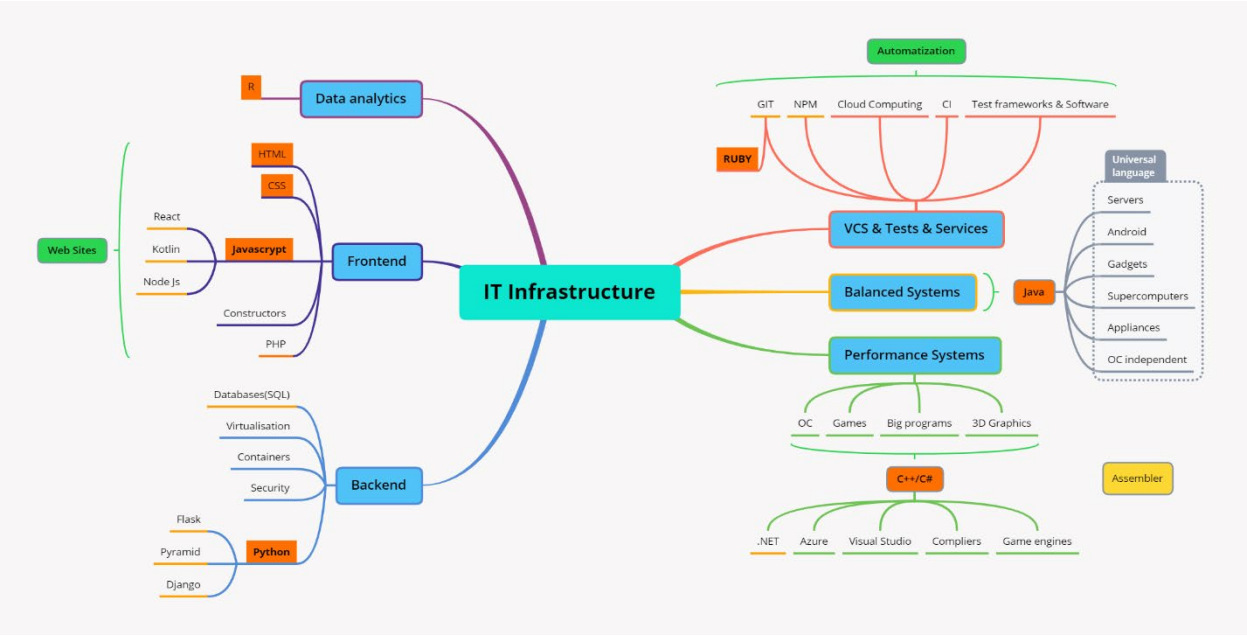
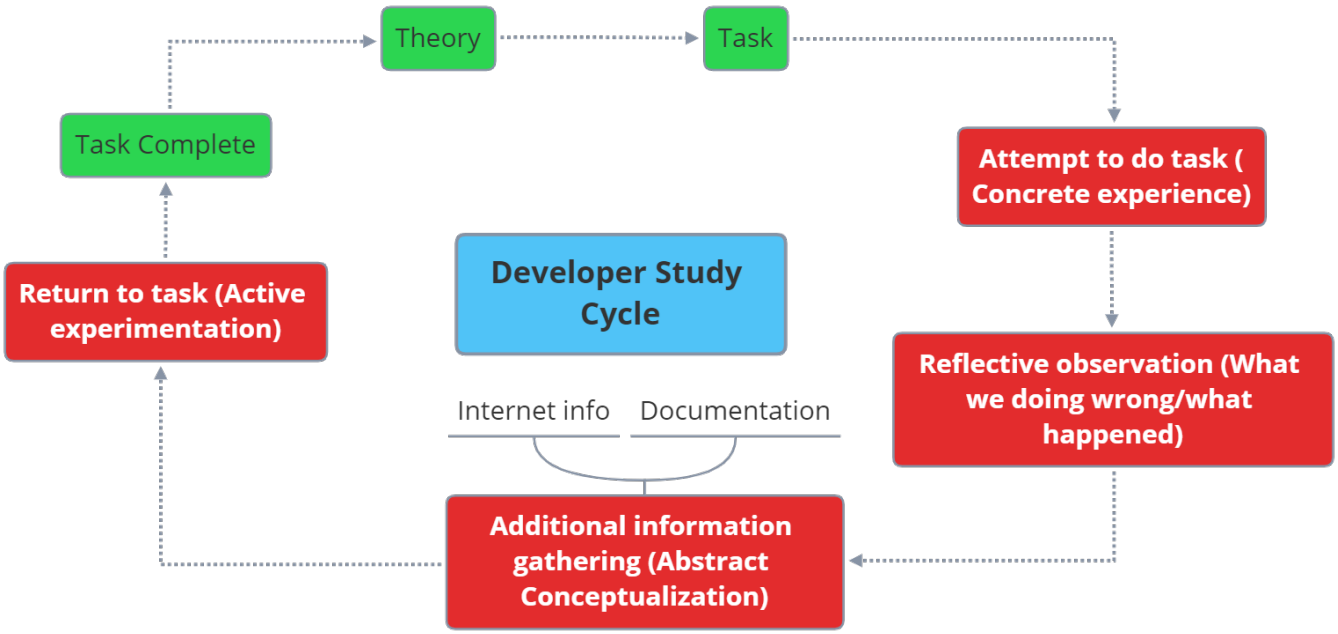
QA Engineer Mind maps

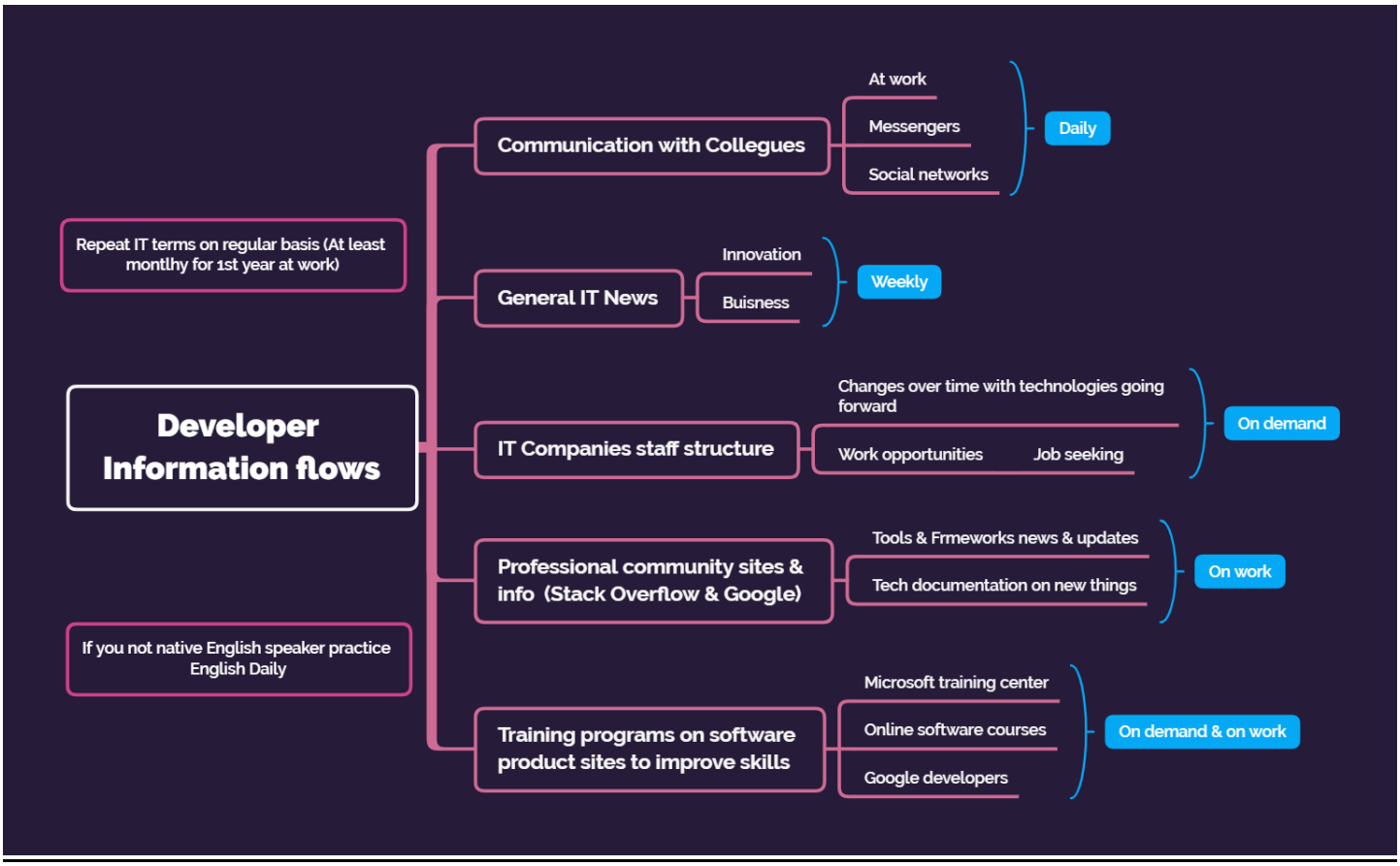
Table of contents:

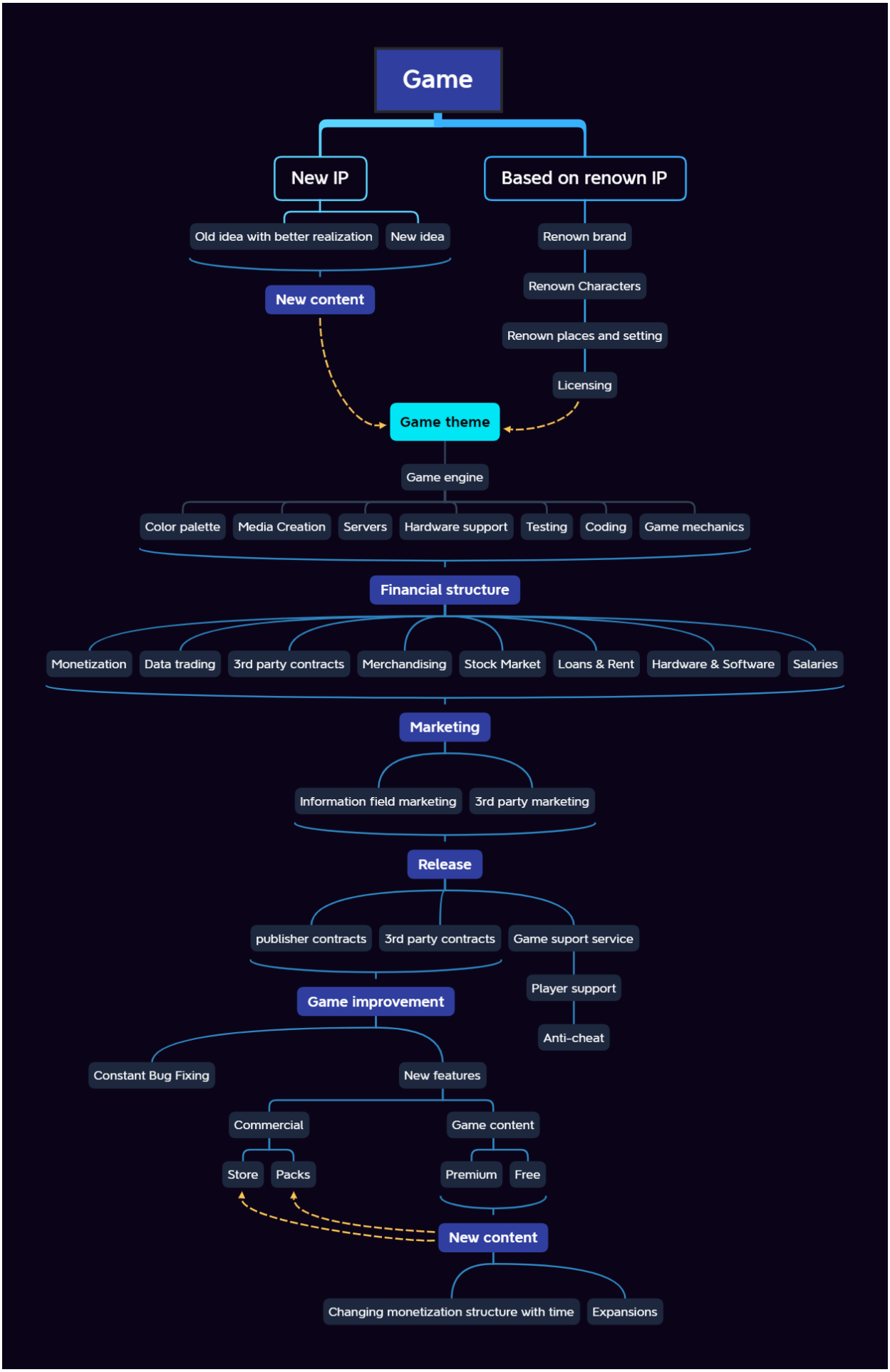
1. IT general
2. QA Theory
3. Software
4. HR
5. Math
6. Code structure
7. SQL and Data
8. Tests writing
9. Pipelines and Dev Ops
10. Aws Azure
11. Coclusion

1. IT general

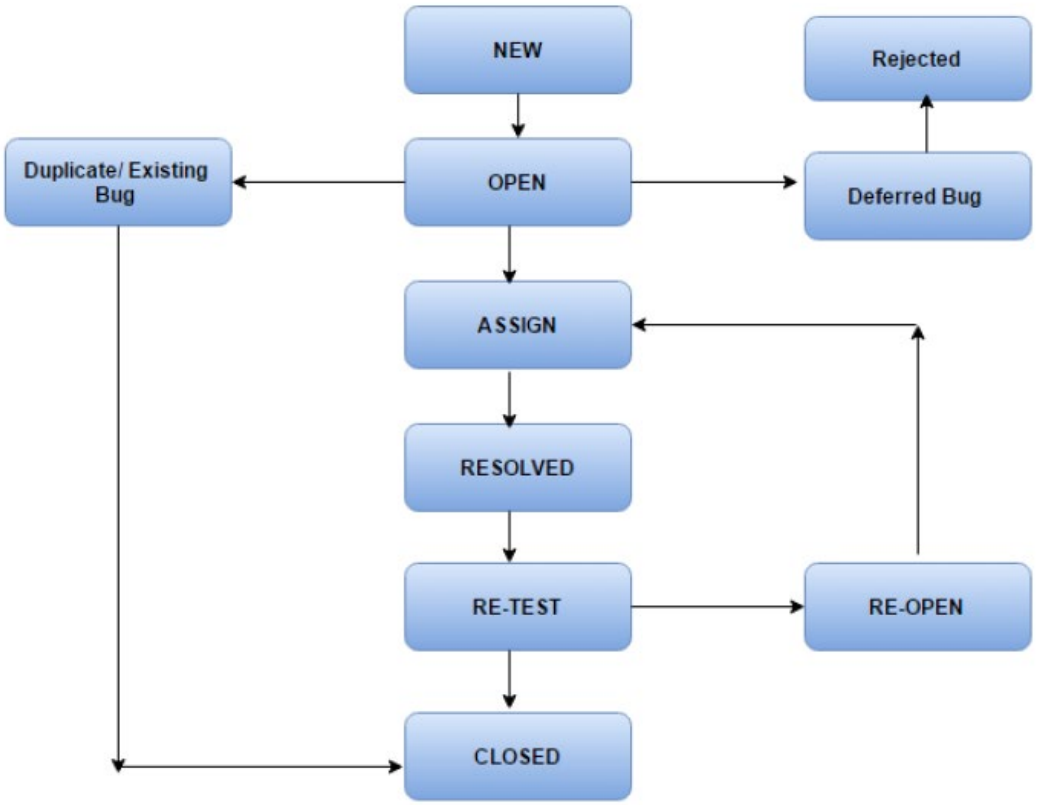



Theory
Task
Task Complete
Attempt to do task (Concrete experience)
Return to task (Active experimentation)
Developer Study Cycle
Reflective observation (What we doing wrong/what happened)
Internet info
Documentation
Additional information gathering (Abstract Conceptualization)

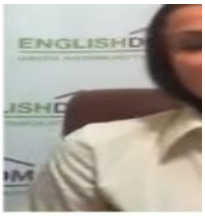

Developer Information flows
Repeat IT terms on regular basis (At least monthy for 1st year at work)
If you not native English speaker practice English Daily
Communication with Collegues
At work
Messengers
Social networks
Daily
General IT News
Innovation
Buisness
Weekly
IT Companies staff structure
Changes over time with technologies going forward
Work opportunities
Job seeking
On demand
Professional community sites & info (Stack Overflow & Google)
Tools & Frmeworks news & updates
Tech documentation on new things
On work
Training programs on software product sites to improve skills
Microsoft training center
Online software courses
Google developers
On demand & on work


Game
New IP
Based on renown IP
Old idea with better realization
New idea
Renown brand
Renown Characters
Renown places and setting
Licensing
New content
Game theme
Game engine
Color palette
Media Creation
Servers
Hardware support
Testing
Coding
Game mechanics
Financial structure
Monetization
Data trading
3rd party contracts
Merchandising
Stock Market
Loans & Rent
Hardware & Software
Salaries
Marketing
Information field marketing
3rd party marketing
Release
publisher contracts
3rd party contracts
Game suport service
Player support
Anti-cheat
Game improvement
Constant Bug Fixing
New features
Commercial
Game content
Store
Packs
Premium
Free
New content
Changing monetization structure with time
Expansions

ЖИЗНЕННЫЙ ЦИКЛ ДЕФЕКТА



STLC – ЖИЗНЕННЫЙ ЦИКЛ ТЕСТИРОВАНИЯ ПО



К незнакомому человеку:

- Dear Sir/Madam,
- Dear Mr/Miss/Mrs (фамилия),
- To whom it may concern,

Коллеге или человеку, с которым вы уже вели переписку в прошлом:

- Dear (имя),
- Hello (имя),

Обязательно вступительное слово:

- Hope you are well!
- Hope everything is fine on your side.
- How was your holiday? Hope you are feeling better.
- Thank you for getting back to me so soon/ Thank you for your prompt response.

2. QA Theory

Чек лист- пример

Проверка	Результат		
	Win XP	XP SP4	Win Vista
Операции с файлами	ok	ok	ok
Создание файла	ok	ok	ok
Открытие файла	ok	ok	ok
Сохранение документа	ok	ok	ok
Печать	ok	ok	ok
Редактирование файлов	bugs	bugs	bugs
Отмена	ok	ok	ok
Копирование	ok	ok	ok
Вырезание	bug #146	bug #146	bug #146
Вставка	ok	ok	ok
Удаление	ok	ok	ok
Поиск	bug #123	bug #133	ok
Поиск с заменой	bug #126	ok	ok

Шаблон тест плана

Что надо тестировать? - описание объекта тестирования: системы, приложения, оборудования

Что будете тестировать? список функций и описание тестируемой системы и её компонент в отдельности

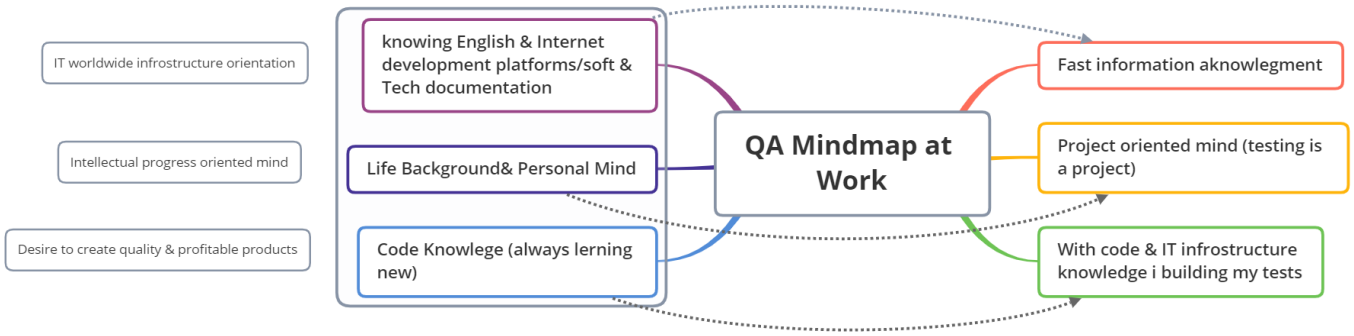
Как будете тестировать? - стратегия тестирования, а именно: **виды тестирования** и их применение по отношению к объекту тестирования

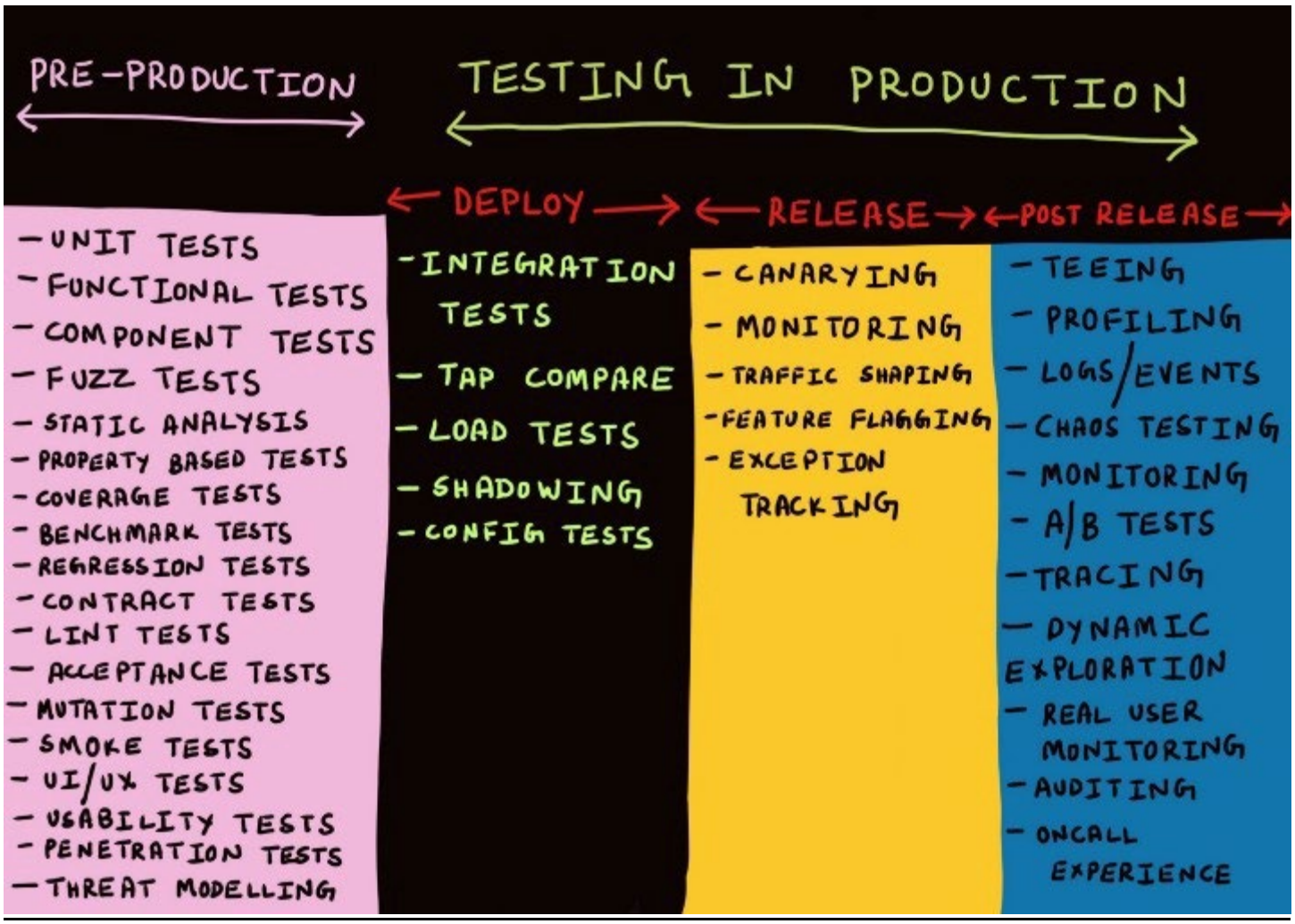
Когда будете тестировать? - последовательность проведения работ: подготовка (Test Preparation), тестирование (Testing), анализ результатов (Test Result Analisys) в разрезе запланированных фаз разработки

Критерии начала тестирования: -готовность тестовой платформы (тестового стенда) , законченность разработки требуемого функционала , наличие всей необходимой документации

Критерии окончания тестирования:

результаты тестирования удовлетворяют критериям качества продукта:


IT worldwide infrostructure orientation
Intellectual progress oriented mind
Desire to create quality & profitable products
knowing English & Internet development platforms/soft & Tech documentation
Life Background& Personal Mind
Code Knowlege (always lerning new)
QA Mindmap at Work
Fast information aknowlegment
Project oriented mind (testing is a project)
With code & IT infrostructure knowledge i building my tests

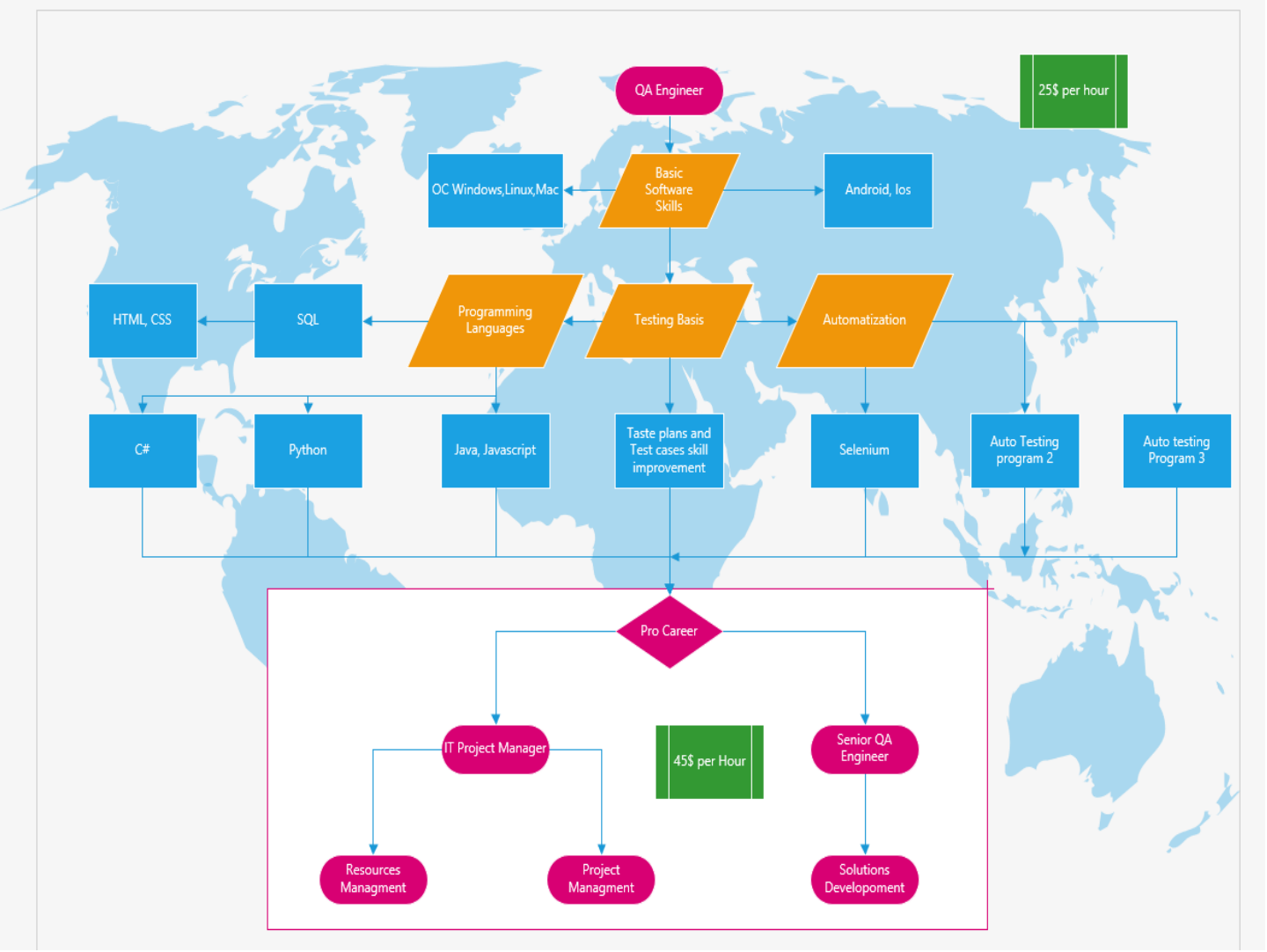

PRE-PRODUCTION
TESTING IN PRODUCTION
DEPLOY
RELEASE
POST RELEASE
- UNIT TESTS
- FUNCTIONAL TESTS
- COMPONENT TESTS
- FUZZ TESTS
- STATIC ANALYSIS
- PROPERTY BASED TESTS
- COVERAGE TESTS
- BENCHMARK TESTS
- REGRESSION TESTS
- CONTRACT TESTS
- LINT TESTS
- ACCEPTANCE TESTS
- MUTATION TESTS
- SMOKE TESTS
- UI/UX TESTS
- USABILITY TESTS
- PENETRATION TESTS
- THREAT MODELLING
- INTEGRATION TESTS
- TAP COMPARE
- LOAD TESTS
- SHADOWING
- CONFIG TESTS
- CANARYING
- MONITORING
- TRAFFIC SHAPING
- FEATURE FLAGGING
- EXCEPTION TRACKING
- TEEING
- PROFILING
- LOGS/EVENTS
- CHAOS TESTING
- MONITORING
- A/B TESTS
- TRACING
- DYNAMIC EXPLORATION
- REAL USER MONITORING
- AUDITING
- ONCALL EXPERIENCE

Пример тест кейса

Атрибуты тест-кейса

1) Номер (ID)
2) Название (Summary/Name
3) Предусловие (PreConditions)
4) Шаги тест кейса и описание (Steps and Descriptions)
5) Ожидаемый результат (Expected result)
6) Пост-условие (PostConditions)
7) Автор (Designer)
8) Статус (Status)
9) Дата создания (Created)

Test Name:		
Status:		
Created Date:		
Designer:		
Pre Conditions:		
Steps	**Description**	**Expected Result**
Step 1		
Step 2		
Step 3		
Step 4		
Post Conditions:		



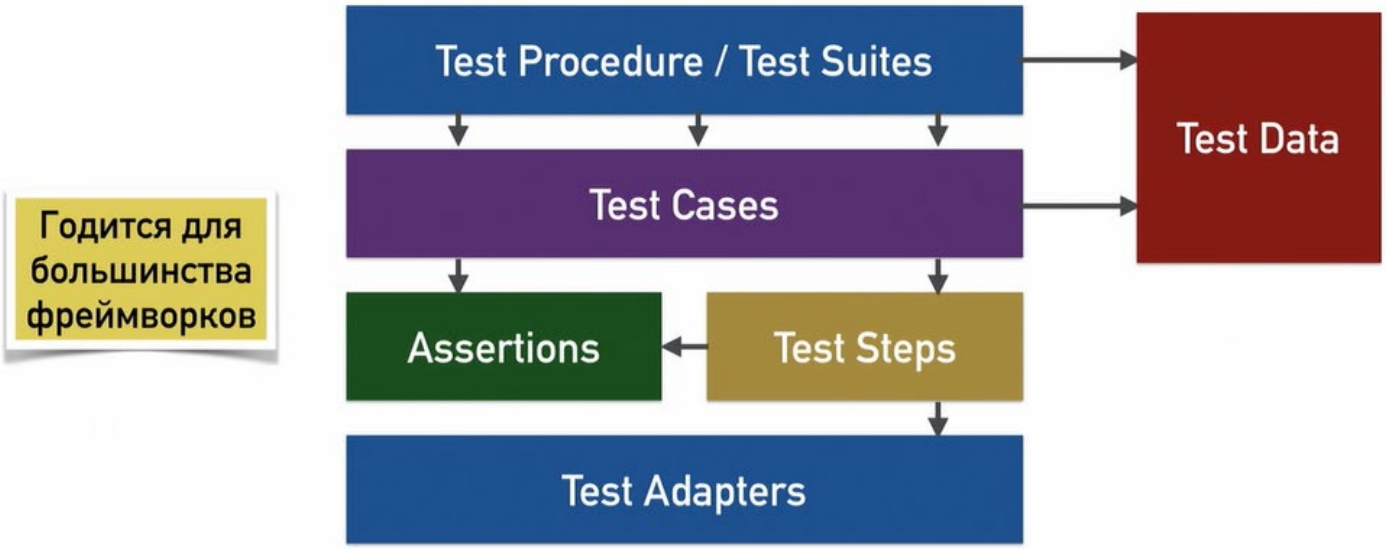
Traceability matrix

Матрица прослеживаемости требований (Requirements traceability matrix) - это двумерная таблица, содержащая соответсвие функциональных требований (functional requirements) продукта и подготовленных тестовых сценариев (test cases)

Requirement Traceability Matrix

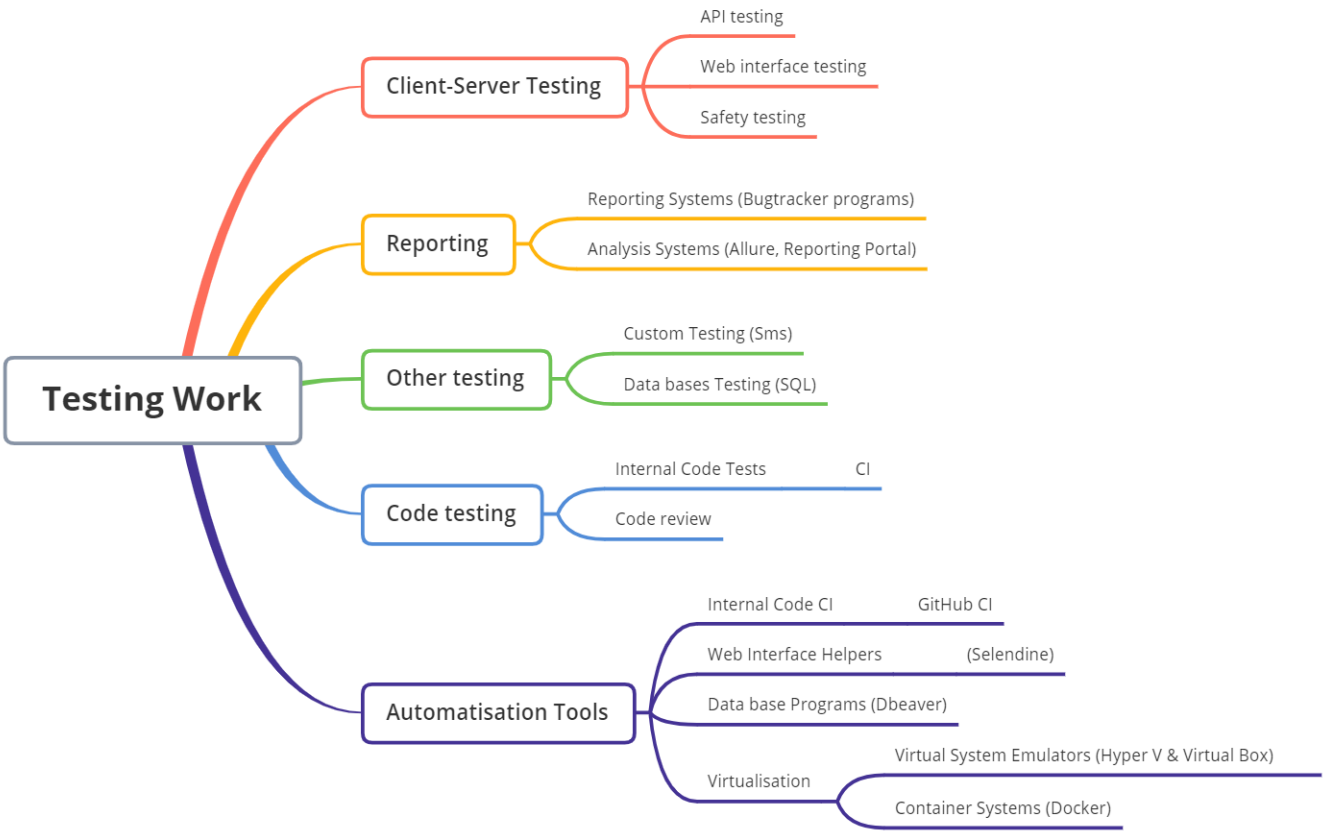
Req. ID \ Test Case ID	TC_1	TC_2	TC_3	TC_4	TC_5	TC_6	TC_7	TC_8	TC_9	TC_10	# Test Cases for respective Requirement
Req_1	✖		✖			✖					3
Req_2		✖			✖						2
Req_3			✖								1
Req_4				✖		✖					2
Req_5					✖		✖				2
Req_6						✖					1
Req_7					✖		✖				2
Req_8								✖			1
Req_9									✖		1
Req_10										✖	1

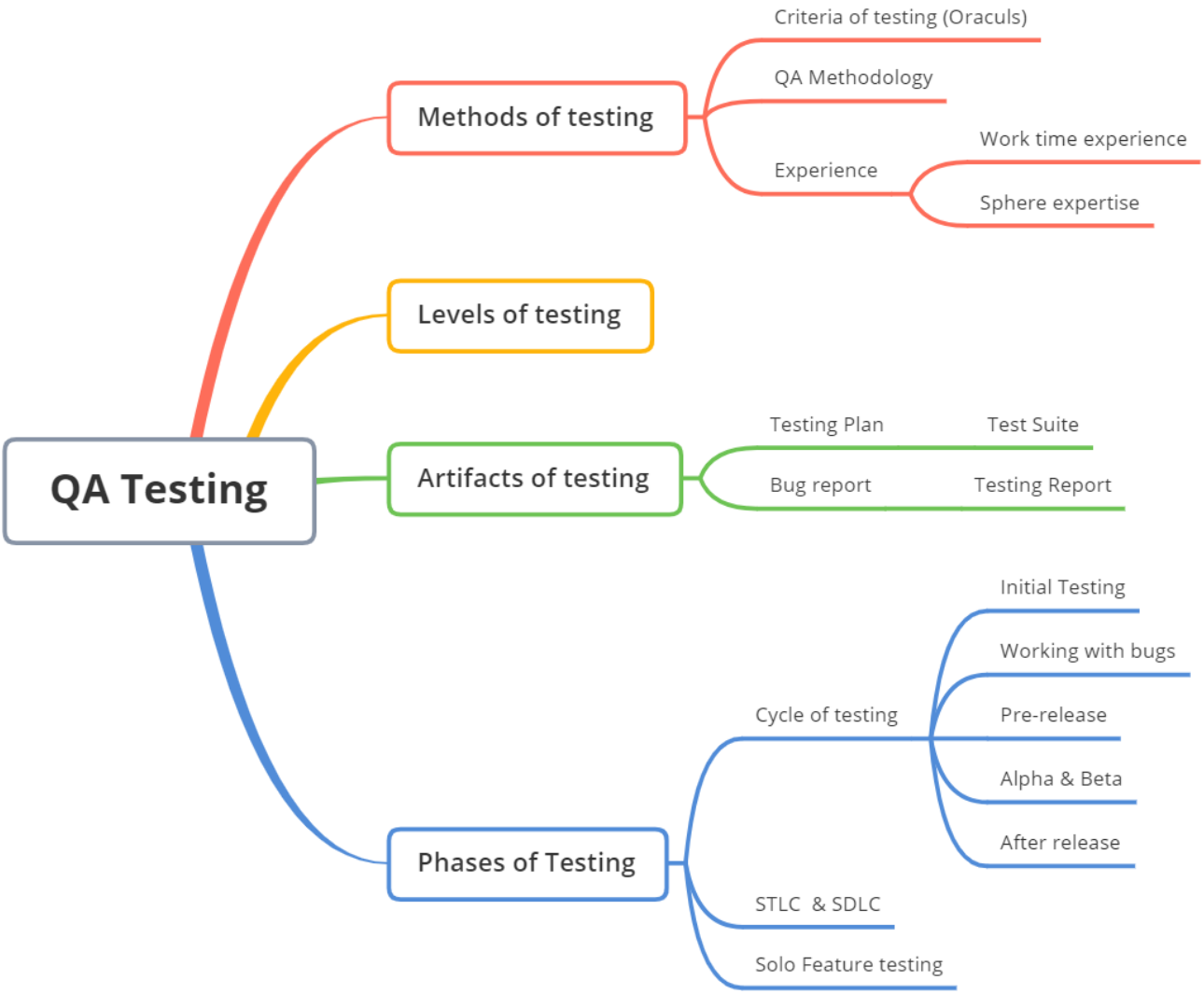
Блоки Построения Тестов

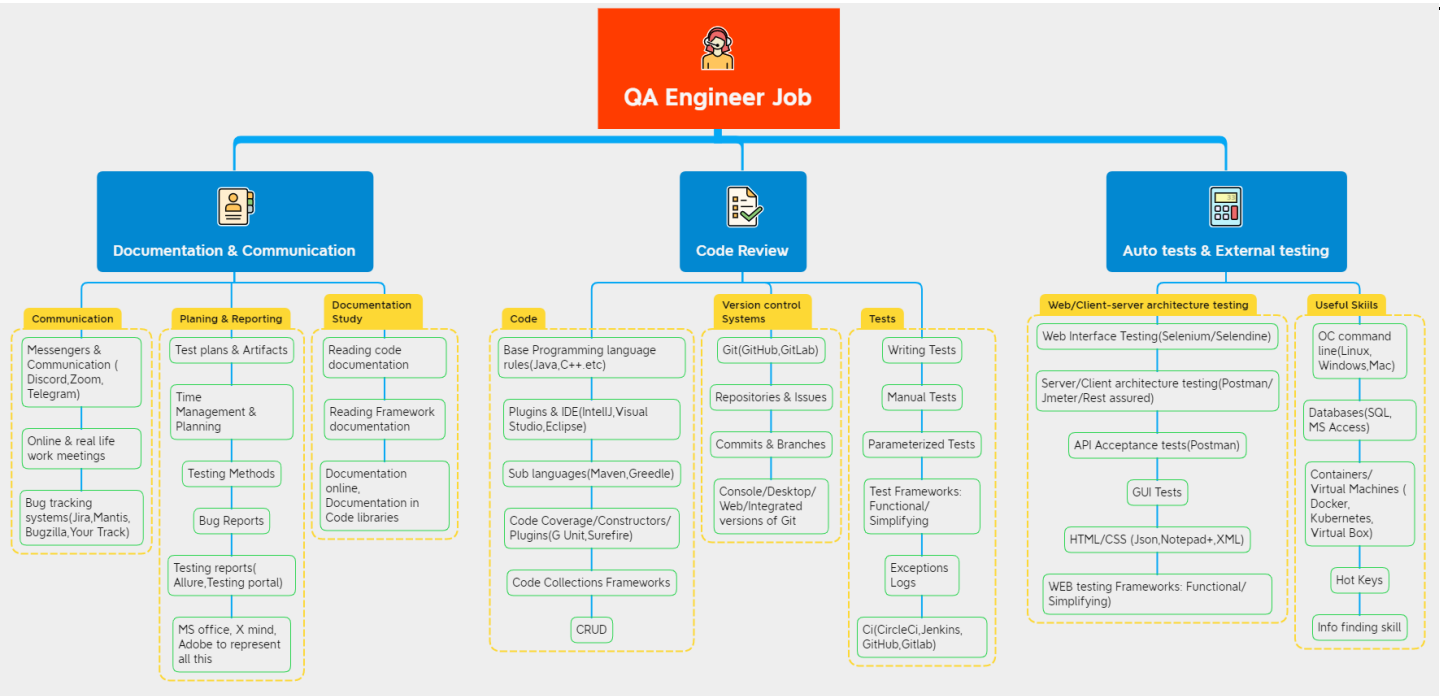


Годится для большинства фреймворков

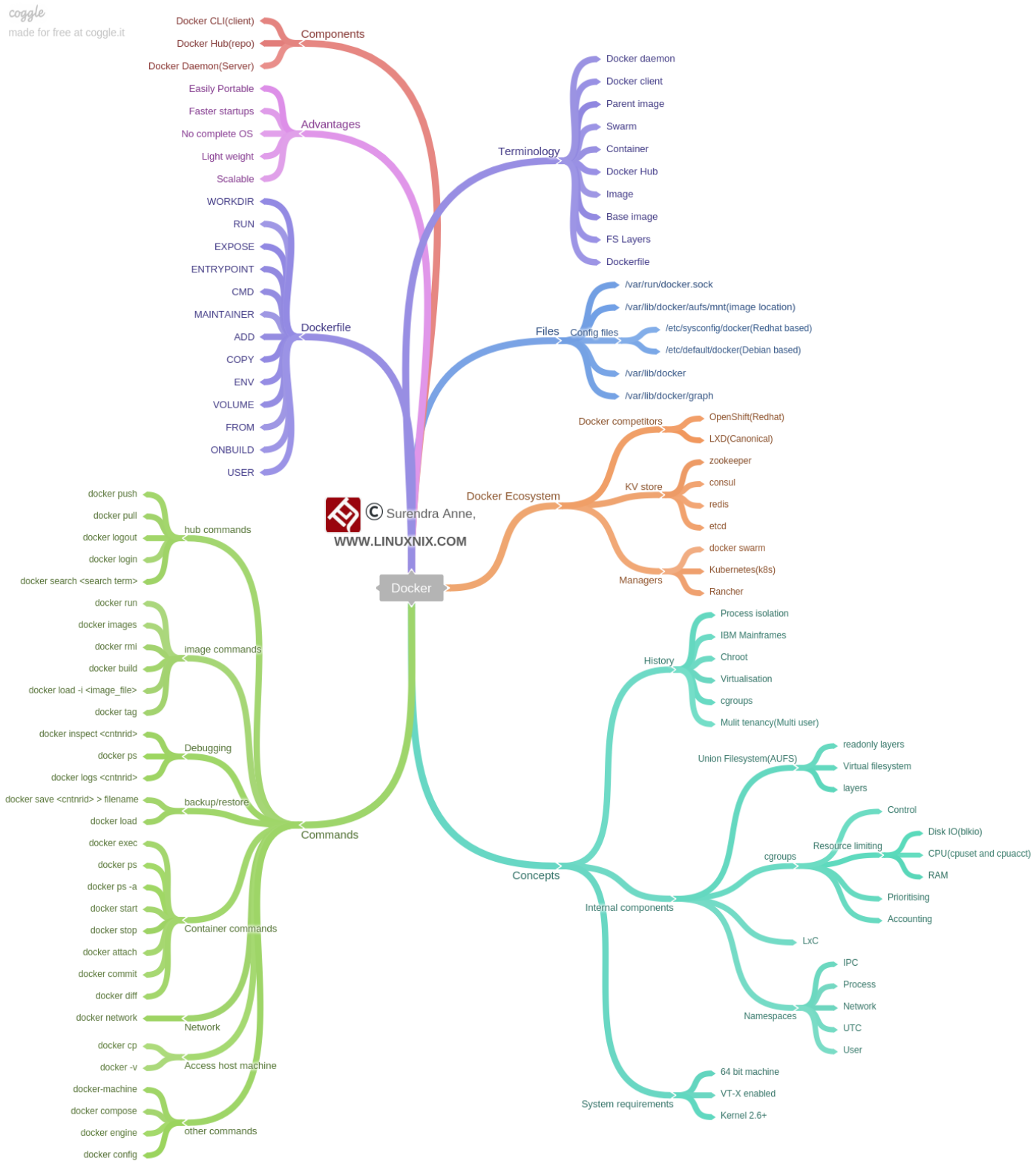
coggle
made for free at coggle.it
Виды тестирования
По знанию системы
Чёрного ящика
Белого ящика
Серого ящика
По позитивности сценария
Позитивное
Негативное
По уровню тестирования
Приемочное
Альфа и бета тестирование
Эксплуатационное
Пользовательское
Контрактное и правовое
Системное
Интеграционное
Модульное
По разработке тестовых сценариев
На основа модели
На основе требований
На основе вариантов использования
По степени автоматизации
Автоматическое
Ручное
Автоматизированное
По исполнению кода
Динамическое
Статическое
Статический анализ кода
Рецензирование исходного кода
Неформальное рецензирование
Сквозной контроль
Технический анализ
Инспекция
По целям
Не функциональное
Совместимости
Исталляционное
UI
Защищённости
Конфигурационное
Надёжности и восстановления после сбоев
Локализации
Производительности
Тестирование изменений
Регрессионное
Подтверждающее
Структурное/архитектурное
Функциональное

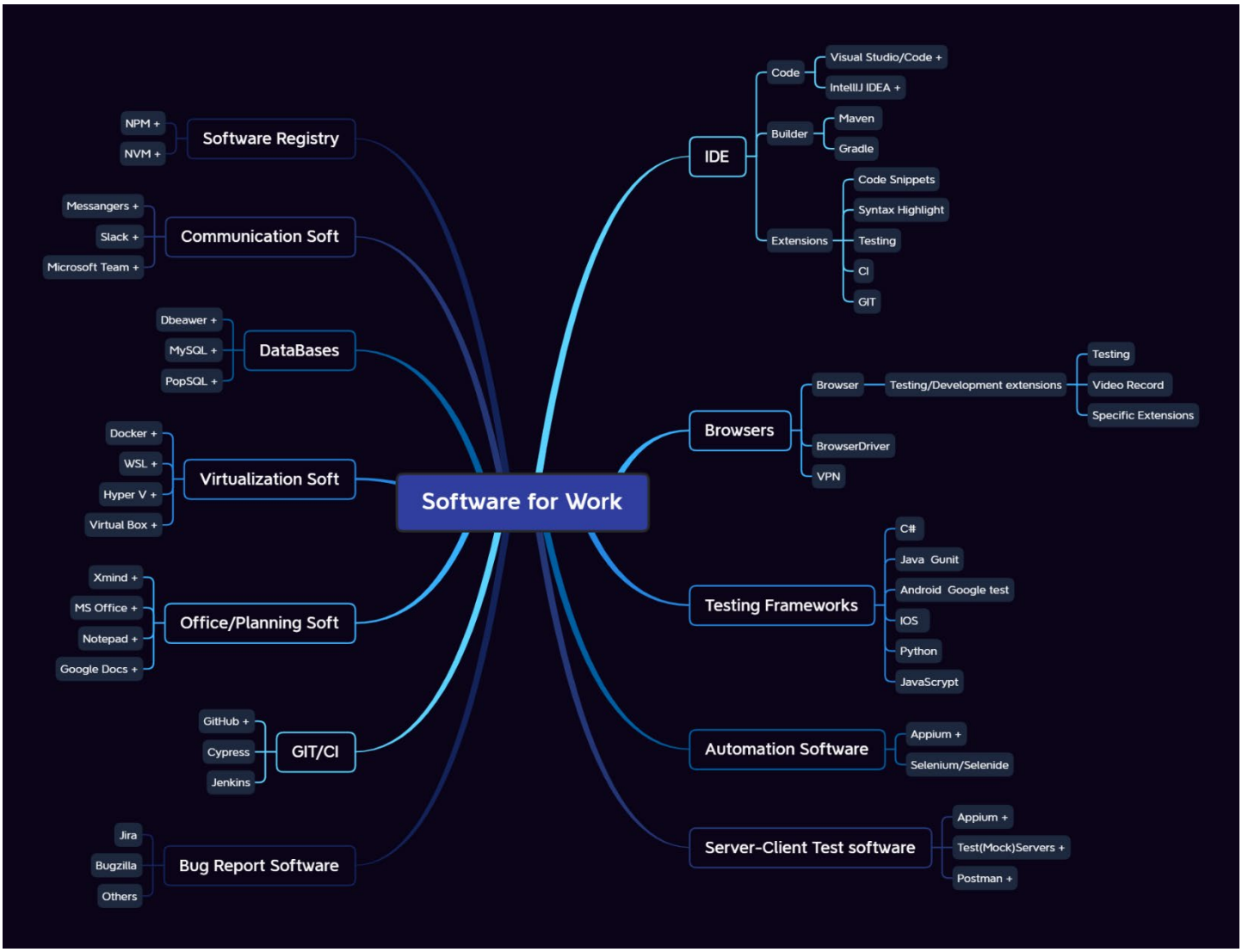

Testing Work
Client-Server Testing
API testing
Web interface testing
Safety testing
Reporting
Reporting Systems (Bugtracker programs)
Analysis Systems (Allure, Reporting Portal)
Other testing
Custom Testing (Sms)
Data bases Testing (SQL)
Code testing
Internal Code Tests
CI
Code review
Automatisation Tools
Internal Code CI
GitHub CI
Web Interface Helpers
(Selendine)
Data base Programs (Dbeaver)
Virtualisation
Virtual System Emulators (Hyper V & Virtual Box)
Container Systems (Docker)

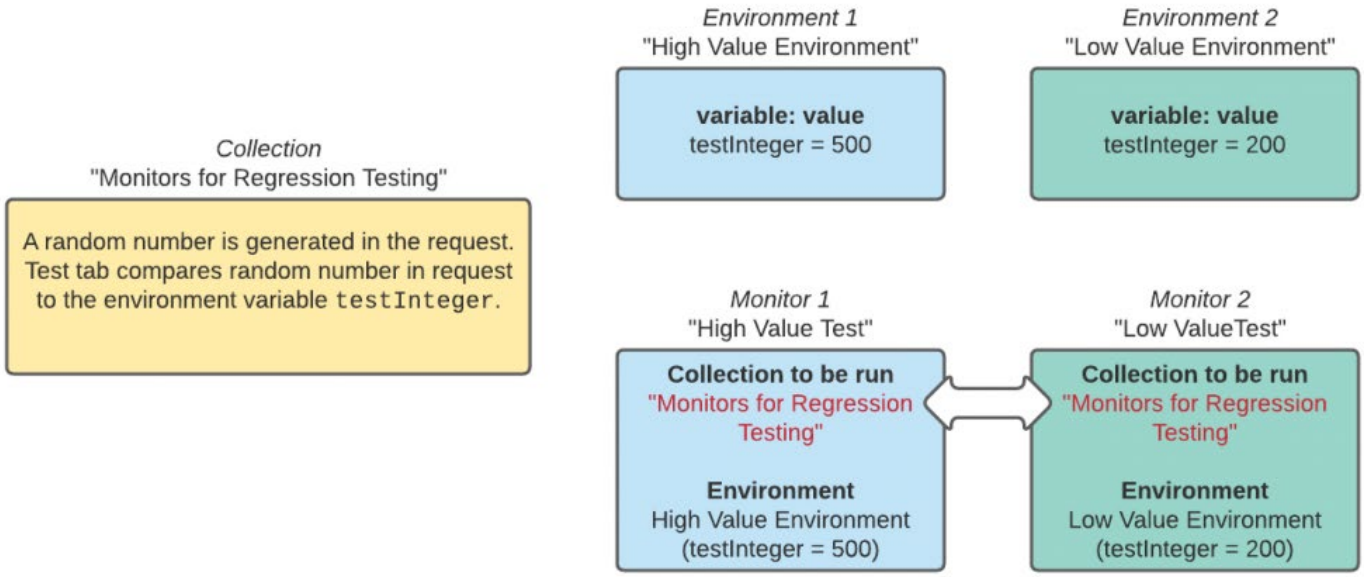

QA Testing
Methods of testing
Criteria of testing (Oraculs)
QA Methodology
Experience
Work time experience
Sphere expertise
Levels of testing
Artifacts of testing
Testing Plan
Test Suite
Bug report
Testing Report
Phases of Testing
Cycle of testing
Initial Testing
Working with bugs
Pre-release
Alpha & Beta
After release
STLC & SDLC
Solo Feature testing


QA Engineer Job
Documentation & Communication
Code Review
Auto tests & External testing
Communication
Messengers & Communication (Discord,Zoom, Telegram)
Online & real life work meetings
Bug tracking systems(Jira,Mantis, Bugzilla,Your Track)
Planing & Reporting
Test plans & Artifacts
Time Management & Planning
Testing Methods
Bug Reports
Testing reports(Allure,Testing portal)
MS office, X mind, Adobe to represent all this
Documentation Study
Reading code documentation
Reading Framework documentation
Documentation online, Documentation in Code libraries
Code
Base Programming language rules(Java,C++.etc)
Plugins & IDE(IntelIJ,Visual Studio,Eclipse)
Sub languages(Maven,Greedle)
Code Coverage/Constructors/ Plugins(G Unit,Surefire)
Code Collections Frameworks
CRUD
Version control Systems
Git(GitHub,GitLab)
Repositories & Issues
Commits & Branches
Console/Desktop/ Web/Integrated versions of Git
Tests
Writing Tests
Manual Tests
Parameterized Tests
Test Frameworks: Functional/ Simplifying
Exceptions Logs
Ci(CircleCi,Jenkins, GitHub,Gitlab)
Web/Client-server architecture testing
Web Interface Testing(Selenium/Selendine)
Server/Client architecture testing(Postman/ Jmeter/Rest assured)
API Acceptance tests(Postman)
GUI Tests
HTML/CSS (Json,Notepad+,XML)
WEB testing Frameworks: Functional/ Simplifying)
Useful Skills
OC command line(Linux, Windows,Mac)
Databases(SQL, MS Access)
Containers/ Virtual Machines (Docker, Kubernetes, Virtual Box)
Hot Keys
Info finding skill

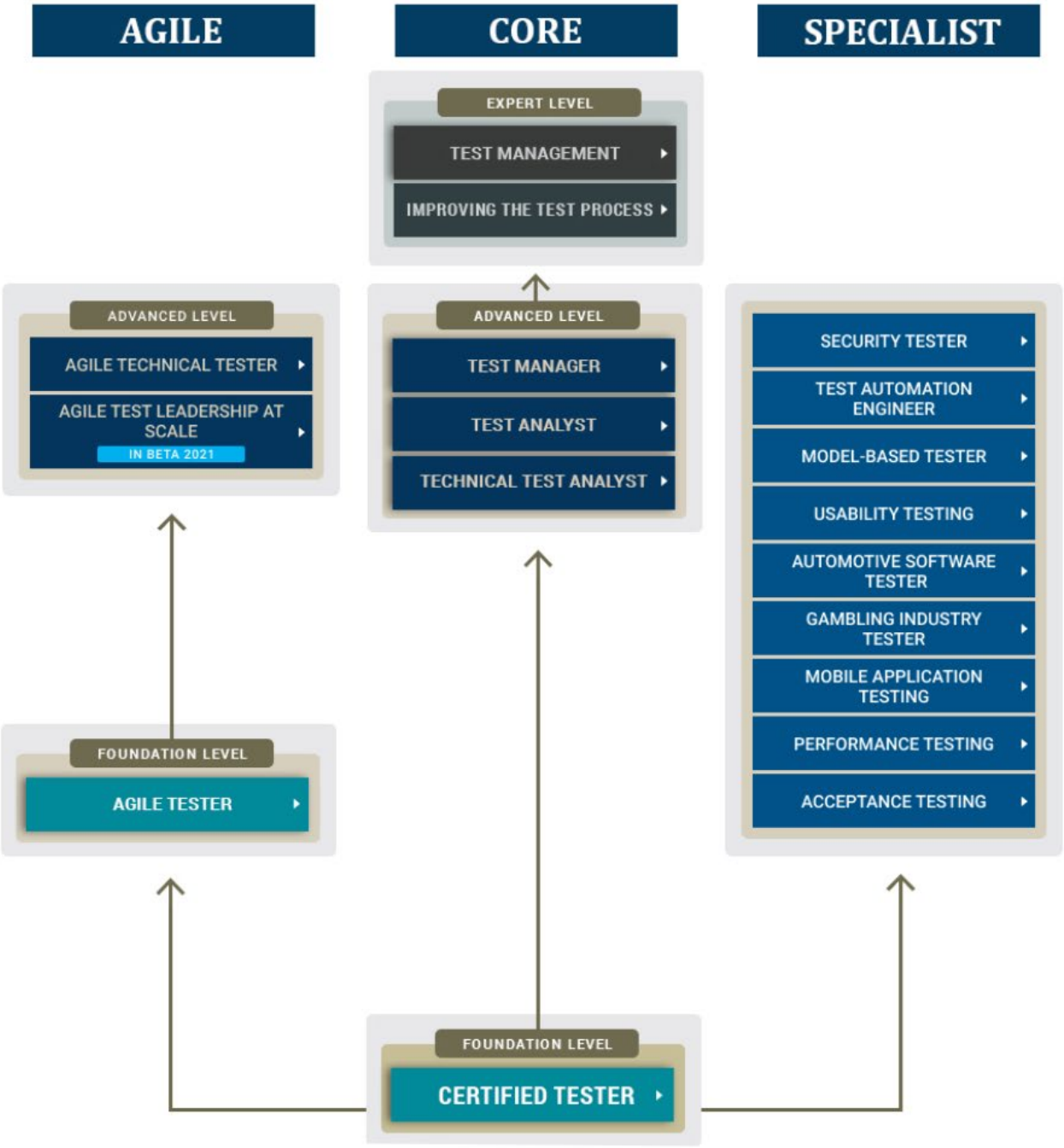
3. Software

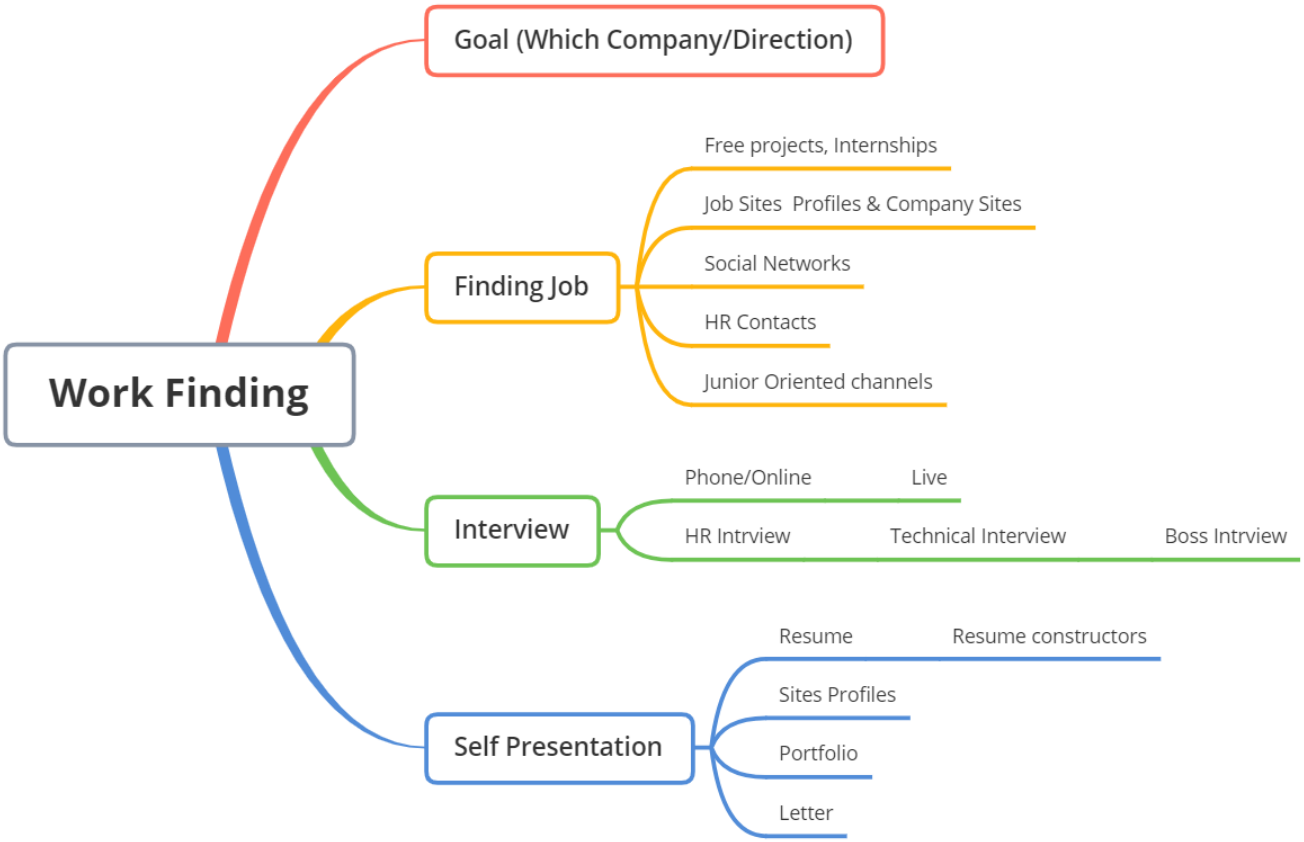

coggle
made for free at coggle.it
Docker
© Surendra Anne,
WWW.LINUXNIX.COM
Components
Docker CLI(client)
Docker Hub(repo)
Docker Daemon(Server)
Advantages
Easily Portable
Faster startups
No complete OS
Light weight
Scalable
Dockerfile
WORKDIR
RUN
EXPOSE
ENTRYPOINT
CMD
MAINTAINER
ADD
COPY
ENV
VOLUME
FROM
ONBUILD
USER
Terminology
Docker daemon
Docker client
Parent image
Swarm
Container
Docker Hub
Image
Base image
FS Layers
Dockerfile
Files
Config files
/var/run/docker.sock
/var/lib/docker/aufs/mnt(image location)
/etc/sysconfig/docker(Redhat based)
/etc/default/docker(Debian based)
/var/lib/docker
/var/lib/docker/graph
Docker Ecosystem
Docker competitors
OpenShift(Redhat)
LXD(Canonical)
KV store
zookeeper
consul
redis
etcd
Managers
docker swarm
Kubernetes(k8s)
Rancher
Commands
hub commands
docker push
docker pull
docker logout
docker login
docker search <search term>
image commands
docker run
docker images
docker rmi
docker build
docker load -i <image_file>
docker tag
Debugging
docker inspect <cntnrid>
docker ps
docker logs <cntnrid>
backup/restore
docker save <cntnrid> > filename
docker load
Container commands
docker exec
docker ps
docker ps -a
docker start
docker stop
docker attach
docker commit
docker diff
Network
docker network
Access host machine
docker cp
docker -v
other commands
docker-machine
docker compose
docker engine
docker config
Concepts
History
Process isolation
IBM Mainframes
Chroot
Virtualisation
cgroups
Mulit tenancy(Multi user)
Internal components
Union Filesystem(AUFS)
readonly layers
Virtual filesystem
layers
cgroups
Resource limiting
Control
Disk IO(blkio)
CPU(cpuset and cpuacct)
RAM
Prioritising
Accounting
LxC
Namespaces
IPC
Process
Network
UTC
User
System requirements
64 bit machine
VT-X enabled
Kernel 2.6+


Software for Work
Software Registry
NPM +
NVM +
Communication Soft
Messangers +
Slack +
Microsoft Team +
DataBases
Dbeawer +
MySQL +
PopSQL +
Virtualization Soft
Docker +
WSL +
Hyper V +
Virtual Box +
Office/Planning Soft
Xmind +
MS Office +
Notepad +
Google Docs +
GIT/CI
GitHub +
Cypress
Jenkins
Bug Report Software
Jira
Bugzilla
Others
IDE
Code
Visual Studio/Code +
IntelliJ IDEA +
Builder
Maven
Gradle
Extensions
Code Snippets
Syntax Highlight
Testing
CI
GIT
Browsers
Browser
Testing/Development extensions
Testing
Video Record
Specific Extensions
BrowserDriver
VPN
Testing Frameworks
C#
Java Gunit
Android Google test
IOS
Python
JavaScrypt
Automation Software
Appium +
Selenium/Selenide
Server-Client Test software
Appium +
Test(Mock)Servers +
Postman +


Collection
"Monitors for Regression Testing"
A random number is generated in the request. Test tab compares random number in request to the environment variable testInteger.
Environment 1
"High Value Environment"
variable: value
testInteger = 500
Environment 2
"Low Value Environment"
variable: value
testInteger = 200
Monitor 1
"High Value Test"
Collection to be run
"Monitors for Regression Testing"
Environment
High Value Environment
(testInteger = 500)
Monitor 2
"Low ValueTest"
Collection to be run
"Monitors for Regression Testing"
Environment
Low Value Environment
(testInteger = 200)

4.HR


AGILE
CORE
SPECIALIST
EXPERT LEVEL
TEST MANAGEMENT
IMPROVING THE TEST PROCESS
ADVANCED LEVEL
AGILE TECHNICAL TESTER
AGILE TEST LEADERSHIP AT SCALE
IN BETA 2021
ADVANCED LEVEL
TEST MANAGER
TEST ANALYST
TECHNICAL TEST ANALYST
SECURITY TESTER
TEST AUTOMATION ENGINEER
MODEL-BASED TESTER
USABILITY TESTING
AUTOMOTIVE SOFTWARE TESTER
GAMBLING INDUSTRY TESTER
MOBILE APPLICATION TESTING
PERFORMANCE TESTING
ACCEPTANCE TESTING
FOUNDATION LEVEL
AGILE TESTER
FOUNDATION LEVEL
CERTIFIED TESTER



5.Math

```
// JAVA Program to find the sum of
// arithmetic series.

class GFG{

    // Function to find sum of series.
    static float sumOfAP(float a, float d,
                                 int n)
    {
        float sum = 0;
        for (int i = 0; i < n; i++)
        {
            sum = sum + a;
            a = a + d;
        }
        return sum;
    }

    // Driver function
    public static void main(String args[])
    {
        int n = 20;
        float a = 2.5f, d = 1.5f;
        System.out.println(sumOfAP(a, d, n));
    }
}

/*This code is contributed by Nikita Tiwari.*/
```

X	Y	X&Y	X|Y	X^Y	~(X)
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

TABLE IV. INTEGER SIGNEDNESS ERRORS REPORTED BY ALL, SIGNFLOW$_{cst}$ AND SIGNFLOW

	#A	#S$_c$	#S	data flow pattern for each excluded error			
				P_{cst}	P_{uncrit}	P_{post}	P_{op}
400.perlbench	48	48	24			18	6
401.bzip2	15	14	7	1		4	4
445.gobmk	17	15	12	2		1	2
458.sjeng	3	3	0				3
462.libquantum	4	4	3		1		
464.h264ref	10	8	7	2			1
483.xalancbmk	9	7	6	2			1
Gzip	1	1	0				1
Dillo	5	5	4			1	
SWFTools	6	6	2			1	3
Total	118	111	65	7	1	25	21

TABLE III. 7 HARMFUL INTEGER SIGNEDNESS BUGS IN REAL WORLD

CVE	Programs	Version	Sign Conv.	Sink
2008-1803	Rdesktop	1.5.0	u→s	bound check
2009-3743	GhostScript	8.70	s→u	memmove
2011-1471	PHP	5.3.6*	s→u	bound check
2012-3368	Dtach	0.8	s→u	bound check
2013-4927	Wireshark	1.10.0	u→s	loop
2013-6489	Pidgin	2.10.11*	s→u	malloc
2014-0045	Mumble	1.2.4	s→u	malloc

Important Trigonometric Formulae

1. $\sin(A+B) = \sin A\cos B + \cos A \sin B$
2. $\sin(A-B) = \sin A\cos B - \cos A \sin B$
3. $\cos(A+B) = \cos A\cos B - \sin A \sin B$
4. $\cos(A-B) = \cos A\cos B + \sin A \sin B$
5. $\tan(A+B) = \dfrac{\tan A + \tan B}{1 - \tan A\tan B}$
6. $\tan(A-B) = \dfrac{\tan A - \tan B}{1 + \tan A\tan B}$
7. $\cot(A+B) = \dfrac{\cot A \cot B - 1}{\cot A + \cot B}$
8. $\cot(A-B) = \dfrac{\cot A \cot B + 1}{\cot B - \cot A}$
9. $\sin 2A =$ (a) $2\sin A\cos A$
   (b) $\dfrac{2\tan A}{1 + \tan^2 A}$
10. $\cos 2A =$ (a) $\cos^2 A - \sin^2 A$
    (b) $1 - 2\sin^2 A$
    (c) $2\cos^2 A - 1$
    (d) $\dfrac{1 - \tan^2 A}{1 + \tan^2 A}$
11. $\tan 2A = \dfrac{2\tan A}{1 - \tan^2 A}$
12. $\sin 3A = 3\sin A - 4\sin^3 A$
13. $\cos 3A = 4\cos^3 A - 3\cos A$
14. $\tan 3A = \dfrac{3\tan A - \tan^3 A}{1 - 3\tan^2 A}$
15. $\sin(-A) = -\sin A$
16. $\cos(-A) = \cos A$
17. $\sin^2 A + \cos^2 A = 1$
18. $1 + \tan^2 A = \sec^2 A$
19. $1 + \cot^2 A = \operatorname{cosec}^2 A$
20. $1 - \cos 2A = 2\sin^2 A$
21. $1 + \cos 2A = 2\cos^2 A$
22. $1 - \sin 2A = (\cos A - \sin A)^2$
23. $1 + \sin 2A = (\cos A + \sin A)^2$
24. $\dfrac{1 + \tan A}{1 - \tan A} = \tan(\frac{\pi}{4} + A)$
25. $\dfrac{1 - \tan A}{1 + \tan A} = \tan(\frac{\pi}{4} - A)$
26. $\sin^2 A - \sin^2 B = \sin(A+B)\sin(A-B)$
27. $\cos^2 A - \sin^2 B = \cos(A+B)\cos(A-B)$
28. $\cos^2 A - \cos^2 B = -\sin(A+B)\sin(A-B)$
29. $2\sin A\cos B = \sin(A+B) + \sin(A-B)$
30. $2\cos A\sin B = \sin(A+B) - \sin(A-B)$
31. $2\cos A\cos B = \cos(A+B) + \cos(A-B)$
32. $2\sin A\sin B = \cos(A-B) - \cos(A+B)$
33. $\sin\alpha + \sin\beta = 2\sin\left(\dfrac{\alpha+\beta}{2}\right)\cos\left(\dfrac{\alpha-\beta}{2}\right)$
34. $\sin\alpha - \sin\beta = 2\cos\left(\dfrac{\alpha+\beta}{2}\right)\sin\left(\dfrac{\alpha-\beta}{2}\right)$
35. $\cos\alpha + \cos\beta = 2\cos\left(\dfrac{\alpha+\beta}{2}\right)\cos\left(\dfrac{\alpha-\beta}{2}\right)$
36. $\cos\alpha - \cos\beta = -2\sin\left(\dfrac{\alpha+\beta}{2}\right)\sin\left(\dfrac{\alpha-\beta}{2}\right)$
37. II | I
    III | IV

(a) With 90^0 & 270^0 ratio will be changed.
(b) Sign of answer, depends on position of intial ratio.

$\sin(90^0 - A) = \cos A$
$\sin(90^0 + A) = \cos A$
$\sin(180^0 + A) = -\sin A$
$\sin(270^0 + A) = -\cos A$
$\sin(360^0 + A) = \sin A$

Excel FLOOR Function

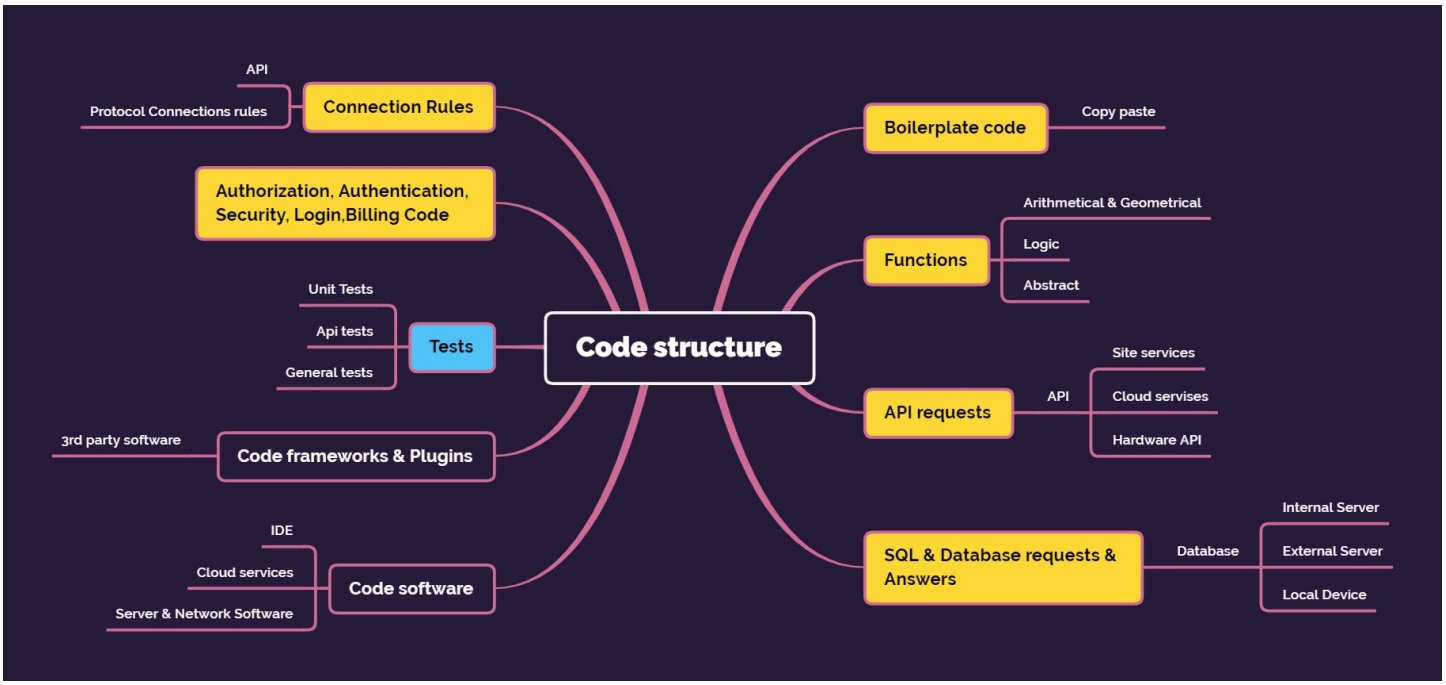
D5 =FLOOR(B5,C5)

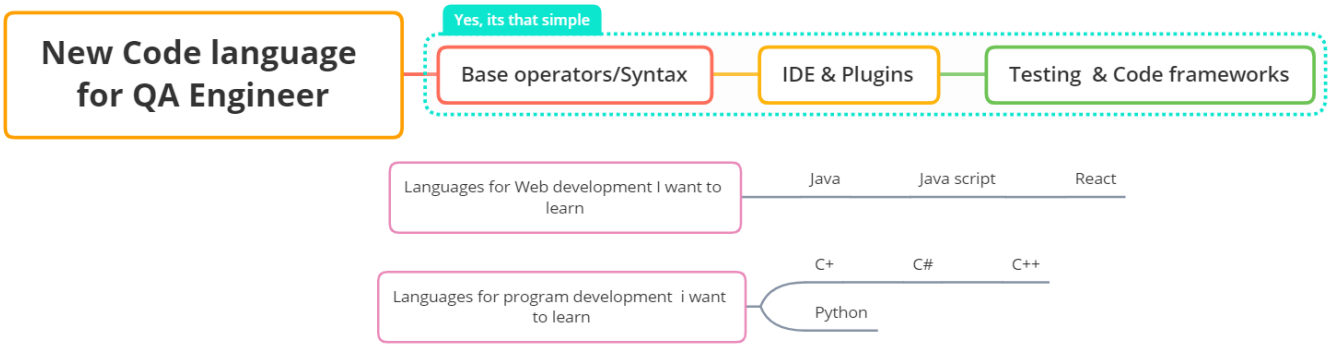
FLOOR function

Number	Multiple	Result	
10	3	9	// round down to nearest 3
40	7	35	// round down to nearest 7
320	25	300	// round down to nearest 25
610	100	600	// round down to nearest 100
5.37	0.05	5.35	// round down to nearest 0.05
5.37	1	5.00	// round down to nearest 1
0	1	0	// no change
-5.4	1	-6	// round away from zero
-5.4	-1	-5	// round toward zero
-320	-25	-300	// round toward zero

EXCELJET

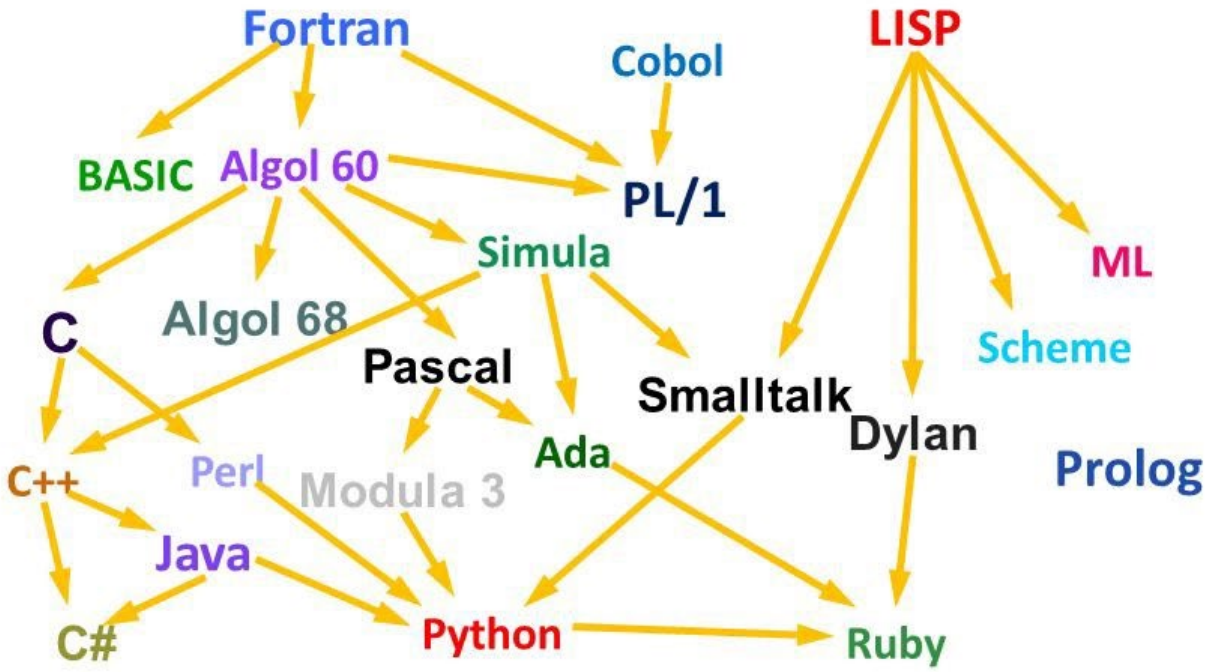
6.Code Structure



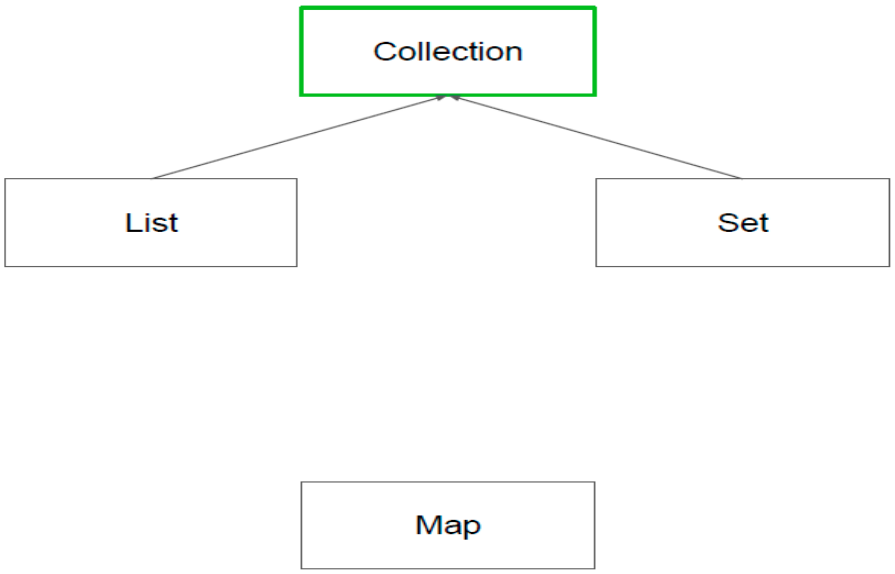


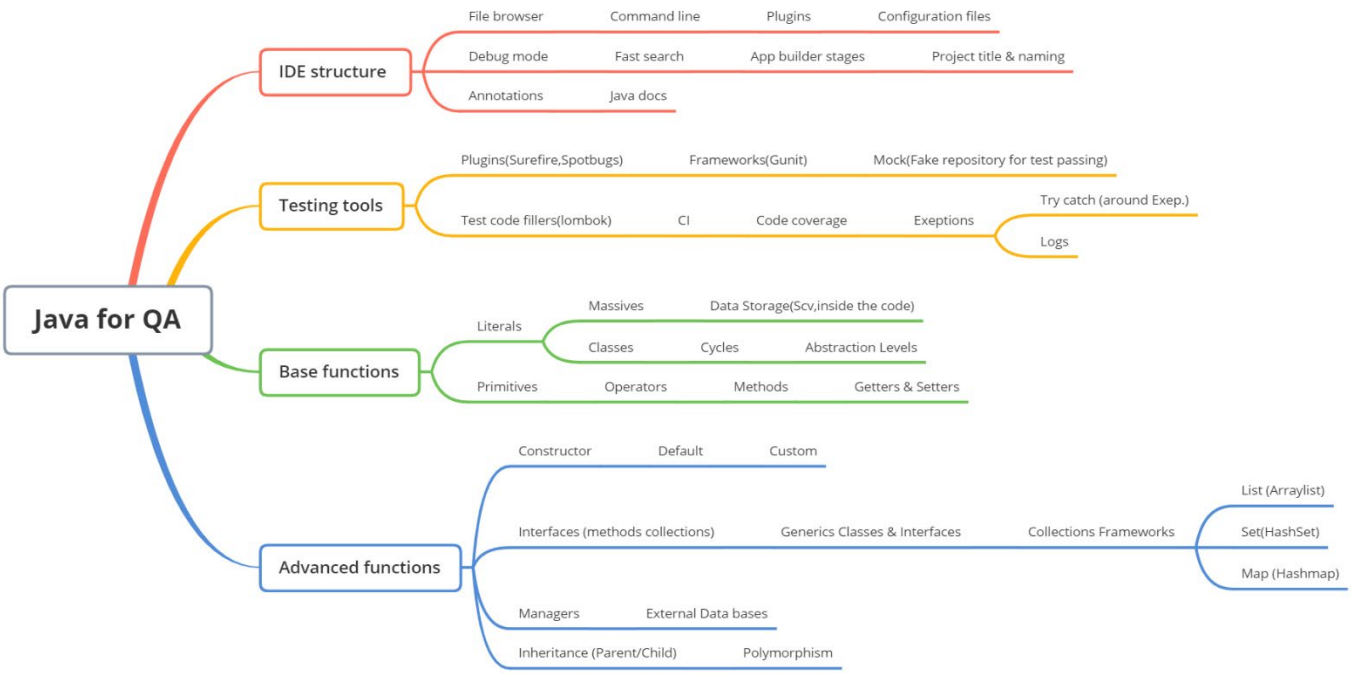
A family tree of languages

Some of the 2400 + programming languages



КЛЮЧЕВЫЕ ИНТЕРФЕЙСЫ




Java for QA
IDE structure
File browser
Command line
Plugins
Configuration files
Debug mode
Fast search
App builder stages
Project title & naming
Annotations
Java docs
Testing tools
Plugins(Surefire,Spotbugs)
Frameworks(Gunit)
Mock(Fake repository for test passing)
Test code fillers(lombok)
CI
Code coverage
Exeptions
Try catch (around Exep.)
Logs
Base functions
Literals
Massives
Data Storage(Scv,inside the code)
Classes
Cycles
Abstraction Levels
Primitives
Operators
Methods
Getters & Setters
Advanced functions
Constructor
Default
Custom
Interfaces (methods collections)
Generics Classes & Interfaces
Collections Frameworks
List (Arraylist)
Set(HashSet)
Map (Hashmap)
Managers
External Data bases
Inheritance (Parent/Child)
Polymorphism

Java programming language structure

Base Functions & Abstract logic & Math to write code

Main Class

Main Manager

Managers

Repository

Tests

Constructor

Immutable

Write protected,faster access

Nested Classes & Interfaces

Inner class can acsess parent class

OBJECT Super Class

Can be overridden

@Override

Generate equals & hash code

Generics <T> (Replacement of values by this class)

- E – element (элемент)
- K – key (ключ)
- V – value (значение)
- T – type (тип)

Utility Classes (AssertEquals,Arrays)

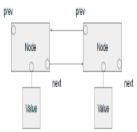
Collections syntax: Collection <E>

contains и remove принимают Object, а add – объект типа E

List

ArrayList (Arrays)

Linked List



Map (Key-Value,no inheritance)

public interface Map<K, V>

тип для ключа (Key) — тип для значения (Value)

Hash Map

Tree Map

Set(No Index)

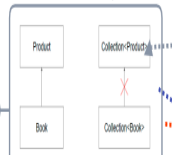
Hash set

Tree set

Product is Parent book is heir

```
public class Book extends Product {
  private String author;
  private int pages;
  private int publishedYear;
}
```

Super - use the parent constructor & methods



Inheritance Rules

```
@Test
void shouldAddMultipleProducts() {
  Collection<Product> products = new ArrayList<>();
  products.add(new Product());
  products.add(new Product());
  repository.addAll(products);

  repository.addAll(List.of(new Product(), new Product()));
}
```

Fabric Methods

If repository has no logic and only data & getters & setters no tests needed

Interfaces (class,uniting the methods to be implemented by object) (instanceof required)

Lombok (Generates Getters & Setters & Constructors)

E.printstacktrace(); Generate Try Catch

Error (general error)

Throwable(class)

Exceptions

ArrayIndexOutOfBoundsExeption

IndexOutOfBoundsExeption

NullPointerException (NPE)

Runtime Exception(Unchecked)

Exception(checked error)

Lambda Expression

() -> service.throwChecked()

() -> service.throwUnhecked()

CI Testing

Mocks (Separates elements for testing)

GunitTests

Jacoco (Statistics & Reporting)

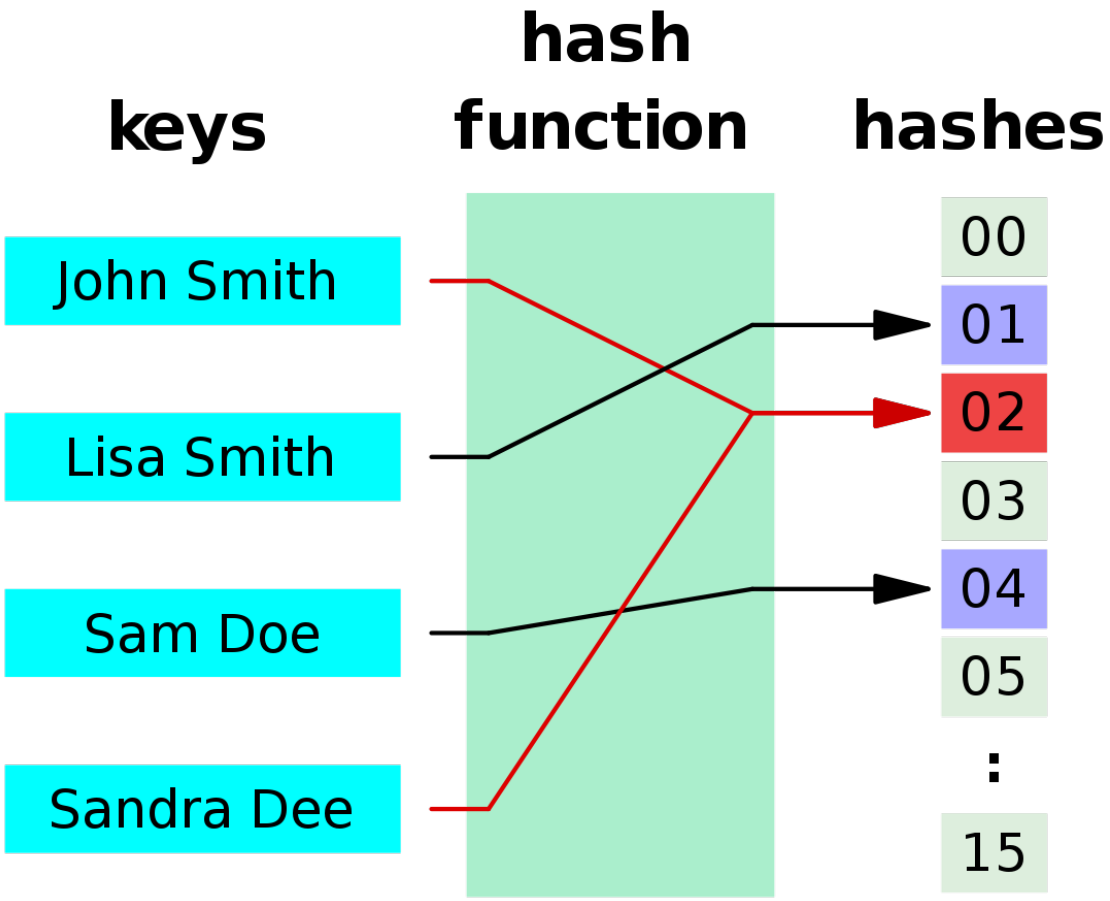
Surefire/Spotbugs (Testing)

```
public interface List<E> extends Collection<E> {
  ...
  static <E> List<E> of(E... elements) {
    ...
  }
  ...
}
```

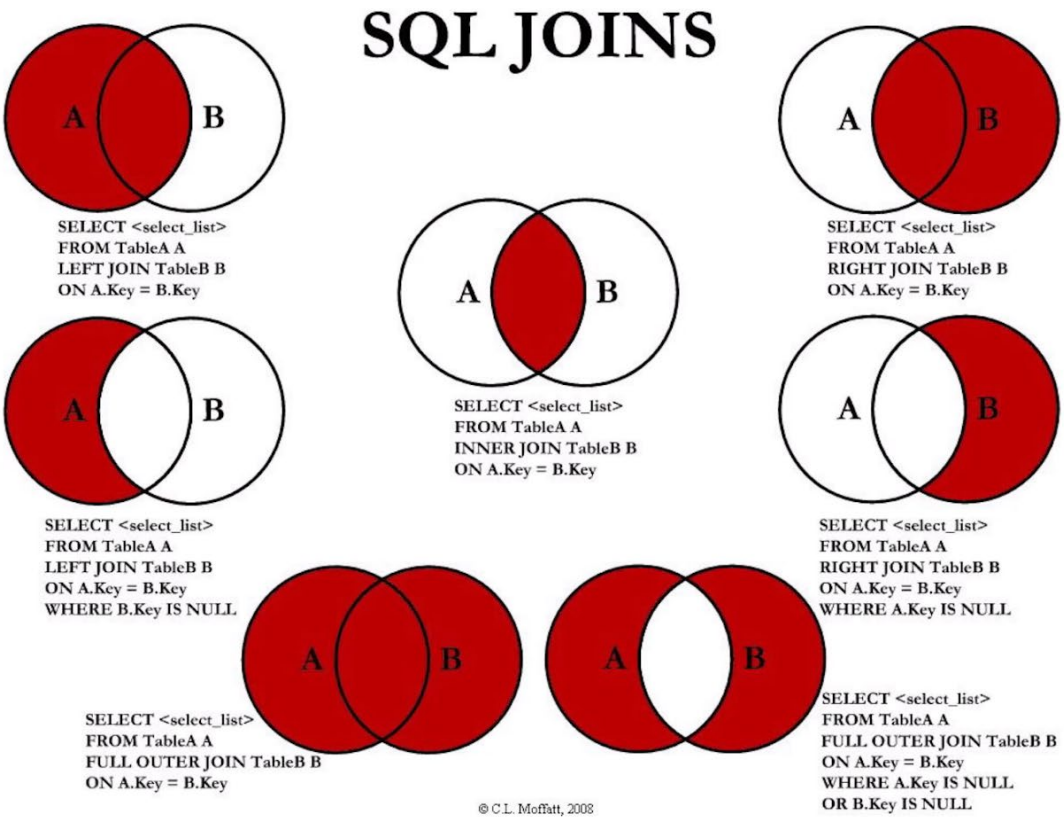
Varargs

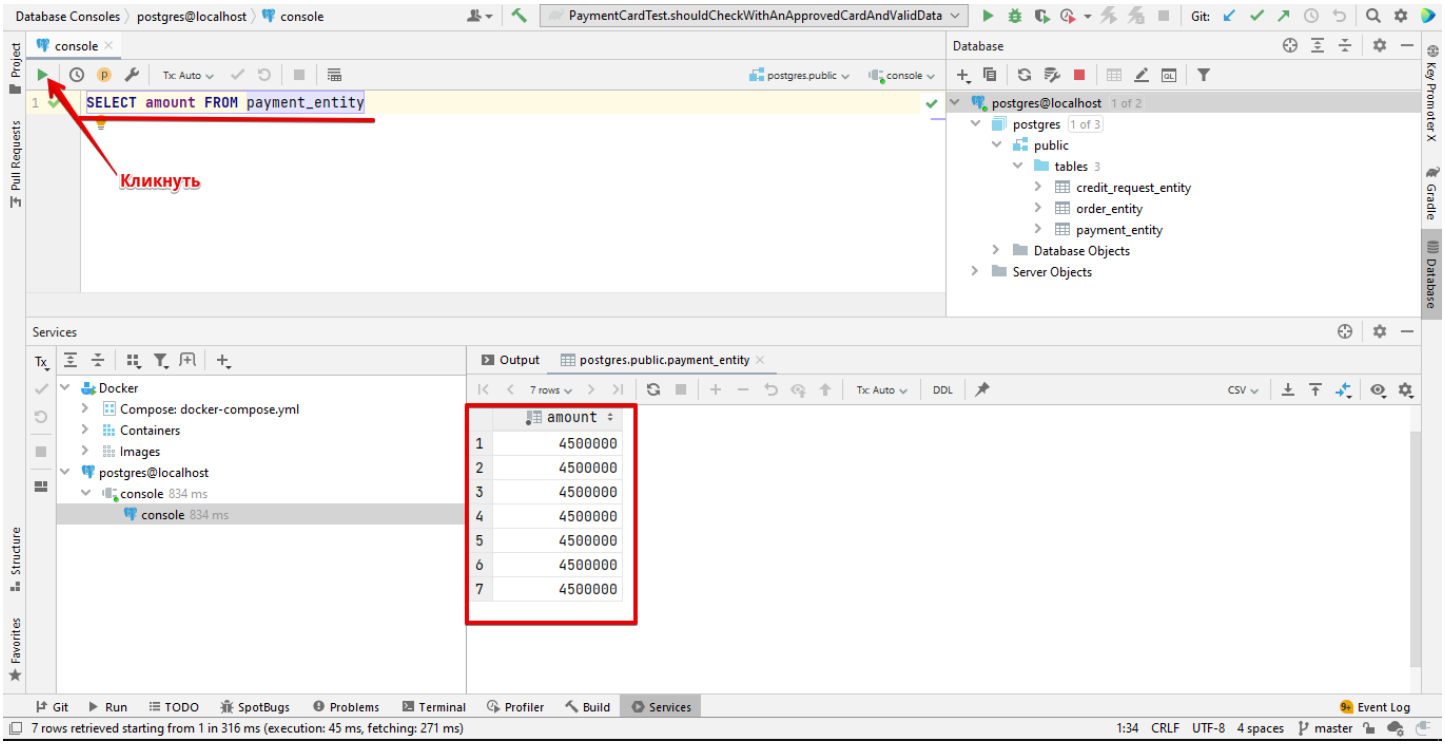
Varargs – это специальная конструкция, когда после типа аргумента пишется <...>. Это означает, что мы можем передавать в вызов этого метода произвольное количество аргументов, и все они будут аккуратно складываться в массив (в данном случае – elements).

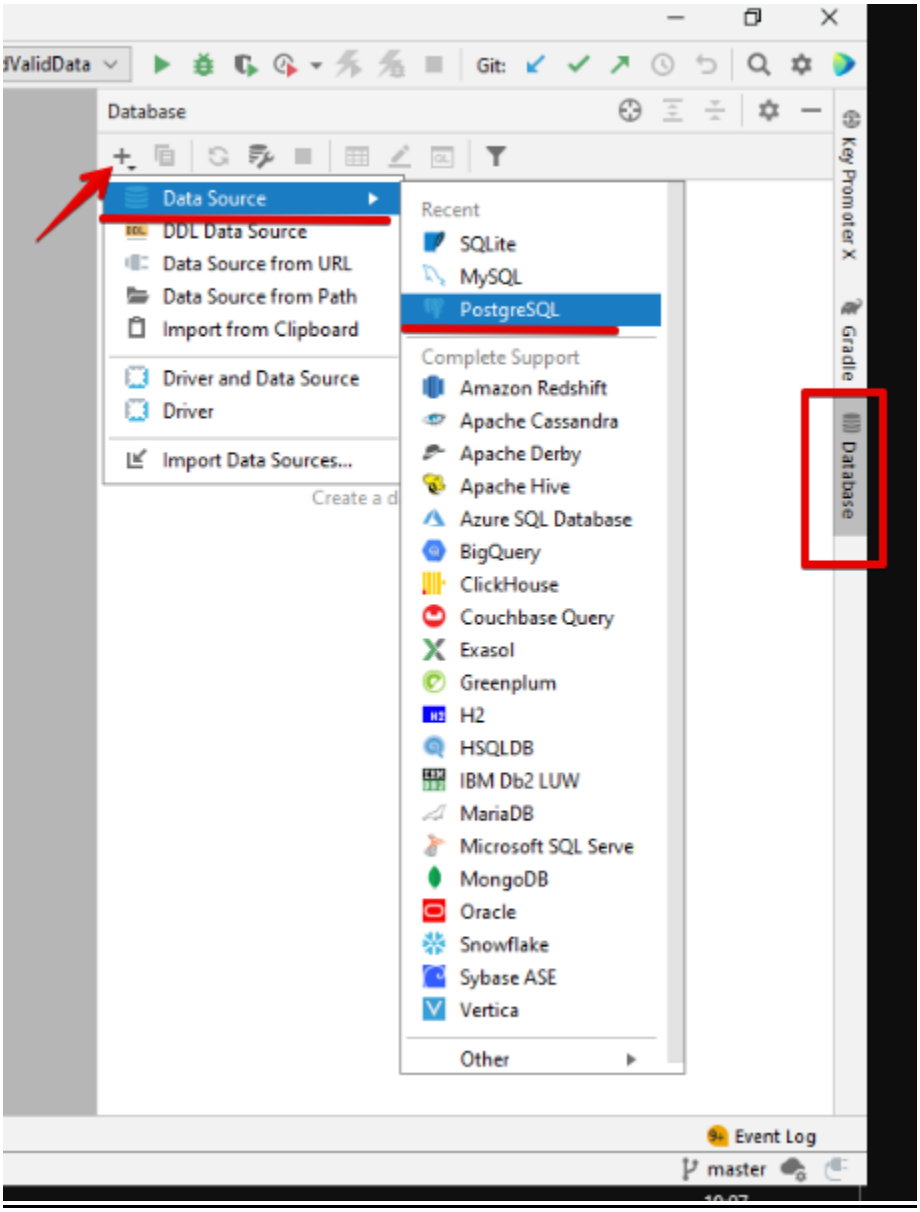
7.SQL and Data



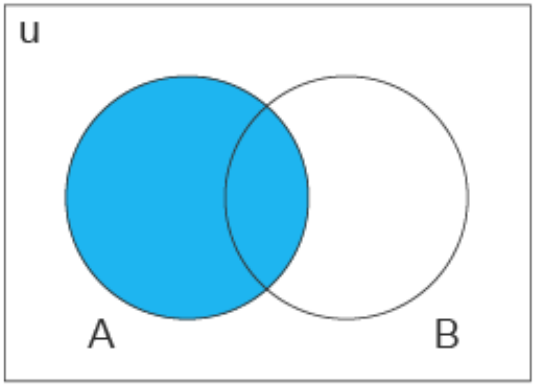
Method	Description
GET	Request to read a Web page
HEAD	Request to read a Web page's header
PUT	Request to store a Web page
POST	Append to a named resource (e.g., a Web page)
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Reserved for future use
OPTIONS	Query certain options


SQL JOINS
A
B
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
© C.L. Moffatt, 2008

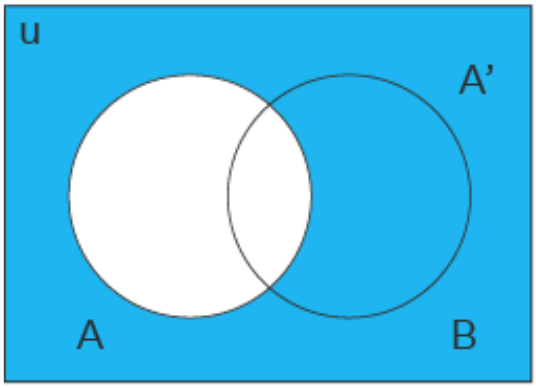

Database Consoles 〉 postgres@localhost 〉 console
PaymentCardTest.shouldCheckWithAnApprovedCardAndValidData
console
SELECT amount FROM payment_entity
Кликнуть
Database
postgres@localhost 1 of 2
postgres 1 of 3
public
tables 3
credit_request_entity
order_entity
payment_entity
Database Objects
Server Objects
Services
Docker
Compose: docker-compose.yml
Containers
Images
postgres@localhost
console 834 ms
console 834 ms
Output
postgres.public.payment_entity
amount
1 4500000
2 4500000
3 4500000
4 4500000
5 4500000
6 4500000
7 4500000
7 rows retrieved starting from 1 in 316 ms (execution: 45 ms, fetching: 271 ms)


Database
Data Source
DDL Data Source
Data Source from URL
Data Source from Path
Import from Clipboard
Driver and Data Source
Driver
Import Data Sources...
Recent
SQLite
MySQL
PostgreSQL
Complete Support
Amazon Redshift
Apache Cassandra
Apache Derby
Apache Hive
Azure SQL Database
BigQuery
ClickHouse
Couchbase Query
Exasol
Greenplum
H2
HSQLDB
IBM Db2 LUW
MariaDB
Microsoft SQL Serve
MongoDB
Oracle
Snowflake
Sybase ASE
Vertica
Other
Key Promoter X
Gradle
Database
Event Log
master

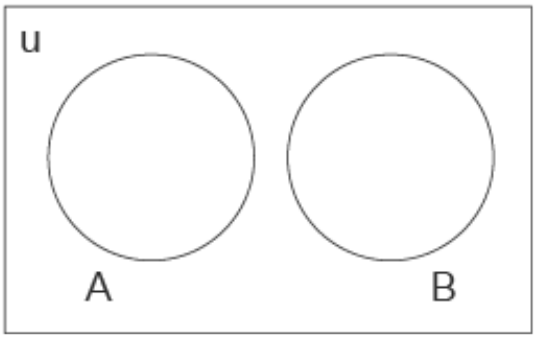
Set Operations



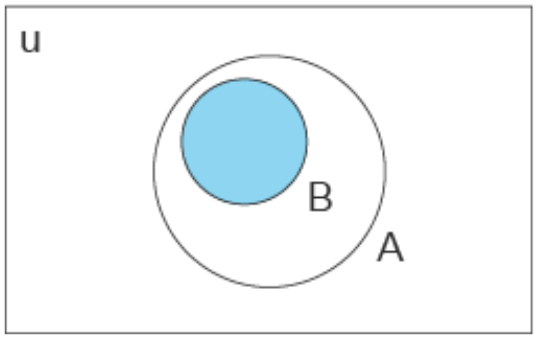
Set A



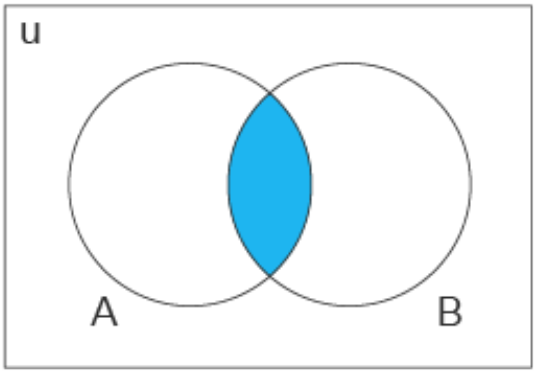
A' the complement of A



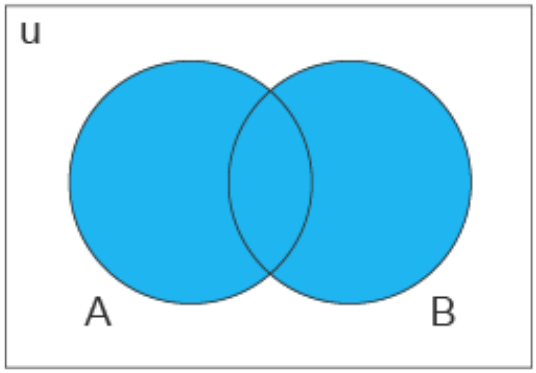
A and B are disjoint sets



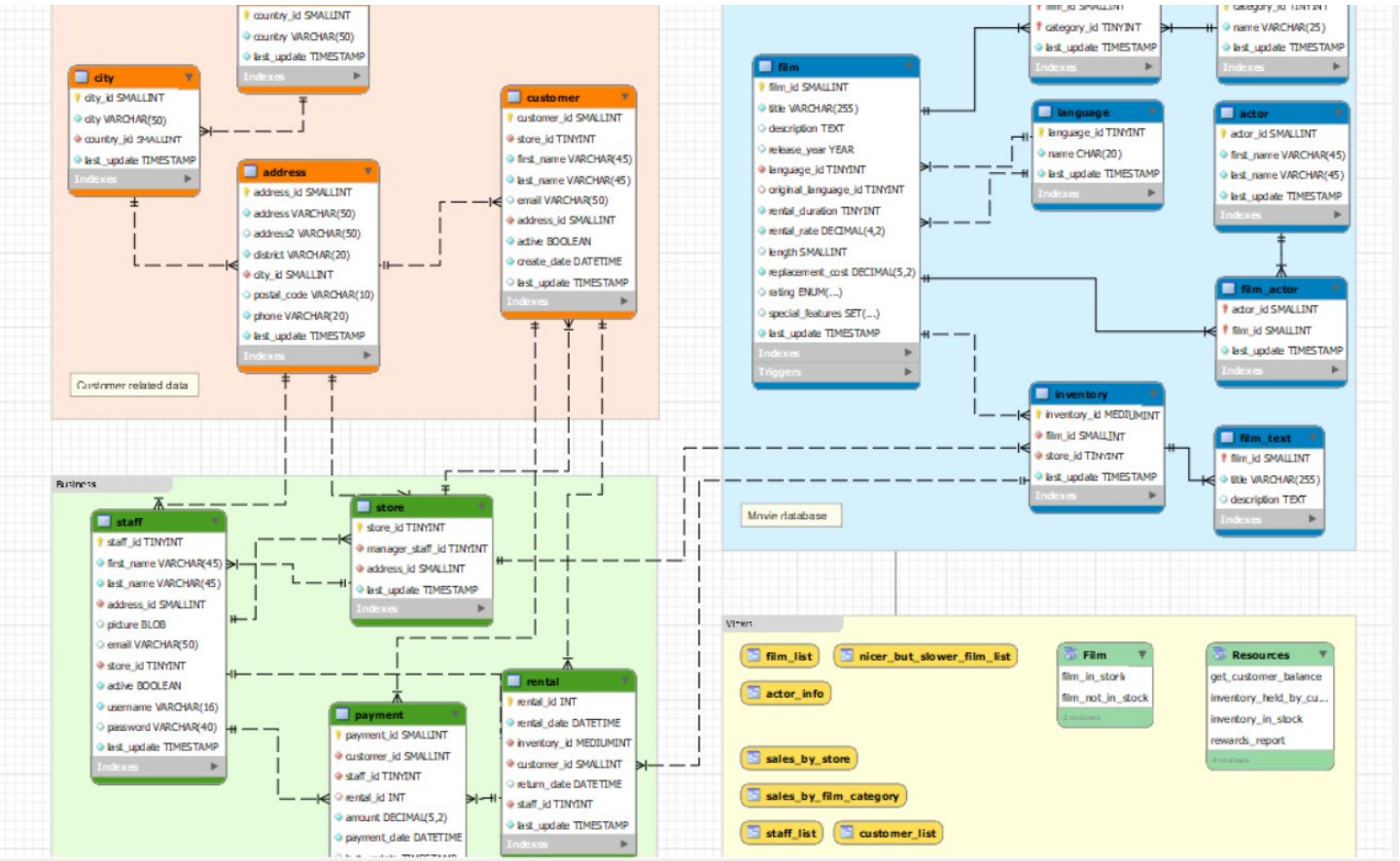
B is proper subset of A $B \subset A$



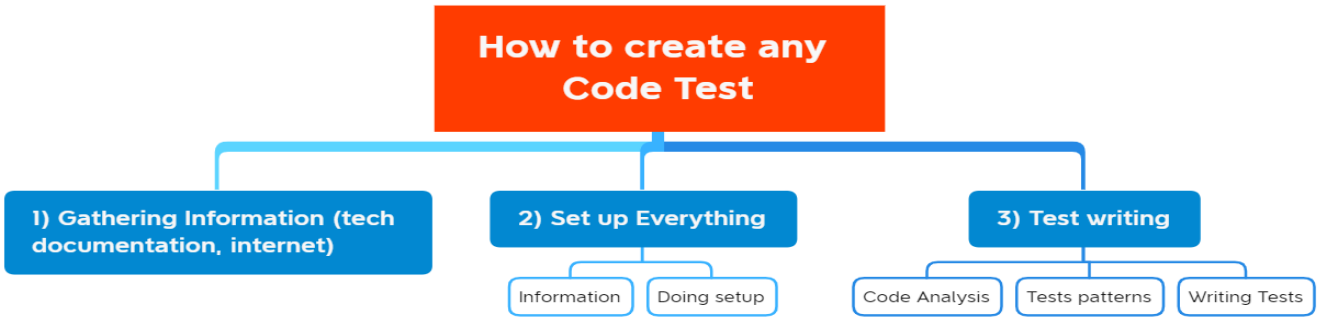
Both A and B
A intersect B $A \cap B$

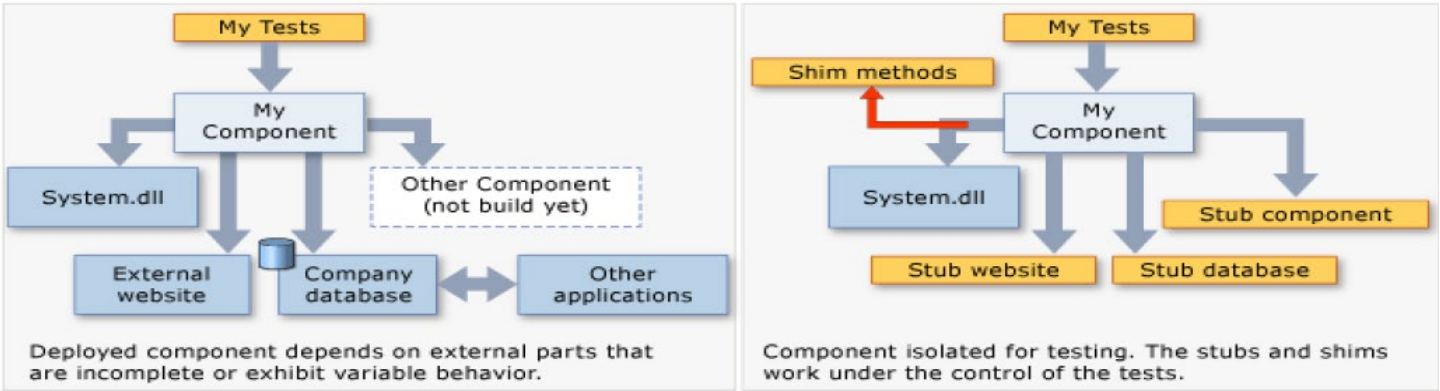


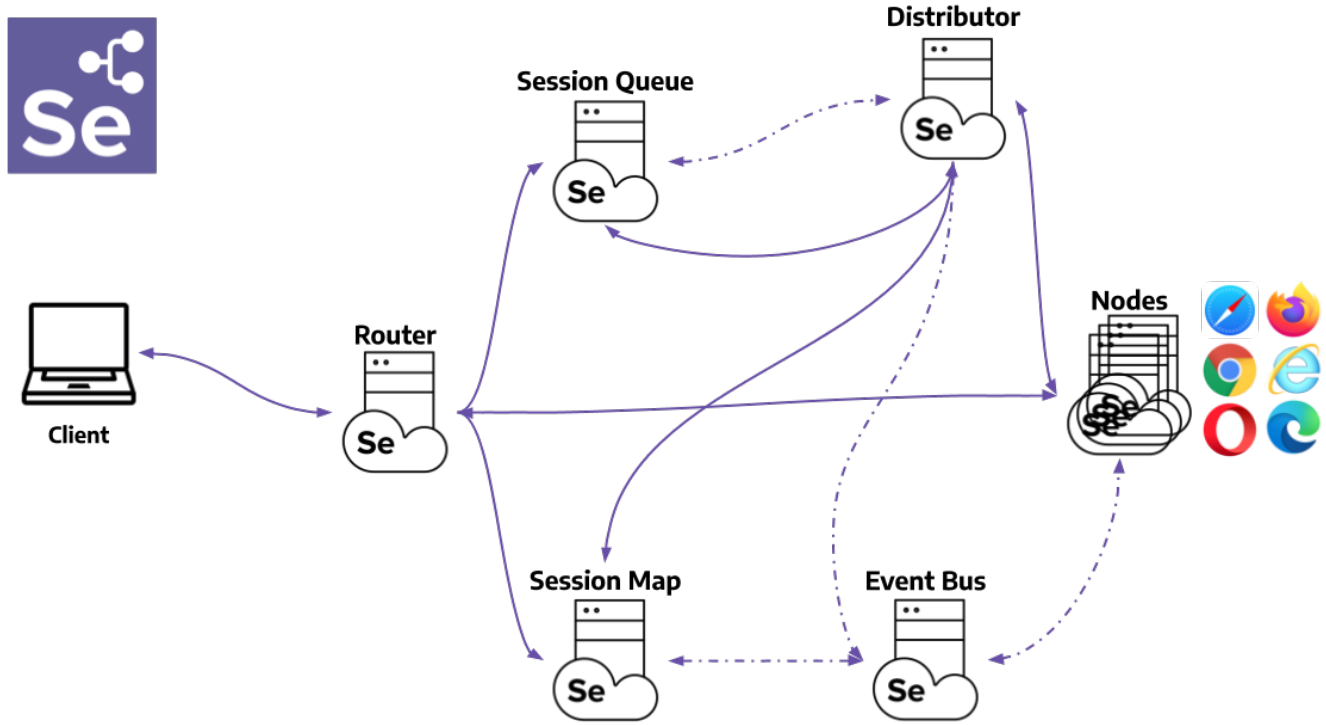
Either A or B
A union B $A \cup B$

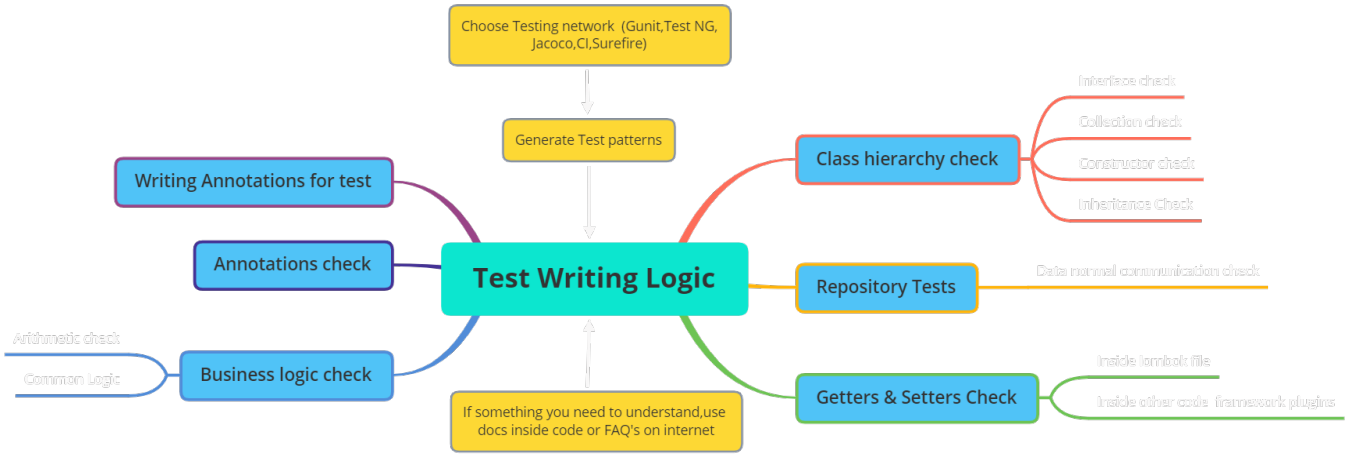


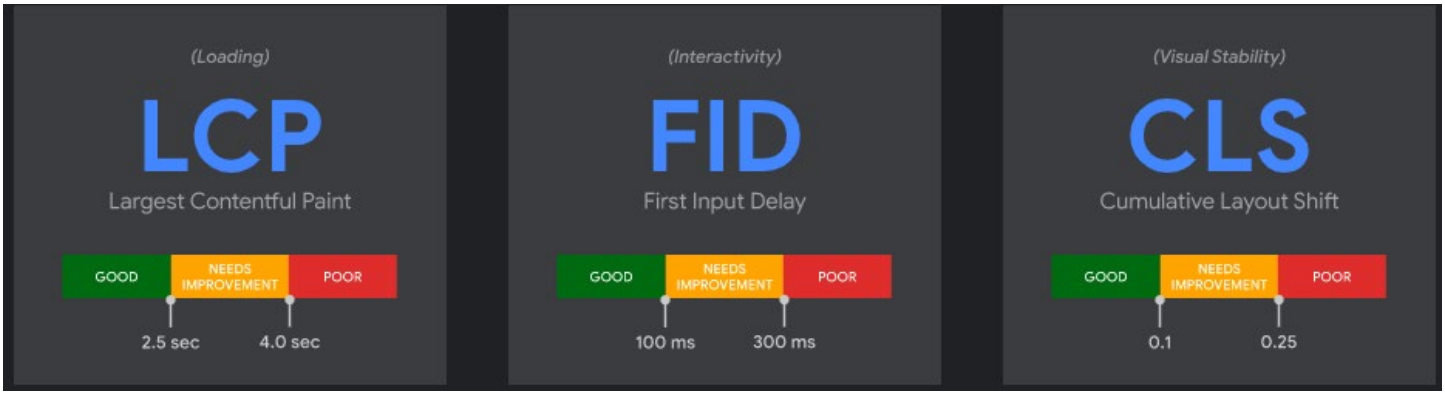
8. Tests Writing

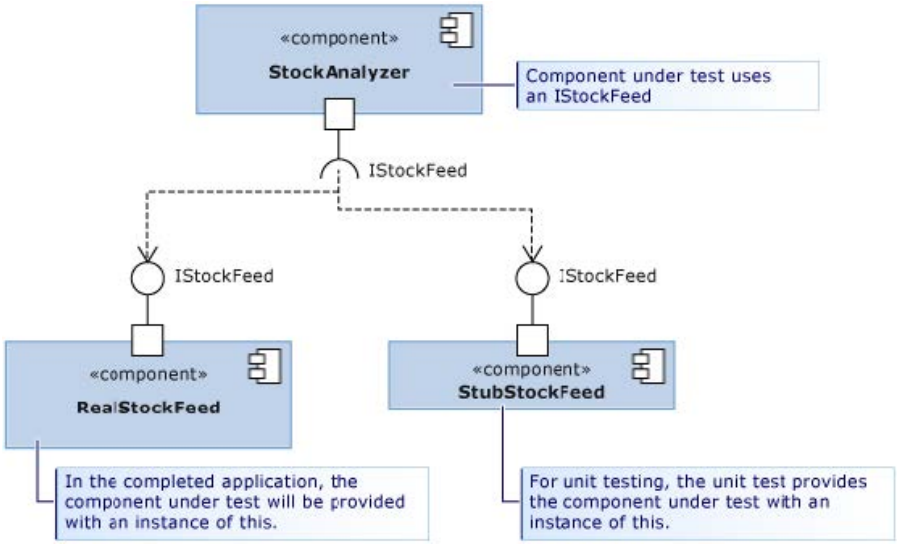


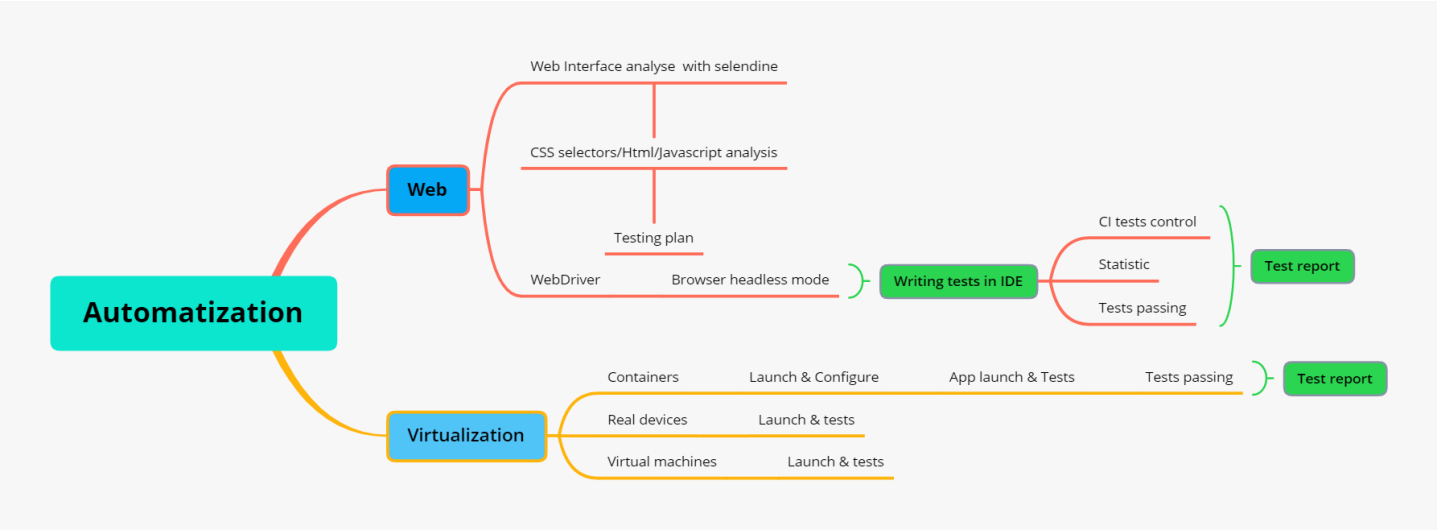


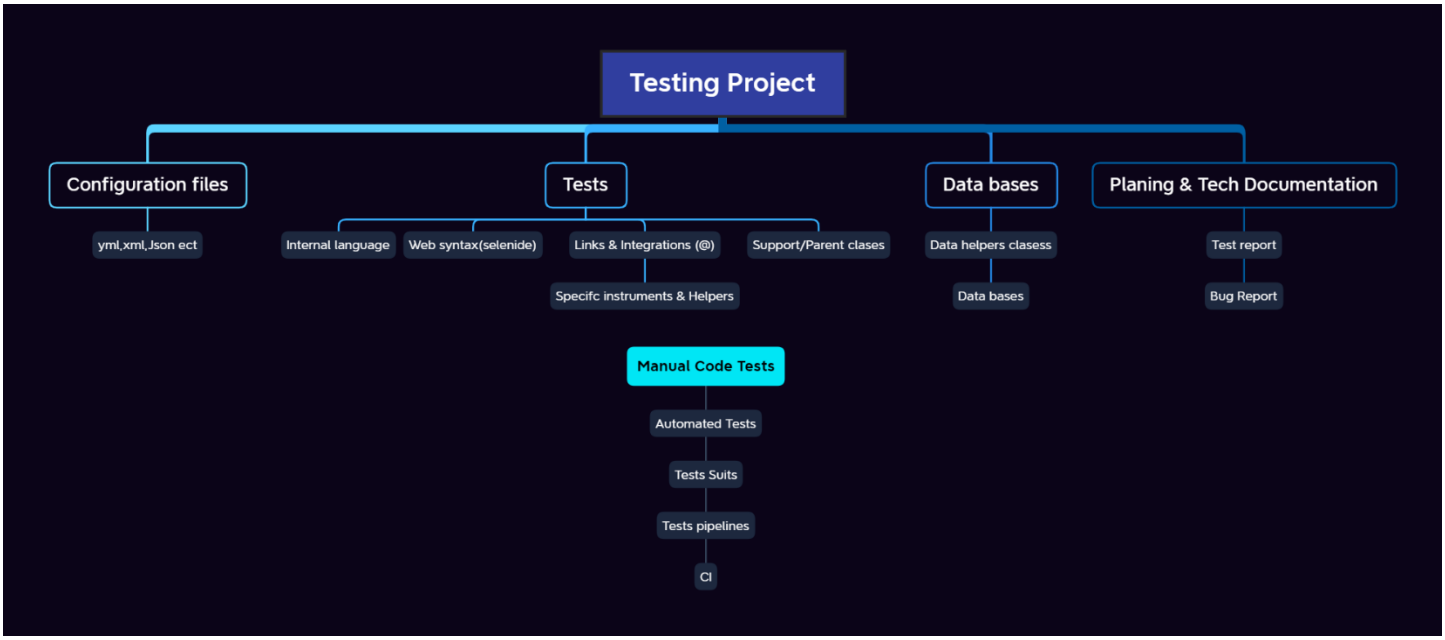

Distributor
Session Queue
Client
Router
Nodes
Session Map
Event Bus

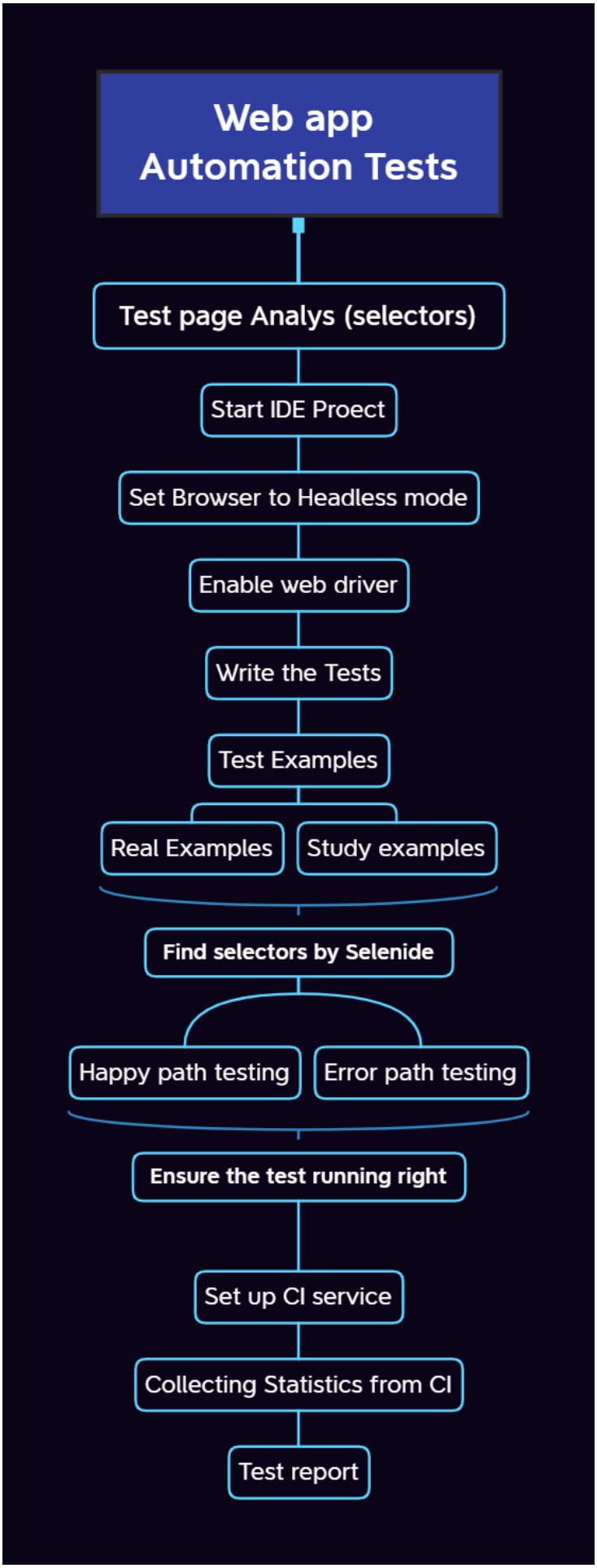

Choose Testing network (Gunit,Test NG, Jacoco,CI,Surefire)
Generate Test patterns
Writing Annotations for test
Annotations check
Test Writing Logic
Business logic check
Class hierarchy check
Repository Tests
Getters & Setters Check
If something you need to understand,use docs inside code or FAQ's on internet

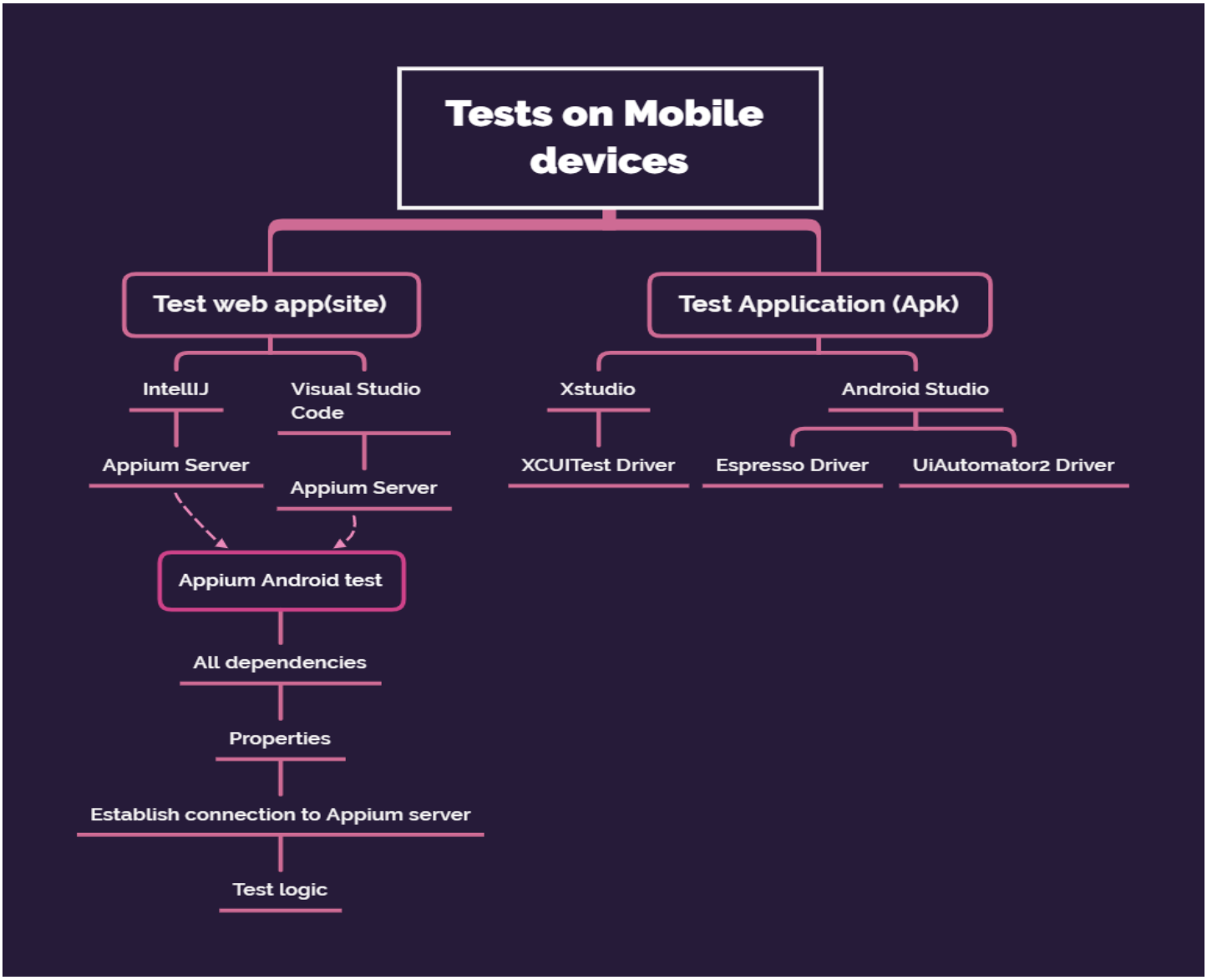

(Loading)
LCP
Largest Contentful Paint
GOOD
NEEDS IMPROVEMENT
POOR
2.5 sec
4.0 sec
(Interactivity)
FID
First Input Delay
GOOD
NEEDS IMPROVEMENT
POOR
100 ms
300 ms
(Visual Stability)
CLS
Cumulative Layout Shift
GOOD
NEEDS IMPROVEMENT
POOR
0.1
0.25

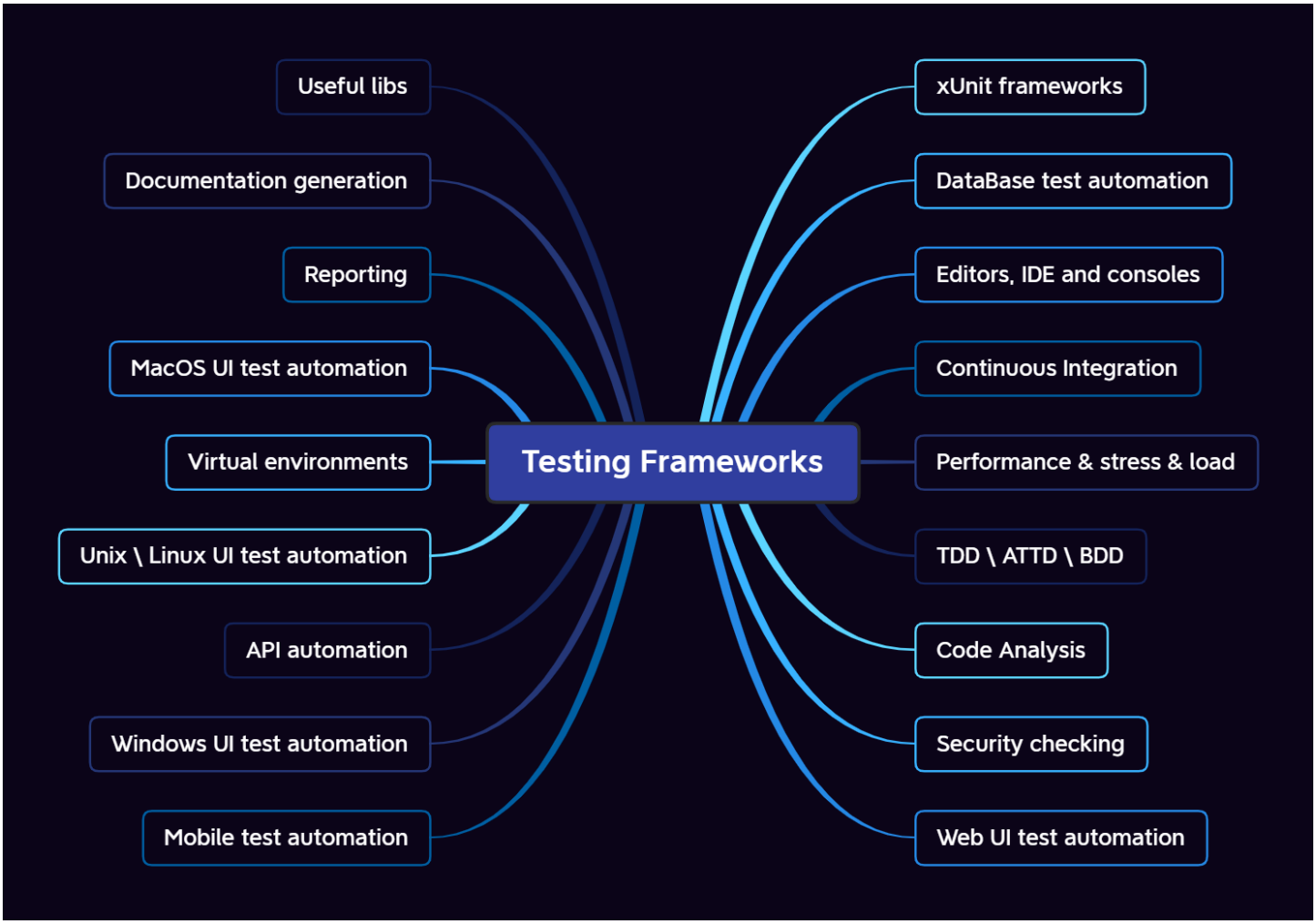

«component»
StockAnalyzer
Component under test uses an IStockFeed
IStockFeed
IStockFeed
IStockFeed
«component»
RealStockFeed
«component»
StubStockFeed
In the completed application, the component under test will be provided with an instance of this.
For unit testing, the unit test provides the component under test with an instance of this.

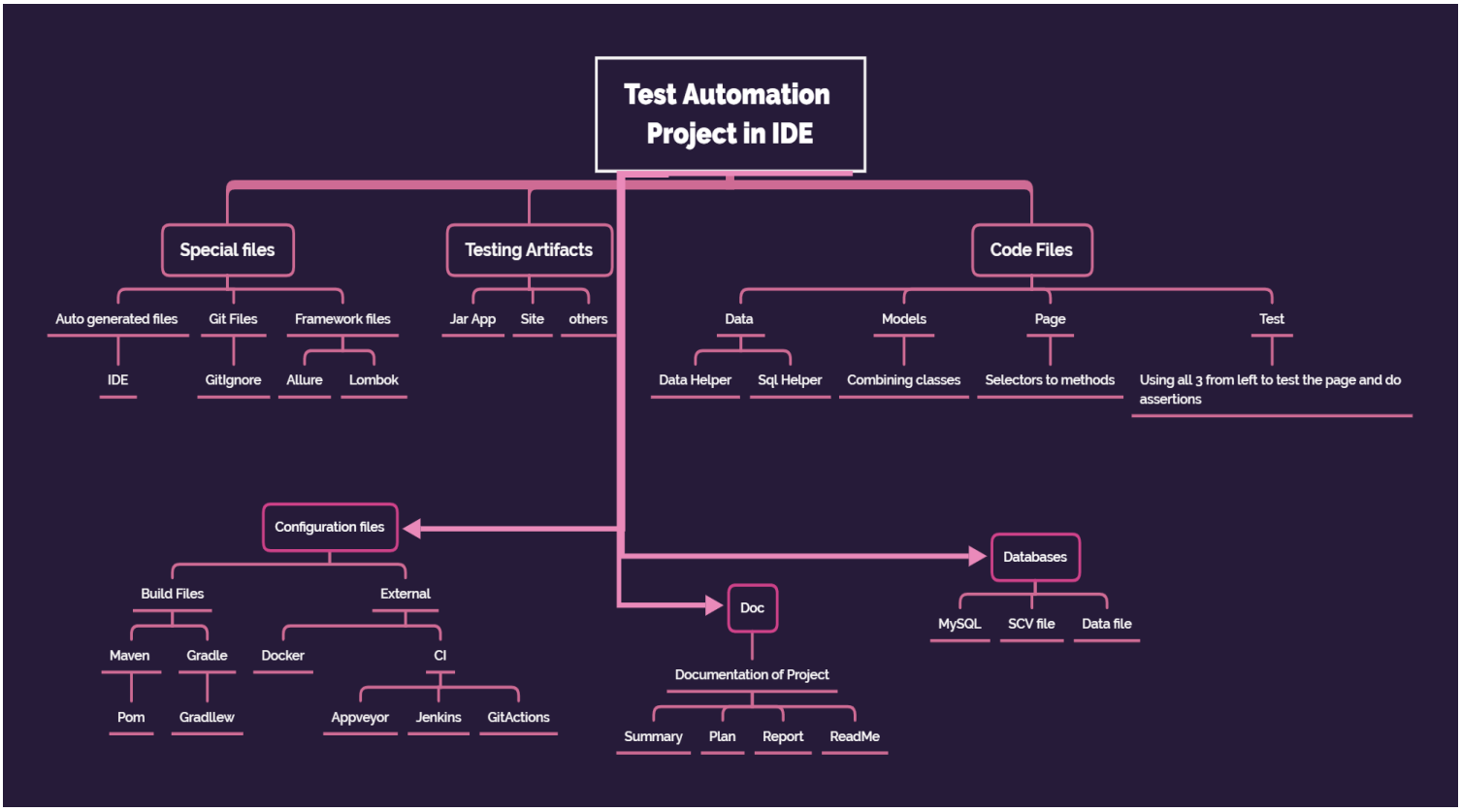

Automatization
Web
Web Interface analyse with selendine
CSS selectors/Html/Javascript analysis
Testing plan
WebDriver
Browser headless mode
Writing tests in IDE
CI tests control
Statistic
Tests passing
Test report
Virtualization
Containers
Launch & Configure
App launch & Tests
Tests passing
Test report
Real devices
Launch & tests
Virtual machines
Launch & tests

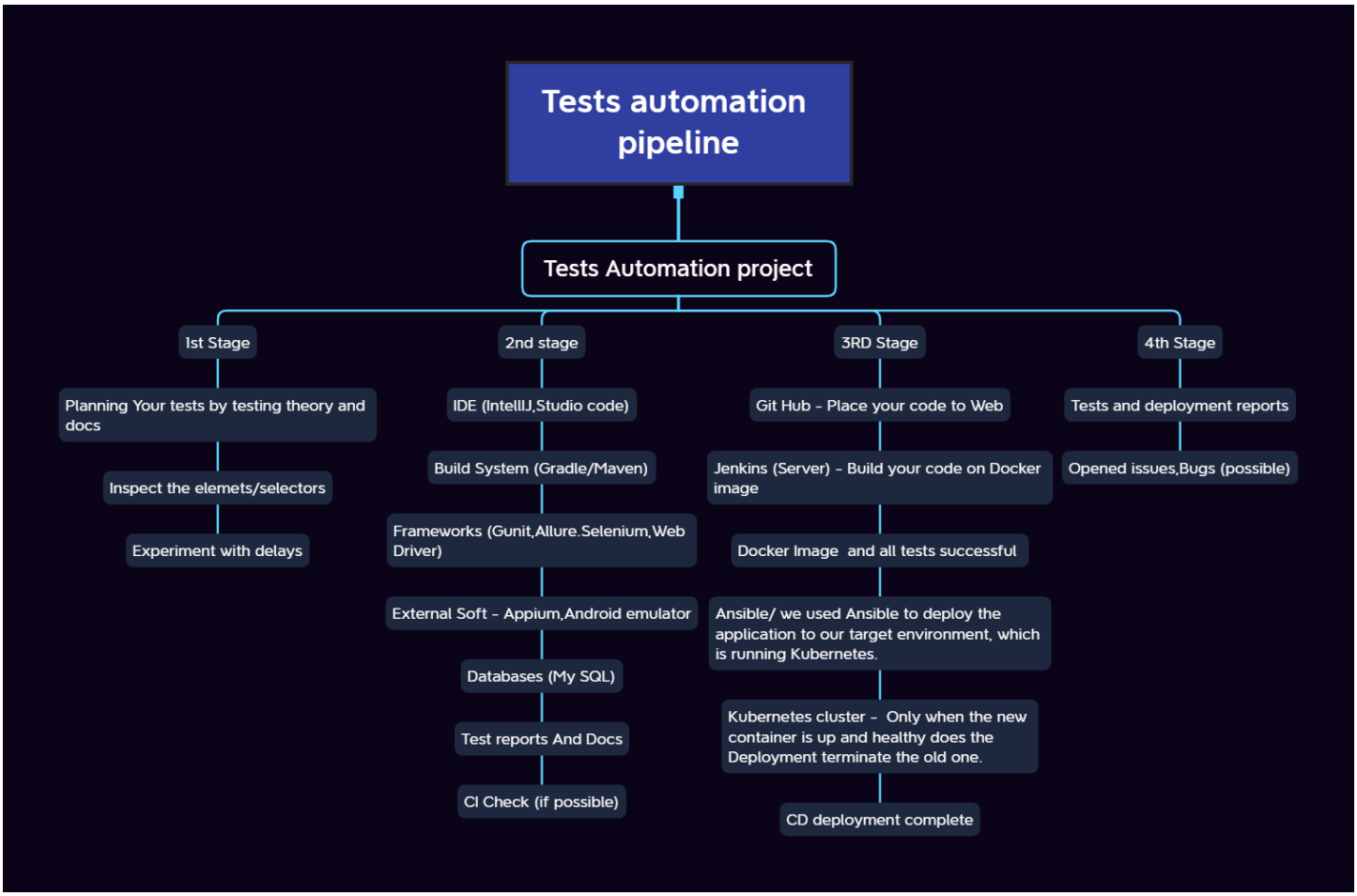

Testing Project
Configuration files
yml,xml,Json ect
Tests
Internal language
Web syntax(selenide)
Links & Integrations (@)
Support/Parent clases
Specifc instruments & Helpers
Data bases
Data helpers clasess
Data bases
Planing & Tech Documentation
Test report
Bug Report
Manual Code Tests
Automated Tests
Tests Suits
Tests pipelines
CI

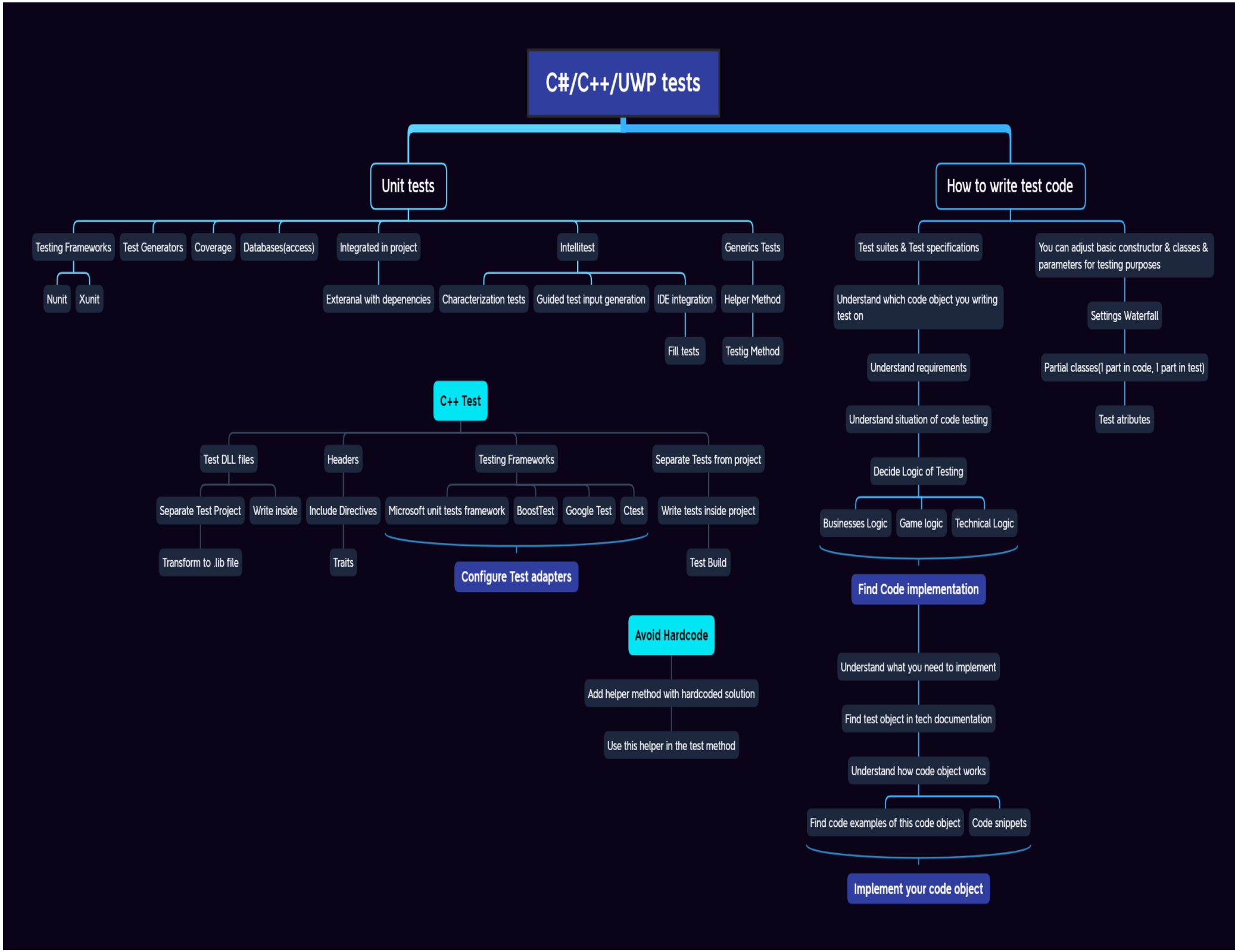

Web app
Automation Tests
Test page Analys (selectors)
Start IDE Proect
Set Browser to Headless mode
Enable web driver
Write the Tests
Test Examples
Real Examples
Study examples
Find selectors by Selenide
Happy path testing
Error path testing
Ensure the test running right
Set up CI service
Collecting Statistics from CI
Test report


Tests on Mobile devices
Test web app(site)
Test Application (Apk)
IntelliJ
Visual Studio Code
Xstudio
Android Studio
Appium Server
Appium Server
XCUITest Driver
Espresso Driver
UiAutomator2 Driver
Appium Android test
All dependencies
Properties
Establish connection to Appium server
Test logic

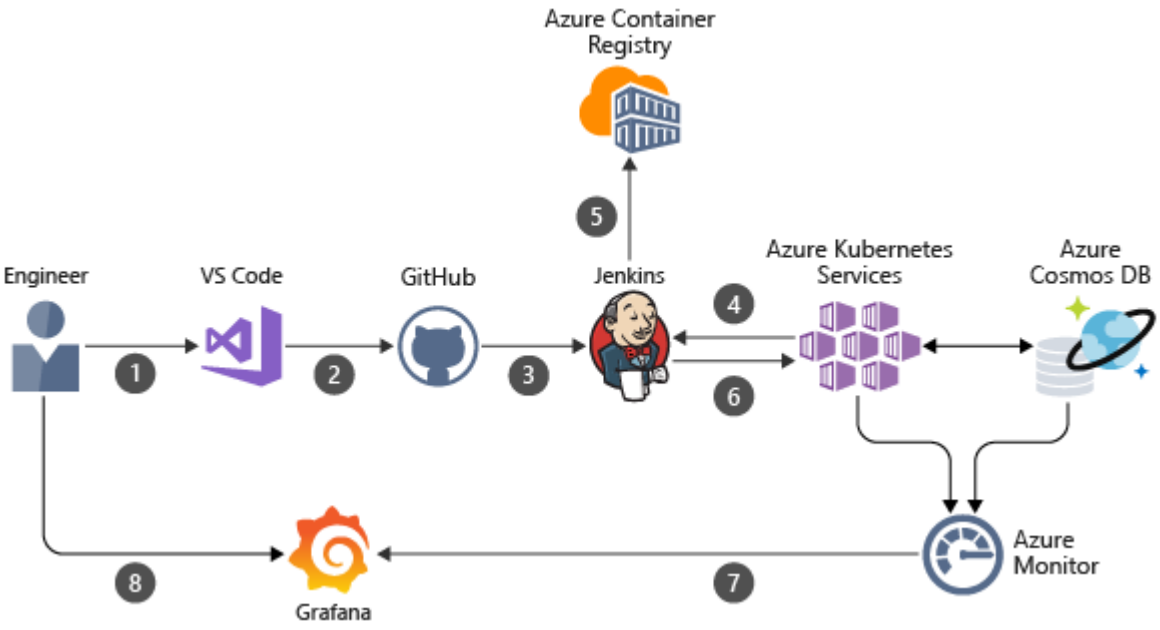

Testing Frameworks
Useful libs
Documentation generation
Reporting
MacOS UI test automation
Virtual environments
Unix \ Linux UI test automation
API automation
Windows UI test automation
Mobile test automation
xUnit frameworks
DataBase test automation
Editors, IDE and consoles
Continuous Integration
Performance & stress & load
TDD \ ATTD \ BDD
Code Analysis
Security checking
Web UI test automation

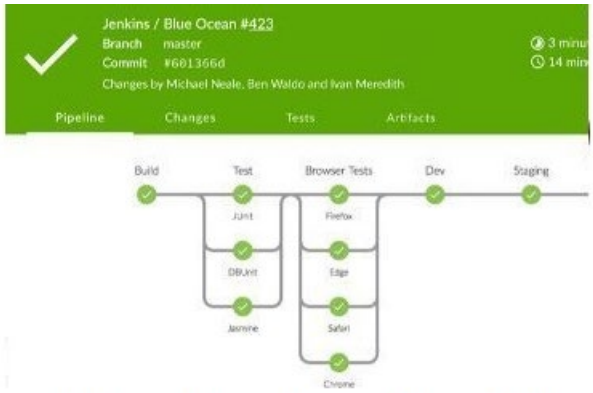

Test Automation Project in IDE
Special files
Auto generated files
IDE
Git Files
GitIgnore
Framework files
Allure
Lombok
Testing Artifacts
Jar App
Site
others
Code Files
Data
Data Helper
Sql Helper
Models
Combining classes
Page
Selectors to methods
Test
Using all 3 from left to test the page and do assertions
Configuration files
Build Files
Maven
Pom
Gradle
Gradllew
External
Docker
CI
Appveyor
Jenkins
GitActions
Doc
Documentation of Project
Summary
Plan
Report
ReadMe
Databases
MySQL
SCV file
Data file


Tests automation pipeline
Tests Automation project
1st Stage
Planning Your tests by testing theory and docs
Inspect the elemets/selectors
Experiment with delays
2nd stage
IDE (IntellIJ,Studio code)
Build System (Gradle/Maven)
Frameworks (Gunit,Allure.Selenium,Web Driver)
External Soft - Appium,Android emulator
Databases (My SQL)
Test reports And Docs
CI Check (if possible)
3RD Stage
Git Hub - Place your code to Web
Jenkins (Server) - Build your code on Docker image
Docker Image and all tests successful
Ansible/ we used Ansible to deploy the application to our target environment, which is running Kubernetes.
Kubernetes cluster - Only when the new container is up and healthy does the Deployment terminate the old one.
CD deployment complete
4th Stage
Tests and deployment reports
Opened issues,Bugs (possible)


C#/C++/UWP tests
Unit tests
Testing Frameworks
Test Generators
Coverage
Databases(access)
Integrated in project
Intellitest
Generics Tests
Nunit
Xunit
Exteranal with depenencies
Characterization tests
Guided test input generation
IDE integration
Helper Method
Fill tests
Testig Method
C++ Test
Test DLL files
Headers
Testing Frameworks
Separate Tests from project
Separate Test Project
Write inside
Include Directives
Microsoft unit tests framework
BoostTest
Google Test
Ctest
Write tests inside project
Transform to .lib file
Traits
Test Build
Configure Test adapters
Avoid Hardcode
Add helper method with hardcoded solution
Use this helper in the test method
How to write test code
Test suites & Test specifications
You can adjust basic constructor & classes & parameters for testing purposes
Understand which code object you writing test on
Settings Waterfall
Understand requirements
Partial classes(1 part in code, 1 part in test)
Understand situation of code testing
Test atributes
Decide Logic of Testing
Businesses Logic
Game logic
Technical Logic
Find Code implementation
Understand what you need to implement
Find test object in tech documentation
Understand how code object works
Find code examples of this code object
Code snippets
Implement your code object

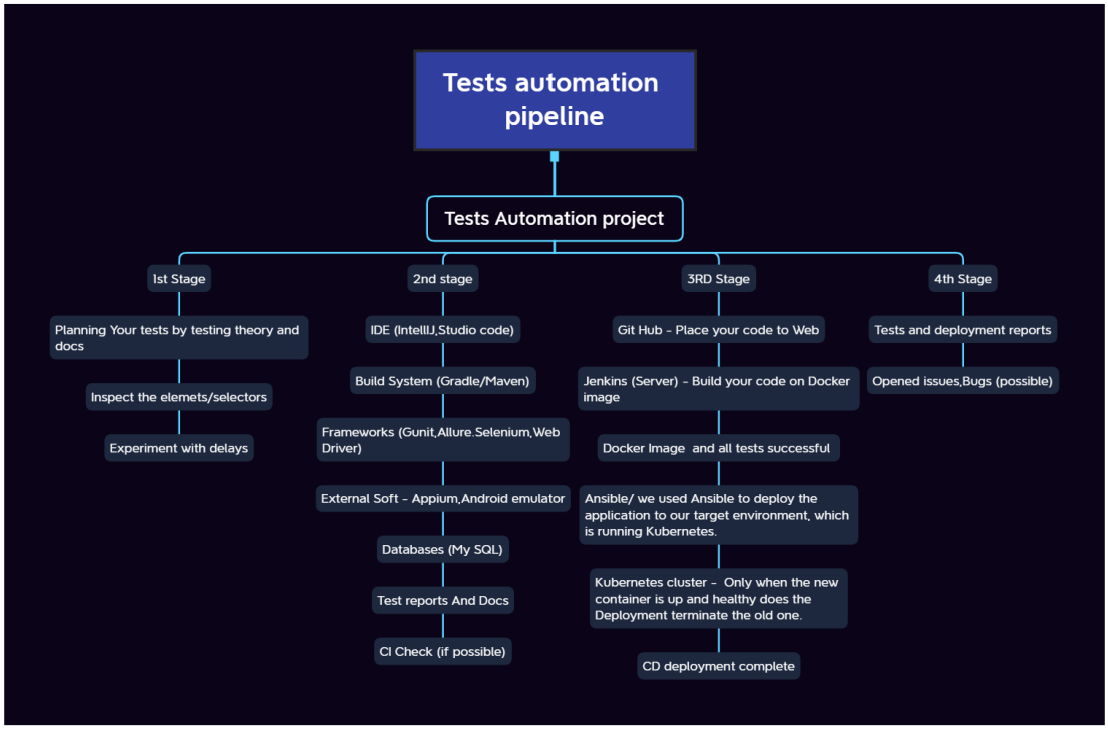
9.Pipelines and Dev Ops



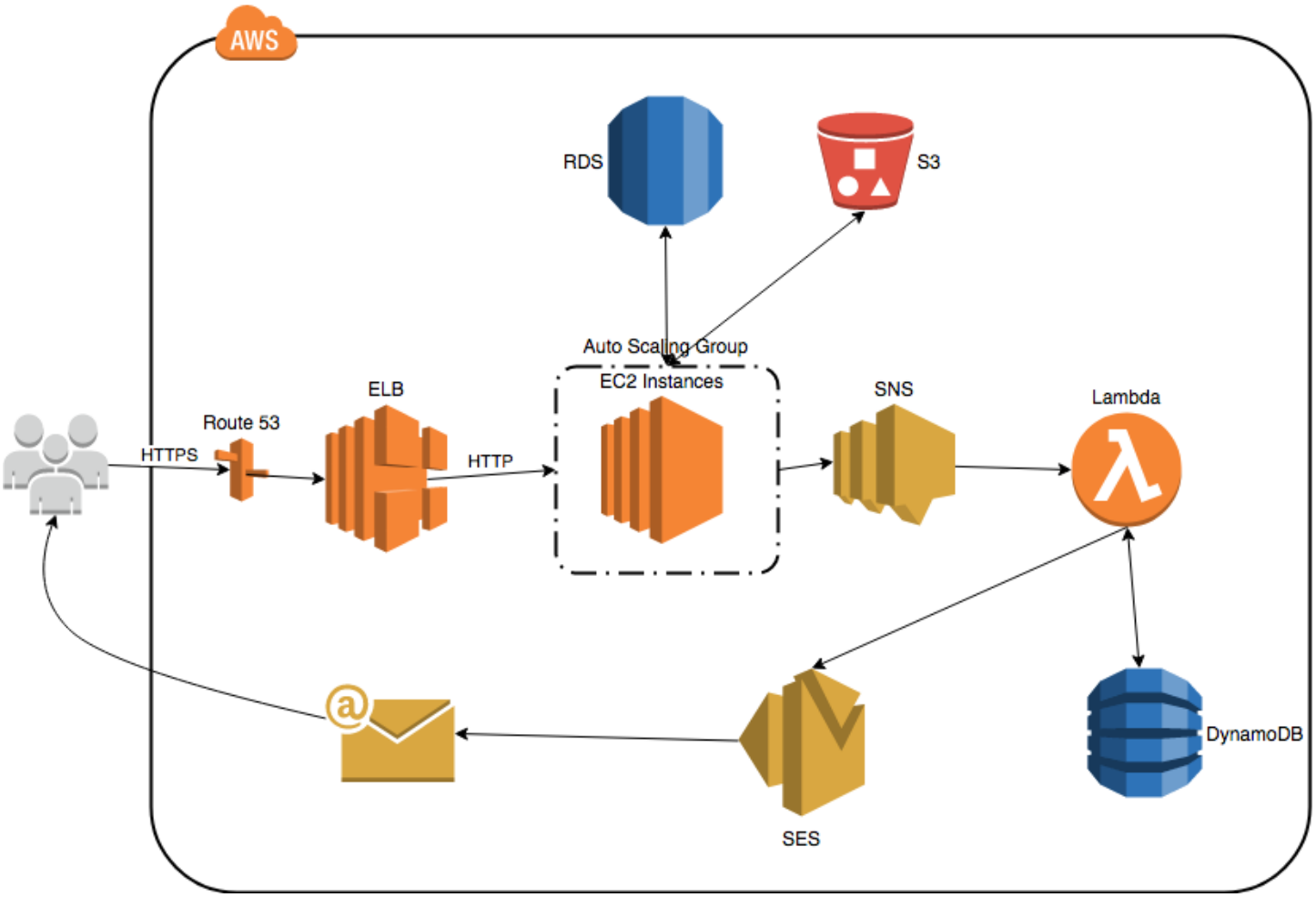


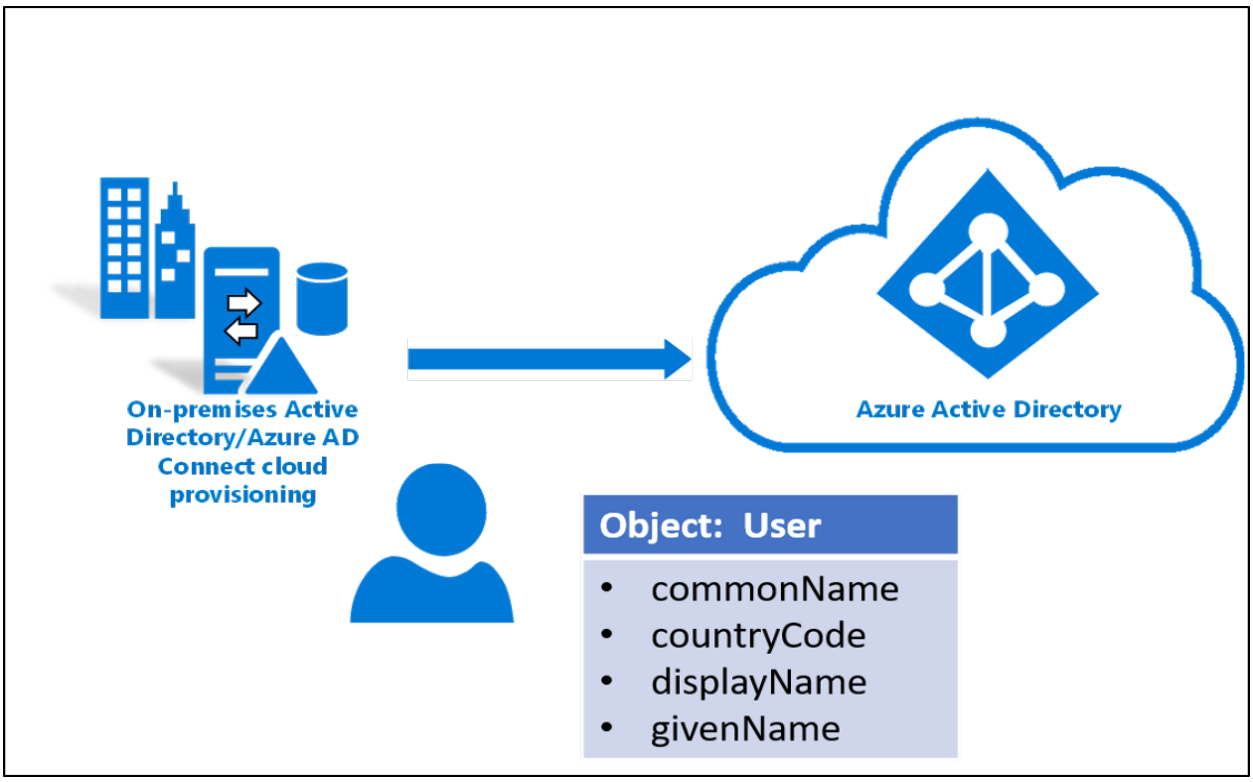
Jenkins Declarative Pipeline

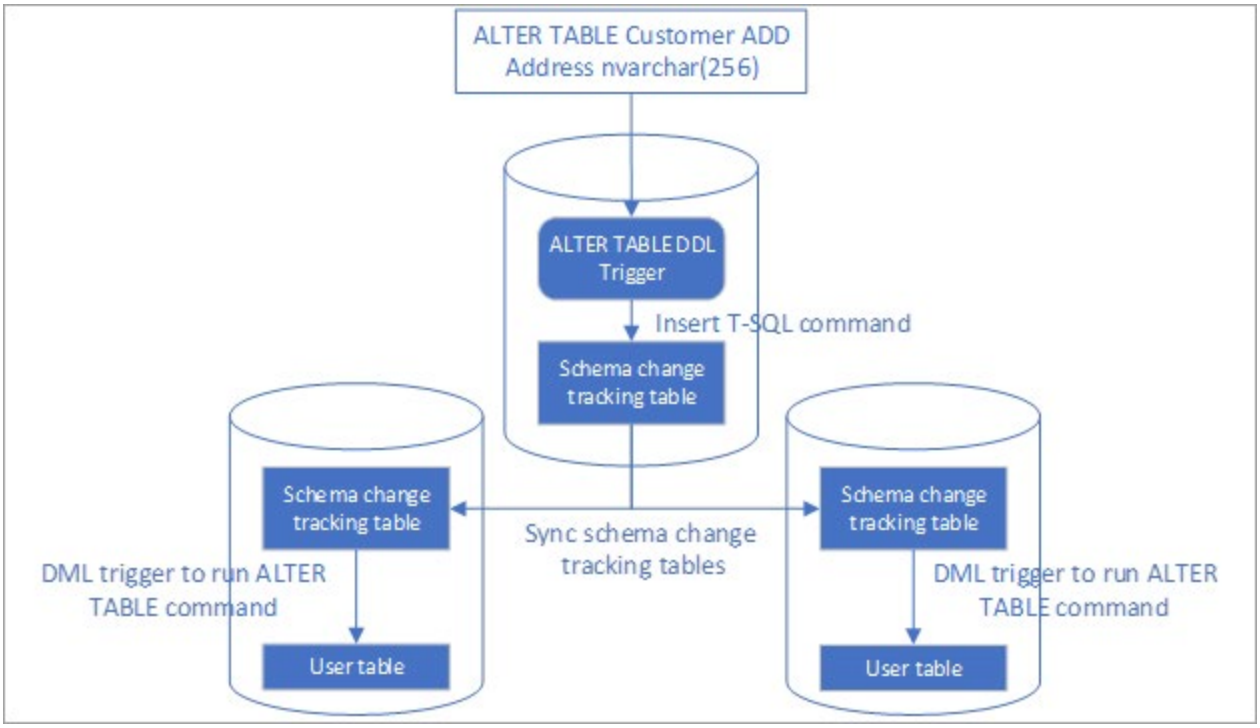
- CI/CD is an integral part of any modern environment that follows the Agile methodology.
- Through pipelines, you can ensure a smooth transition of code from the version control system to the target environment (testing/staging/production/etc.) while applying all the necessary testing and quality control practices.
- In this article, we had a practical lab where we built a continuous delivery pipeline to deploy a Golang application.
- Through Jenkins, we were able to pull the code from the repository, build and test it using a relevant Docker image.
- Next, we Dockerize and push our application - since it passed our tests - to Docker Hub.
- Finally, we used Ansible to deploy the application to our target environment, which is running Kubernetes.
- Using Jenkins pipelines and Ansible makes it very easy and flexible to change the workflow with very little friction. For example, we can add more tests to the Test stage, we can change the version of Go that's used to build and test the code, and we can use more variables to change other aspects in deployment and service definitions.
- The best part here is that we are using Kubernetes deployments, which ensures that we have zero downtime for the application when we are changing the container image. This is possible because Deployments use the rolling update method by default to terminate and recreate containers one at a time. Only when the new container is up and healthy does the Deployment terminate the old one.

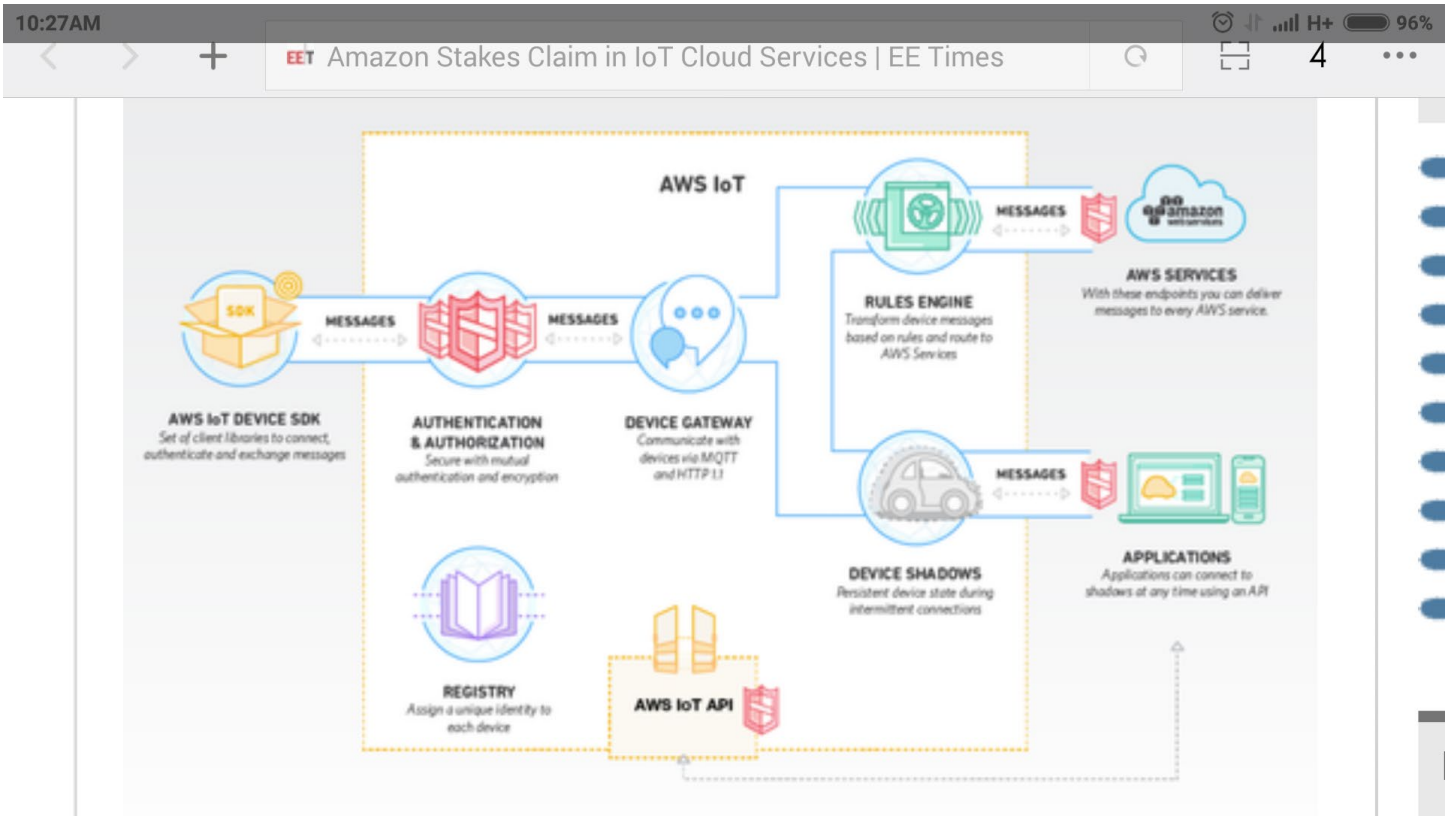


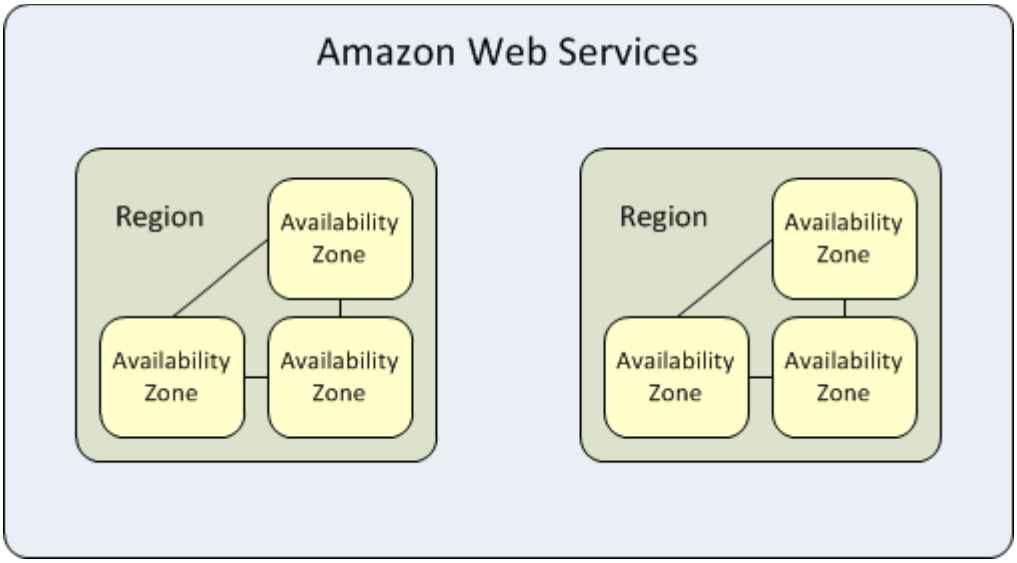
10.AWS Azure (Example of cloud platform)

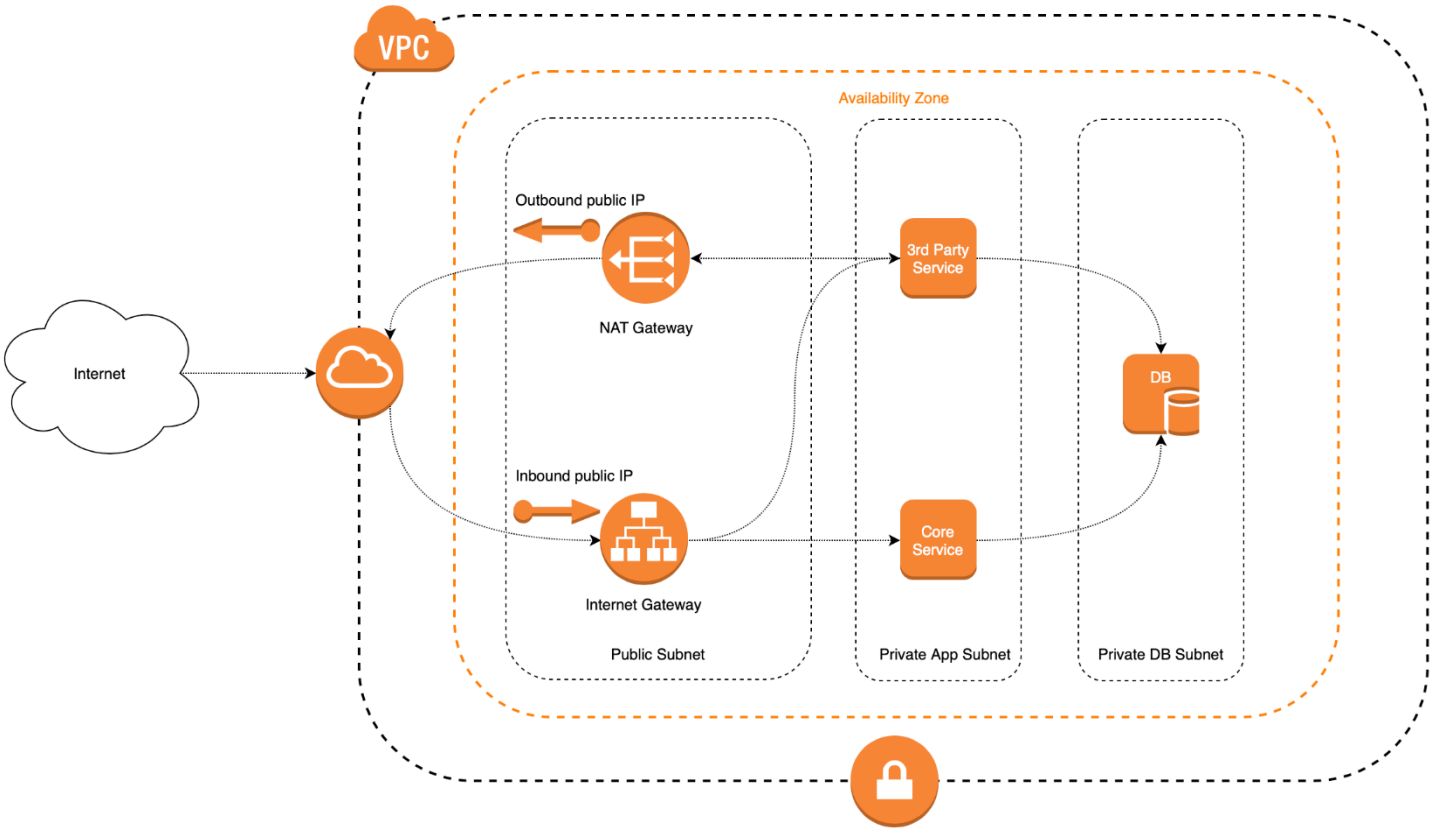


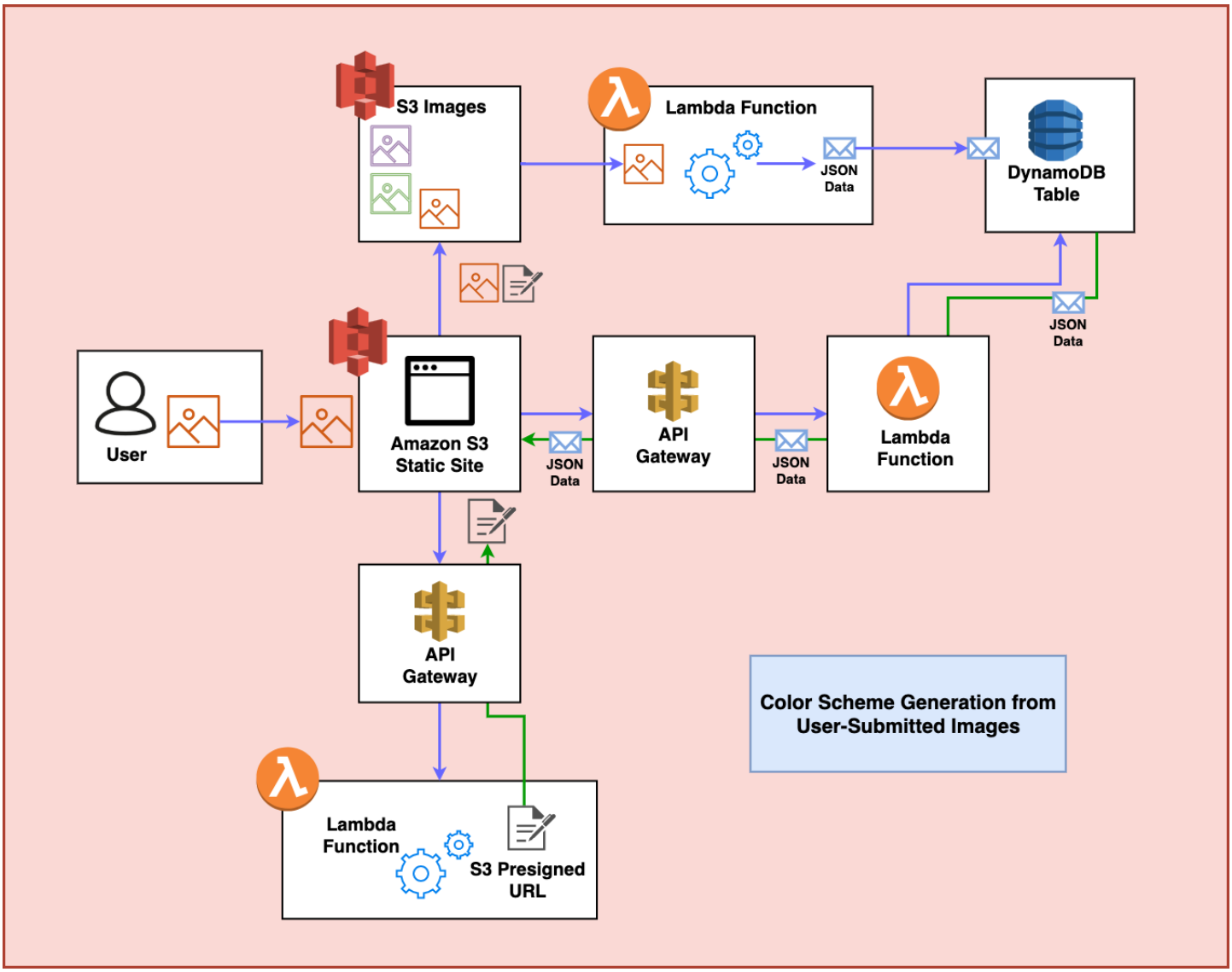


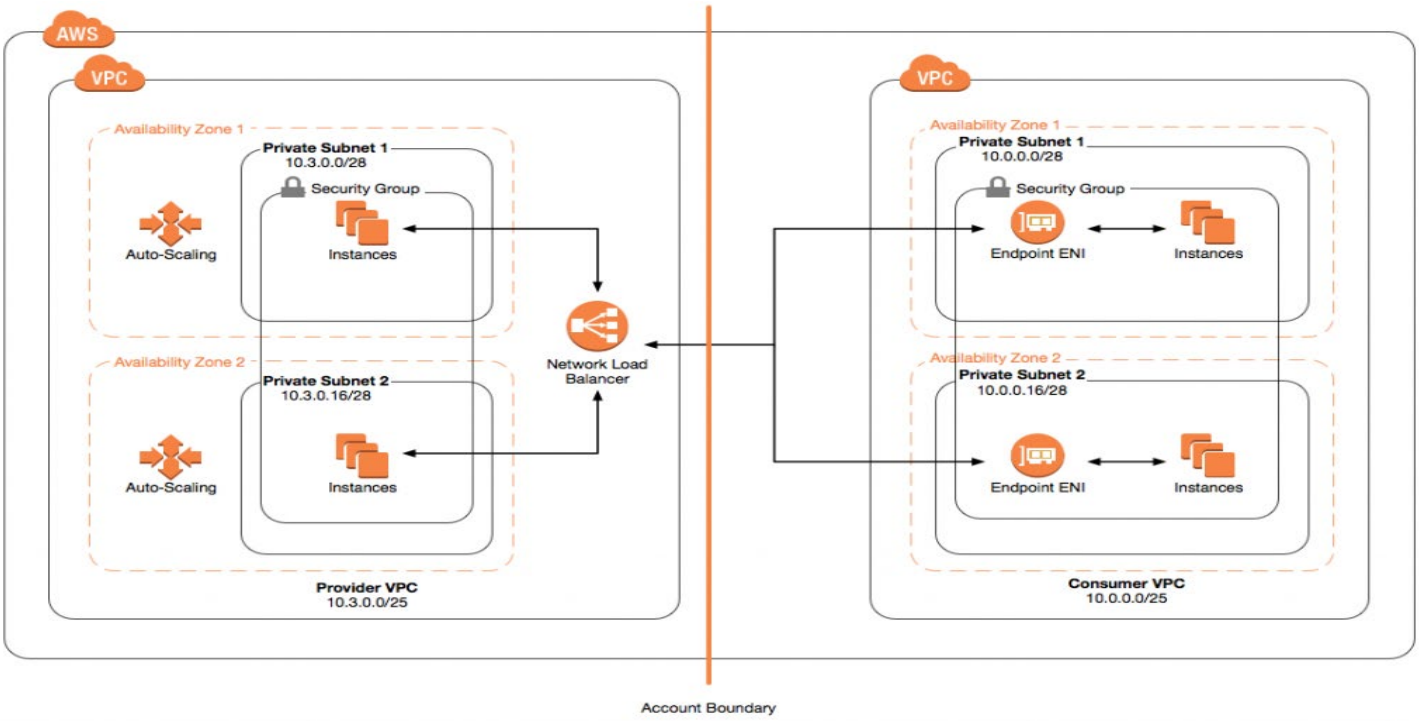

ALTER TABLE Customer ADD Address nvarchar(256)
ALTER TABLE DDL Trigger
Insert T-SQL command
Schema change tracking table
Schema change tracking table
Schema change tracking table
Sync schema change tracking tables
DML trigger to run ALTER TABLE command
DML trigger to run ALTER TABLE command
User table
User table

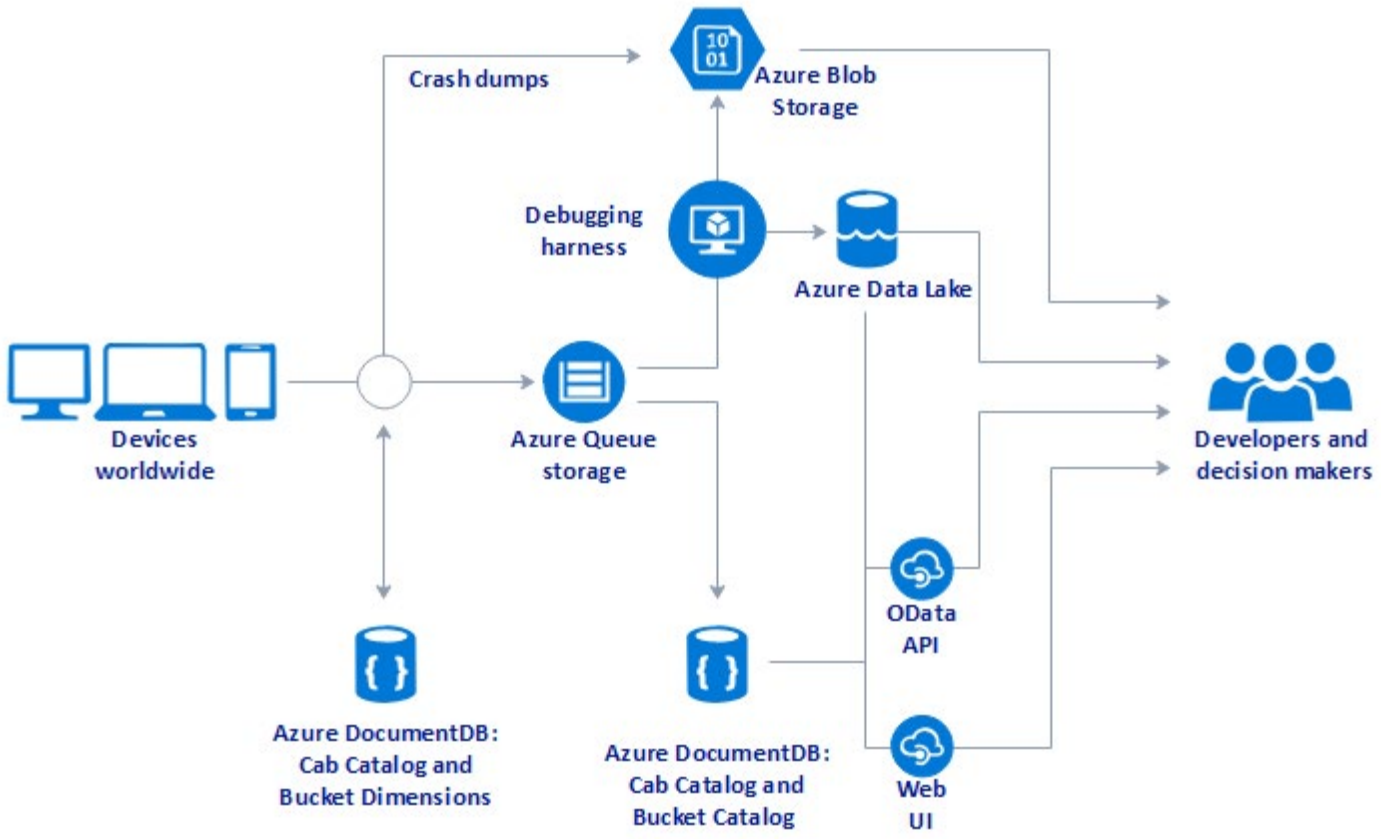

10:27AM
96%
Amazon Stakes Claim in IoT Cloud Services | EE Times
4
AWS IoT
MESSAGES
MESSAGES
MESSAGES
MESSAGES
SDK
AWS IoT DEVICE SDK
Set of client libraries to connect, authenticate and exchange messages
AUTHENTICATION & AUTHORIZATION
Secure with mutual authentication and encryption
DEVICE GATEWAY
Communicate with devices via MQTT and HTTP 1.1
RULES ENGINE
Transform device messages based on rules and route to AWS Services
AWS SERVICES
With these endpoints you can deliver messages to every AWS service.
DEVICE SHADOWS
Persistent device state during intermittent connections
APPLICATIONS
Applications can connect to shadows at any time using an API
REGISTRY
Assign a unique identity to each device
AWS IoT API

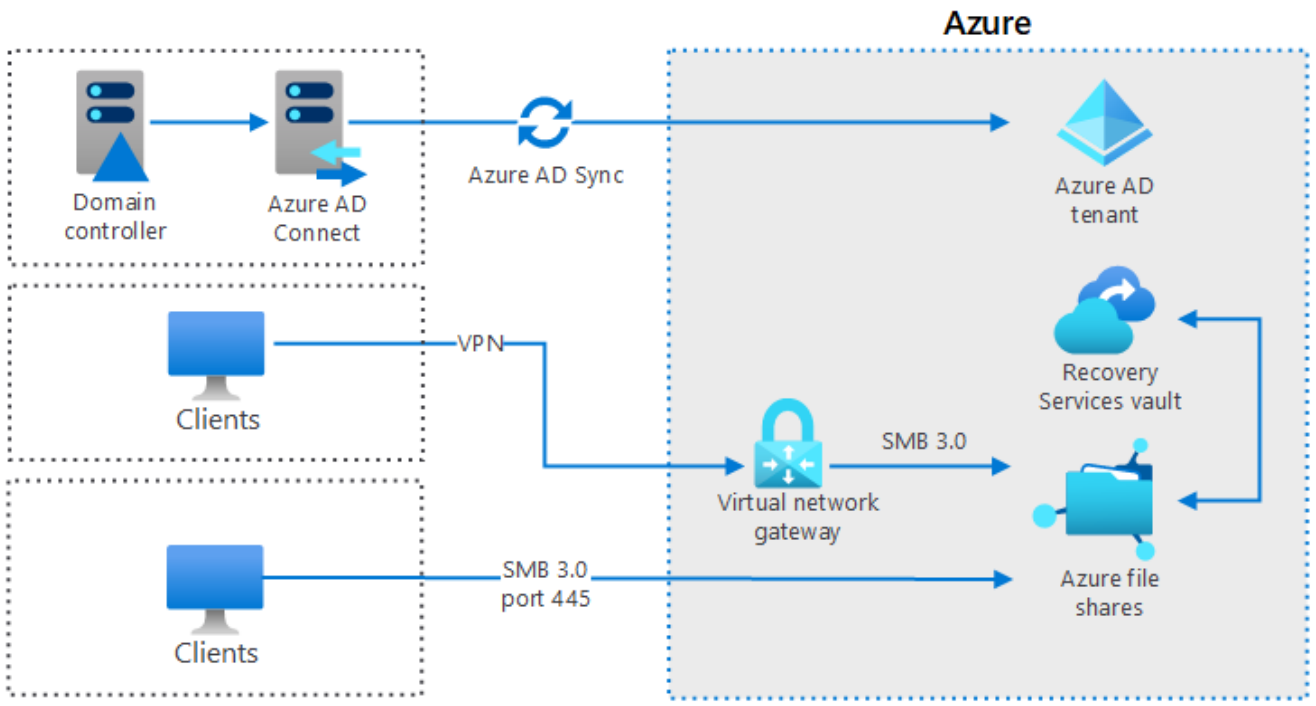

Amazon Web Services
Region
Availability Zone
Availability Zone
Availability Zone
Region
Availability Zone
Availability Zone
Availability Zone

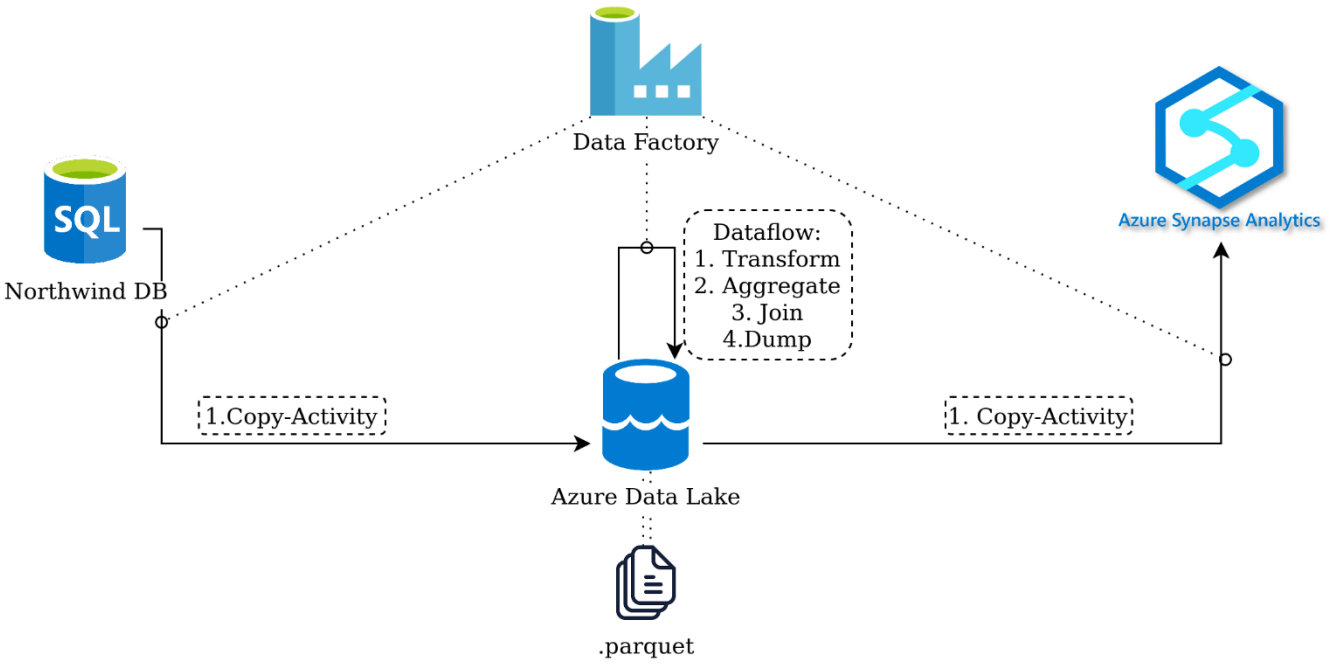

VPC
Availability Zone
Outbound public IP
NAT Gateway
3rd Party Service
Internet
DB
Inbound public IP
Internet Gateway
Core Service
Public Subnet
Private App Subnet
Private DB Subnet

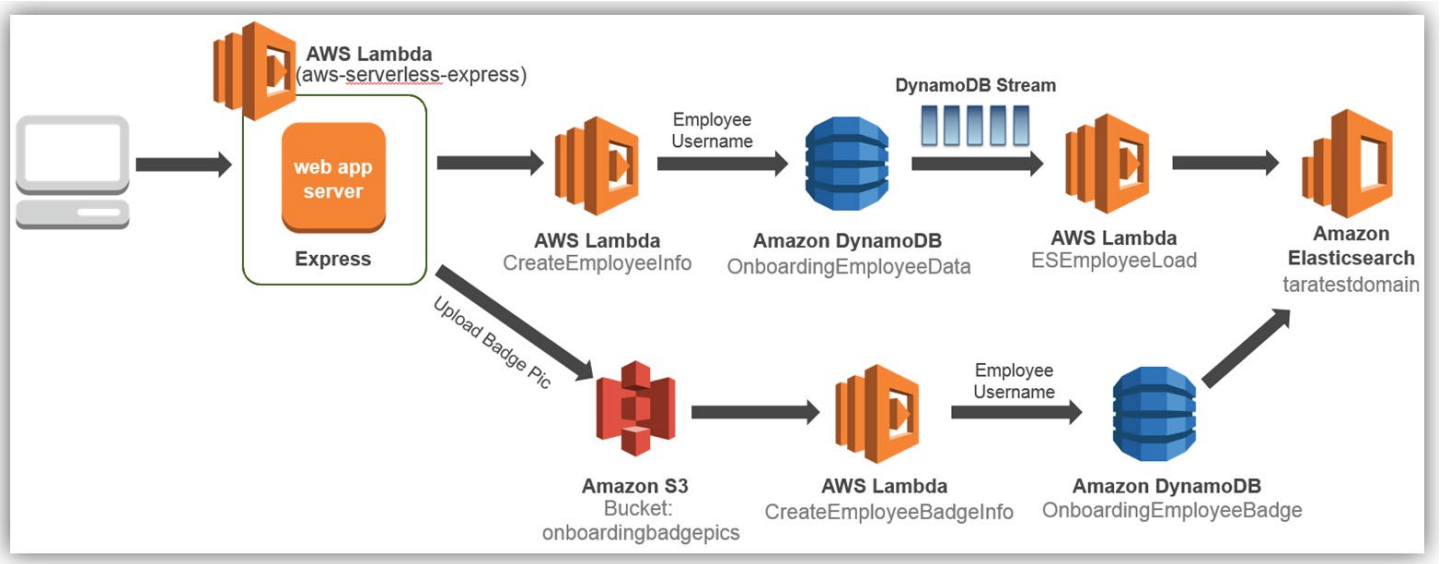

S3 Images
Lambda Function
JSON Data
DynamoDB Table
JSON Data
User
Amazon S3 Static Site
JSON Data
API Gateway
JSON Data
Lambda Function
API Gateway
Color Scheme Generation from User-Submitted Images
Lambda Function
S3 Presigned URL

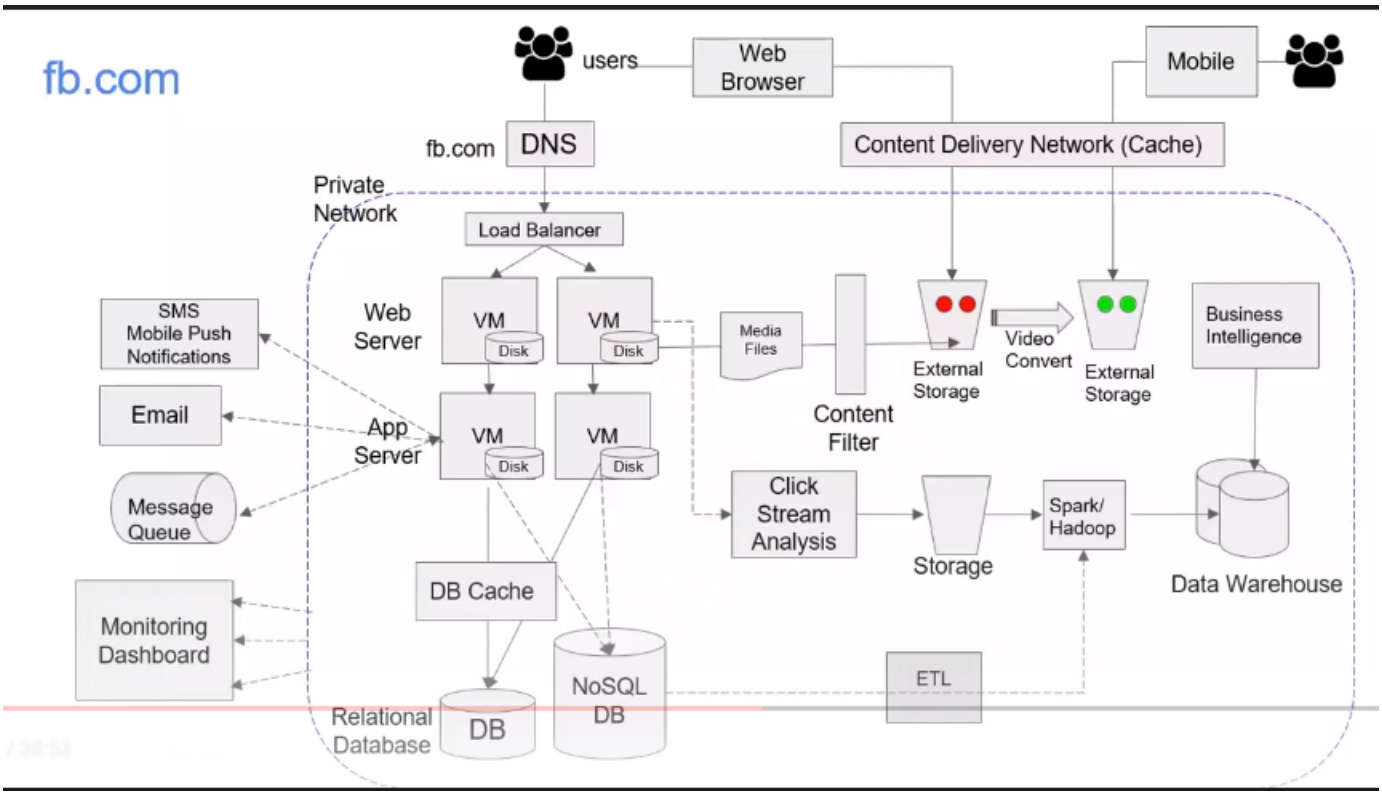

AWS
VPC
Availability Zone 1
Private Subnet 1
10.3.0.0/28
Security Group
Auto-Scaling
Instances
Availability Zone 2
Private Subnet 2
10.3.0.16/28
Auto-Scaling
Instances
Provider VPC
10.3.0.0/25
Network Load Balancer
VPC
Availability Zone 1
Private Subnet 1
10.0.0.0/28
Security Group
Endpoint ENI
Instances
Availability Zone 2
Private Subnet 2
10.0.0.16/28
Endpoint ENI
Instances
Consumer VPC
10.0.0.0/25
Account Boundary

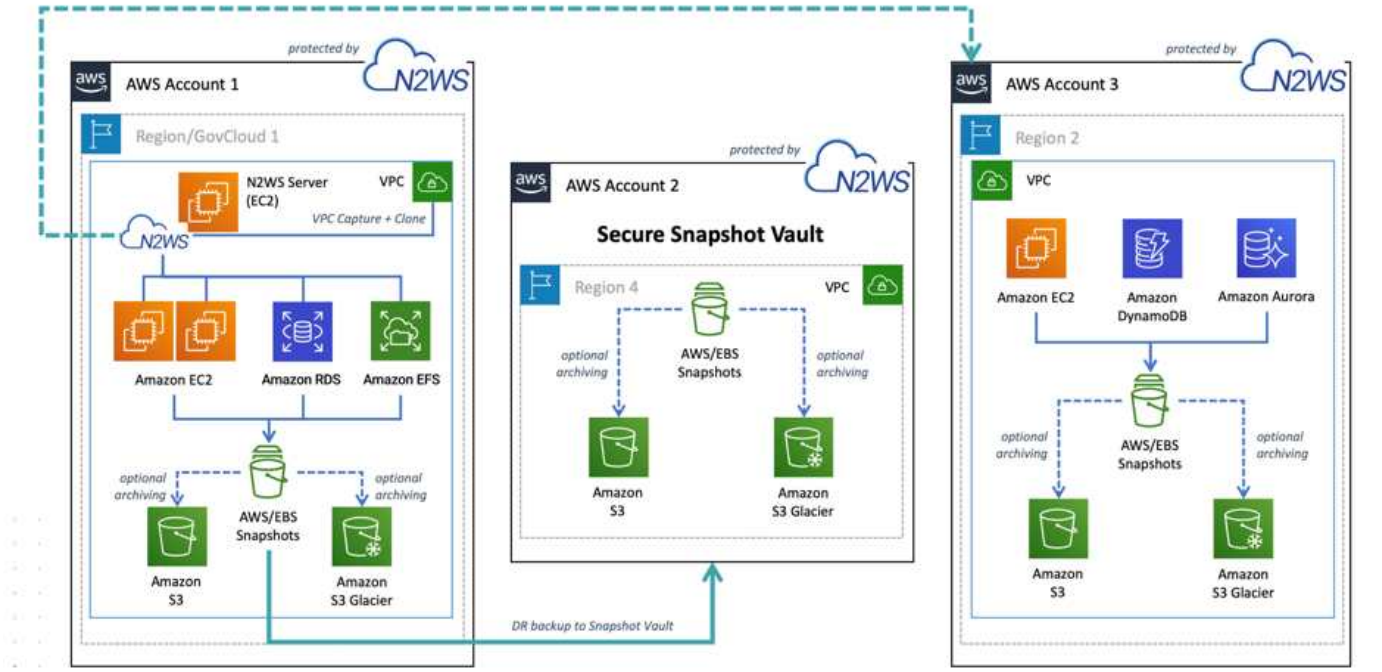

Crash dumps
Azure Blob Storage
Debugging harness
Azure Data Lake
Devices worldwide
Azure Queue storage
Developers and decision makers
OData API
Azure DocumentDB: Cab Catalog and Bucket Dimensions
Azure DocumentDB: Cab Catalog and Bucket Catalog
Web UI

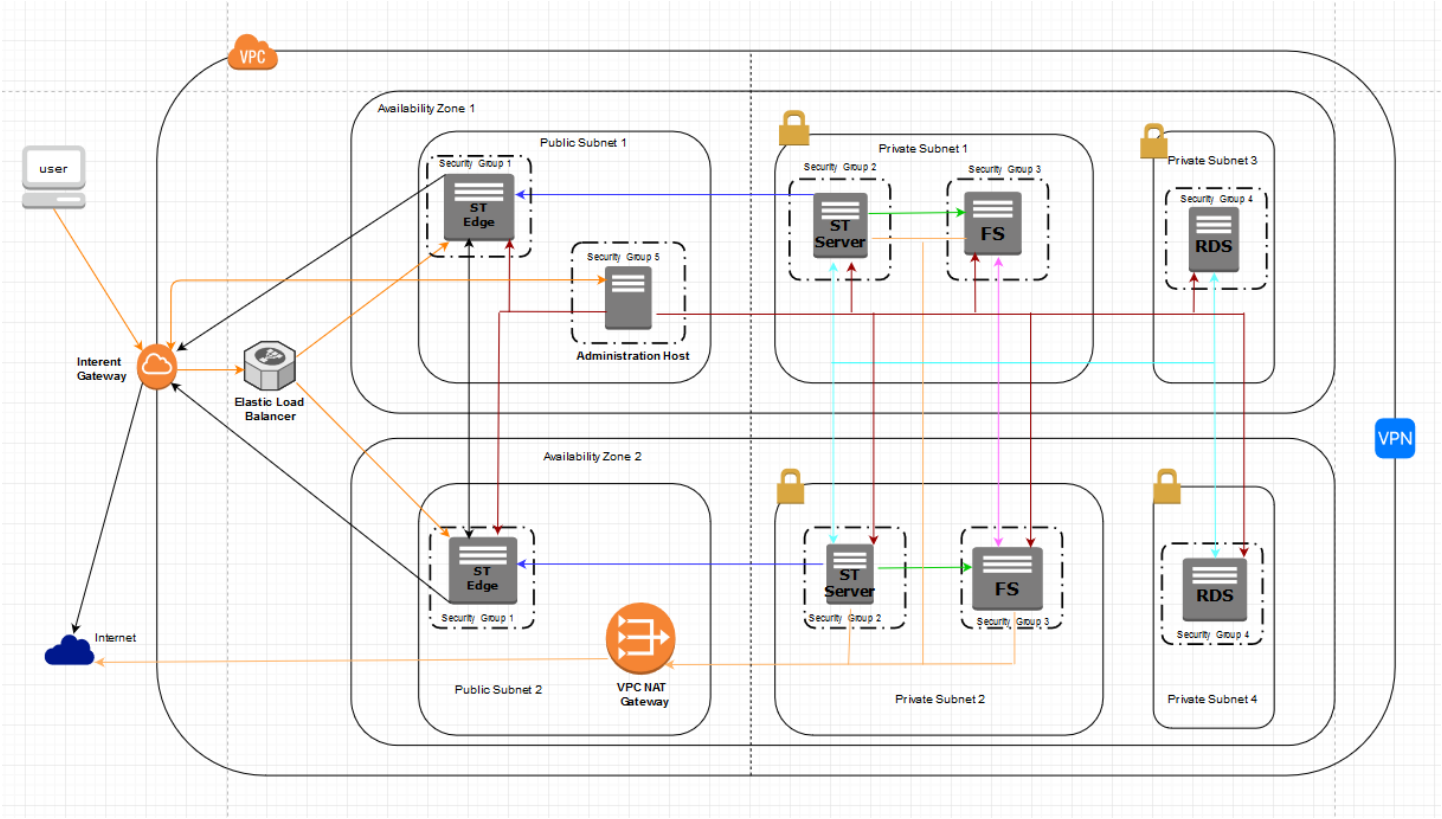

Azure
Domain controller
Azure AD Connect
Azure AD Sync
Azure AD tenant
Clients
VPN
Recovery Services vault
SMB 3.0
Virtual network gateway
Clients
SMB 3.0 port 445
Azure file shares

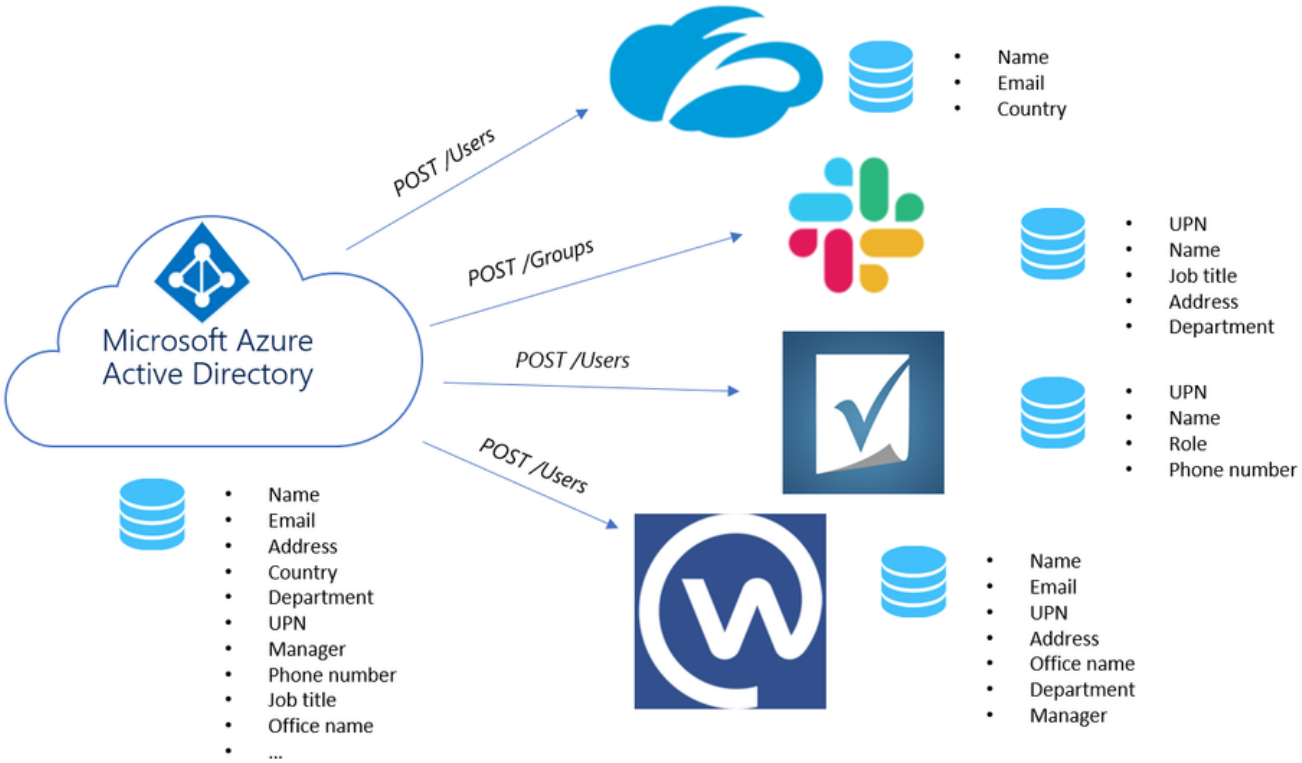

Data Factory
SQL
Northwind DB
Dataflow:
1. Transform
2. Aggregate
3. Join
4.Dump
Azure Synapse Analytics
1.Copy-Activity
1. Copy-Activity
Azure Data Lake
.parquet


AWS Lambda
(aws-serverless-express)
web app server
Express
Employee Username
DynamoDB Stream
AWS Lambda
CreateEmployeeInfo
Amazon DynamoDB
OnboardingEmployeeData
AWS Lambda
ESEmployeeLoad
Amazon Elasticsearch
taratestdomain
Upload Badge Pic
Amazon S3
Bucket:
onboardingbadgepics
AWS Lambda
CreateEmployeeBadgeInfo
Employee Username
Amazon DynamoDB
OnboardingEmployeeBadge

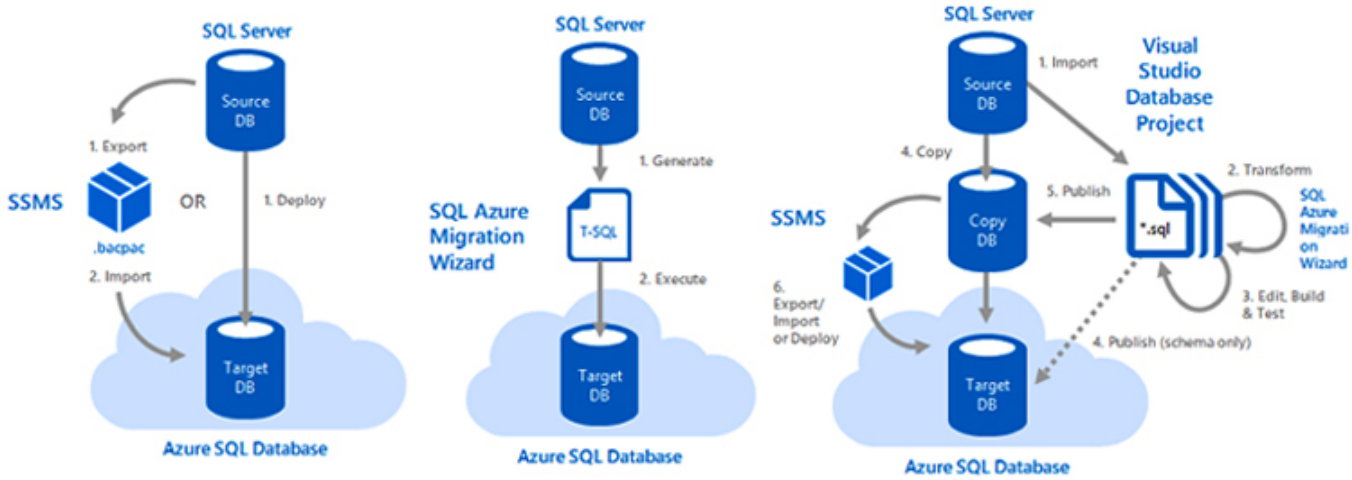

fb.com
users
Web Browser
Mobile
fb.com
DNS
Content Delivery Network (Cache)
Private Network
Load Balancer
Web Server
VM
Disk
VM
Disk
Media Files
External Storage
Video Convert
External Storage
Business Intelligence
SMS Mobile Push Notifications
Email
App Server
VM
Disk
VM
Disk
Content Filter
Message Queue
Click Stream Analysis
Storage
Spark/ Hadoop
Data Warehouse
DB Cache
Monitoring Dashboard
NoSQL DB
ETL
Relational Database
DB


protected by
N2WS
AWS Account 1
Region/GovCloud 1
N2WS Server (EC2)
VPC
VPC Capture + Clone
N2WS
Amazon EC2
Amazon RDS
Amazon EFS
optional archiving
AWS/EBS Snapshots
optional archiving
Amazon S3
Amazon S3 Glacier
protected by
N2WS
AWS Account 2
Secure Snapshot Vault
Region 4
VPC
optional archiving
AWS/EBS Snapshots
optional archiving
Amazon S3
Amazon S3 Glacier
DR backup to Snapshot Vault
protected by
N2WS
AWS Account 3
Region 2
VPC
Amazon EC2
Amazon DynamoDB
Amazon Aurora
optional archiving
AWS/EBS Snapshots
optional archiving
Amazon S3
Amazon S3 Glacier


VPC
Availability Zone 1
Public Subnet 1
Security Group 1
ST Edge
Security Group 5
Administration Host
Private Subnet 1
Security Group 2
ST Server
Security Group 3
FS
Private Subnet 3
Security Group 4
RDS
user
Interent Gateway
Elastic Load Balancer
VPN
Availability Zone 2
ST Edge
Security Group 1
ST Server
Security Group 2
FS
Security Group 3
RDS
Security Group 4
Internet
Public Subnet 2
VPC NAT Gateway
Private Subnet 2
Private Subnet 4


POST /Users
POST /Groups
POST /Users
POST /Users
Microsoft Azure Active Directory
Name
Email
Country
UPN
Name
Job title
Address
Department
UPN
Name
Role
Phone number
Name
Email
Address
Country
Department
UPN
Manager
Phone number
Job title
Office name
...
Name
Email
UPN
Address
Office name
Department
Manager

Migrate SQL Server Database to Azure SQL Database



11.Conclusion

Mind maps and Schemes are a powerful professional instrument to organize all your knowledge and have easy access to multiple knowledge in 1 image. They also helping free your brain from too much information.

Created by Alexander Berg 20/06/2022.