如右图所示，点击右上角的像三明治的图标，在下拉菜单中选择管理调色板菜单，进行相关节点的安装。

As indicated in the right figure, click the sandwich icon at the right top corner to bring up the drop-down menu. Select the Mange palette menu to install the relevant nodes.

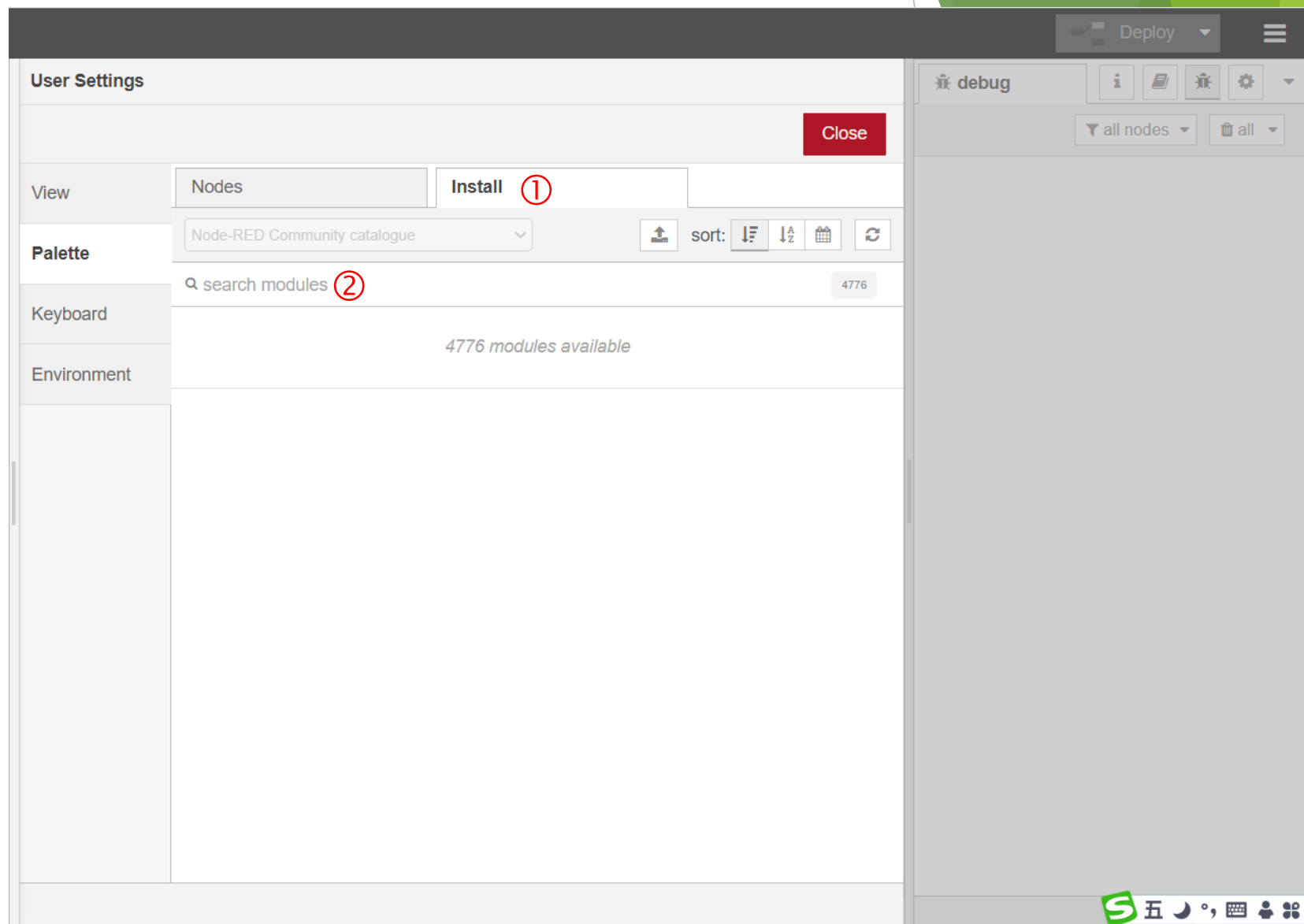


# 仪表盘 Dashboard

- ▶ 然后在弹出的对话框中，选择安装面板，安装合适的软件包，我们需要安装的有两项：

Then select the install tag from the popped up dialog, to install appropriate packages. There are two packages we have to install:

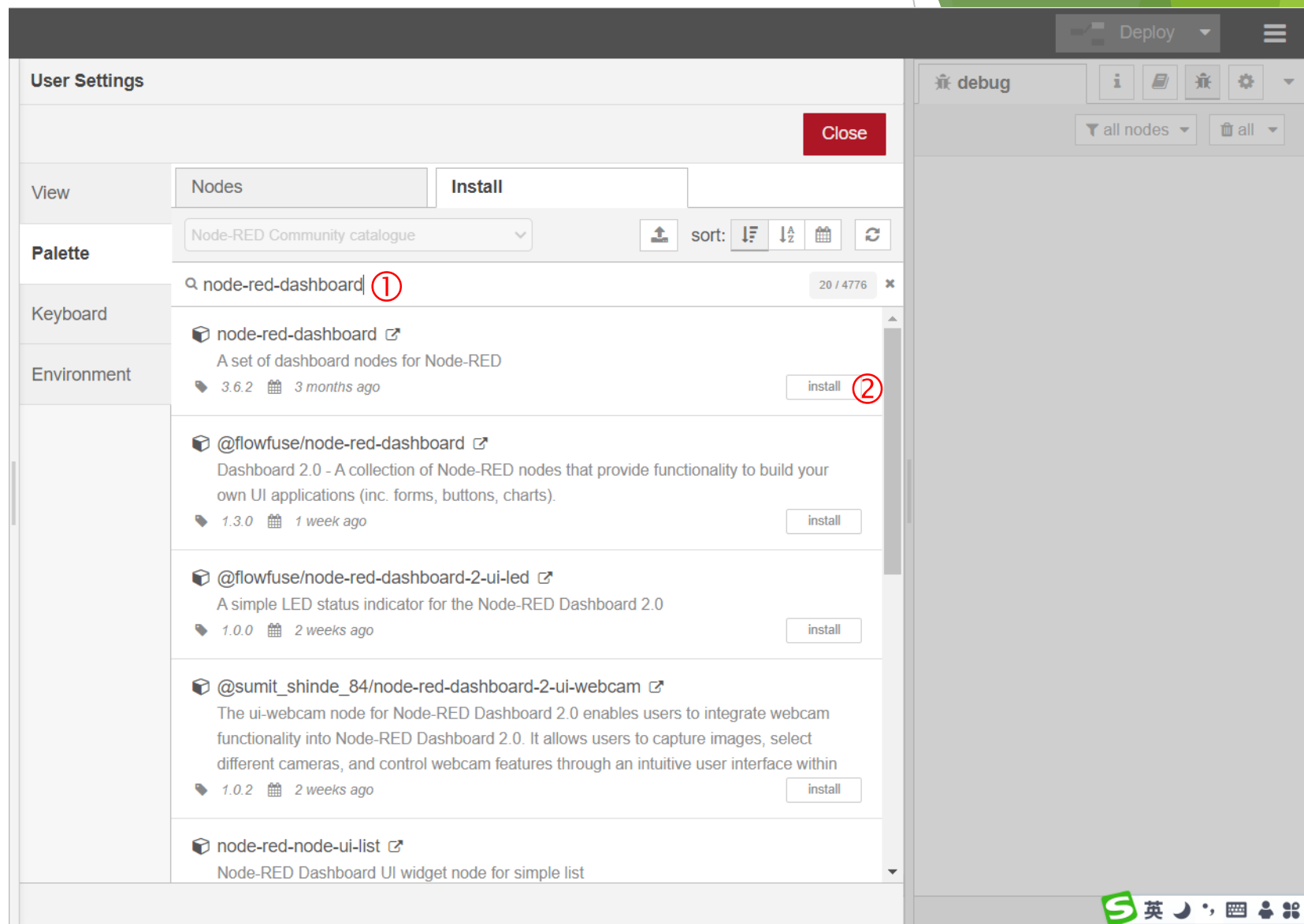
- ▶ node-red-dashboard
- ▶ node-red-contrib-open



# 仪表盘 Dashboard

- 通常，当用户键入包的名称之后，系统会自动进行匹配，因此，一般情况下用户不必键入完整的包名。当符合要求的包出现之后，用户单击安装即可。

Usually, the system tries to match the package when users start to text the package name, which means partial package name might be sufficient to filter out a specific package. After the desired package name appears, the user can hit the install button to install it.

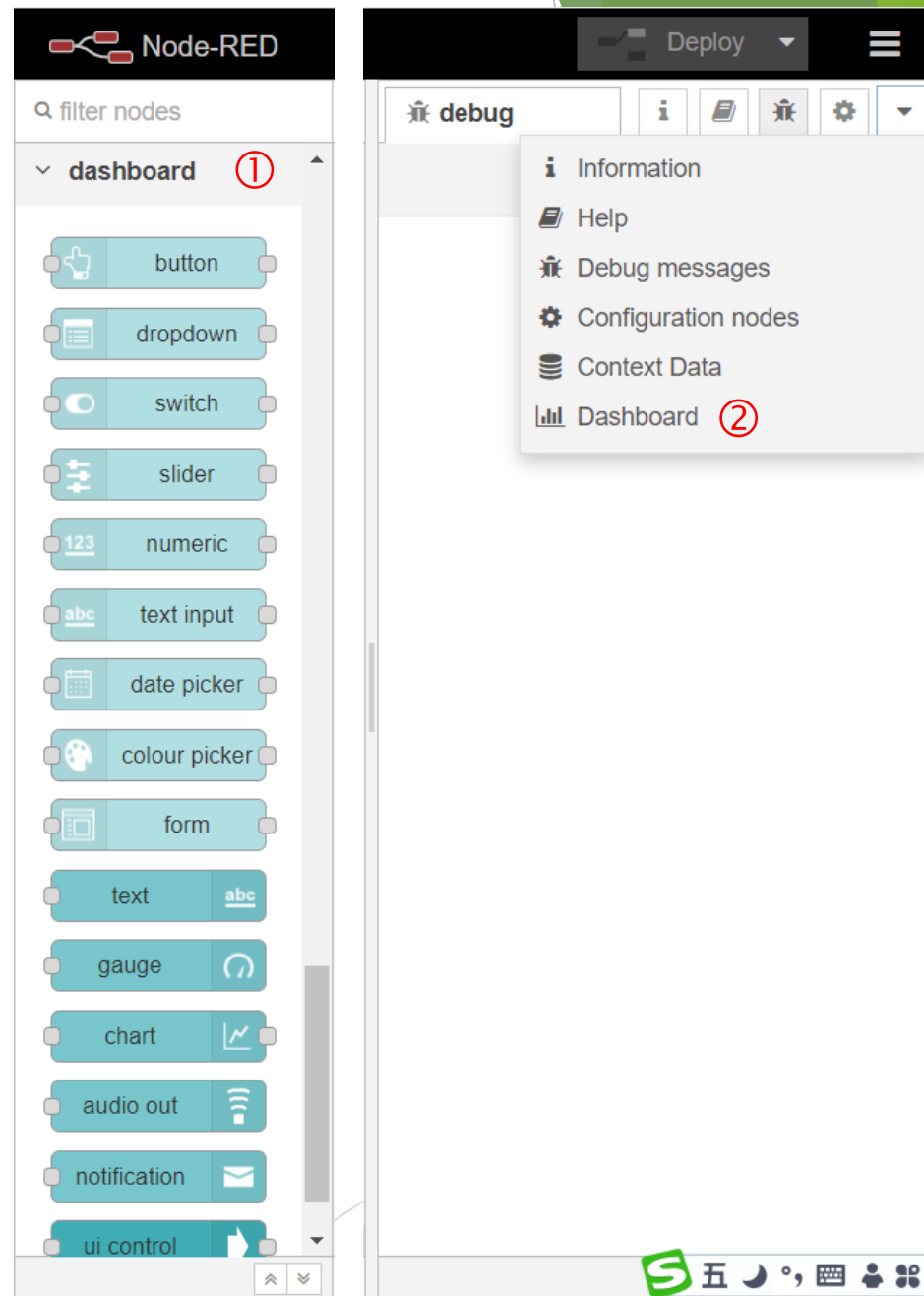


# 仪表盘

## Dashboard

- ▶ 当成功安装仪表盘包后，读者可以首先在左侧调色板栏发现有仪表盘组与相关节点被加入。同时，右侧的工具栏也会多出仪表盘栏。

After successfully installed the dashboard package, the user can find the dashboard group and the relevant nodes already added into the palette panel. Meanwhile, the right side bar also integrates the dashboard menu.



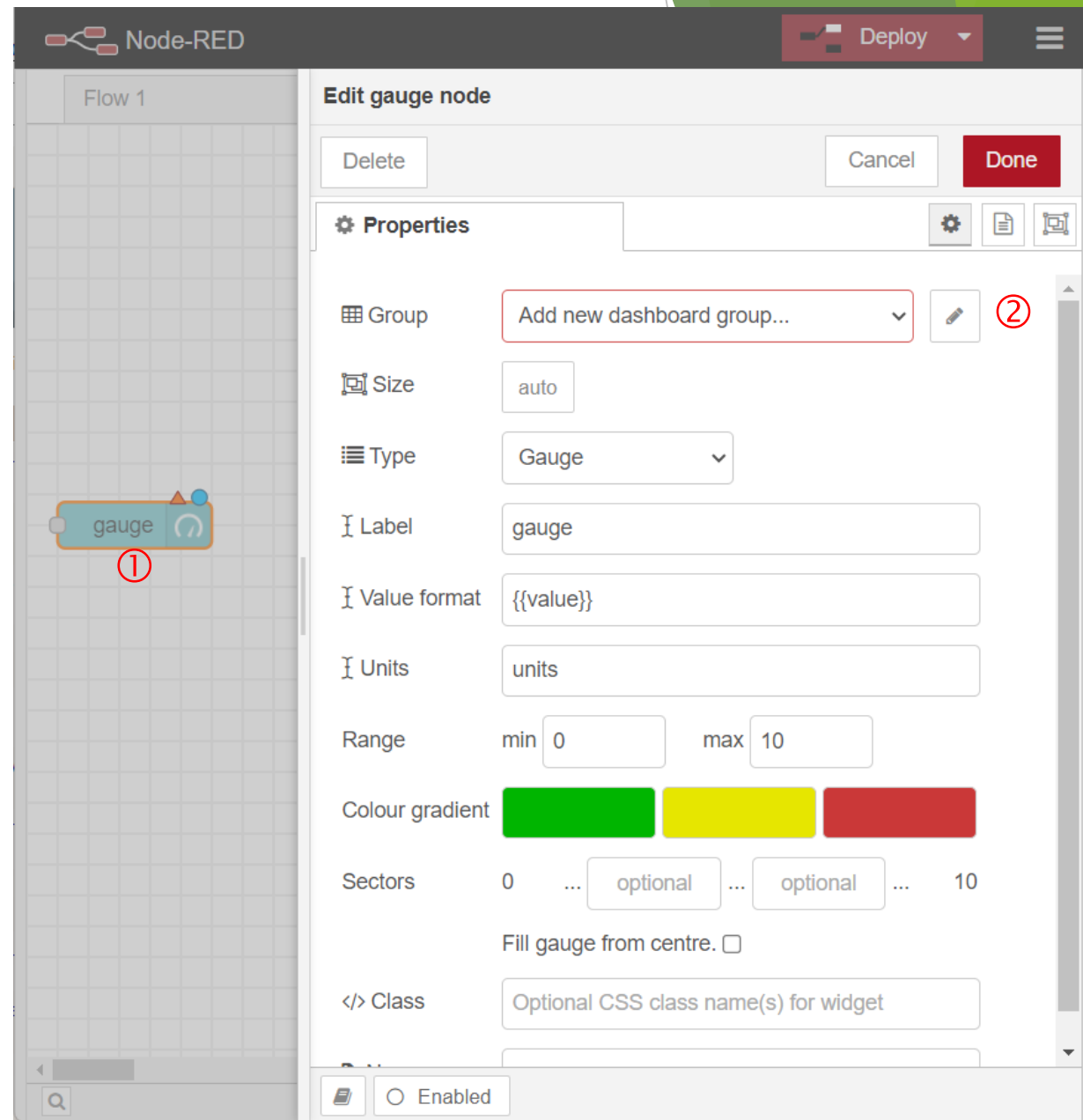
# 仪表盘 Dashboard

- 我们可以创建一个仪表盘来丰富上一讲的例子，例如展示并记录温度信息。仪表盘小部件的用法与一般节点没有实质的差别，即需要拖入工作区后进行配置。

We can create an instance of dashboard to extent our previous example, for example, display and log the temperature. The widget nodes of dashboard make almost no difference with the nodes we introduced before. They need to be dragged into the workspace then configured.

- 当节点未被配置时，会显示红色小三角，可以双击节点打开编辑对话框。

There will be a red triangle appears on the top-right of the node if the node is unconfigured. You can double-click the node to bring up the editing dialog.



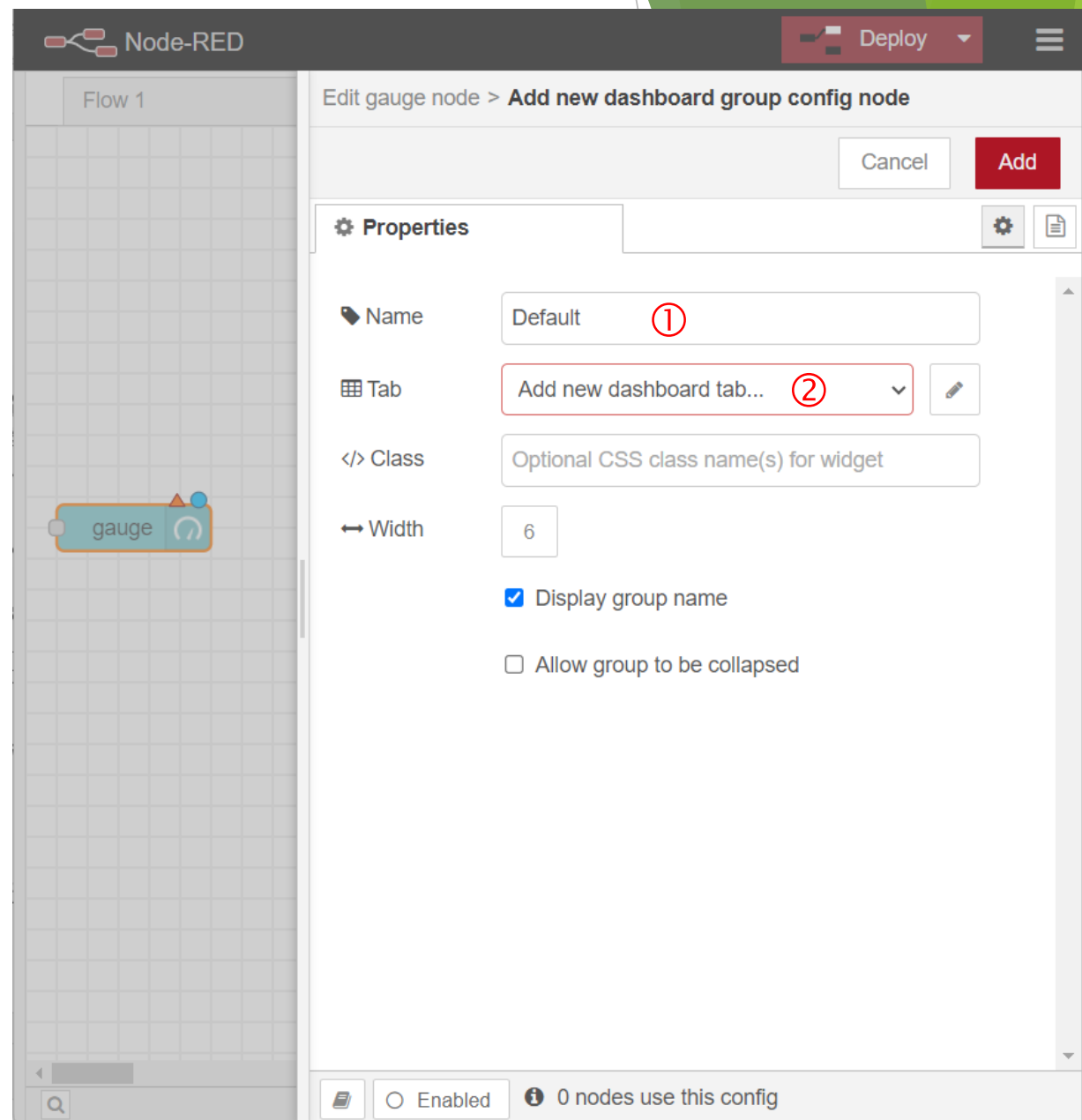


# 仪表盘

## Dashboard

- 通常仪表盘需要涵盖多个组件，因此需要创建一个组来容纳这些组件。实际上这有两个工作，一个是创建一个组，从观念上容纳这些组件，另外是创建一个页面，来展示这些组件，因此，如图所示，我们需要给这个组起一个名称，并创建一个页。

Usually, a dashboard needs many widgets, so we need to create a group to hold them together. Actually, there are two kind of tasks. One is to group these widgets together conceptually, the other is to create a tab for displaying these widgets.

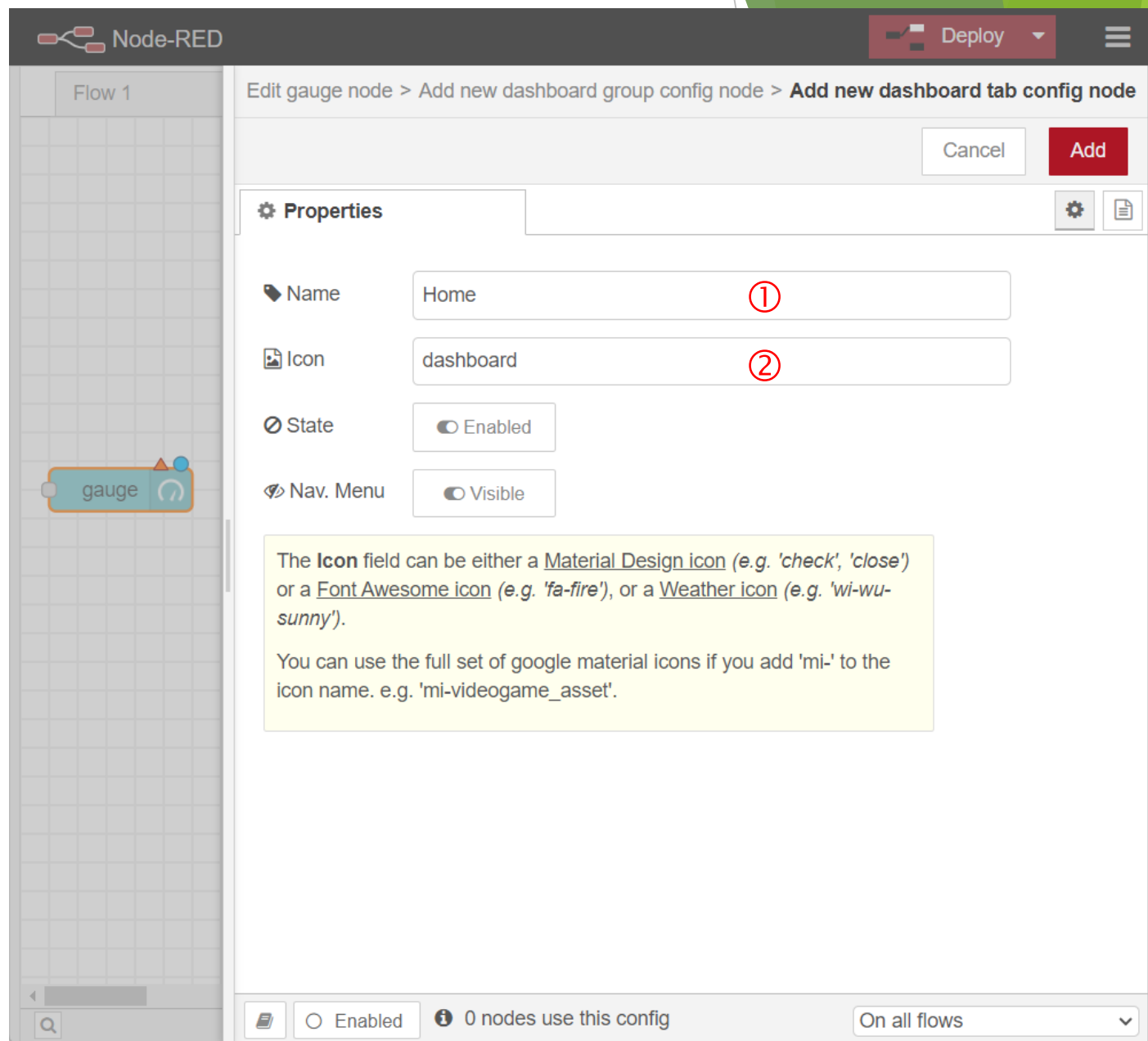


# 仪表盘

## Dashboard

- 当单击上面添加Tab页，会弹出如右所示的属性对话框。其中名称是指该Tab的名称，Icon在下面有解释，用户可以通过名称选择对应的图标被展示。在这里我们选择默认设置。

When user clicks the add new dashboard tab, a property dialog will be brought up. The name refers to the name of the tab, and for icon, there is a detailed explanation below. By specifying the name of the icon, the corresponding icon will be displayed. We adopt the default setting here.

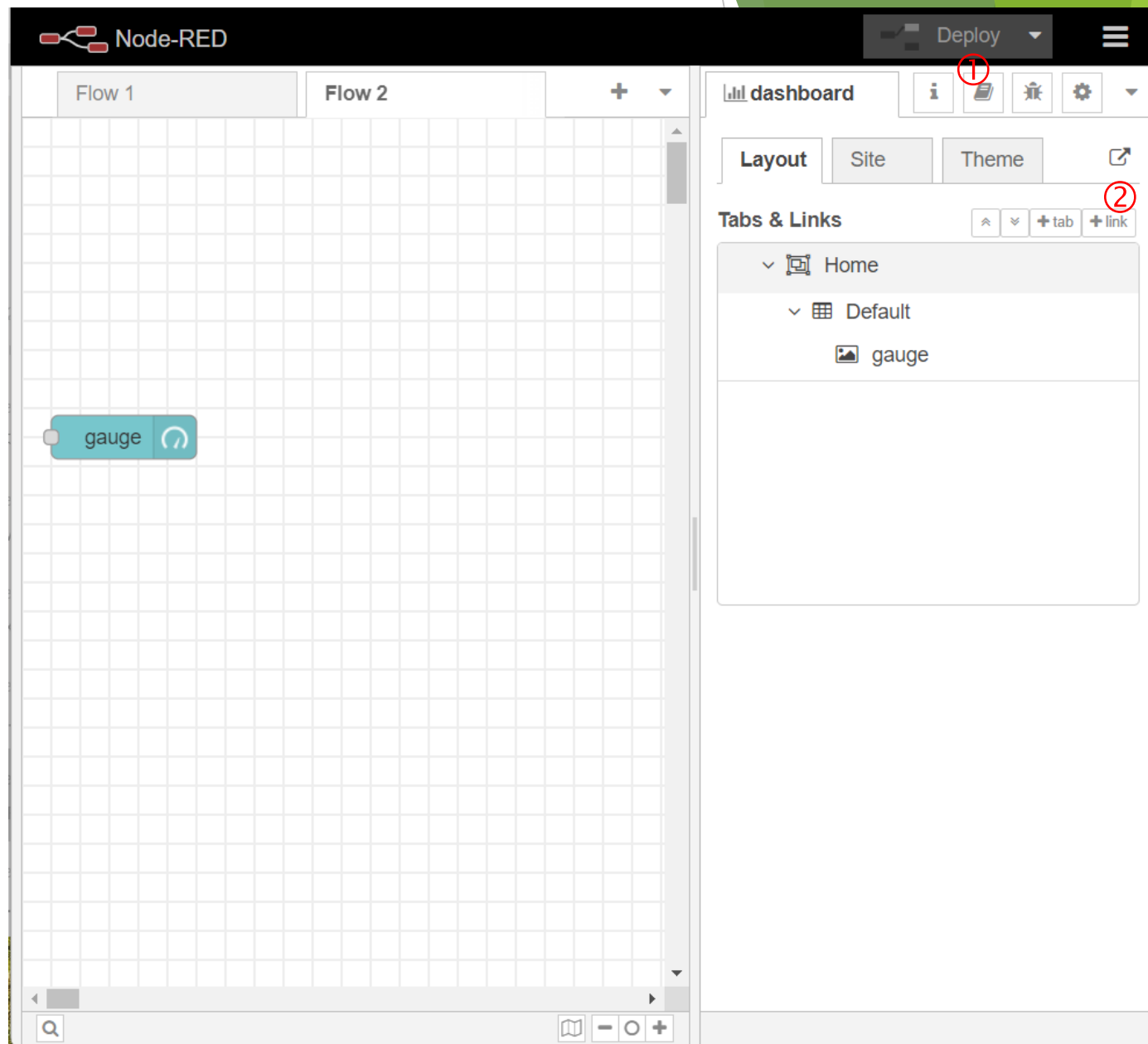


# 仪表盘

## Dashboard

- 为了初步体验仪表盘的功能，我们所有的选项都选择默认设置。此时，gauge上的红三角会消失。我们首先部署节点，然后单击右侧工具栏的外部链接按钮，如②所示，会打开新建的仪表盘Tab页。

To have a taste of the functionality provided by dashboard, we just accept all the default setting. Now, the red triangle on the top right of the gauge icon will disappear. In the following, we deploy the nodes, then click the link out button on the top right corner of the sidebar to bring up the dashboard webpage.

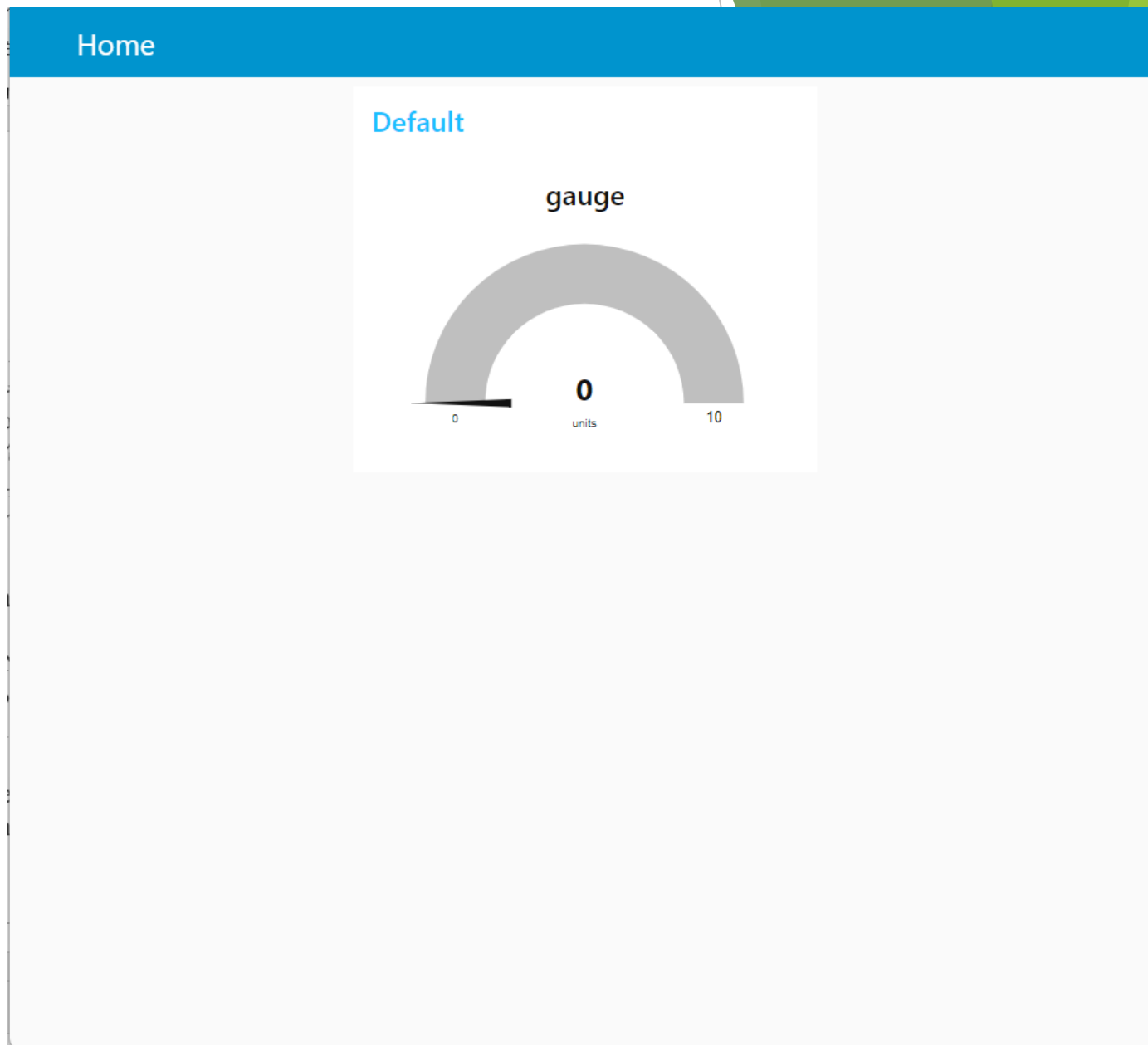


# 仪表盘

## Dashboard

- ▶ 右图是打开后的仪表盘的页面，由于只有一个小部件，其展示如右图所示。

The right figure is webpage of the dashboard. Due to the fact that we just add one gauge widget, the overall display is as displayed.

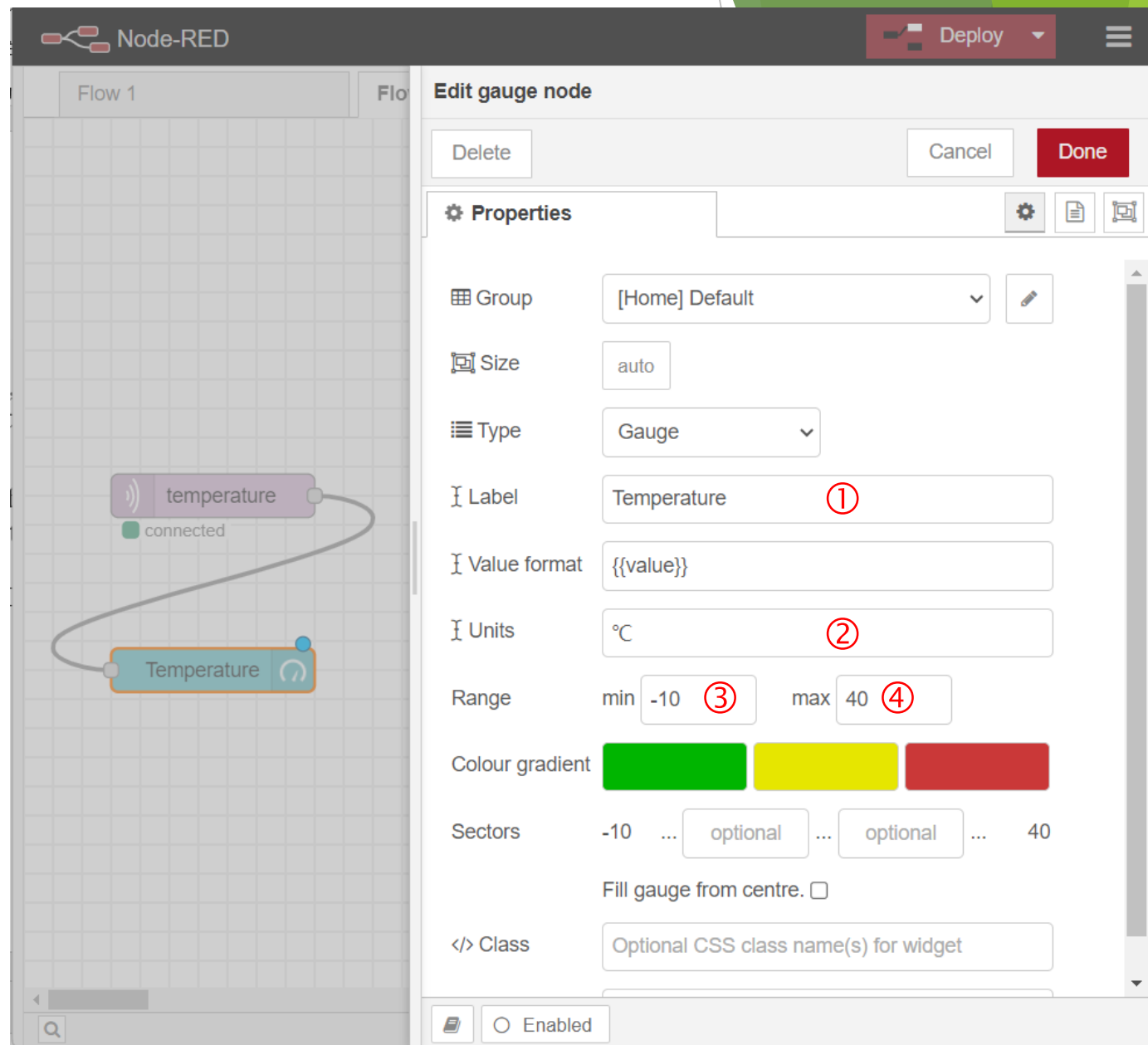


# 仪表盘

## Dashboard

- 下面，我们展示如何在仪表盘上展示数据，如通过温度感测器接收的温度值。实际上，我们可以对小部件进行更加精细的配置，例如更改部件名称，更改数量单位，显示范围等。

Now we demonstrate how to display data on the widgets of the dashboard, for example, displaying the temperature received from the DHT sensor. In fact, we can customize like the appearance of the widget in a more delicate way, such the alter the label of it, the unit for data and the range for displaying.

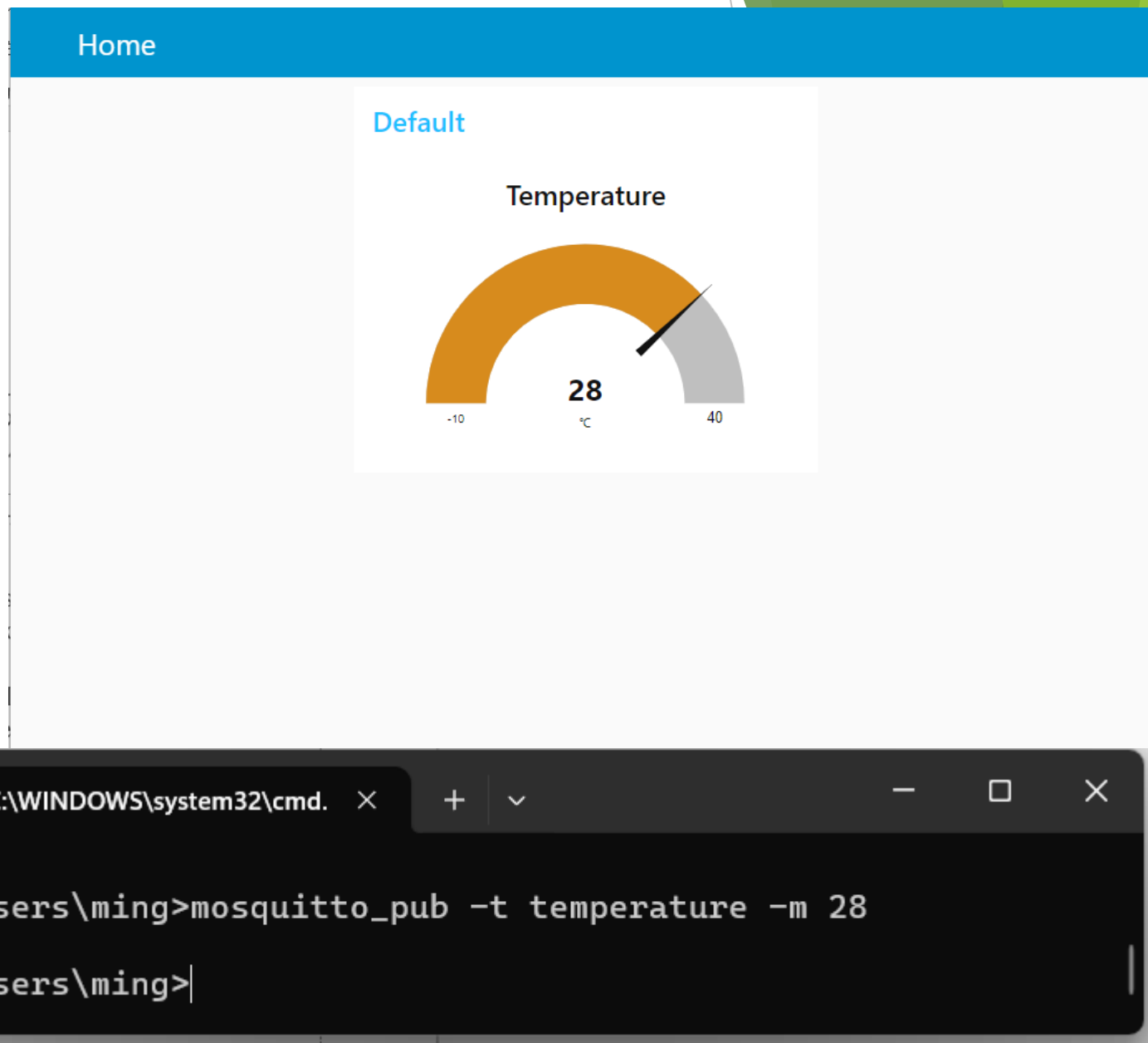


# 仪表盘

## Dashboard

- ▶ 如上张幻灯片所示，当添加mqtt in 节点并配置后，将两个节点链接并部署，当例行用mosquitto\_pub进行测试时，则显示的温度如右所示。

Just as indicated in the last slide, after adding the mqtt in node and configuring it, user can connect them then deploy the flow. The temperature is shown in the right figure when a usual test via mosquitto\_pub is carried out.



# 工程管理

## Project Management

- ▶ 由于Node-RED需要与特定的MQTT服务器绑定共同完成物联网应用，所以Node-RED最初设计时，工程的概念并不明显。实际上，对于HTTP服务，不同的业务逻辑用不同的路径进行隔离即可。针对MQTT服务，可以用主题（Topic）进行隔离。这对工业应用来讲可能问题不大，但对教学教育来讲，可能独立的工程更方便课堂教学。

Due to the fact that Node-RED should be coupled with MQTT proxy for IoT application, due to some reason, the concept of project is not that obviously. Actually, for HTTP service, different business logic are segregated via different locations. For MQTT service, it can be segregated via different topics. This might be not a big problem for industrial applications, however, for education and lecturing purposes, it is more convenient to manage everything via projects.

- ▶ 在Node-RED提供工程管理之前，Node-RED提供了相关功能进行流管理。

Before project management provided by Node-RED, it provides relevant functionalities for managing flows.

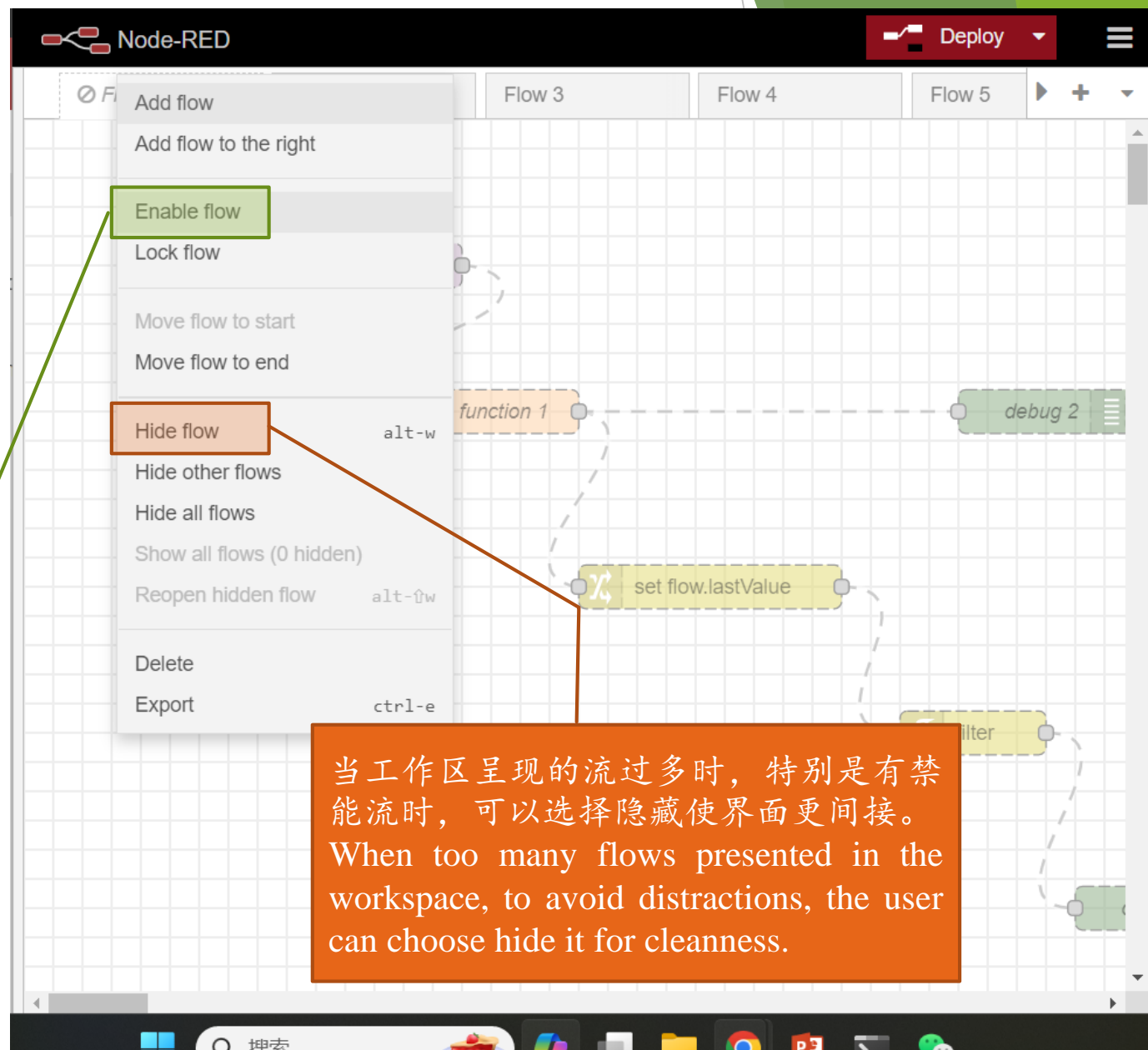
# 流管理

## Flow Management

► 右图展示常见的流管理功能。

The right figure shows the commonly-used flow management options.

可以选择特定流，比如流1、流2，进行使能或禁能。  
Select specific flows, for example, flow 1, flow 2, to enable or disable.



当工作区呈现的流过多时，特别是有禁能流时，可以选择隐藏使界面更间接。  
When too many flows presented in the workspace, to avoid distractions, the user can choose hide it for cleanness.



# 工程管理

## Project Management

- ▶ 虽然通过项目是管理流文件目前是试行性功能，但确实提供了现代观念上的项目管理。例如其支持Git存储库，即所有文件是完全版本控制的，允许开发人员使用熟悉的工作流程与其他人协作。同时，项目不是将流视为简单的文件，而是代表创建可再发行Node-RED应用程序所需的一切。

Although managing flows via project is current in preview mode, however, it indeed provides modern concept project management. For example, it supports Git repository, meaning all of the files are fully version controlled and allow developers to use familiar workflows to collaborate with others. At the same time, instead of treating flows as a simple pair of files, they represent everything you need to create a redistributable Node-RED application.

- ▶ 要启用项目功能，需要编辑settings.js文件。该设置文件会在Node-RED启动时被解析，导出生成JavaScript对象。具体地，在module.exports块中的编辑器风格属性中，添加相关选项，然后重新启动Node-RED。

To enable the project feature, you need edit the settings.js file. When Node-RED starts, it will parse the settings.js file into a JavaScript object. Specifically, add the following option for the editor theme property within the module.exports block and restart Node-RED.

# 工程管理

## Project Management

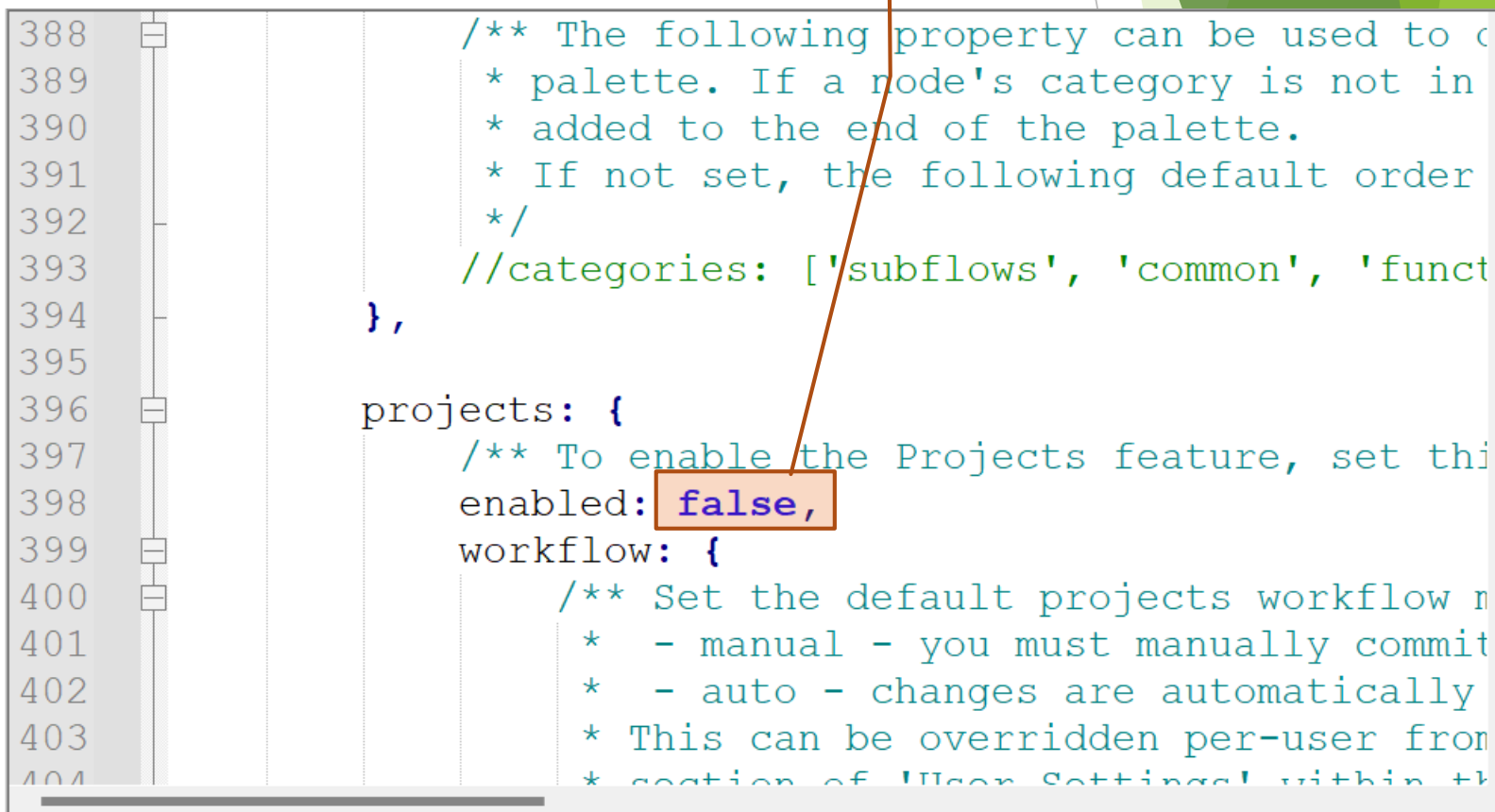
- 配置文件的路径是在当前用户的目录下，如下所示：

The path for the configuration file is as follows:

C:\Users\ming\.node-red\settings.js

- 要修改的内容如右图所示：

The intended modification is shown as right figure:



```
388 /** The following property can be used to c  
389 * palette. If a node's category is not in  
390 * added to the end of the palette.  
391 * If not set, the following default order  
392 */  
393 //categories: ['subflows', 'common', 'funct  
394 },  
395  
396 projects: {  
397   /** To enable the Projects feature, set thi  
398   enabled: false,  
399   workflow: {  
400     /** Set the default projects workflow n  
401     * - manual - you must manually commit  
402     * - auto - changes are automatically  
403     * This can be overridden per-user from  
404     * section of 'User Settings' within th
```

# 工程管理

## Project Management



- ▶ 当将工程使能并重新启动Node-RED后，在浏览器会首先看到新建或克隆工程界面：

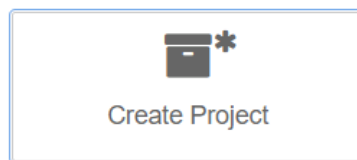
After enable the project and relaunch Node-RED, you will see from web browser the first create or clone project page:

Hello! We have introduced 'projects' to Node-RED.

This is a new way for you to manage your flow files and includes version control of your flows.

To get started you can create your first project or clone an existing project from a git repository.

If you are not sure, you can skip this for now. You will still be able to create your first project from the 'Projects' menu at any time.



Open existing project

Not right now

# 工程管理

## Project Management

- 接下来，用户需要填写与项目相关的信息，如项目名称，项目描述等。

In the following, the user need to fill information related to the project being created, such as name of the project and description of the project.



### Create your project

A project is maintained as a Git repository. It makes it much easier to share your flows with others and to collaborate on them.

You can create multiple projects and quickly switch between them from the editor.

To begin, your project needs a name and an optional description.

Project name

Must contain only A-Z 0-9 \_ -

Description

Optional

Back

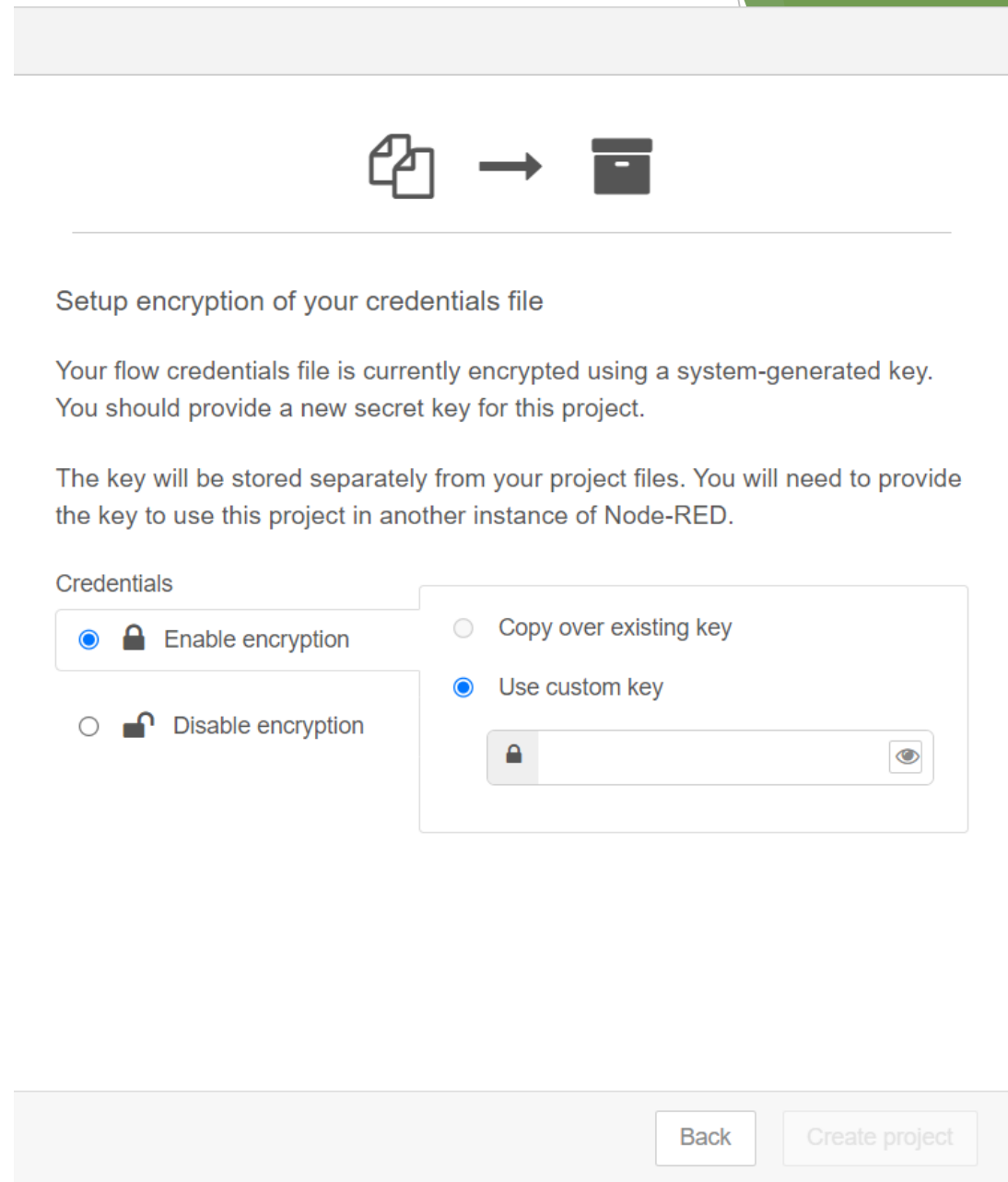
Next

# 工程管理

## Project Management

- ▶ 由于在Node-RED中，某些节点需要配置诸如密码、API 密钥和其他节点在流中所需的凭据等信息，这些敏感信息需要被安全地存储。因此，Node-RED会有凭据文件，用于加密 Node-RED 流配置文件中存储的凭据。显然，访问凭据文件需要密码。

In Node-RED, some nodes need to configure such as passwords, API keys, and other information required by nodes within a flow. So, an credential file is provided to securely store these sensitive information. Since credential file is used to encrypt credentials stored within Node-RED's flow configuration files, obviously you need password to access the credential file.



The screenshot shows a dialog box titled "Setup encryption of your credentials file". At the top, there is a visual representation of a file being moved into a folder. Below this, the text states: "Your flow credentials file is currently encrypted using a system-generated key. You should provide a new secret key for this project." It further explains: "The key will be stored separately from your project files. You will need to provide the key to use this project in another instance of Node-RED." The "Credentials" section contains two main options: "Enable encryption" (selected with a blue radio button) and "Disable encryption" (unselected). Under "Enable encryption", there are two sub-options: "Copy over existing key" (unselected) and "Use custom key" (selected). Below "Use custom key", there is a text input field with a lock icon on the left and an eye icon on the right, indicating it is a password field.

Setup encryption of your credentials file

Your flow credentials file is currently encrypted using a system-generated key. You should provide a new secret key for this project.

The key will be stored separately from your project files. You will need to provide the key to use this project in another instance of Node-RED.

Credentials

☒ Enable encryption

☐ Disable encryption

☐ Copy over existing key

☒ Use custom key

Back Create project

# 工程管理

## Project Management



- ▶ 当然，如果不涉及部署问题，可以选择不加密凭据文件。

If deployment is not a concern, users can choose not to encrypt the credential file.

- ▶ 当进行完上面所有工作之后，单击创建项目。若创建过程没有问题，则会出现创建成功提示框。

After all the previous steps, click the create project button. If everything goes smoothly, finally a dialog will prompt up indicating everything's succeeded.

You have successfully created your first project!

You can now continue to use Node-RED just as you always have.

The 'info' tab in the sidebar shows you what your current active project is. The button next to the name can be used to access the project settings view.

The 'history' tab in the sidebar can be used to view files that have changed in your project and to commit them. It shows you a complete history of your commits and allows you to push your changes to a remote repository.

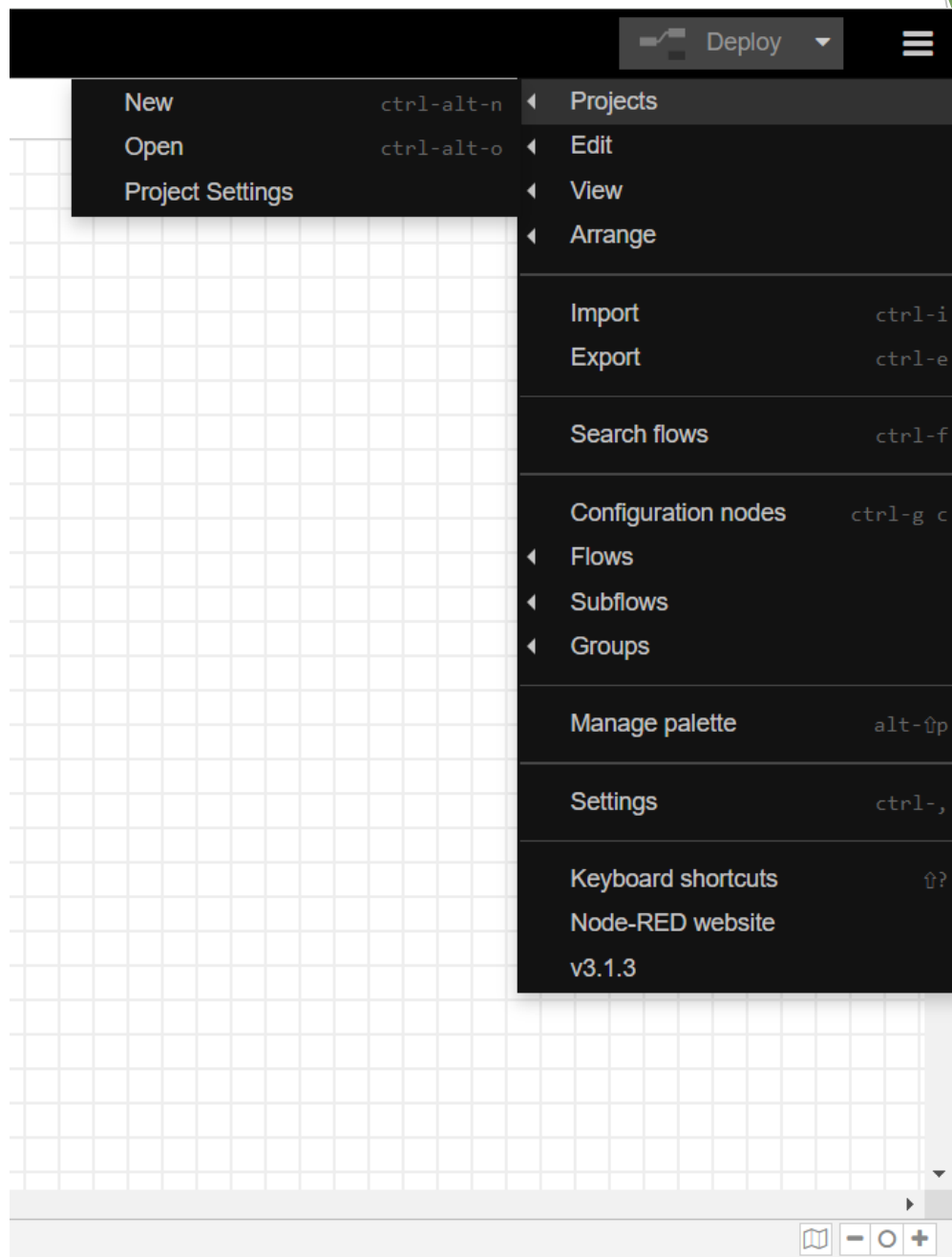
Done

# 工程管理

## Project Management

- ▶ 当启动项目管理后，与项目管理相关的一些菜单与功能项也会出现，用户可以根据之前使用经验探索使用。

Actually after enable the project management, the related menus and function icons will emerge accordingly. Users can based on their experience to try these functionalities.

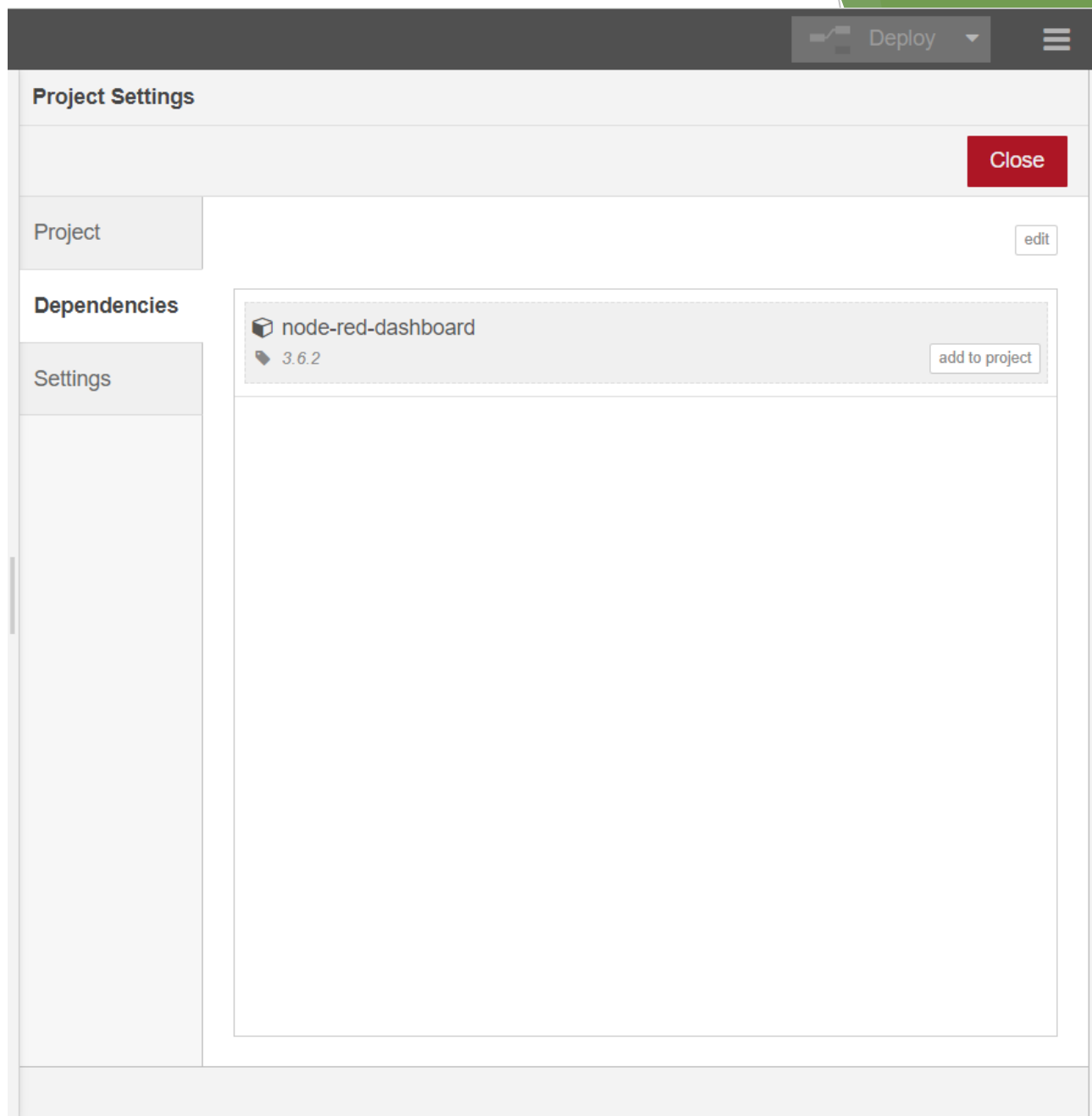


# 工程管理

## Project Management

- ▶ 尽管系统会自动感知当前项目的依赖，但如果涉及到部署，还是需要用户仔细检查项目依赖并进行添加。

Although the system can automatically detect the dependency for the current project, however, user should check the dependency list and add the items to project manually.





# 工程管理

## Project Management

- ▶ 同时，侧栏也会出现与工程相关的文件的修改与提交情况，这为在现代软件工程理念下进行项目实施与项目协作提供了很大的帮助。

At the same time, the sidebar will also show the history of file modification and submission. This provides the great convenience and assistance for implement and collaborate on projects under the modern software engineering disciplines.

- ▶ 注意，新建的工程的路径是在用户目录的.node-red文件夹下的projects目录里，如下例所示：

Note, the newly created project is resides in the projects folder, which under the .node-red directory of the user's home directory, as exemplified below:

C:\Users\ming\.node-red\projects

