PORT-FORWARDING

Definition:

Port forwarding is a networking process that **redirects traffic** from one IP address and port number combination to another, **allowing external devices to access services** on a private network.

Scenario :

Imagine you're running a **web server** on your laptop inside your home network (Private IP: `192.168.1.100`, Port `80`).

- Normally, people **outside** (on the internet) **can't access** it, because your home network uses **NAT** (Network Address Translation).
- So, you tell your router:
  "When someone connects to my **public IP** on **port 8080**, forward that to my **laptop** on **port 80**."

So:

`Incoming: 203.0.113.5:8080 → Forwarded to: 192.168.1.100:80`

Now anyone visiting `http://203.0.113.5:8080` sees your local web server.

Workflow:

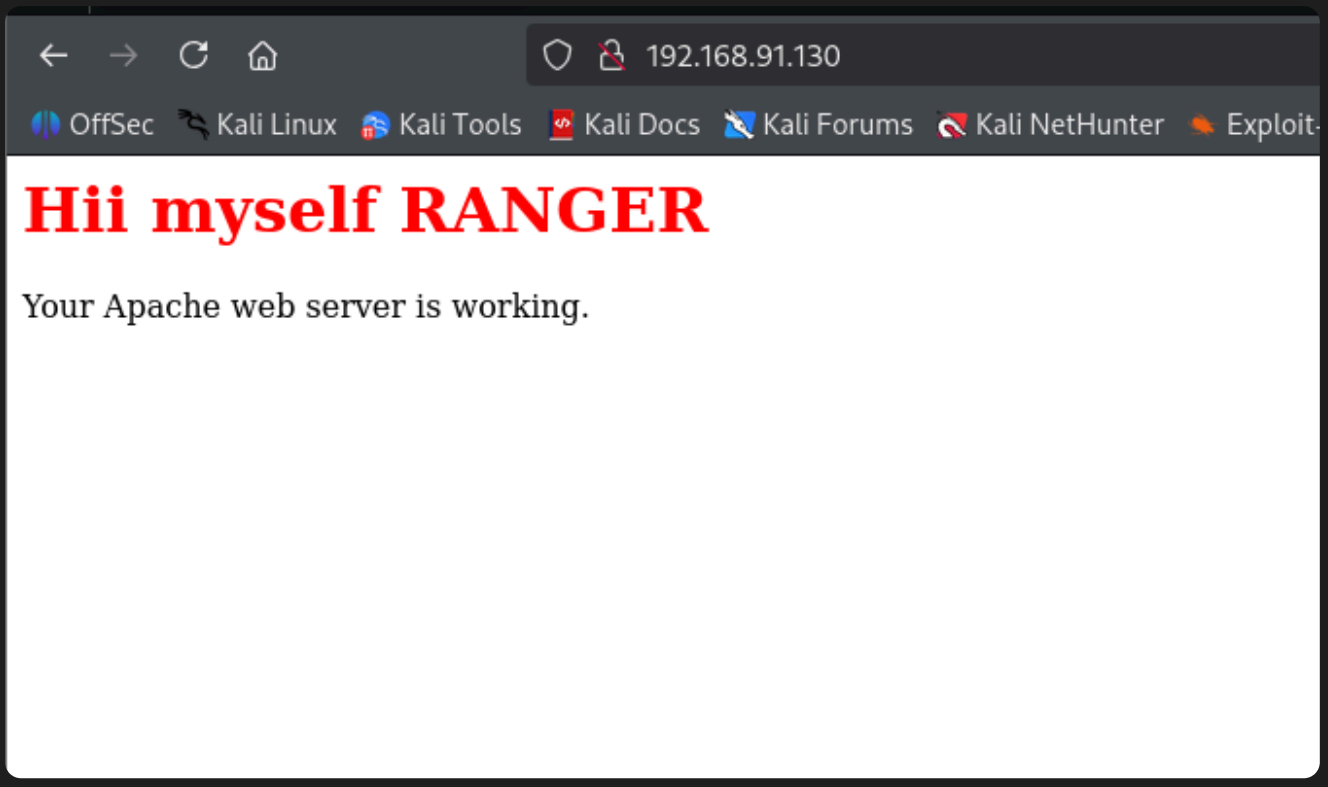
[INTERNET USER]
|
v
Public IP:203.0.113.5:8080
|
| (Router receives incoming packet)
|
[ROUTER: Port Forwarding Rule]
|
|---> Forward to --> [LOCAL SERVER 192.168.1.100:80]
|

|---> Response goes back via router
|
<--- To Internet User

Example:

For example we are running a **apache web server** on our desktop inside your home private network
Private IP: `192.168.91.130`, Port `80`

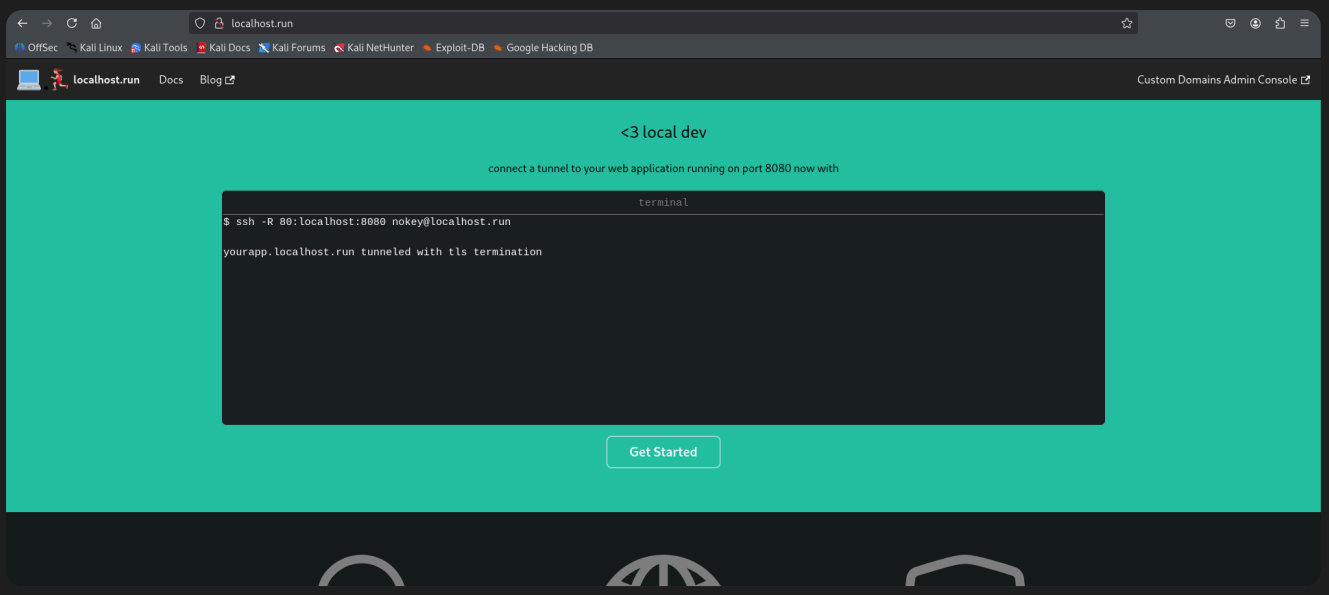
```
[ranger@localhost ~]$ sudo systemctl status httpd
[sudo] password for ranger:
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2025-10-05 12:51:35 EDT; 4min 39s ago
     Docs: man:httpd(8)
           man:apachectl(8)
 Main PID: 1444 (httpd)
   Status: "Total requests: 0; Current requests/sec: 0; Current traffic:   0 B/sec"
   CGroup: /system.slice/httpd.service
           ├─1444 /usr/sbin/httpd -DFOREGROUND
           ├─2596 /usr/sbin/httpd -DFOREGROUND
           ├─2598 /usr/sbin/httpd -DFOREGROUND
           ├─2599 /usr/sbin/httpd -DFOREGROUND
           ├─2601 /usr/sbin/httpd -DFOREGROUND
           └─2602 /usr/sbin/httpd -DFOREGROUND

Oct 05 12:51:31 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
Oct 05 12:51:35 localhost.localdomain httpd[1444]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using lo... message
Oct 05 12:51:35 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
[ranger@localhost ~]$
```



But we cant access the web server from a external system or in a public ip,to resolve the issue what can we do we just use port forwarding method.

To do that we can use some websites to forward our port and host our website for some time like--
http://localhost.run/
https://portmap.io/

step 1 :



Copy the command we got from this website and paste it into our host machine
ssh -R 80:localhost:8080 nokey@localhost.run here we change the command little bit,that was we change the localhost port from 8080 to 80 because the web server currently runs on port 80.

Step 2 :

final command is----

> !! ssh -R 80:localhost:80 nokey@localhost.run

This command:

> !! Creates a **tunnel** from your computer → to a remote server (localhost.run).
> That remote server gives you a **public URL** so others can see your site.

So basically:

> !! You make your local website available on the internet instantly, using SSH.

Breaking Down the Command

---

ssh The **Secure Shell** program — used to connect securely to another computer.
-R Remote port forward

80 The **port on the remote server** (localhost.run)

localhost:80 The **destination on your local machine** (your local server running on port 80)

nokey@localhost.run This means:

Connect to the server **localhost.run**

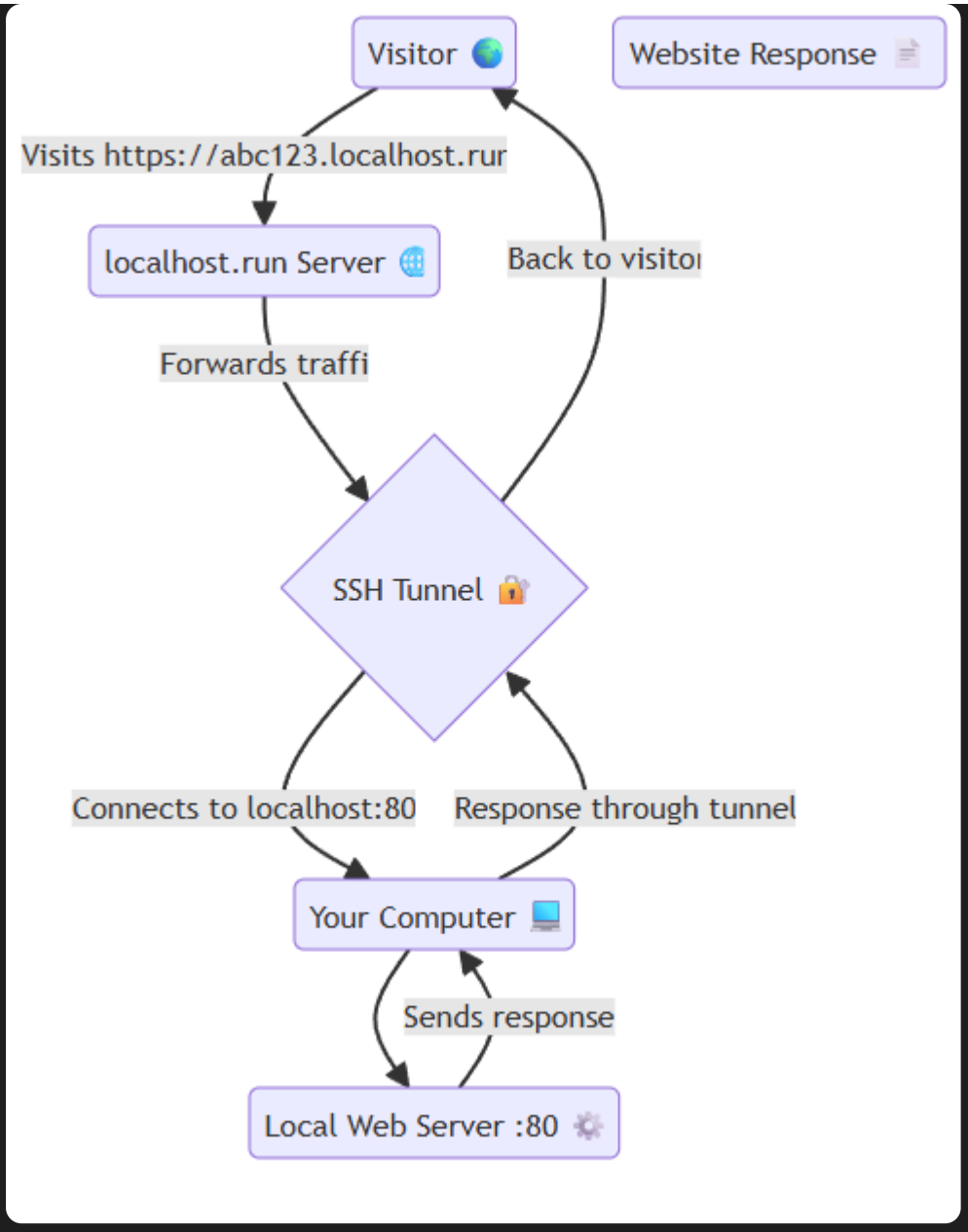
As the user **nokey** (no password needed — it's a public tunnel service)

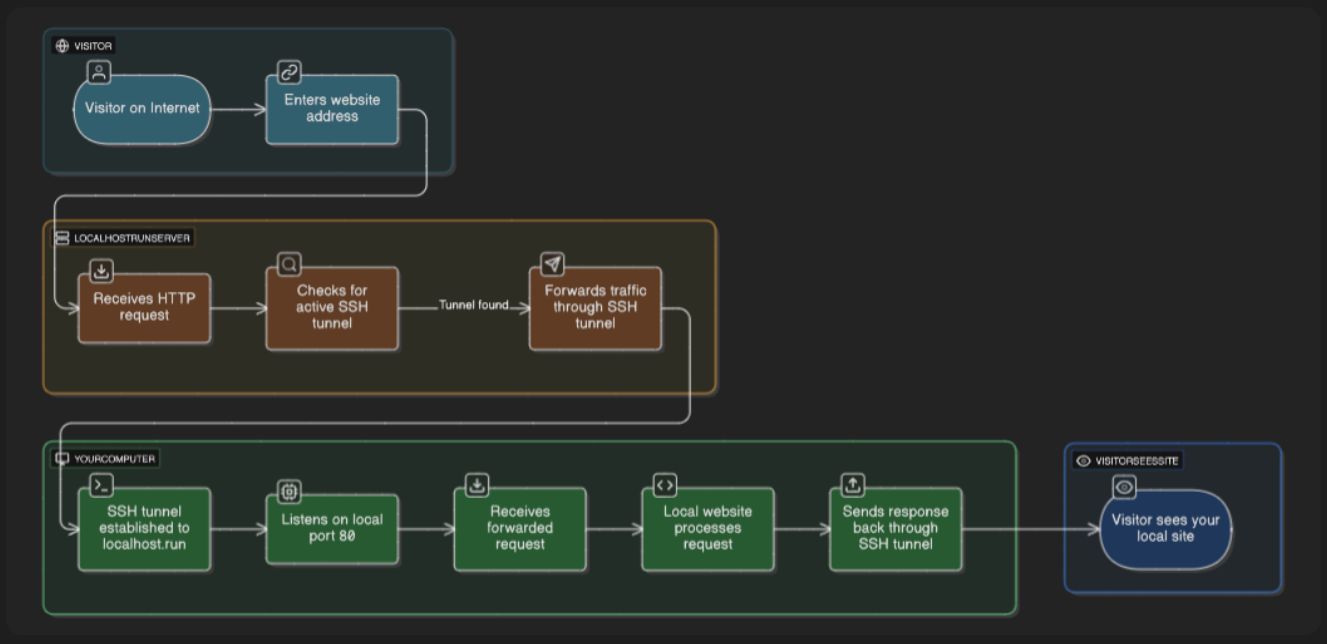
`localhost.run` is a free tunneling service — it creates a **temporary public address** (like `https://abc123.localhost.run`) that forwards to your local web server.

Step 3 :

Now if we scan the **qr** we can visit the **website publically**.

Flowchart :


Visitor
Website Response
Visits https://abc123.localhost.rur
localhost.run Server
Back to visitor
Forwards traffi
SSH Tunnel
Connects to localhost:80
Response through tunnel
Your Computer
Sends response
Local Web Server :80


VISITOR
Visitor on Internet
Enters website address
LOCALHOSTRUNSERVER
Receives HTTP request
Checks for active SSH tunnel
Tunnel found
Forwards traffic through SSH tunnel
YOURCOMPUTER
SSH tunnel established to localhost.run
Listens on local port 80
Receives forwarded request
Local website processes request
Sends response back through SSH tunnel
VISITORSEESSITE
Visitor sees your local site