Web Developer

Foundation knowledge + FrontEnd + BackEnd + DevOps = Full Stack

<https://github.com/kamranahmedse/developer-roadmap>

# Foundation Knowledge

## Internet

#### - How does the internet works ?

The Internet is a global network of computers connected to each other which communicate through a standardized set of protocols.

○ the internet explained

○ how does the internet work ?

#### - What is HTTP ?

HTTP is the TCP/IP based application layer communication protocol which standardizes how the client and server communicate with each other. It defines how the content is requested and transmitted across the internet.

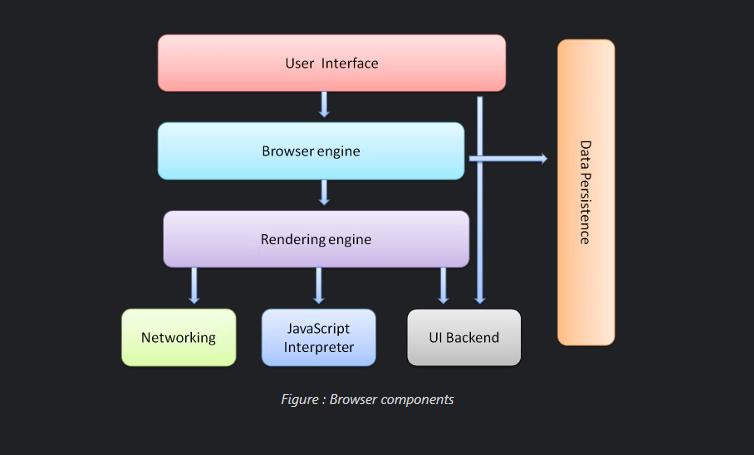
○ what is HTTP

○ an overview of HTTP

#### - Browsers and how they works ?

A web browser is a software application that enables a user to access and display web pages or other online content through its graphical user interface.

○ How Browsers work

○

#### - DNS and how it works ?

The Domain Name System (DNS) is the phonebook of the Internet. Humans access information online through domain names, like nytimes.com or espn.com. Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to IP addresses so browsers can load Internet resources.

○ what is DNS ?

○

#### - what is Domain Name ?

A domain name is a unique, easy-to-remember address used to access websites, such as ‘google.com’, and ‘facebook.com’. Users can connect to websites using domain names thanks to the DNS system.

○ what is a domain name

○ what is a domain name ? domain name vs. Url

A uniform resource locator (URL), sometimes called a web address, contains the domain name of a site as well as other information, including the transfer protocol and the path. For example, in the URL ‘https://cloudflare.com/learning/’, ‘cloudflare.com’ is the domain name, while ‘[https](https://www.cloudflare.com/learning/ddos/glossary/hypertext-transfer-protocol-http/)’ is the protocol and ‘/learning/’ is the path to a specific page on the website.

#### - what is hosting ?

Web hosting is an online service that allows you to publish your website files onto the internet. So, anyone who has access to the internet has access to your website.

## Configuration

### Version Control Systems

- Repo hosting servies

GitHub, GitLab, Bitbucket

- SSH

- Basic Terminal Usage

### Web Security Knowledge

#### - HTTP/HTTPS and APIs

HTTPS is a secure way to send data between a web server and a browser.

○ what is https ?

○ why https matters

○ enabling https on your servers

#### - Content Security Policy

Content Security Policy is a computer security standard introduced to prevent cross-site scripting, clickjacking and other code injection attacks resulting from execution of malicious content in the trusted web page context.

○ MDN - content security policy ( CSP )

○ google devs ( content security policy )

#### - CORS

Cross-Origin Resource Sharing (CORS) is an HTTP-header based mechanism that allows a server to indicate any origins (domain, scheme, or port) other than its own from which a browser should permit loading resources.

- OWASP Security Risks

OWASP or Open Web Application Security Project is an online community that produces freely-available articles, methodologies, documentation, tools, and technologies in the field of web application security.

- Learn to Research

- Data Structures and Algorithms

- Character Encoding

- Design Patterns

# FrontEnd

## Learn the Basics

https://www.knowledgehut.com/blog/web-development/front-end-developer-roadmap

### Learn HTML

<https://www.knowledgehut.com/blog/web-development/front-end-developer-roadmap>

- Learn the basics and how to write Semantic HTML

- Form and Validations

- Conventions and Best Practices

- Accessibility

- SEO Basics

### Basic of CSS

- Learn the basics of CSS

- Making Layouts

○ Floats

○ Positioning

○ Display

○ Box Model

○ CSS Grid

○ Flex Box

- Responsive Design and media Queries

- Style the HTML Pages that you made in last step

### Basics of JavaScript

- Learn the syntax and basic constructs

- Learn how to manipulate DOM

- Learn Fetch API / Ajax ( XHR )

- Learn Es6 + new features and writing modular JavaScript

- Understand the concepts such as Hoisting, Event bubbling, Scope, Prototype, shadow DOM, strict etc

## 2) CSS Frameworks & Processors

### Responsive Web

### Choose Framework

- Foundation

- Bootstrap \*

- Reactstrap \*

- Material UI \*

- Tailwind CSS \*

- Chakra UI \*

- Materialize CSS

- Semantic UI

- Bulma

### Preprocessors

- Sass

- Less

- PostCSS

- Stylus

## **3) JS Client Frameworks + Tools**

### ES6

### Build Tools

- Task Runner

○ npm scripts

○ Gulp

- Module Bundlers

○ Webpack

○ Rollup

○ Parcel

- Module Loaders

○ Webpack

○ Require JS/ AMD

○ Browserify

- Linters and Formatters

○ Prettier

○ ESLint

○ StandardJS

### Package Manager

- npm

- yarn

### Choose a Framework

- Angular

○ RxJS

○ NgRx

- React

○ Flux

○ Redux

○ MobX

- Vue.js

○ VueX

- Ember JS

- Preact

- Inferno

### Testing

- Jest

- Mocha

- Jasmine

### Type Checkers

- TypeScript

- Flow

### Progressive Web Apps

- Learn different Web APIs used in PWAs

○ Storage

○ Web Sockets

○ Service Workers

○ Location

○ Notifications

○ Device orientation

○ payments

○ Credentials

- Calculating, Measuring and improving performance

○ PRPL Pattern

○ RAIL Model

○ Performance Metrics

○ Using Lighthouse

○ Using Dev Tools

### Server Side Rendering ( SSR )

- React.js

○ Next.js

○ After.js

- Angular ( Universal )

- Vue.js

○ Nuxt.js

### Static Site Generators

- Next,js, GatsbyJS, Nuxt.js, Vuepress, Jekyll, Hugo

### Mobile Applications

- React Native, NativeScript, Flutter, Ionic

### Desktop applications

- Electron, NWjs, Carlo, Proton Native

## 4) CSS Mastery, Graphics & Visualization

### CSS

- Methodologies

○ SUITCSS

○ BEM

○ OOCSS

○ SMACSS

○ Systematic CSS

- Mordern CSS

○ Styled components

○ CSS Modules

○ Styles JSX

○ Emotion

○ Radium

○ Glamorous

- Dive Deep CSS3

○ Flexbox

○ Gradients

○ Grids

○ Rotate

○ Transform

○ Skew

○ Scale

○ Transitions

○ …etc

### SVG

- D3, paper, raphael,

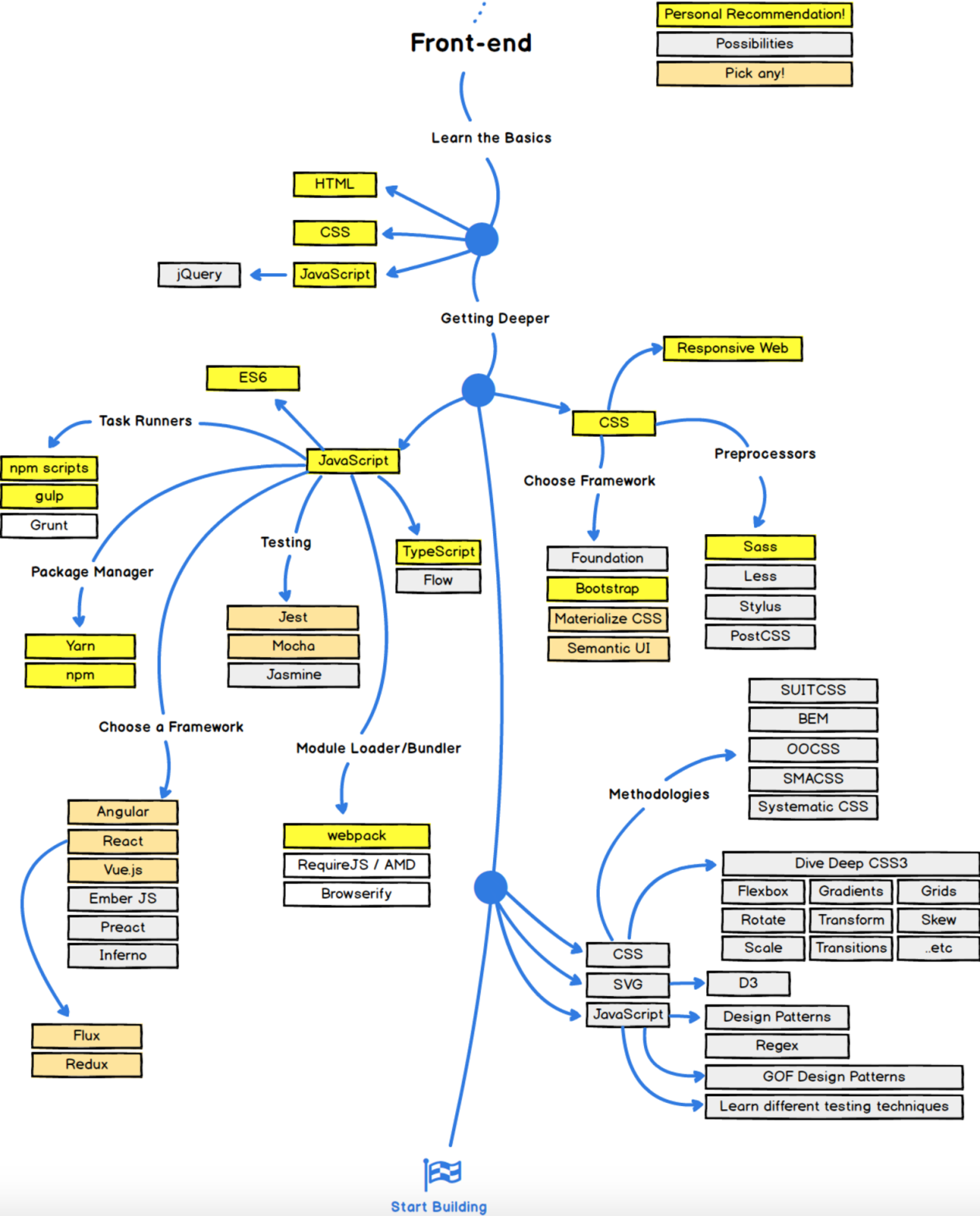
### JavaScript

- Design Patterns

- Regex

- GOF Design Patterns

- Learn different testing techniques



# Backend

## OS and General Knowledge

### Terminal Usage

### How OSs work in General

### Process Management

### Threads and Concurrency

### Basic Terminal Commands

- grep, awk, sed, isof, curl, wget, tail, head, less, find, ssh, kill

### Memory Management

### Interprocess Communication

### I/O Management

### POSIX Basics

- stdin, stdout, stderr, pipes

### Basic Networking Concepts

## Lean a Language

### PHP

### C#

### JAVA

### JavaScript

### Python

### Ruby

### Rust

### Go

## Node.js

### Package Manager

- npm

- yarn

### Framework

- Express

- hapi

- Koa

- Sails.js

### Testing

- Jest

- Mocha

- Jasmine

- Chai

- should.js

## Infrastructure + Key Techniques

### Web Server

- Coddy

- Apache

- Nginx

### RESTful APIs

- JSON Web Token ( JWT )

- OAuth 2.0

### Read about MVC

### Authentication

### SOLID, YAGNI, KISS, …

### Regular Expressions

### Security

### GraphQL

- Apllo

- Relay Modern

### Docker

## Database

### Caching

- Memcached

- Redis

### Relational Databases

- Oracle

- PostgreSQL

- MariaDB

- MySQL

- MS SQL

### NoSQL Databases

- MongoDB

- RethinkDB

- CouchDB

- DynamoDB

## Patterns

### Search engines

- Solr

- Shpinx

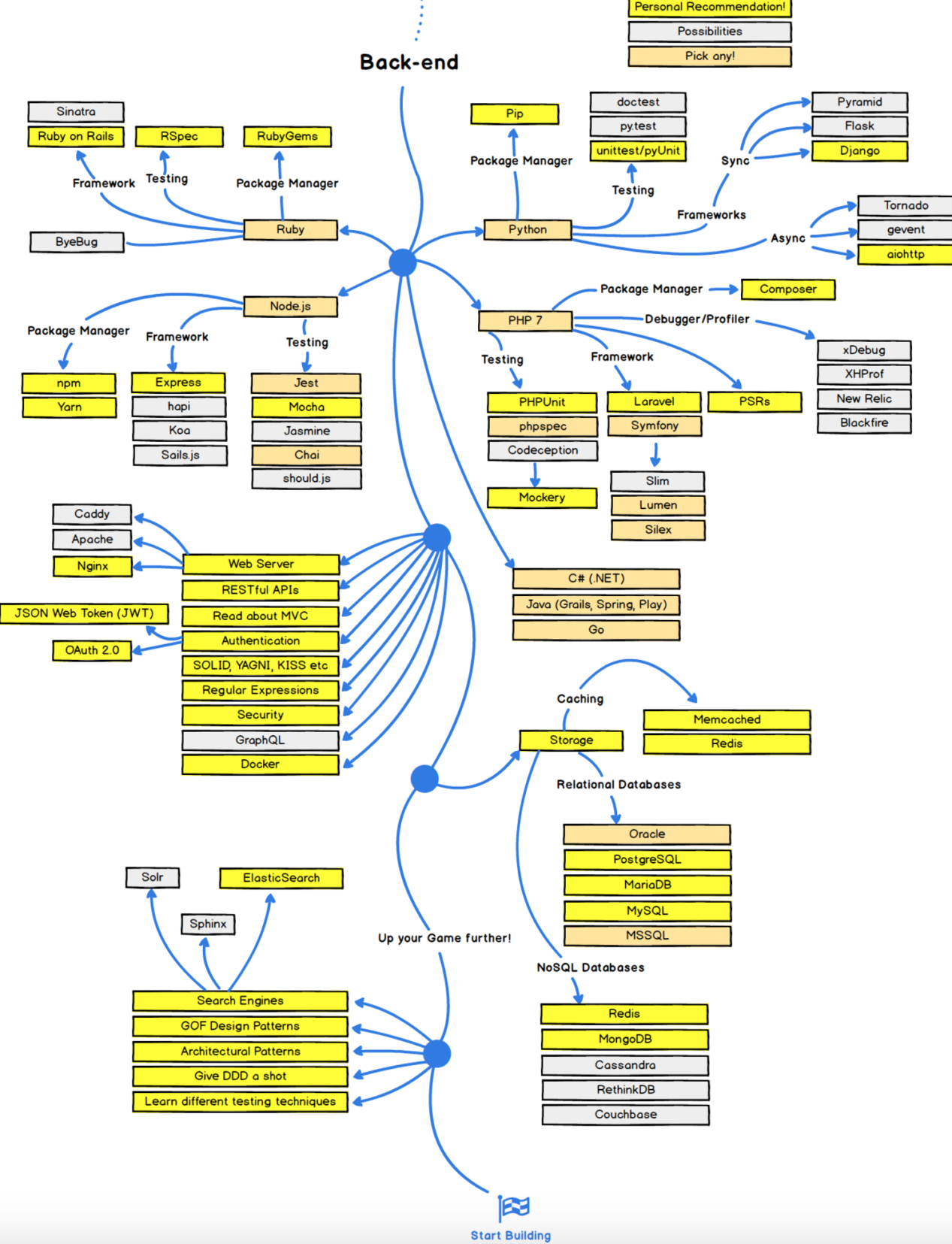
- ElasticSearch

### GOF Design Patterns

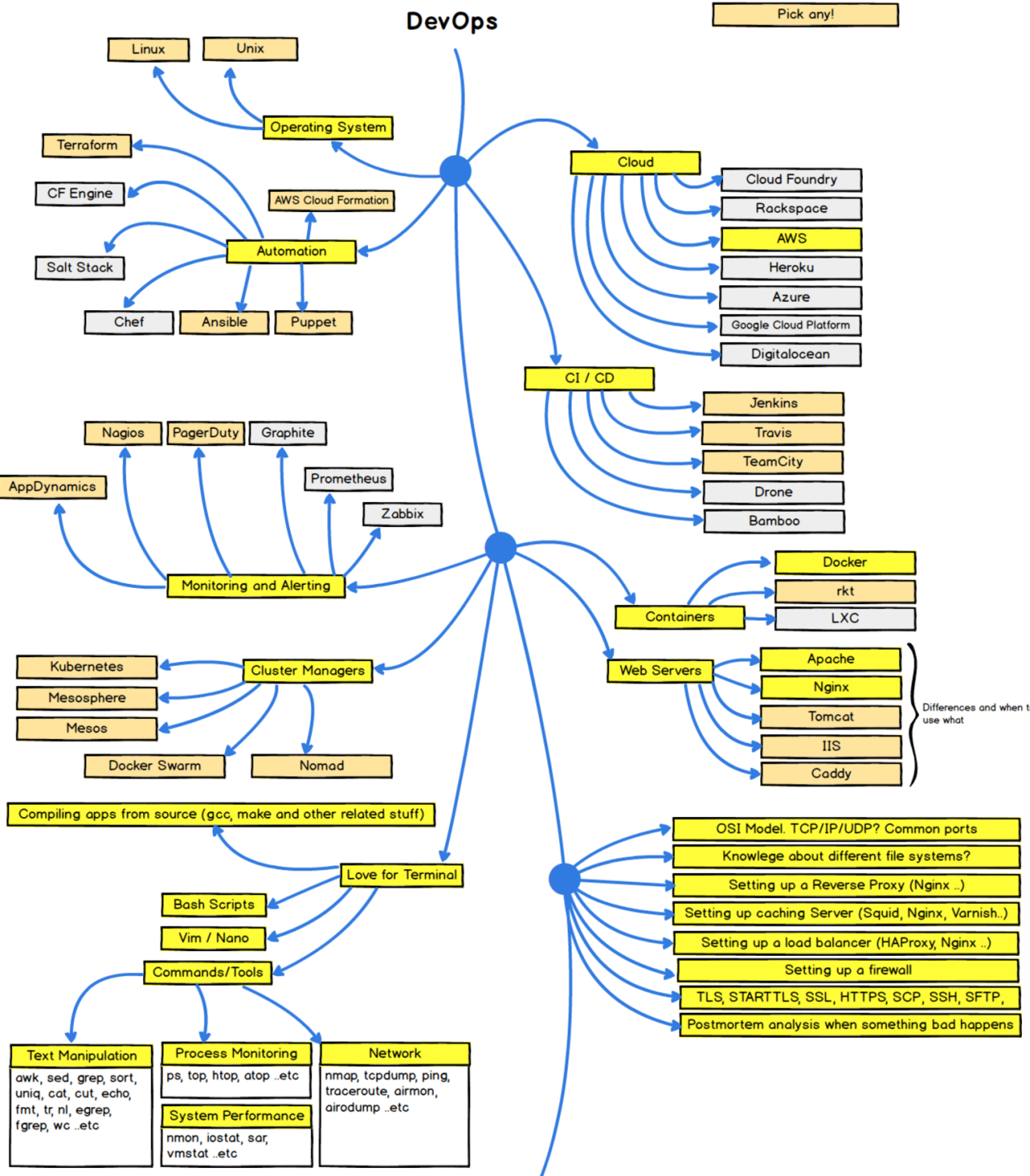
### Architectural Patterns

### Give DDD a shot

### Learn different testing techniques



# DevOps



1. Xampp
2. Wamp