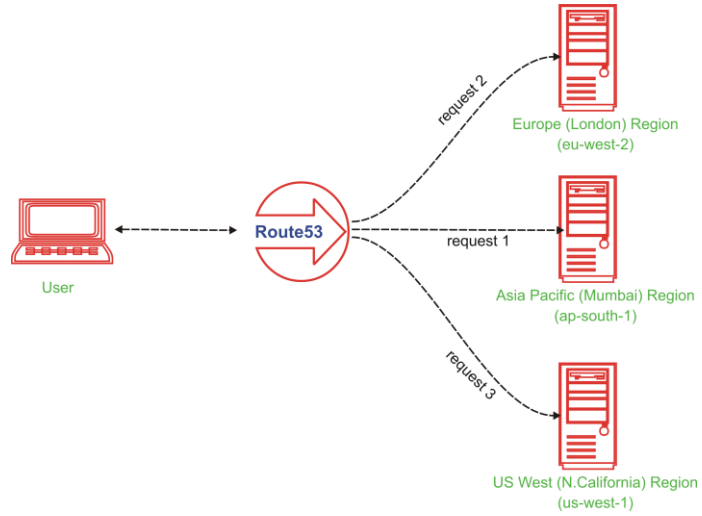


AWS Route53 Introduction

(AWS Route53 and DNS Basics)

Vipin Gupta
BE,RHCE,CEH,CCNA,MCSE,MCSA
vipin2411@gmail.com
Mobile: 93563-10379
www.linuxexpert.in

What is DNS

DNS (Domain Name System) is used for name resolution i.e. getting IP addresses by specifying FQDN (Fully Qualified Domain Name). For example if we are querying “www.example.com” then DNS will return IP address (called forward name resolution)

Why the need for DNS? It is very easy to remember names. Another benefit is that it isolates you from the effect of changes in IP addresses. DNS is also used for reverse name resolution i.e. getting name by supplying IP addresses. DNS is also used for providing information about mail servers.

Types of Resolution

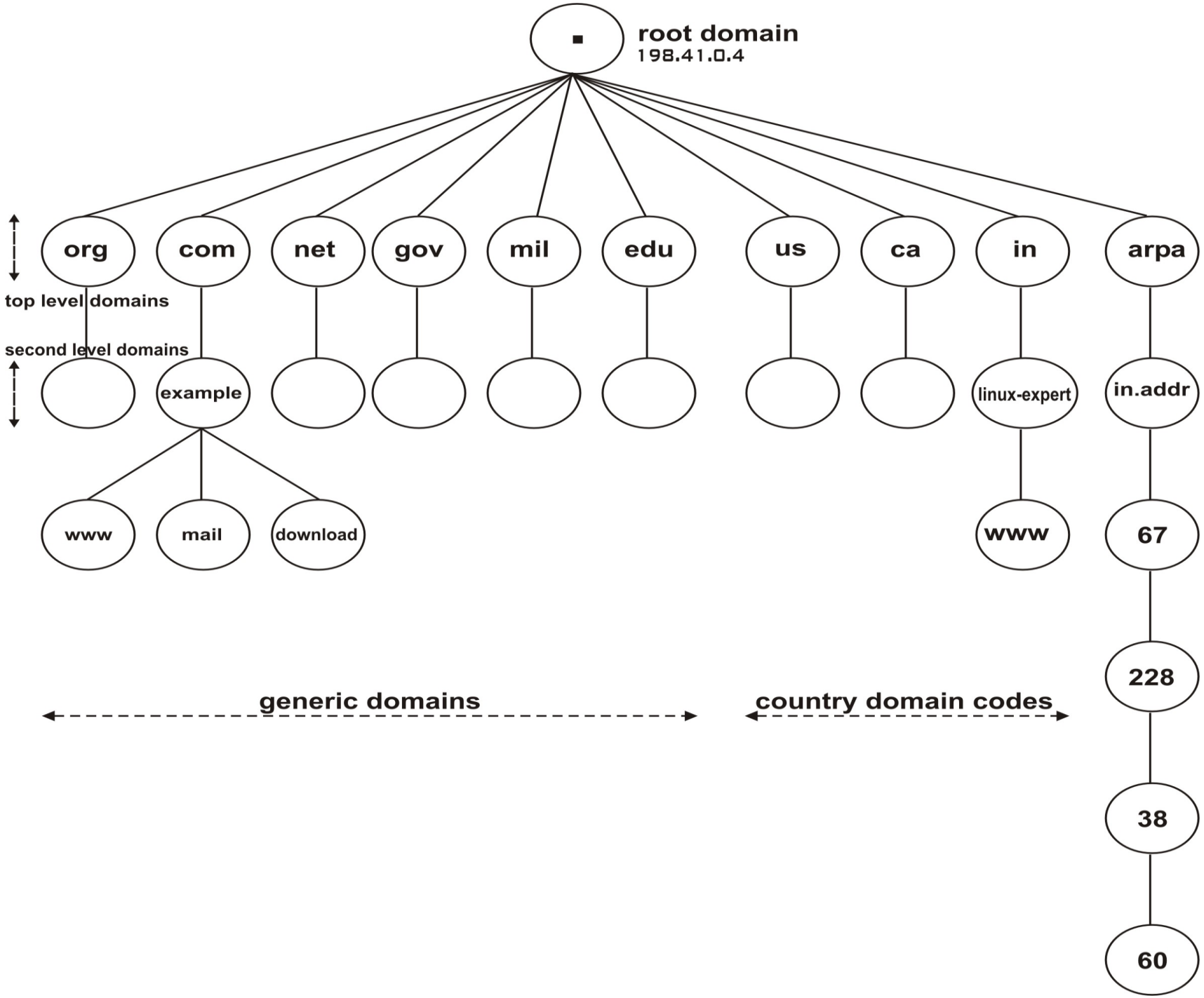
Forward Resolution: When we query “www.example.com” and get IP address (192.168.32.10).

Reverse Resolution: When we query “192.168.32.10” and get name “www.example.com”.

Top Level Domains

Top Level Domain (TLD)	Meaning
gov	government
net	networking
com	commercial
org	non-profit organizations
mil	military
edu	educational
arpa	internet infrastructure
biz	business
coop	cooperative associations
name	for individuals
us	USA
uk	United Kingdom
in	India

Domain Structure



DNS is hierarchical in nature. The whole internet is controlled by root servers (also called dot servers). Under root servers are generic top level domains, country specific domains and one internet infrastructure domain called “arpa”.

DNS Namespace

The whole DNS name space can be divided into 3 categories

- generic Top Level Domains (gTLDs) contains “com”, “net”, ”edu”, ”mil”, “gov”, “org”, “int”. Since the internet is expanding rapidly, few more like “biz”, “info”, “coop”, “aero”, “museum”, “name”, “pro” have been added. Every “gTLD” represents specific category.

- country code Top Level Domains (ccTLDs) uses 2-character country code. For example “us” means USA and “in” means India.

- “arpa” is a special domain used for reverse resolution.

Types of Name Servers

Master Servers: Get zone data from locally stored files. We generally have one master server for a zone. All changes should be made at master servers. These give authoritative information to clients.

Slave Servers: Get mirror zone data (read only i.e. you can not change zone data at secondary servers) from master servers through zone transfer. Slave servers periodically checks for updates in master server (as specified in SOA record) and if there is any change, they will update their data. There can be multiple slave servers for a zone. These give authoritative information to clients.

Master and Slave servers are also called Primary and Secondary servers, but these days, we are no longer using these terms.

Caching Only Servers: They do not have any data in zone files. They simply accept request and forward them. They keep results in cache. These give non-authoritative information to clients.

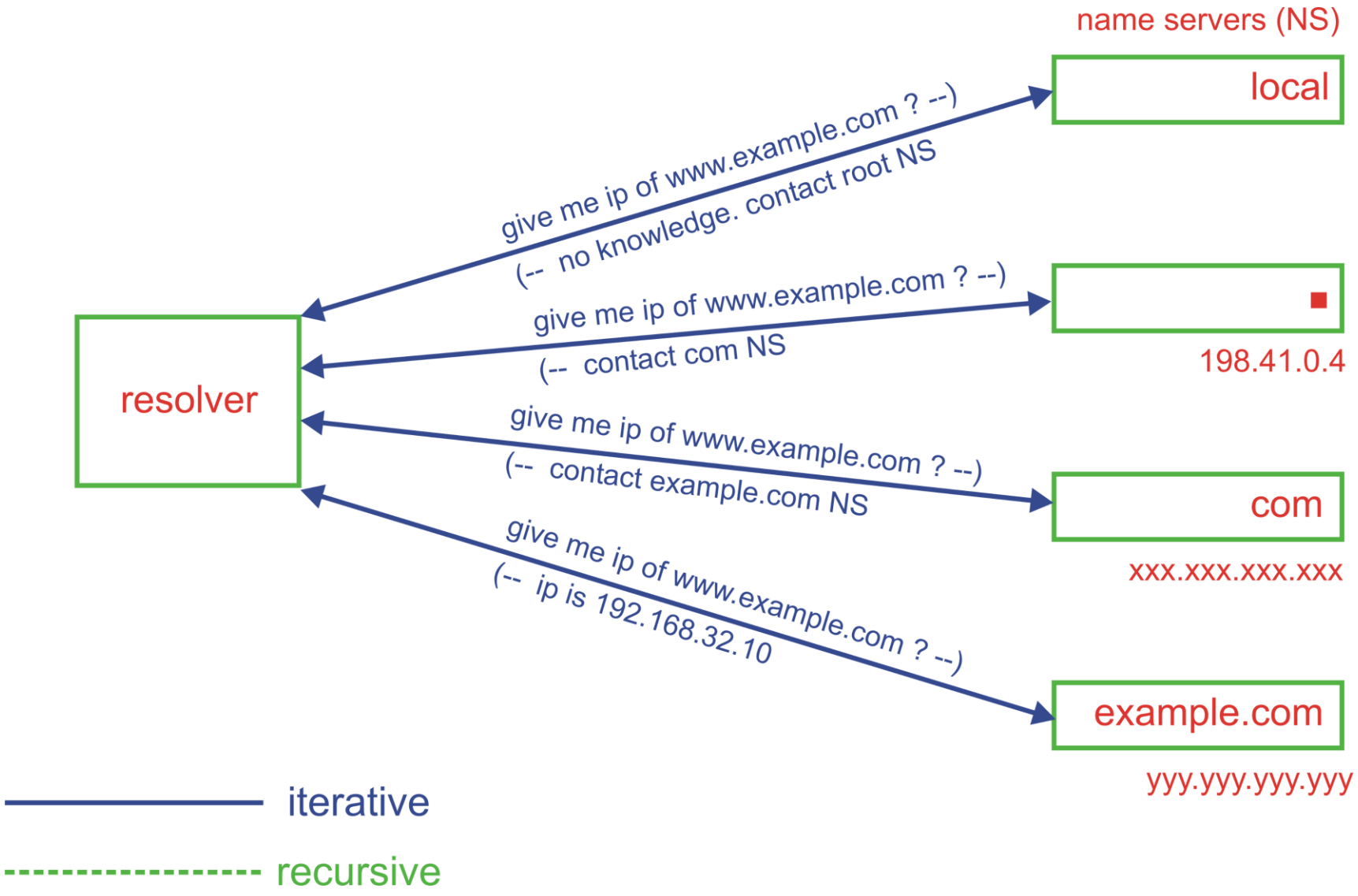
Types of Query

- Iterative Query

- Recursive Query

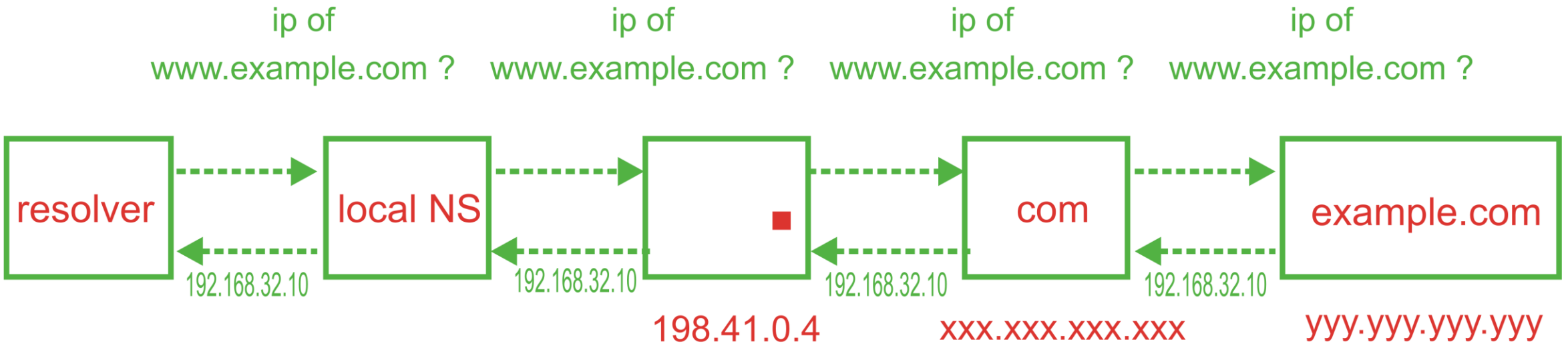
- Mixed (Hybrid) Query

Iterative Query



iterative query: In this the resolver has the responsibility to resolve the name to IP address. As illustrated in the diagram, resolver will query local NS for IP address of "www.example.com". If local NS has got the information, it will give that information to resolver, otherwise it will tell the resolver to contact root servers. The resolver then queries "root" NS (in this case 198.41.0.4). If root NS does not have the information, it will tell the resolver to contact "com" NS (in this case xxx.xxx.xxx.xxx) and so on till resolver gets the IP address "192.168.32.10". Most of the load is on resolver instead of on top level name servers.

Recursive Query



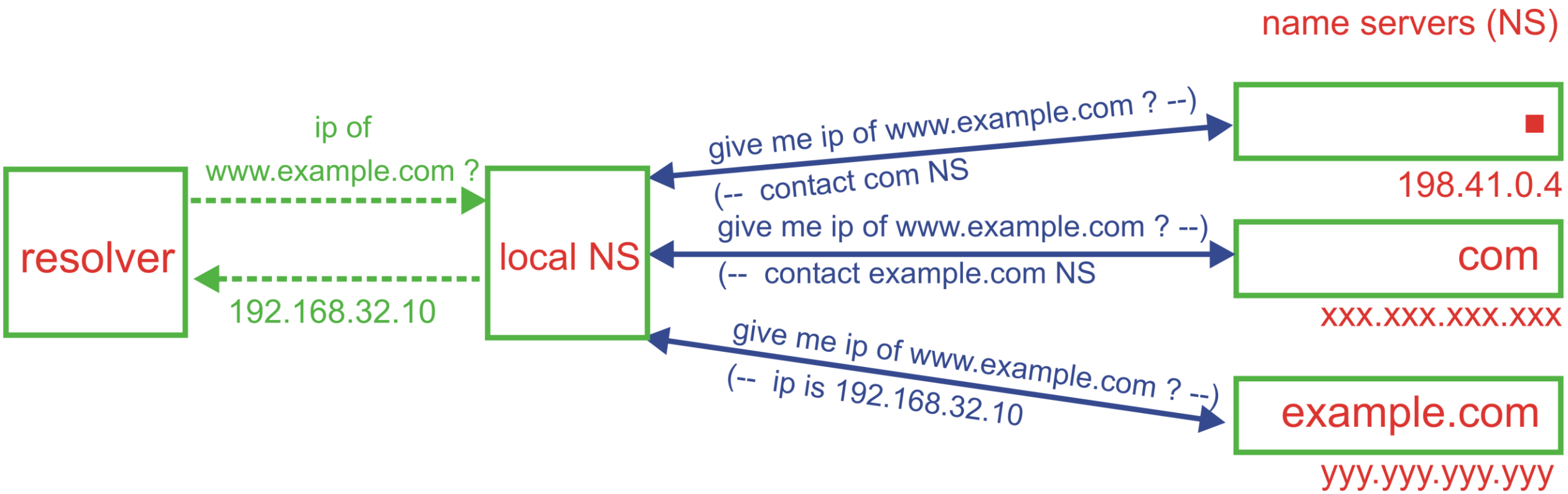
iterative

recursive

Recursive: In this resolver will send recursive query to "local" NS. If it has the information, the response will be sent to resolver. If not, it will send recursive query to "root" name server. If it has the information, the response will be sent to "local" NS. If not, it will send recursive query to "com" name server and so on.

There is lot of load on top level name servers.

Hybrid (Recursive-Iterative) Query



Hybrid: In this resolver sends recursive query to **"local"** NS. Depending up the information it have, it will return the result to resolver or send iterative query to **"root"** name server. In real world scenarios, often this type of query is used.

Resource Records and Directives

Resource Records (RR) are basic building blocks of zone files and used for answering DNS queries.

The syntax of resource records(RR) is

domain ttl class type rdata

domain: current domain or some specific domain.

ttl: for how long the record will be cached.

class: record classification mostly “IN”. if left empty, “IN” is assumed.

type: type of record such as “SOA”, “A”, “PTR”, “CNAME”, “NS”, “MX” etc.

rdata: particular record data

Start of Authority (SOA)

SOA (Start of Authority): it specifies the information regarding FQDN of name server, email of domain administrator, serial number, various refreshing timers.

serial number: is used to control the changes in zone data. Whenever we are going to make changes in zone data, we will increase the serial number, thus indicating to slave servers the change in data. Generally serial number is specified in dates.

refresh: the time after which slave servers should contact master servers for updates.

retry: if slave fails to contact master server after "refresh" interval, retry time.

expire: if master server is down, for how much time slave will handle queries.

minimum: for how much time negative information will be cached. Sometimes instead of specifying "http://www.example.com", we might be entering "http://www.exampl.com" in our web browser. Since there is no site "www.exampl.com", so resolver will get "-ve" answer.

NS, MX, A, PTR, CNAME

NS (Name Server): shows authoritative name servers for the domain. When we create hosted zone in AWS Route53, NS record automatically get created. It shows 4 name servers which are authoritative for the domain.

MX (Mail Exchanger): is used to give the information about mail servers responsible for a particular domain. We can also specify priority. The lower priority mean more importance.

A (Address): define IPv4 address for the resource.

PTR (Pointer): maps IP address to domain name.

CNAME (Canonical Name): maps a name to another name.

Useful Utilities for Testing DNS Server

host: it will show lot of useful information. It reads the resolver file "/etc/resolv.conf".

```
[root@master ~]# host master
master.example.com has address 172.24.0.1
[root@master ~]#
[root@master ~]# host www.example.com
www.example.com is an alias for master.example.com.
master.example.com has address 172.24.0.1
```

Useful Utilities for Testing DNS Server

nslookup: is the standard utility used in "Linux/Unix/Windows" systems. It can also be used in interactive mode. This utility is now being depreciated.

```
[root@master ~]# nslookup www.example.com
Server:         172.24.0.1
Address:        172.24.0.1#53

www.example.com canonical name = master.example.com.
Name:   master.example.com
Address: 172.24.0.1
```

Useful Utilities for Testing DNS Server

dig (domain information groper): very powerful utility which sends queries directly to DNS server instead of through resolvers.

```
[root@master ~]# dig www.example.com

; <<>> DiG 9.11.4-P2-RedHat-9.11.4-26.P2.el7_9.2 <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18635
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        10800   IN      CNAME   master.example.com.
master.example.com.     10800   IN      A       172.24.0.1

;; AUTHORITY SECTION:
example.com.            10800   IN      NS      master.example.com.

;; Query time: 0 msec
;; SERVER: 172.24.0.1#53(172.24.0.1)
;; WHEN: Mon Nov 30 12:23:35 IST 2020
;; MSG SIZE  rcvd: 95
```