



دانشگاه حکیم سبزواری

مستندات نهایی پروژه مقایسه و remote access در node red

استاد مربوطه:

دکتر مینا ملک زاده

گردآوری شده توسط:

پوریا آزاد و شایان نیکروش

خرداد

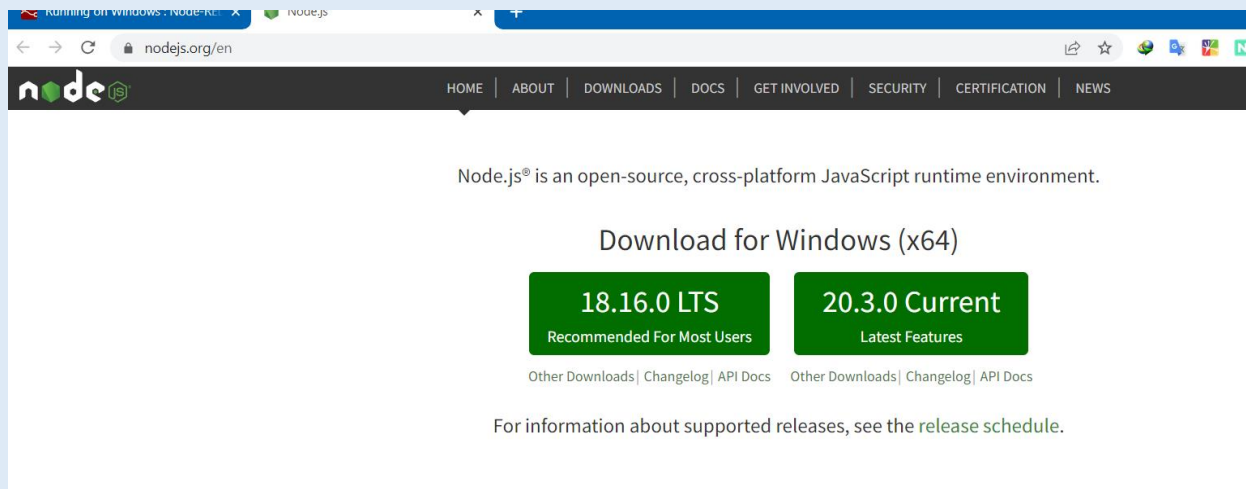
1402

فهرست

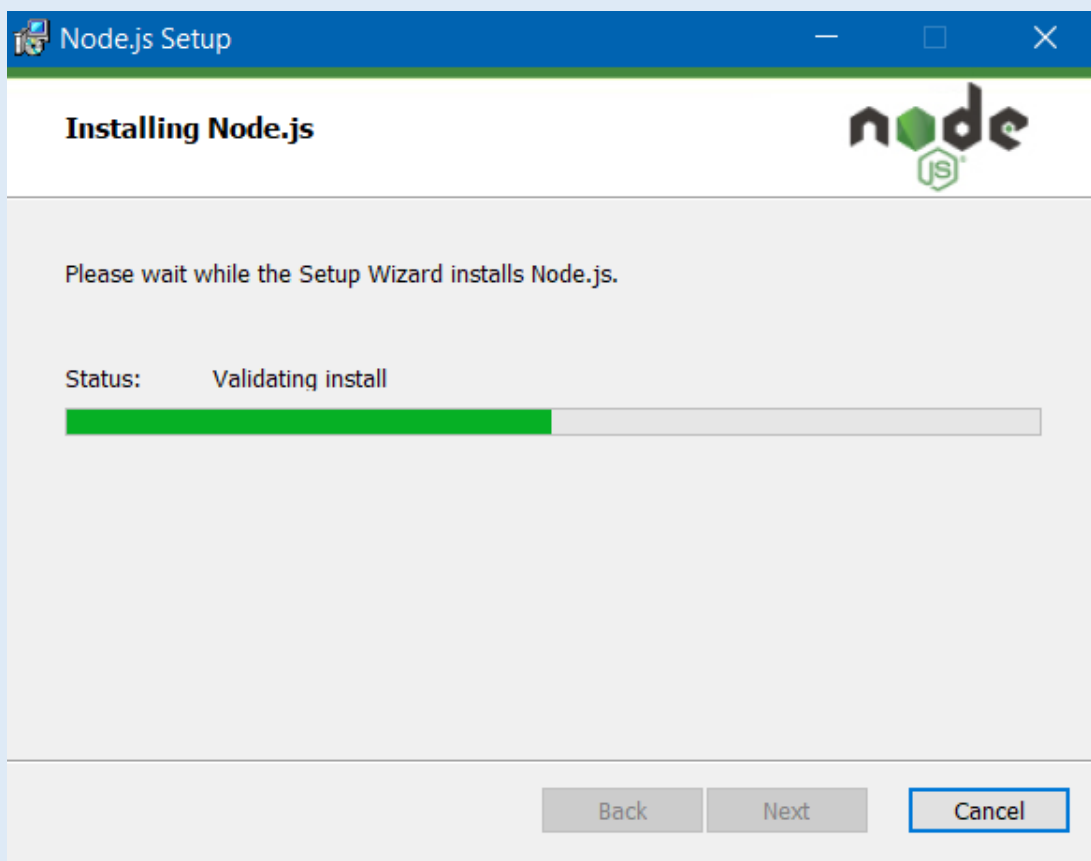
2	فهرست
3	نصب و راه اندازی node-red
6	connection-oriented و connection-less در TCP
10	connection-oriented و connection-less در UDP
12	Reliable in TCP
17	Reliable in UDP
20	پیاده سازی Acknowledge
24	تست سرعت TCP و UDP
28	Fragmentation for TCP
37	Remote

نصب و راه اندازی *node-red*

1- نصب Node Js



ابتدا باید نسخه Node Js LTS رو از سایت رسمی دانلود کنیم و نصب کنیم



اگه به درستی انجام شده بود با کامند های زیر در power shell باید ورژن node js رو بده

```
PS E:\Daneshgah\net\project> node --version; npm --version
v18.16.0
9.5.1
PS E:\Daneshgah\net\project>
```

حالا با وارد کردن کامند زیر میتونیم node-red رو نصب کنیم

```
9.5.1
PS E:\Daneshgah\net\project> npm install -g --unsafe-perm node-red

added 1 package, and changed 292 packages in 1m

41 packages are looking for funding
  run `npm fund` for details
npm notice
```

در آخر با وارد کردن node-red ، node-red بر روی لوکال هاست اجرا میشه

```
node-red
PS E:\Daneshgah\net\project> node-red
10 Jun 15:01:46 - [info]

Welcome to Node-RED
=====

10 Jun 15:01:46 - [info] Node-RED version: v3.0.2
10 Jun 15:01:46 - [info] Node.js version: v18.16.0
10 Jun 15:01:46 - [info] Windows_NT 10.0.19045 x64 LE
10 Jun 15:01:48 - [info] Loading palette nodes
10 Jun 15:01:48 - [info] Settings file : C:\Users\sadeg\.node-red\settings.js
10 Jun 15:01:48 - [info] Context store : 'default' [module=memory]
10 Jun 15:01:48 - [info] User directory : C:\Users\sadeg\.node-red
10 Jun 15:01:48 - [warn] Projects disabled : editorTheme.projects.enabled=false
10 Jun 15:01:48 - [info] Flows file : C:\Users\sadeg\.node-red\flows.json
10 Jun 15:01:48 - [info] Server now running at http://127.0.0.1:1880/
10 Jun 15:01:48 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
```

نکته) تاوقتی که میخواهیم با node-red کار کنیم نباید power shell یا CMD رو ببندیم

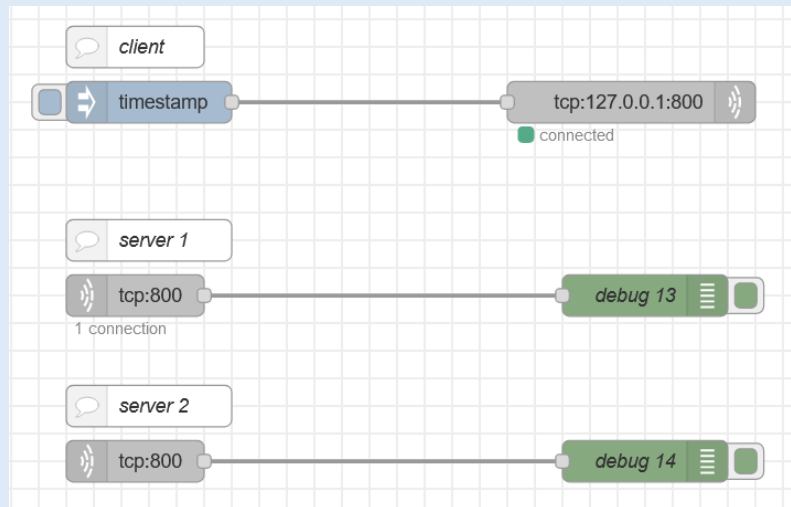
TCP connection-less و connection-oriented در

Connection-oriented و Connection-less دو روش مختلف برای ارسال داده‌ها در شبکه‌های کامپیوتری هستند.

در Connection-oriented، ابتدا یک ارتباط بین دو دستگاه برقرار می‌شود. این ارتباط توسط یک پروتکل مانند TCP مدیریت می‌شود و اطمینان حاصل می‌شود که داده‌ها به ترتیب درستی و بدون از دست دادن به مقصد می‌رسند. در این روش، قبل از ارسال داده‌ها، یک دستور handshake بین دو دستگاه انجام می‌شود تا اطمینان حاصل شود که ارتباط بین دو دستگاه برقرار شده است. این روش برای ارسال داده‌های حساس و مهم مانند پرداخت‌های آنلاین و انتقال فایل‌های بزرگ به کار می‌رود.

در Connection-less، هیچ ارتباطی بین دو دستگاه برقرار نمی‌شود و داده‌ها بدون هیچ گونه تضمینی برای رسیدن به مقصد ارسال می‌شوند. در این روش، داده‌ها به صورت پیام‌های مستقل ارسال می‌شوند و هیچ گونه تضمینی برای رسیدن به مقصد وجود ندارد. این روش برای ارسال داده‌هایی که نیاز به سرعت و کارایی دارند مانند تصاویر و ویدئوها به کار می‌رود.

برای مقایسه نحوه کار connection-oriented و connection-less ابتدا چارت زیر را برای tcp پیاده سازی میکنیم



The 'Edit inject node' dialog box is shown. It has a 'Delete' button, a 'Cancel' button, and a 'Done' button. Below these is a 'Properties' tab. The 'Name' field is empty. The 'msg. payload' field is set to 'timestamp'. The 'msg. topic' field is set to 'a_z'. There is an 'add' button and an 'inject now' button. The 'Inject once after 0.1 seconds, then' checkbox is unchecked. The 'Repeat' dropdown is set to 'none'. The 'Enabled' checkbox is checked.

تنظیمات inject node

Edit tcp out node

Delete Cancel Done

Properties

Type Connect to port 800

at host 127.0.0.1

☐ Enable secure (SSL/TLS) connection

☐ Close connection after each message is sent?

☐ Decode Base64 message?

Name Name

تنظیمات tcp out

Edit tcp in node

Delete Cancel Done

Properties

Type Listen on port 800

☐ Enable secure (SSL/TLS) connection

Output stream of Buffer payload(s)

Topic Topic

Name Name

تنظیمات tcp in

Edit debug node

Delete

Cancel

Done

⚙️ Properties

⚙️ 📄 🖨️

Output

▼ msg. payload

🔗 To

☒ debug window

☐ system console

☐ node status (32 characters)

🏷️ Name

debug 13

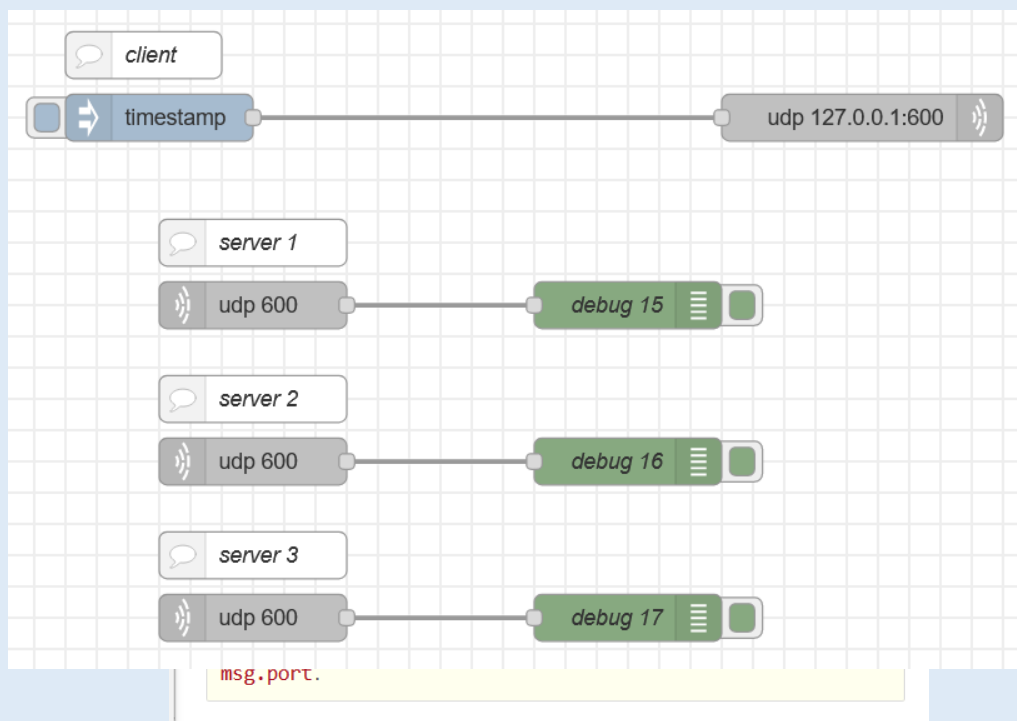
تنظیمات debug node

حال بعد از ارسال یک timestamp میبینیم که فقط به یکی از server 1 ها متصل شده است

9 | Page

UDP در connection-less و connection-oriented

برای مقایسه نحوه کار connection-less و connection-oriented در udp چارت زیر را پیاده سازی میکنیم



تنظیمات udp out

Edit udp in node

Delete
Cancel
Done

⚙️ Properties

Listen for
udp messages

on Port
600
using
ipv4

Output
a Buffer

Name
Name

Tip: Make sure your firewall will allow the data in.

Ports already in use: 76,600,9910

تنظیمات udp in

دو udp in دیگر به شکل بالا پیاده سازی میکنیم

در اینجا میبینیم که client زمانی که یک timestamp برای پورتهی خاص میفرستیم سه سرور جواب ما را میدهند و ارتباط چک نشده است به عبارتی هر سروری که بتواند جواب ما را میدهد اگر هم چند سرور بتوانند همه شان جواب ما را میدهد

11 | Page

Reliable in TCP

مفهوم Reliable به معنای قابل اطمینان بودن یک پروتکل یا خدمت است. در شبکه‌های کامپیوتری، reliable بودن یک پروتکل به معنای تضمین این است که داده‌ها به صورت کامل، در ترتیب درست و بدون خطا از یک محل به مقصد منتقل شوند.

برای داشتن یک پروتکل reliable، به چند ویژگی اساسی نیاز است:

- 1- تأیید دریافت: هنگامی که یک بسته داده ارسال می‌شود، باید تأیید دریافت آن بسته از سمت مقصد دریافت شود. این اطمینان می‌دهد که بسته داده به درستی به مقصد رسیده است.
 - 2- دریافت داده‌ها به ترتیب درست: اگر بسته‌های داده به ترتیب درست دریافت نشوند، ممکن است باعث ایجاد خطا و اشتباه در داده‌های دریافت شده شود.
 - 3- بازبینی خطا: در صورت بروز خطا در ارسال و دریافت داده‌ها، باید قابلیت بازبینی و اصلاح خطا وجود داشته باشد تا داده‌های درست دریافت شوند.
 - 4- کنترل جریان: باید از طریق کنترل جریان اطمینان حاصل شود که مقصد قادر به پردازش و دریافت داده‌ها است.
 - 5- تحمل از دست دادن بسته‌ها: در شبکه‌های پرترافیک، ممکن است بسته‌ها گم شوند. پروتکل باید قابلیت تحمل از دست دادن بسته‌ها را داشته باشد و بازسازی داده‌ها را به درستی انجام دهد.
- با توجه به این ویژگی‌ها، پروتکل TCP یک پروتکل reliable است که برای ارسال داده‌ها در شبکه استفاده می‌شود. این پروتکل اطمینان حاصل می‌کند که داده‌ها به ترتیب درستی دریافت می‌شوند و هر بسته داده‌ای که ارسال می‌شود دریافت شده است.

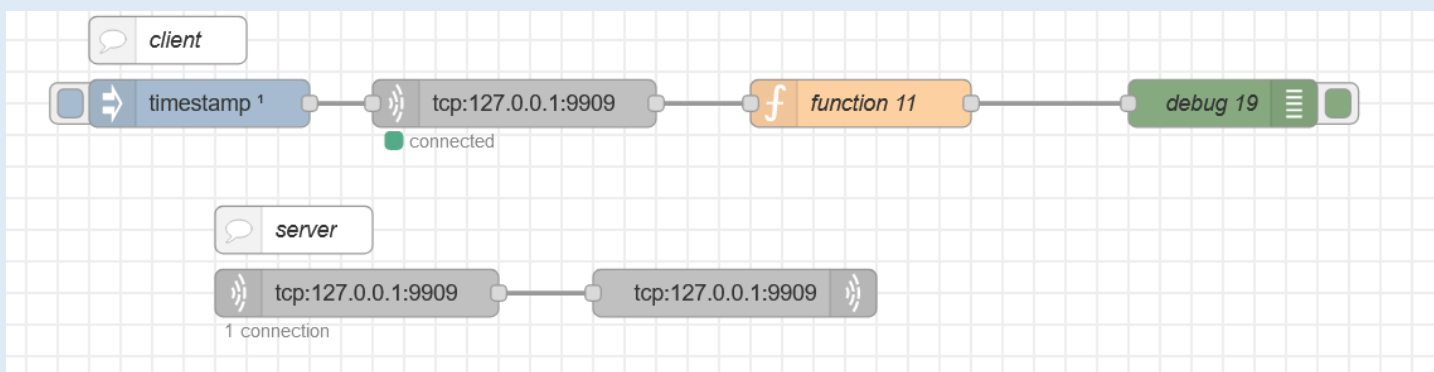
UDP یک پروتکل ناقص (unreliable) است که برای انتقال داده ها در شبکه استفاده می شود. در UDP، داده ها به دو صورت بدون تأیید دریافت (unacknowledged) و بدون ترتیب (unordered) ارسال می شوند، به عبارت دیگر، این پروتکل به صورت ناقص و سریع عمل می کند.

از این رو، در UDP، برخلاف TCP، هیچ گونه مکانیزم تأیید دریافت (ACK) یا بازبینی خطا (error checking) وجود ندارد. به همین دلیل، UDP به عنوان یک پروتکل unreliable شناخته می شود.

با این حال، در برخی از کاربردهای شبکه، ممکن است که نیاز به سرعت بالا و حجم کم داده ها باشد و مهم نباشد که بعضی از داده ها گم شوند. به عنوان مثال، در برخی از بازی های آنلاین، قابلیت تحمل از دست دادن برخی از بسته های داده به عنوان یک ویژگی مطلوب شناخته می شود.

در نتیجه، برای کاربردهایی که نیاز به سرعت بالا و حجم کم داده ها دارند و از اهمیت کمتری برخوردارند، می توان از UDP به جای TCP استفاده کرد. اما برای کاربردهایی که دقت و اطمینان از رسیدن داده ها بسیار مهم هستند، بهتر است از پروتکل reliable مانند TCP استفاده شود.

برای پیاده سازی ابتدا چارت زیر را برای tcp پیاده میکنیم



Edit inject node

Delete Cancel Done

Properties

Name

msg. payload = timestamp

msg. topic = a_z

+ add inject now

☒ Inject once after 1 seconds, then

Repeat interval

every 1 seconds

تنظیمات inject

Edit tcp request node

Delete Cancel Done

Properties

Server 127.0.0.1 port 9909

☐ Enable secure (SSL/TLS) connection

Return String

Close never - keep connection open

delimited by (optional)

☐ re-attach delimiter

Name

تنظیمات tcp request

Edit tcp in node

Delete

Cancel

Done

⚙️ Properties

⚙️ 📄 🖼️

Type

Listen on

port

9909

☐ Enable secure (SSL/TLS) connection

➡️ Output

stream of

String

payload(s)

delimited by

(optional)

☐ re-attach delimiter

📄 Topic

Topic

🔖 Name

Name

تنظیمات tcp in

Edit tcp out node

Delete

Cancel

Done

⚙️ Properties

⚙️ 📄 🖼️

Type

Reply to TCP

☐ Decode Base64 message?

🔖 Name

Name

تنظیمات tcp out

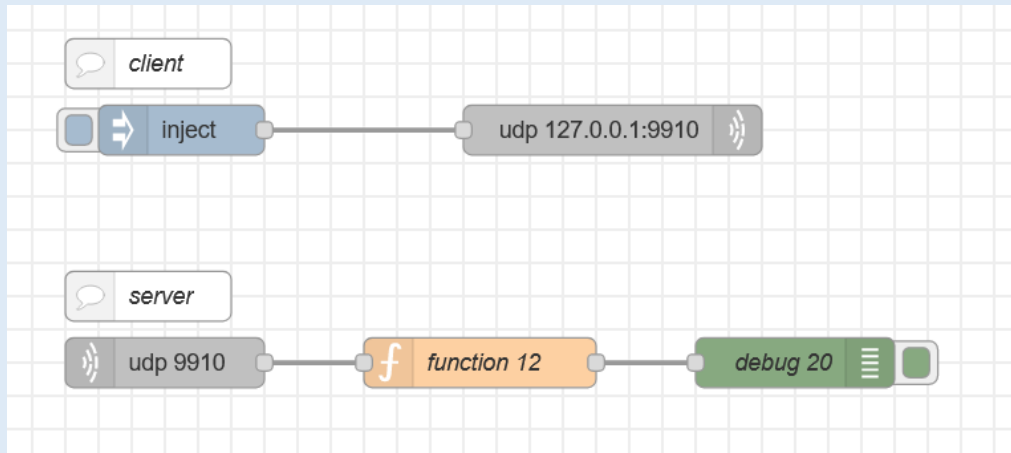
15 | Page

<div> <div>all nodes</div> <div>all</div> </div>	
6/25/2023, 7:52:34 PM	node: debug 19 msg.payload : number 1
6/25/2023, 7:52:35 PM	node: debug 19 msg.payload : number 2
6/25/2023, 7:52:36 PM	node: debug 19 msg.payload : number 3
6/25/2023, 7:52:37 PM	node: debug 19 msg.payload : number 4
6/25/2023, 7:52:38 PM	node: debug 19 msg.payload : number 5
6/25/2023, 7:52:39 PM	node: debug 19 msg.payload : number 6

در خروجی شماره پکت ها را بترتیب میبینیم

Reliable in UDP

در مرحله بعد چارت udp را بصورت زیر پیاده سازی میکنیم



Edit udp out node

Delete Cancel Done

Properties

Send a to port

Address

☐ Decode Base64 encoded payload?

Name

Tip: leave address and port blank if you want to set using **msg.ip** and **msg.port**.

تنظیمات udp out

Edit udp in node

Delete Cancel Done

Properties

Listen for: udp messages

on Port: 9910 using: ipv4

Output: a String

Name: Name

Tip: Make sure your firewall will allow the data in.

Ports already in use: 76,600,9910

تنظیمات udp in

Edit function node

Delete Cancel Done

Properties

Name: function 12

Setup On Start On Message On Stop

```

1 context.count = context.count || 0;
2 context.count++;
3
4 return { payload: context.count};

```

تنظیمات function

6/25/2023, 8:00:54 PM	node: debug 20
msg.payload : number	
2	
6/25/2023, 8:00:55 PM	node: debug 20
msg.payload : number	
3	
6/25/2023, 8:00:56 PM	node: debug 20
msg.payload : number	
4	
6/25/2023, 8:00:57 PM	node: debug 20
msg.payload : number	
5	
6/25/2023, 8:00:58 PM	node: debug 20
msg.payload : number	
6	
6/25/2023, 8:00:59 PM	node: debug 20
msg.payload : number	
7	

به ظاهر که دو پروتکل tcp و udp به خوبی عمل میکنند و هیچ پکت گمشده ای نداریم چون در لوکال هاست خودمان این را اجرا میکنیم اما اگر در محیط اینترنت این را اجرا کنیم چون پروتکل udp connection-oriented نیست پس ممکن است پکت هایی به مقصد نرسند و در نتیجه tcp reliable هست

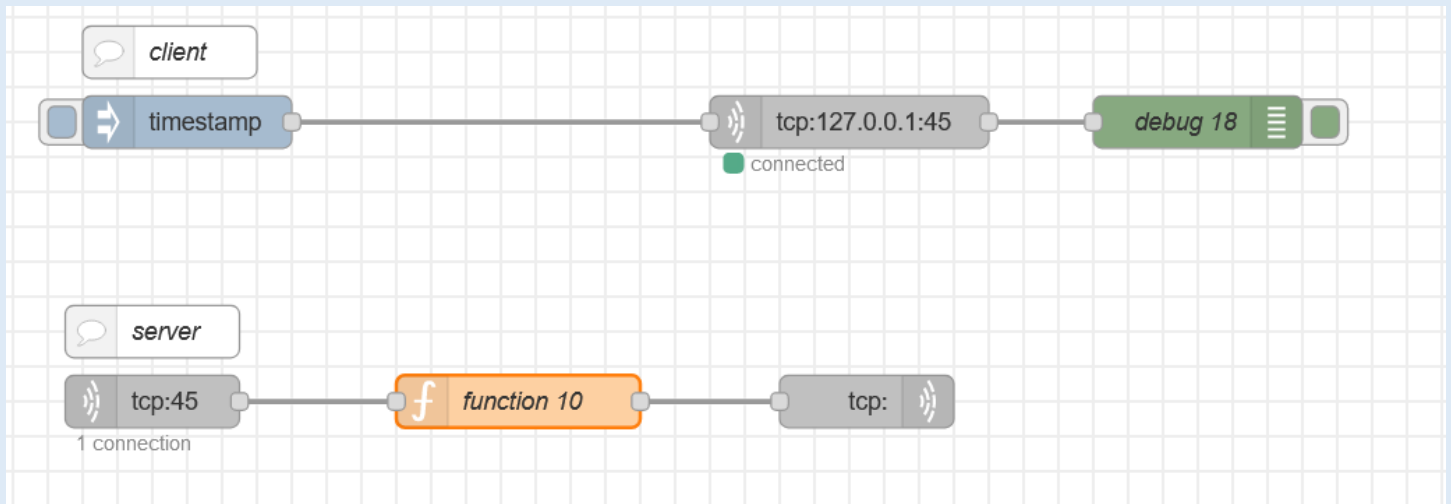
پیاده سازی Acknowledge

در پروتکل TCP، برای اطمینان از دریافت درست و کامل داده‌ها، از مکانیزم تأیید دریافت (Acknowledgment) یا به اختصار ACK) استفاده می‌شود. هر بسته‌ای که ارسال می‌شود، باید توسط دستگاه دریافت کننده تأیید دریافت شود. این تأیید دریافت، به فرستنده برای ادامه ارسال داده‌ها، اطمینان می‌دهد که بسته‌های قبلی به درستی دریافت شده‌اند و در صورت بروز خطا در ارسال بسته‌ها، فرستنده می‌تواند بسته‌های از دست رفته را مجدداً ارسال کند.

مکانیزم ACK در پروتکل TCP به این صورت عمل می‌کند که هر بسته‌ای که دریافت می‌شود، با شماره ترتیبی (Sequence number) بسته‌ها که در بسته‌ها قرار دارد، به فرستنده پاسخ داده می‌شود. در واقع، در هر بسته ACK که به فرستنده ارسال می‌شود، شماره ترتیبی آخرین بسته‌ای که به درستی دریافت شده، قرار دارد. به عنوان مثال، اگر دستگاه دریافت کننده بسته‌های با شماره ترتیبی 1، 2 و 3 را دریافت کرده باشد، باید یک ACK با شماره ترتیبی 3 به فرستنده ارسال کند تا فرستنده بداند که بسته‌های 1 تا 3 درست و کامل دریافت شده‌اند و می‌تواند بسته‌های بعدی را ارسال کند.

به طور کلی، مکانیزم ACK در پروتکل TCP برای اطمینان از دریافت درست و کامل داده‌ها، و همچنین کنترل خطا و بازیابی خطا در ارسال داده‌ها، بسیار مهم است.

برای پیاده سازی acknowledge بدین صورت عمل میکنیم



Edit tcp request node

Delete Cancel Done

Properties

Server 127.0.0.1 port 45

☐ Enable secure (SSL/TLS) connection

Return String

Close never - keep connection open

delimited by (optional)

☐ re-attach delimiter

Name Name

تنظیمات tcp request

Edit tcp in node

Delete Cancel Done

Properties

Type Listen on port 45

☐ Enable secure (SSL/TLS) connection

Output stream of String payload(s)

delimited by (optional)

☐ re-attach delimiter

Topic Topic

Name Name

تنظیمات tcp in

Edit function node

Delete Cancel Done

Properties

Name function 10

Setup On Start On Message On Stop

```

1 msg.payload = "I am the server and I got " + msg.payload ;
2 return msg;

```

تنظیمات function

Edit tcp out node

Delete

Cancel

Done

⚙️ Properties

⚙️ 📄 🖼️

⦿ Type

Reply to TCP ▾

☐

Decode Base64 message?

🏷️ Name

Name

تنظیمات tcp out

all nodes ▾

all ▾

6/25/2023, 8:15:24 PM node: debug 18

msg.payload : string[39]

"I am the server and I got 1687711524802"

6/25/2023, 8:15:29 PM node: debug 18

msg.payload : string[39]

"I am the server and I got 1687711529187"

6/25/2023, 8:15:29 PM node: debug 18

msg.payload : string[39]

"I am the server and I got 1687711529574"

6/25/2023, 8:15:29 PM node: debug 18

msg.payload : string[39]

"I am the server and I got 1687711529921"

و بدین صورت بعد از هر دریافت پیام از طرف سرور یک تایید به همراه timestamp دریافتی را میگیریم

تست سرعت TCP و UDP

علت تفاوت سرعت ارسال داده در پروتکل TCP و UDP، به دو عامل اصلی بستگی دارد:

1- مکانیزم تأیید دریافت:

در پروتکل TCP، برای هر بسته‌ای که ارسال می‌شود، باید تأیید دریافت آن بسته نیز به فرستنده ارسال شود. این مکانیزم باعث افزایش تعداد بارهای ارتباطی بین دو دستگاه و کاهش سرعت ارسال داده می‌شود. در مقابل، در پروتکل UDP، بدون تأیید دریافت، بسته‌ها به صورت پیوسته و بدون انتظار برای تأیید دریافت بسته‌ها به سمت مقصد ارسال می‌شوند و بنابراین سرعت بیشتری دارند.

2- کنترل خطا:

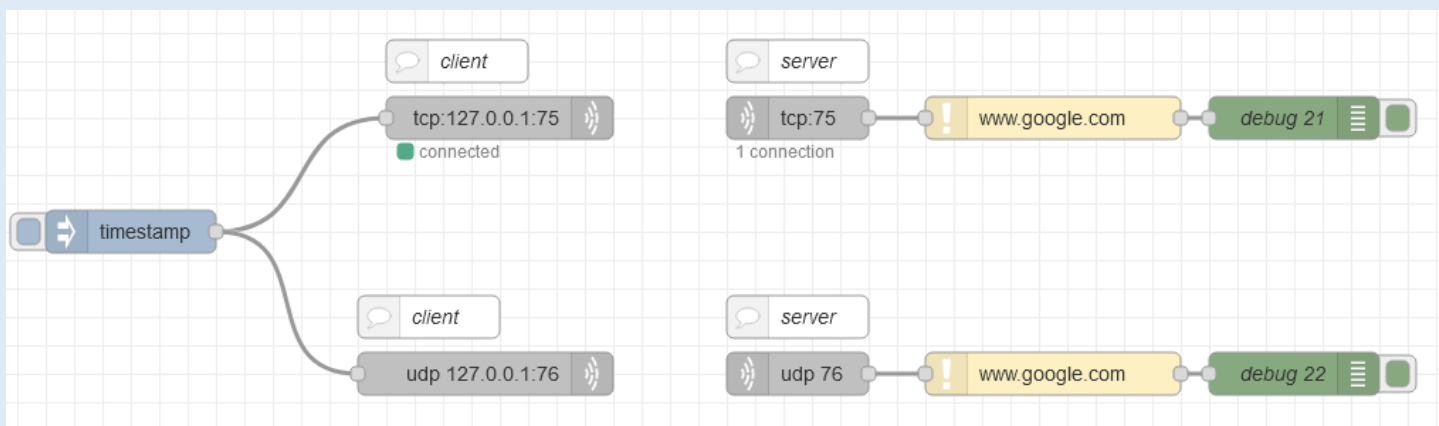
در پروتکل TCP، برای کنترل خطا در ارسال داده، از یک مکانیزم به نام روش بازیابی خطا (error recovery) استفاده می‌شود. در این روش، اگر بسته‌ای در ارتباط از دست برود، فرستنده مجدداً آن بسته را ارسال می‌کند. این کار باعث افزایش تعداد بارهای ارتباطی می‌شود و در نتیجه سرعت ارسال داده کاهش پیدا می‌کند. در پروتکل UDP، به دلیل عدم وجود مکانیزم بازیابی خطا، در صورتی که بسته‌ای در ارتباط از دست برود یا به صورت ناقص دریافت شود، این مشکل را نمی‌توان به صورت خودکار برطرف کرد و باید در سطح برنامه‌نویسی این مشکل را مدیریت کرد. در نتیجه، پروتکل UDP برای داده‌هایی که سرعت و کیفیت در انتقال آن‌ها مهم است، مانند داده‌های صوتی و تصویری، مناسب است.

برای تست سرعت این دو پروتکل کافی است که از adv ping برای اینکار استفاده نماییم

ابتدا با استفاده از دستور زیر در یک Palette نصب میکنیم

```
npm install node-red-contrib-advanced-ping
```

حال کافیت چارت زیر را پیاده سازی کنیم



Edit tcp out node

Delete Cancel Done

Properties

Type Connect to port 75

at host 127.0.0.1

☐ Enable secure (SSL/TLS) connection

☐ Close connection after each message is sent?

☐ Decode Base64 message?

Name Name

تنظیمات tcp out

Edit tcp in node

Delete Cancel Done

Properties

Type Listen on port 75

☐ Enable secure (SSL/TLS) connection

Output stream of Buffer payload(s)

Topic 127.0.0.1

Name Name

تنظیمات tcp in

Edit adv ping node

Delete Cancel Done

Properties

Host www.google.com

Name Name

تنظیمات adv ping

Edit udp out node

Delete Cancel Done

Properties

✉ Send a to port

📄 Address

☐ Decode Base64 encoded payload?

🏷 Name

Tip: leave address and port blank if you want to set using **msg.ip** and **msg.port**.

تنظیمات udp out

Edit udp in node

Delete Cancel Done

Properties

➡ Listen for

➡ on Port using

➡ Output

🏷 Name

Tip: Make sure your firewall will allow the data in.

Ports already in use: 76,600,9910

تنظیمات udp in

برای تنظیمات adv ping هم مثل قبلی عمل میکنیم

```
6/25/2023, 8:31:02 PM node: debug 22  
www.google.com : msg.payload : number  
238
```

```
6/25/2023, 8:31:02 PM node: debug 21  
www.google.com : msg.payload : number  
261
```

حال میبینیم که پینگ پایینی که مربوط به tcp ماست کندتر در آمد در صورتی که بالایی که مربوط به udp است کمتر در آمد و در نتیجه udp سریعتر است

Fragmentation for TCP

فرگمنتاسیون یک فرایند در شبکه‌های کامپیوتری است که در آن داده‌ها به بخش‌های کوچکتر، به نام فرگمنت‌ها، تقسیم می‌شوند تا بتوانند بهتر در شبکه منتقل شوند.

در پروتکل TCP، داده‌های ارسالی به بخش‌های کوچکتر تقسیم شده و به صورت جداگانه از هم به سمت مقصد ارسال می‌شوند. این فرایند به دلیل محدودیت اندازه بسته‌های شبکه و برخی محدودیت‌های فیزیکی شبکه، مانند اندازه حداکثر فریم در لایه دو، انجام می‌شود.

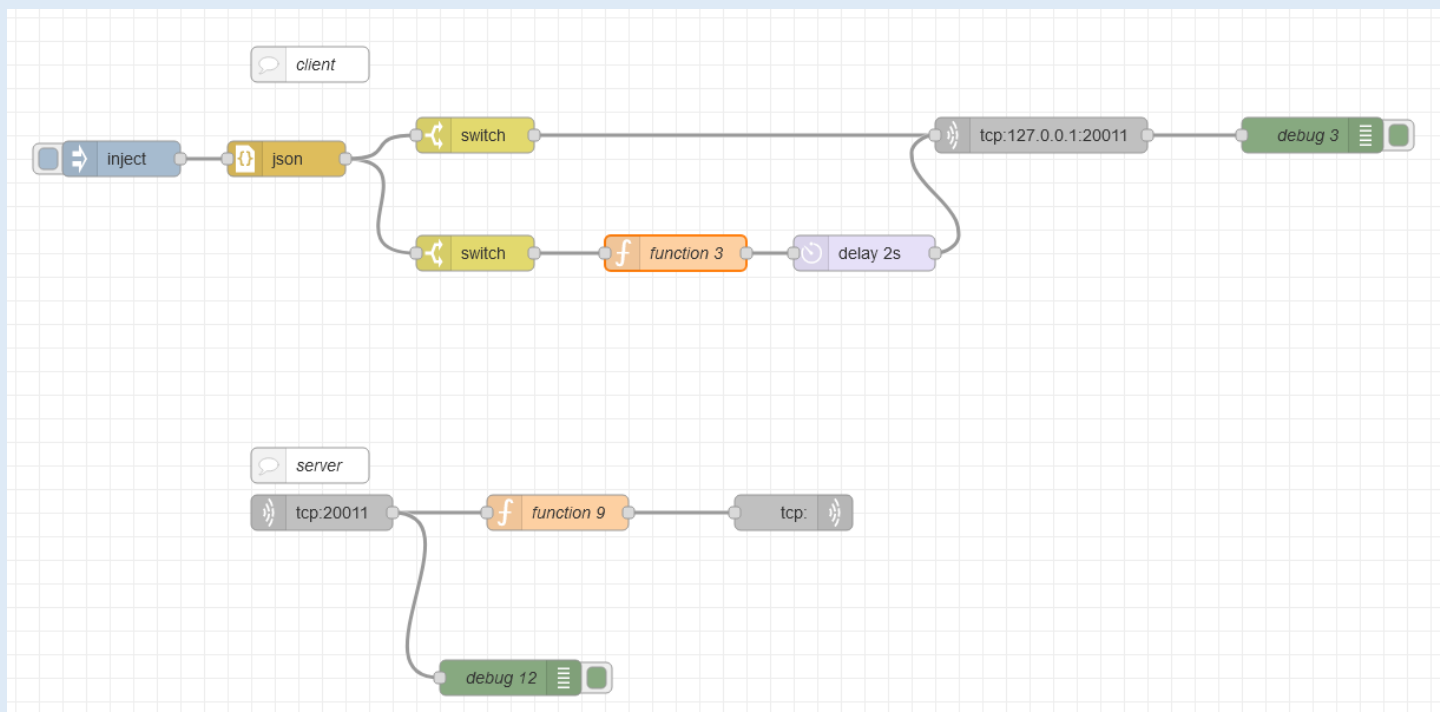
با استفاده از فرگمنتاسیون، بسته‌های بزرگ تر، مانند فایل‌های بزرگ، می‌توانند به چند بخش کوچکتر تقسیم شوند و به صورت جداگانه از هم ارسال شوند. همچنین، به کمک فرگمنتاسیون می‌توان بسته‌هایی که در شبکه از بین می‌روند یا به صورت ناقص دریافت می‌شوند، دوباره ساخته و از سمت مقصد دریافت کرد.

فرایند فرگمنتاسیون در TCP به این صورت انجام می‌شود که ابتدا بسته‌هایی به اندازه مشخصی به نام Maximum Segment Size (MSS) تقسیم می‌شوند. MSS بیانگر حداکثر اندازه داده‌ای است که می‌توان در یک بسته TCP ارسال کرد، بدون این که بسته‌ای بزرگ‌تر از حد مجاز شبکه ایجاد شود. سپس، هر یک از بسته‌های ایجاد شده به یک Header جدید از پروتکل TCP اضافه می‌شود که شامل اطلاعاتی مانند شماره دنباله بایت، شماره پنجره گیرنده و شماره تأیید است. در نهایت، هر بسته با یک شماره دنباله بایت و شماره تأیید مشخص به سمت مقصد ارسال می‌شود.

در سمت مقصد، بسته‌های دریافت شده توسط پروتکل TCP مجدداً ترکیب می‌شوند تا داده اصلی بازسازی شود. این کار با استفاده از شماره دنباله بایت و شماره تأیید پروتکل TCP انجام می‌شود. در صورتی که یک بسته در شبکه از بین رود یا به صورت ناقص دریافت شود، پروتکل TCP تلاش می‌کند برای دوباره ارسال آن بسته تا داده‌ها به طور کامل و سالم دریافت شوند.

برای پیاده سازی فرگمنتیشن tcp در نود رد میتوانیم بدین صورت عمل کنیم که ابتدا با استفاده از قطعه سوییچ اندازه قطعه را مقایسه میکنیم اگر درست بود که آن را مستقیم ارسال میکنیم و اگر از یک حدی بیشتر بود آن را به

تابعی میفرستیم که در آن جا اندازه پکت رو به بخش های کوچک تری تقسیم میکند و آن را به خروجی ارسال میکند
چارت زیر نحوه پیاده سازی آن است



Edit inject node

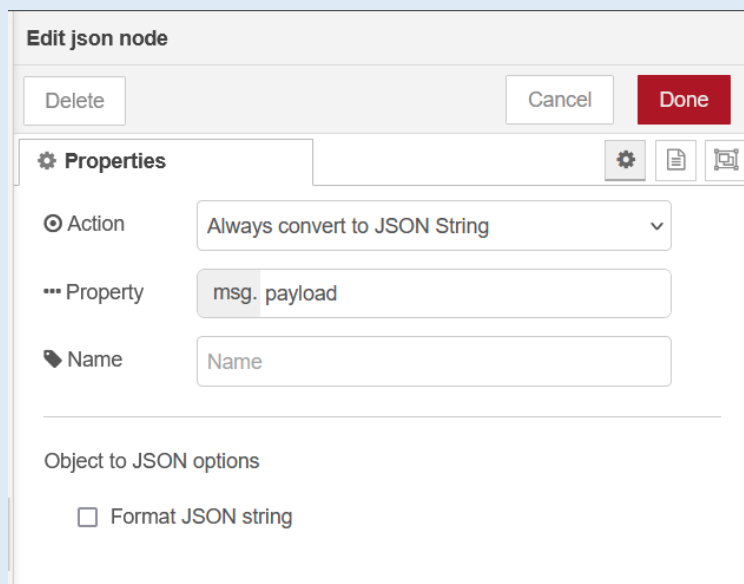
Delete
Cancel
Done

Properties

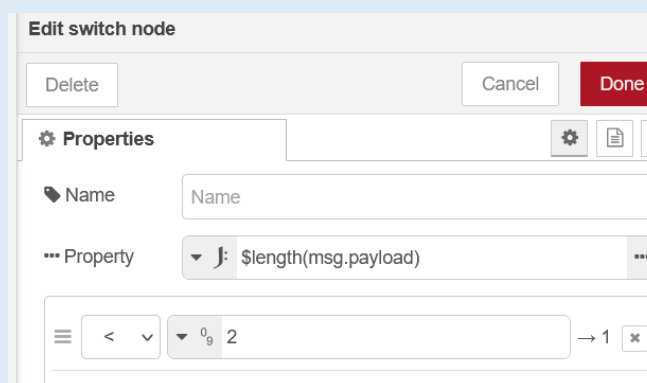
Name
Name

msg. payload
=
{"squadName":"Super hero squa

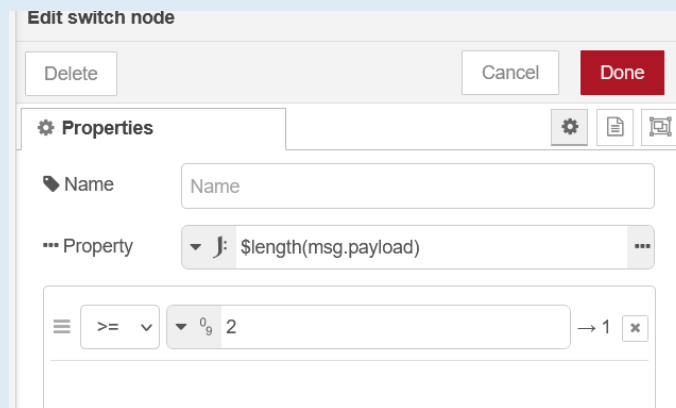
تنظیمات inject که در payload آن یک فایل json نوشته ایم



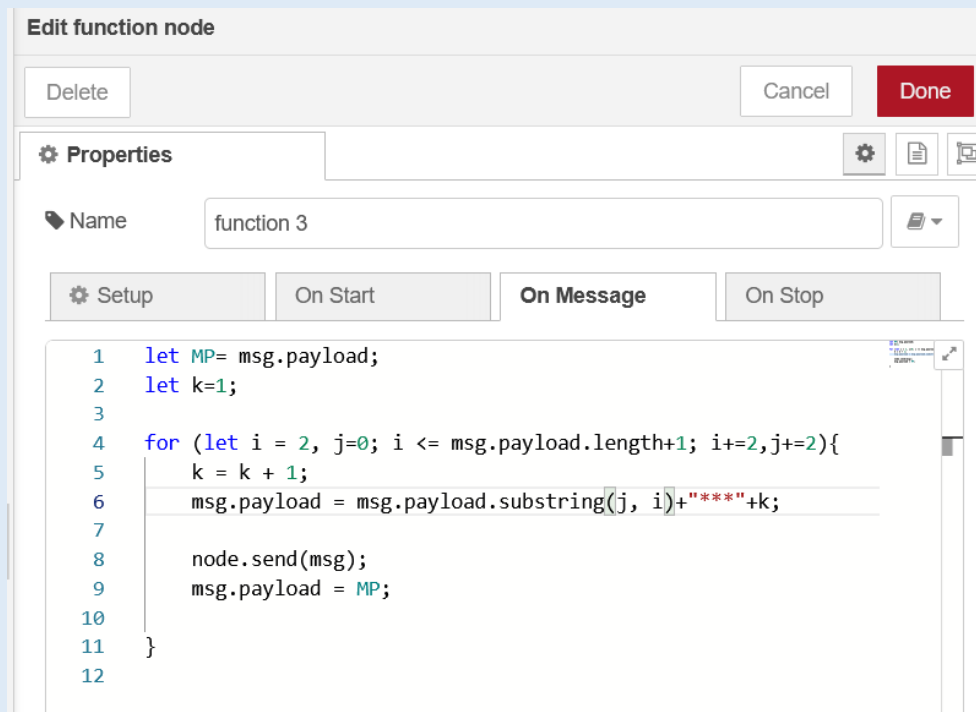
تنظیمات json



تنظیمات سویچ بالایی



تنظیمات سویچ پایینی



تنظیمات function

Edit delay node

Delete Cancel Done

Properties

⚙️ Action Rate Limit

All messages

⌚ Rate 1 msg(s) per 2 Seconds

☐ allow msg.rate (in ms) to override rate

Queue intermediate messages

🔖 Name Name

تنظیمات delay

Edit tcp request node

Delete Cancel Done

Properties

🌐 Server 127.0.0.1 port 20011

☐ Enable secure (SSL/TLS) connection

➡️ Return String

🔒 Close after a fixed timeout of 0 ms

🔖 Name Name

تنظیمات tcp request

Edit tcp in node

Delete Cancel Done

Properties

Type Listen on port 20011

☐ Enable secure (SSL/TLS) connection

Output stream of String payload(s)

delimited by (optional)

☐ re-attach delimiter

Topic Topic

Name Name

تنظیمات tcp in

Edit function node

Delete Cancel Done

Properties

Name function 9

Setup On Start On Message On Stop

```

1 let VALUE= msg.payload;
2 msg.payload = "i listen "+VALUE;
3 let split = msg.payload.split("***");
4 msg.payload = split[1];
5 return msg;

```

تنظیمات function

Edit tcp out node

Delete

Cancel

Done

⚙️ Properties

⚙️

📄

🔗

🕒 Type

Reply to TCP

▼

☐ Decode Base64 message?

🏷️ Name

Name

تنظیمات tcp out

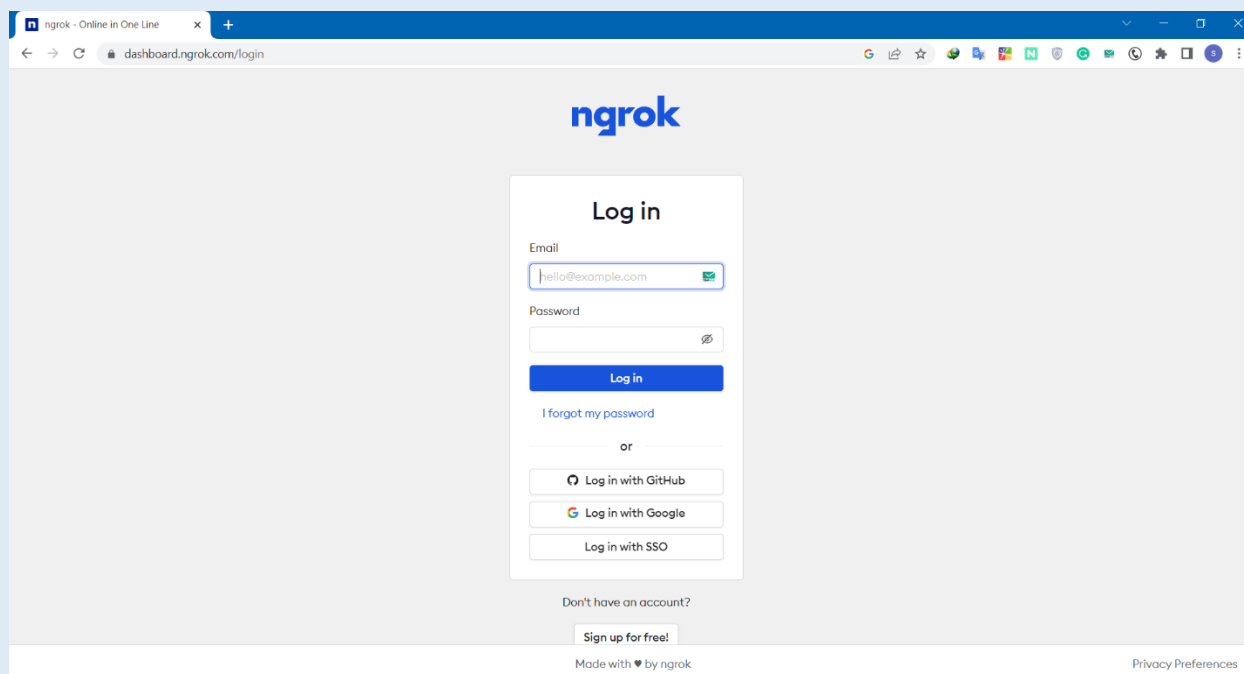
نتیجہ:

"7"
6/25/2023, 9:14:00 PM node: debug 12 msg.payload : string[6] ": ***8"
6/25/2023, 9:14:00 PM node: debug 3 msg.payload : string[1] "8"
6/25/2023, 9:14:01 PM node: debug 12 msg.payload : string[6] "Su***9"
6/25/2023, 9:14:01 PM node: debug 3 msg.payload : string[1] "9"
6/25/2023, 9:14:02 PM node: debug 12 msg.payload : string[7] "pe***10"
6/25/2023, 9:14:02 PM node: debug 3 msg.payload : string[2] "10"
6/25/2023, 9:14:03 PM node: debug 12 msg.payload : string[7] "r ***11"
6/25/2023, 9:14:03 PM node: debug 3 msg.payload : string[2] "11"

Remote

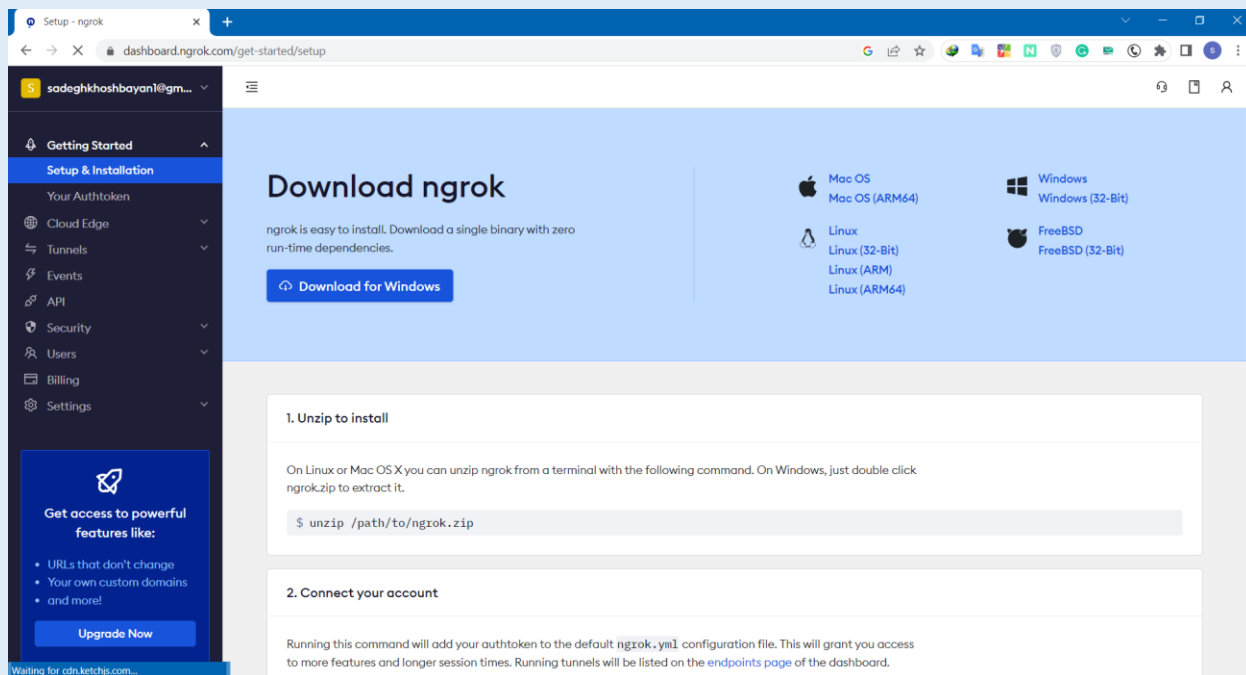
برای اینکه node-red رو به صورت remote فعال کنیم از سرویس ngrok استفاده میکنیم که مراحل فعال سازی به صورت زیر هست

ایتما باید در سایت ngrok ، log in کنیم

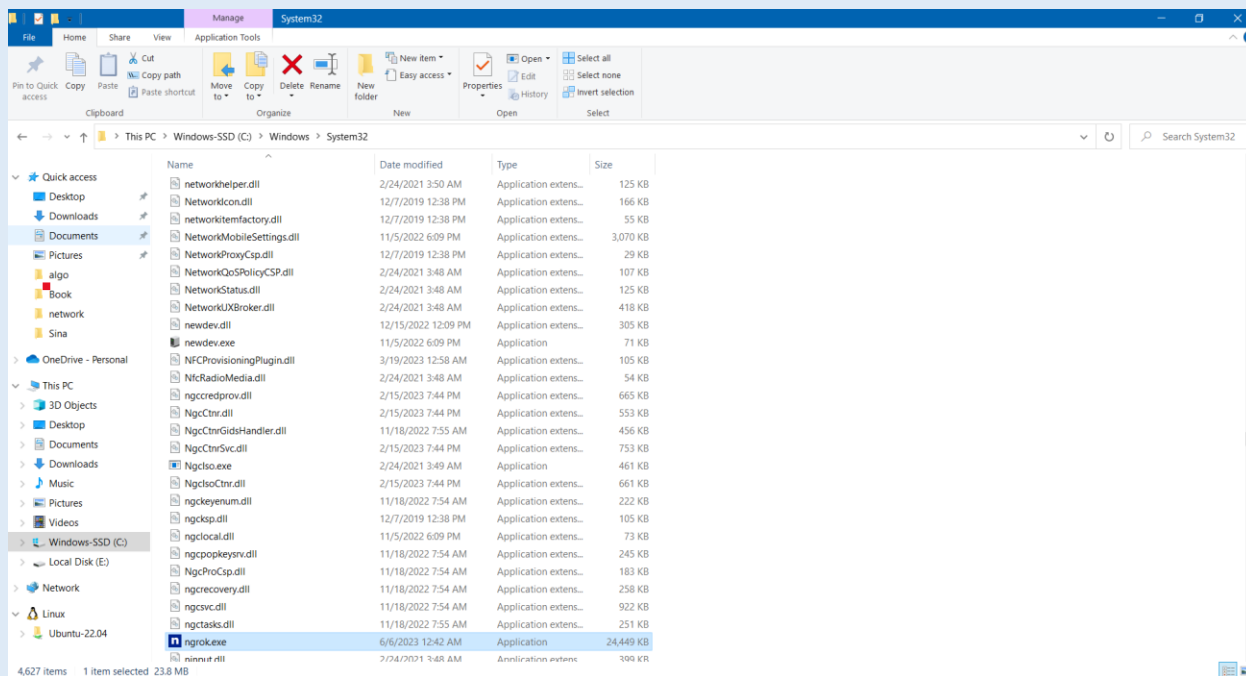


میشه از اکانت github یا google برای log شدن استفاده کرد

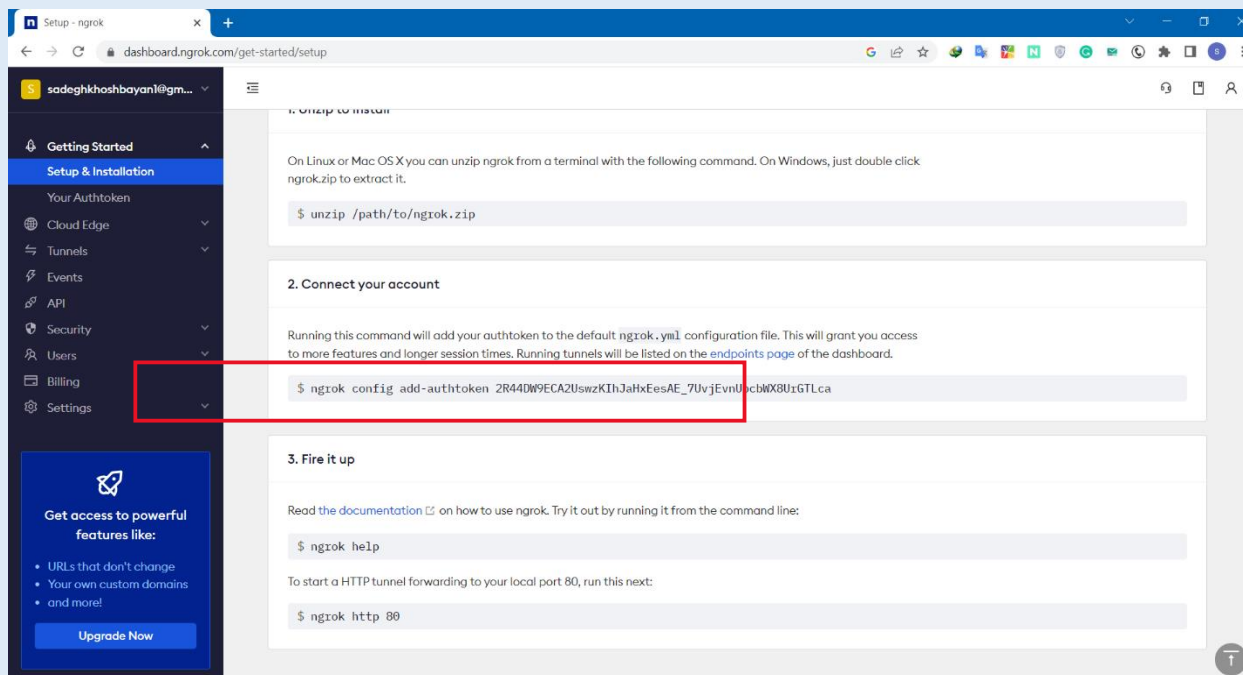
حالا فایل نصبی ngrok رو از سایتش دانلود میکنیم



بعد باید فایل نصبی رو به پوشه C:/Window/System32 منتقل کنیم



حالا باید با استفاده از key داده، ngrok رو config کنیم



برای config کردن ngrok ، cmd در ویندوز یا Terminal در لینوکس باز میکنیم و کد مشخص شده رو paste میکنیم

```
Command Prompt
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sadeg>ngrok config add-authtoken 2R44DW9ECA2UswzKIhJaHxEesAE_7UvjEvnUpcbWX8UrGTLca
Authtoken saved to configuration file: C:\Users\sadeg\AppData\Local\ngrok\ngrok.yml

C:\Users\sadeg>
```

حالا باید ابتدا node-red رو فعال کنیم

```
node-red
18 Jun 23:25:18 - [info] Windows_NT 10.0.19045 x64 LE
18 Jun 23:25:20 - [info] Loading palette nodes
18 Jun 23:25:20 - [info] Settings file : C:\Users\sadeg\.node-red\settings.js
18 Jun 23:25:20 - [info] Context store : 'default' [module=memory]
18 Jun 23:25:20 - [info] User directory : \Users\sadeg\.node-red
18 Jun 23:25:20 - [warn] Projects disabled : editorTheme.projects.enabled=false
18 Jun 23:25:20 - [info] Flows file : \Users\sadeg\.node-red\flows.json
18 Jun 23:25:20 - [info] Server now running at http://127.0.0.1:1880/
18 Jun 23:25:20 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----

18 Jun 23:25:20 - [info] Starting flows
18 Jun 23:25:20 - [info] Started flows
18 Jun 23:25:20 - [info] [tcp in:040e313299c4f3cd] listening on port 2000
18 Jun 23:25:20 - [error] [tcp out:745f91d69f9b2df6] unable to listen on port 2000, error: Error: listen EADDRINUSE: address already in use :::2000
18 Jun 23:25:20 - [info] [udp in:udp in] udp listener at 0.0.0.0:2001
18 Jun 23:25:20 - [info] [udp out:udp out] udp re-use socket: -> 127.0.0.1:2001
```

بعد با دستور ngrok http 1880 ، port forwarding انجام میدیم

```
Command Prompt
C:\Users\sadeg>ngrok http 1880
```

(نکته) 1880 پورت که node-red روش فعال هست

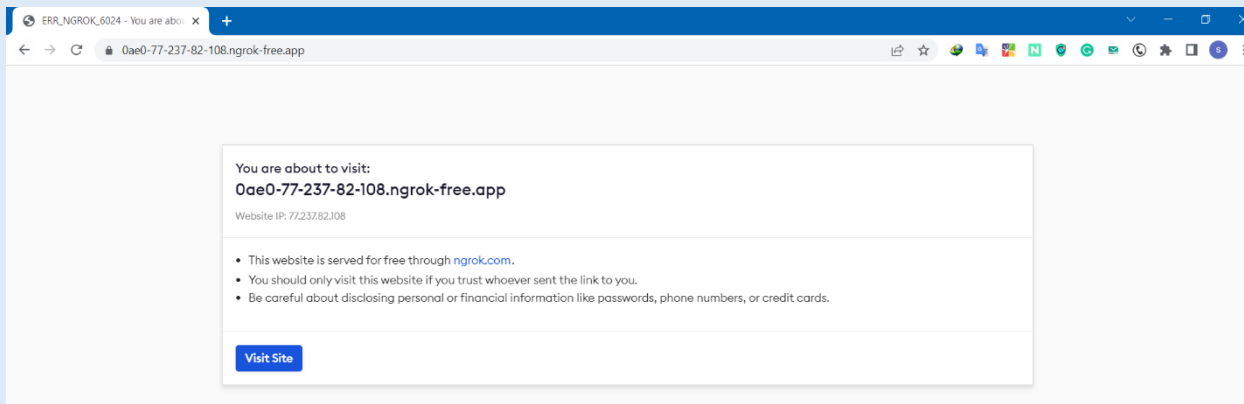
```
Command Prompt - ngrok http 1880
ngrok
(Ctrl+C to quit)

Send your ngrok traffic logs to Datadog: https://ngrok.com/blog-post/datadog-logs

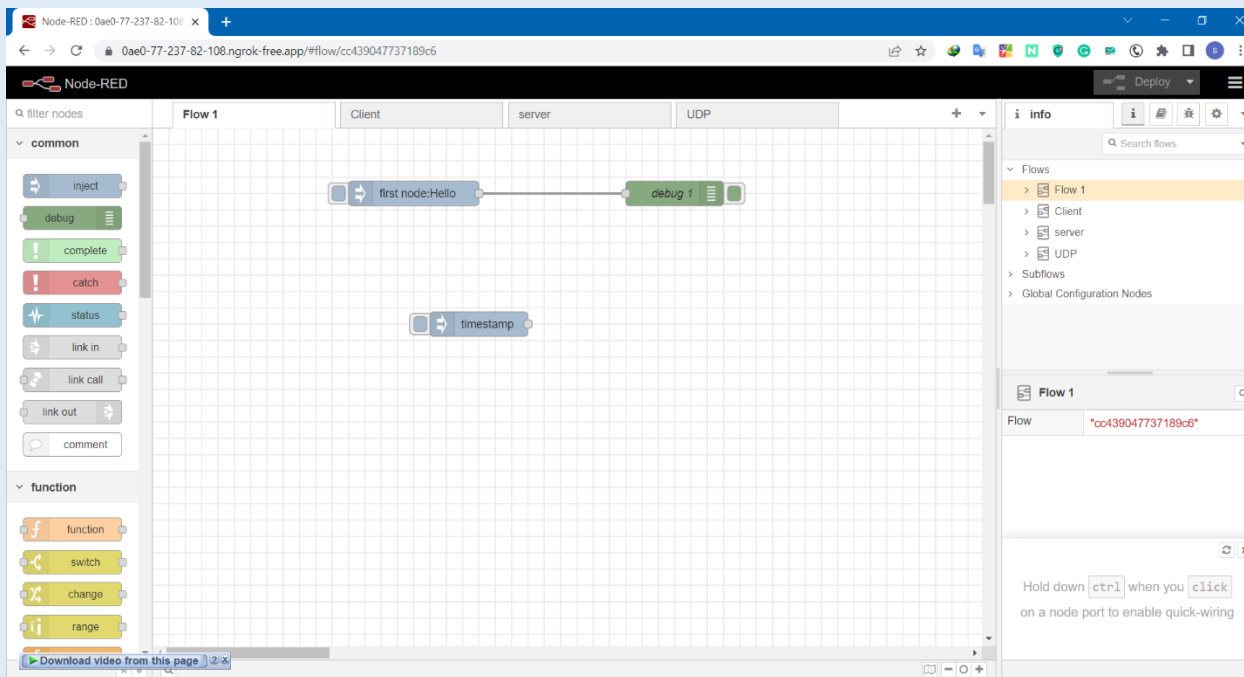
Session Status      online
Account             sadeghkhoshbayan1@gmail.com (Plan: Free)
Version             3.3.1
Region              Europe (eu)
Latency              -
Web Interface       http://127.0.0.1:4040
Forwarding           https://0ae0-77-237-82-108.ngrok-free.app -> http://localhost:1880

Connections
  ttl    opn    rt1    rt5    p50    p90
   0      0     0.00   0.00   0.00   0.00
```

حالا به صورت remote با url مشخص شده میتونیم به node-red متصل شیم



این صفحه باز میشه و با زدن vist site میتونیم به صفحه node-red مورد نظر رو مشاهده کنیم



پایان