01 - Basic Networking

Intro to DevOps

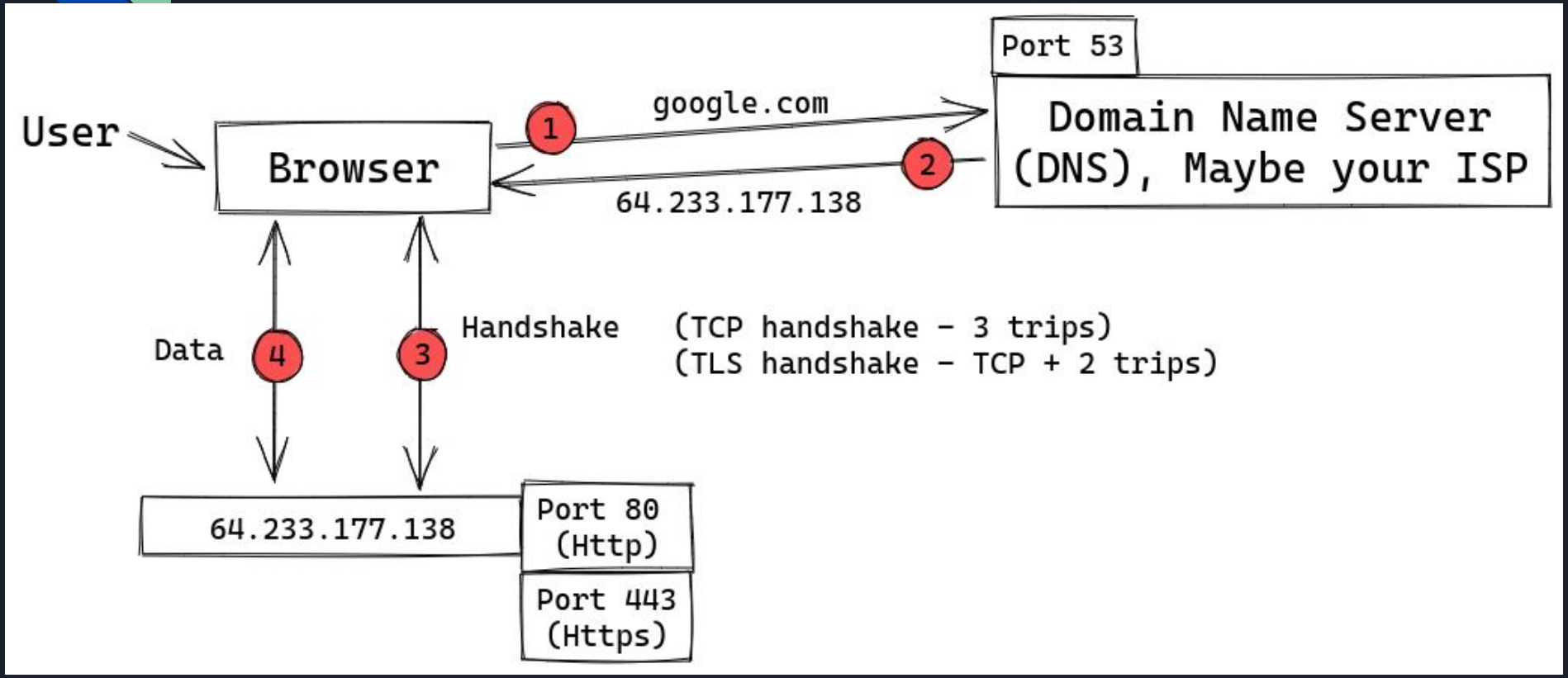
Why Networking?

- The web and the cloud are at the heart of everything nowadays.
- You need a solid general understanding of how it works to get your work done, even as a regular software engineer.

Terminology

- Internet Protocol (IP) Address; used to identify a server and connect to it over the internet. Has to be globally unique, no two public servers can share the same IP over the internet.
  - IPv4 is 32 bits = 4,294,967,296 Possible Addresses - We're running out of them
  - IPv6 is 128 bits > 340282367000000000000000000000000000000 Addresses
- Port; used to connect to a specific process on a machine. Has to be unique on the machine, no two apps can share the same port on a single machine.
- Open Systems Interconnection (OSI) Model & TCP/IP Model
  - Just two models that define how two things on the internet can communicate from hardware low-levels (electricity in the wire) up to application high-levels (a button click in an app).

What happens when you visit a website?

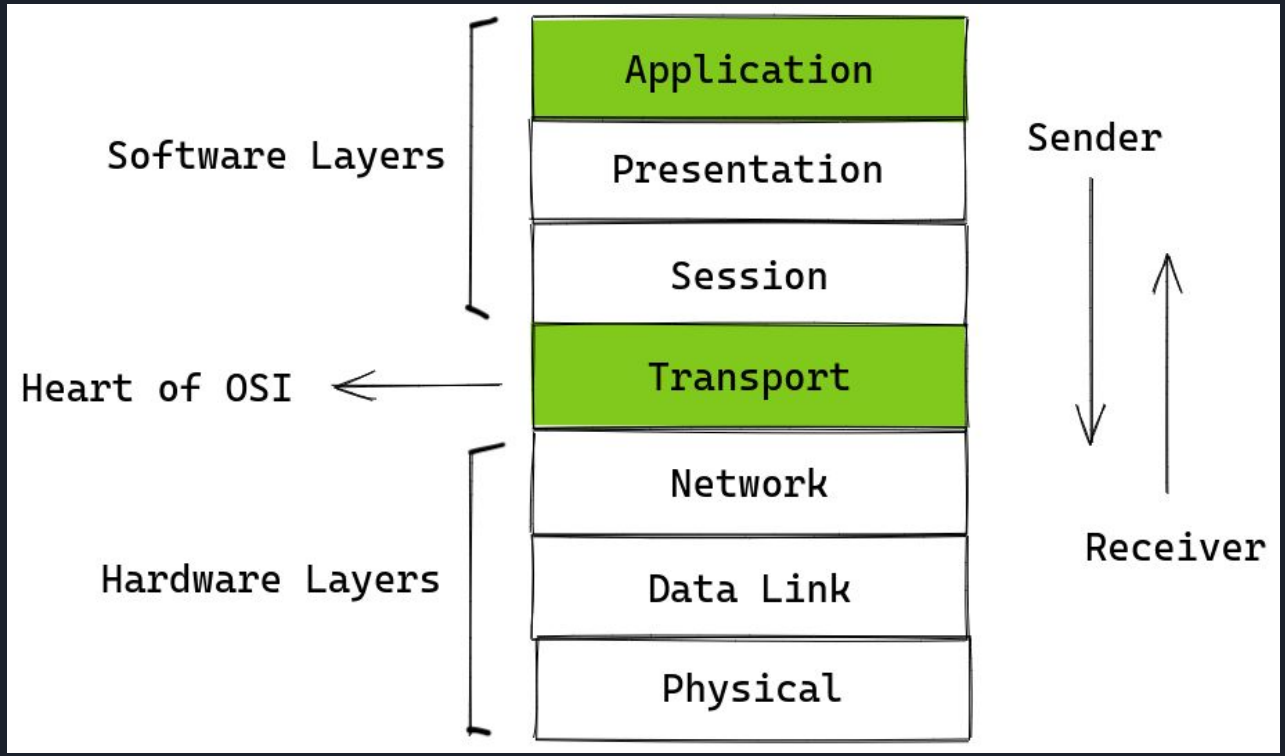


What does a DNS Server have?

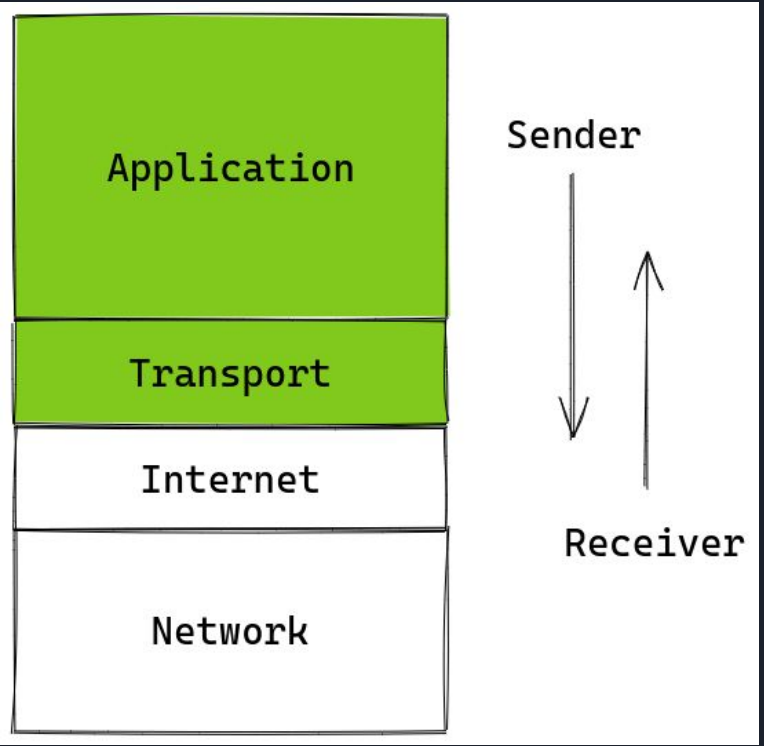
Mapping of Names to Internet Protocol (IP) Addresses

- google.com -> 64.233.177.138 (A Record - IPv4)
- google.com -> 2607:f8b0:4002:c09::8b (AAAA Record - IPv6)
- www.google.com -> 64.233.174.150 (A Record - IPv4)
- docs.google.com -> 64.233.176.38 (A Record - IPv4)
- forms.google.com -> google.com (CNAME Record - Subdomains, two things pointing to the same IP)
- mail.google.com -> google.com (MX Record - eMail, is looked up when people send an email to myuser@google.com)
- _verify.google.com -> SOME_RANDOM_VALUE_TO_VERIFY_ME (TXT Record)

OSI Model - Theoretical reference-ish



TCP/IP - De facto



Transport Layer

Usually referred to as Layer 4

- Port-to-Port communication between processes (meaning that you'll need an IP/Host and a Port)
- Doesn't understand the data/application. This means that it can't do any decisions based on data; only IP address & port-based decisions.
- Example Protocols: TCP (Reliable but Slower) & UDP (Faster but not as reliable)
- Example Applications: Network Load Balancers (NLBs)

Application Layer

Usually referred to as Layer 7

- App-to-App communication; mostly user-facing things.
- Can understand the data.
  It can make decisions based on the data (like what?).
- Example Protocols: HTTP, DNS, SSH
- Example Applications: Web browsers, Online Video Games, Application Load Balancers (ALBs), Web Application Firewalls (WAFs).

What does HTTP do?

- Transfers plain text (not encrypted) over TCP – that's it.
  - It doesn't care if you're sending HTML, CSS, JS, Python, JSON, zip, …
- People use it to serve
  - Websites & WebApps (Frontends) - Browsing
  - APIs & Microservices (Backends) - Manipulating Data (Adding, Updating, Deleting, Retrieval)
- Not secure; you'll need HTTPs for that
  - Uses TLS to secure itself and encrypt communication in both ways.
  - You'll need a "certificate" and a "key"

Well-Known Protocols & Ports

HTTP - 80 (on top of TCP)

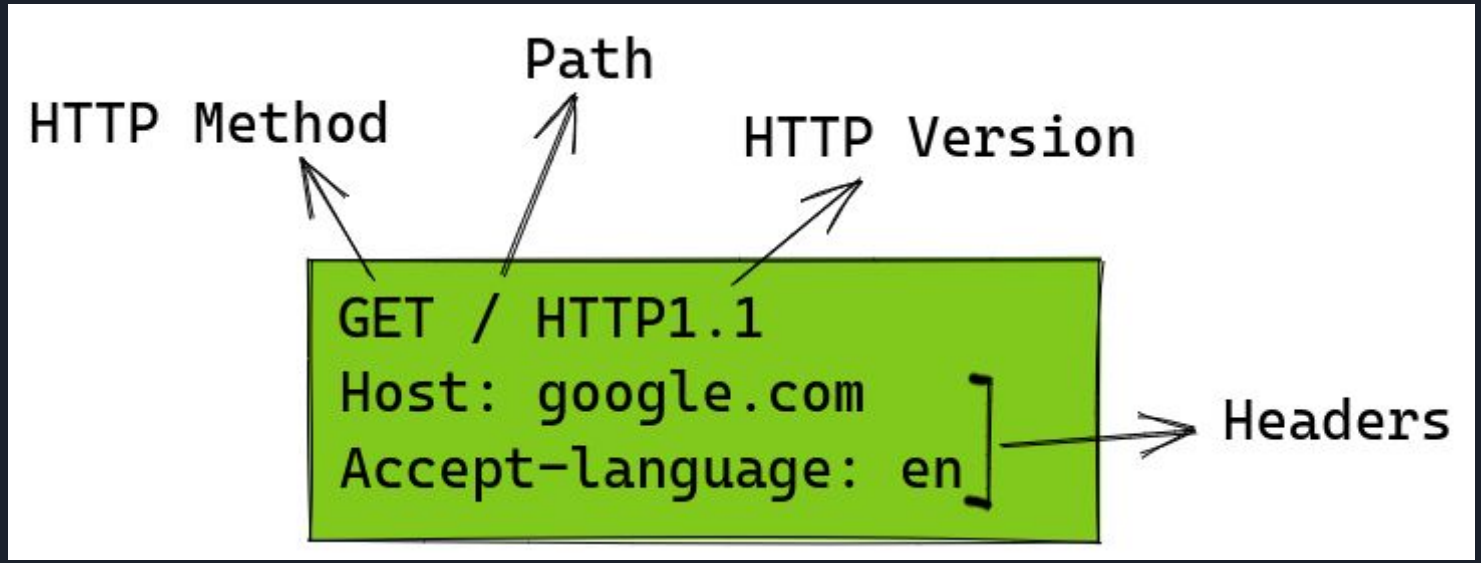
HTTPs - 443 (TCP)

SSH - 22 (TCP)

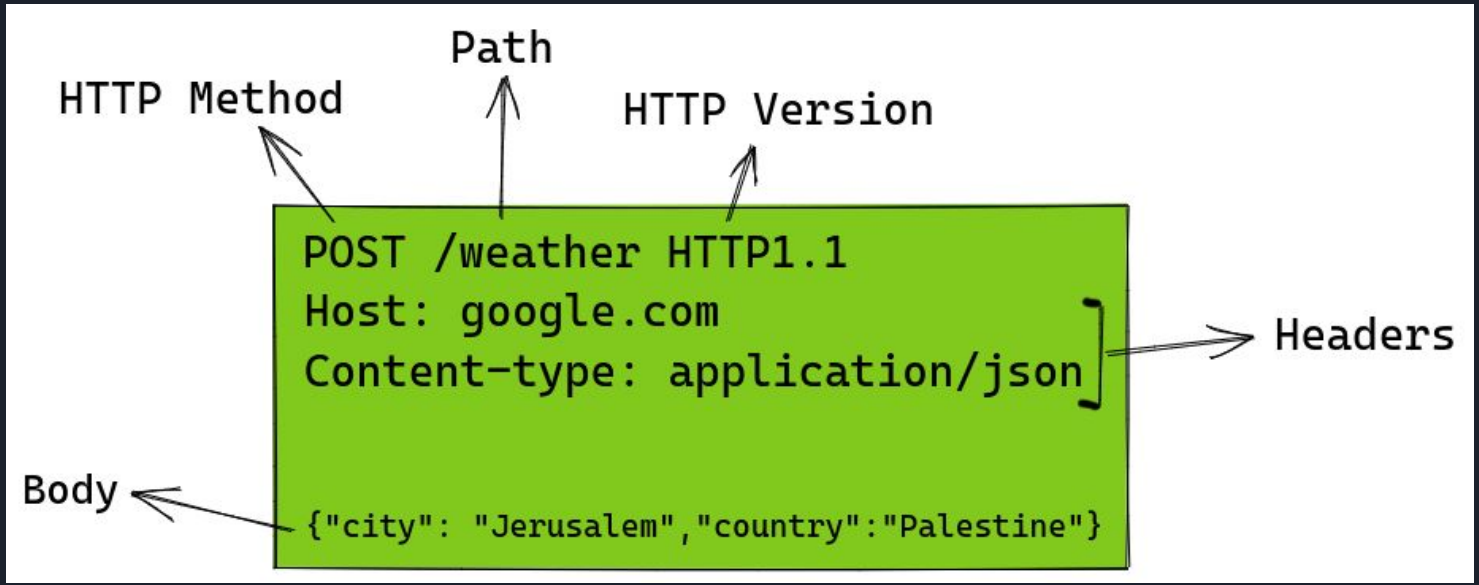
DNS - 53 (TCP & UDP)

SMTP - 25 (TCP)

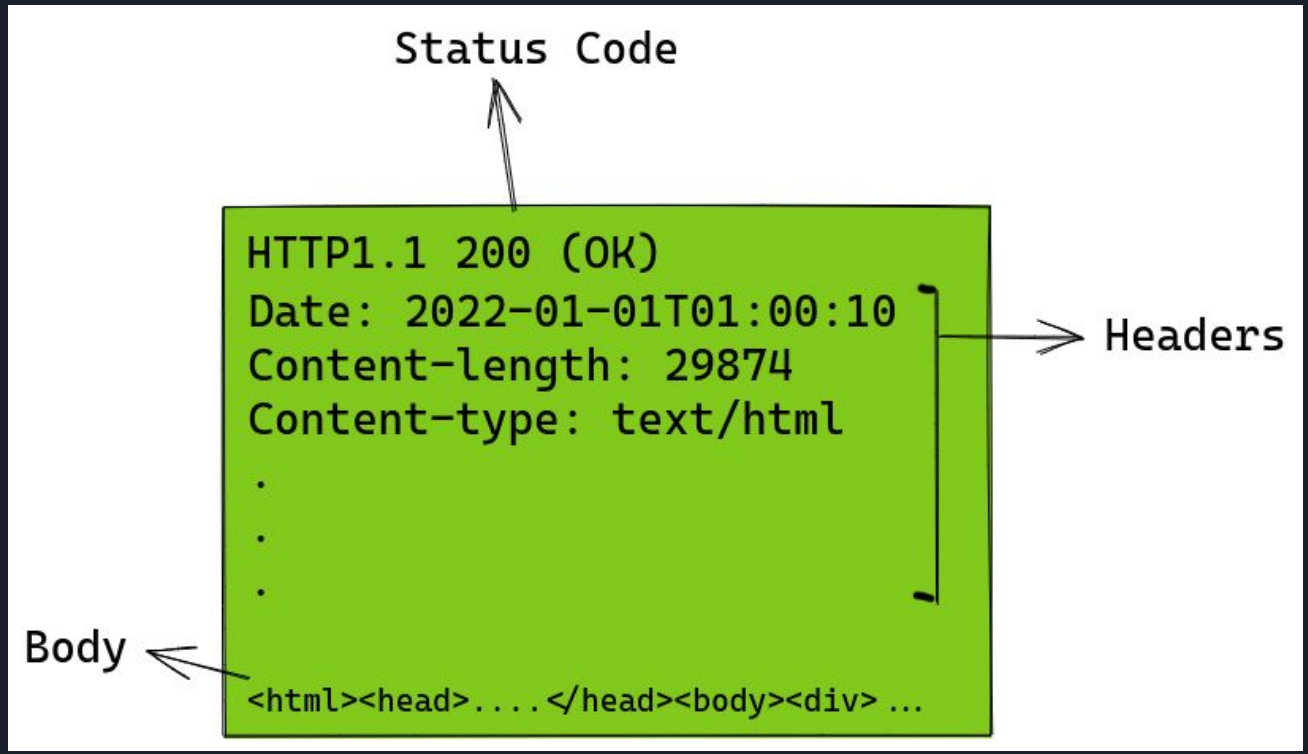
HTTP Request Example



HTTP Request Example



HTTP Response Example



HTTP Status codes

- Informational responses (100–199) - usually you don’t have to worry about those
- Successful responses (200–299) - assures you that thing went fine
- Redirection messages (300–399) - if you go to http://google.com you’ll be redirected to https://google.com through such status codes
- Client error responses (400–499) - The problem is caused by the user
- Server error responses (500–599) - The problem is caused by our code

Why do you need networking in DevOps?

You'll setup & configure:

- DNS so that it points to the right server(s)
- Allow access to certain protocols & ports on your server (http/80, https/443, ssh/22, …) or certain IP addresses.
- TLS (securing client-server connection) using a certificate and a key.
- Load balancers to distribute the load on your app.
- A lot of times you'll have to deal with http.

That’s it!

- This is all what we need for now.
- If you want to learn more, you can read about:
  - HTTP; headers & status codes
  - Subnets & CIDRs
  - Load Balancers
  - WAFs

02 - Linux Intro

Intro to DevOps

Why Linux?

- 90% of all cloud infrastructure operates on Linux
- Free Open Source Software; free as in freedom, not just cost-wise
- Built on top of very simple concepts
- Extensibility

Concepts to be aware of

- Files – Everything in linux is a file (and their names are case-sensitive)
- Normal Files - just like **`hello.txt`** or **`index.html`**
- Directory - just like **`/home/mydevopscourse/`**
- Package Manager - `apt, yum, apk, pacman, yay, ...`
  - Used to install anything you want, from a programming library to a web browser.
- Root; aka superuser
- Shell; it's a command-line interface for running programs
  - Most famous ones are **`bash`** and **`zsh`**
- Terminal; it's a GUI program that hosts the shell

What are you we doing today?

- Filesystem operations & navigation
- File permissions
- Data manipulation
- General commands
- Redirection and Pipes
- Environment variables
- System commands

Enough talking, let’s go to the command-line