[day-19] AWS+ Selenium

→ online Banking App → Login Page., QA Requirements,

- → app working as per requirements.

* testing

- → Without testing - Fixing Bug a problem
- → no one ~~to~~ use it, it shoud be bug free

↳ manual	Automatic
→ missing tab → QA gets bored same things again	→ Need to have Script, Run against app. — Login, Run all tests.

→ Functional test
→ Regression test
→ Unit-test
(Static code analysis)

} → Cost Reduction
→ IT,
→ Devops doesn't need to go for testing

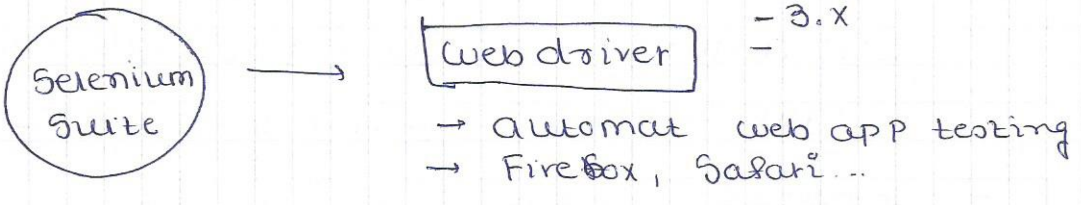
* Selenium :→ support multiple programming Lang

- → it's for Browser based app.
- → Robo - kind of.

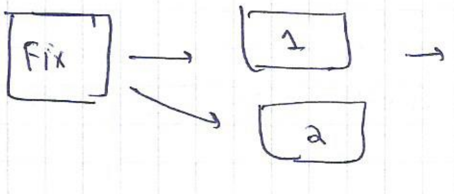
→ Open source tool- ALM (
→ good performance
} Best tool for Functional programming

→ Programming Lang. - operating system →

* Component of Selenium.



* Regression testing →



→ It should not disturb other Part of app. after Fix.

→ Selenium (script testNG (Next Generation)
(Reports 2)
↳ Cucumber (integration with Selelum)

→ Webdriver

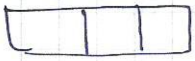
* Addressbook → http:// Locahost:8080/...

↳ Add New Contact –

First Name:	Rajat
Last Name:	a
phone:	—

→ Save.

↳ Selenium shoud do that

→  → element – inspect element – role="button" class=" v-button".

→ "gwt-uid-4" – For every element.

→ System. setProperty ("webdriver. chrome. driver", " E:// ... // cromedrive.exe");

driver. get (" URL:// address book")

↳ driver goes & finds element

driver. findElement (By. id(" ")).

* Referconce Library → Import (Reference Libararies) – selenium – server – standalone. jar.

* testNG

↳ @Before Method
public __

@ Before Class
public void class() { }

↳ test-output → testing-results.xml

↳ creates report out of the statics.

→ Export → Runnable . jar file (with selenium code)

C:// address / ravi.jar (packaged Runnable)

• jar → Click – open the Brower → Run the test →

→ generate . jar file → give to jenkins → add in git & [Run it from jenkins job]

* jenkins →

execute : ~~jar~~ ↲ java -jar [jenkins. jar]

* AWS *

→ on demand delivery → In organization, we need more time for resources.

→ pay as you use. (couple of click & vm is ready).

→ cloud increase server, CPU → you don't need any other stuff.

IAAS (Infrasture as a Service) → to run a app. → infrastructure (EC2, VM)

- ~~Cloud~~ (kubernetes AWS, or gogle)

→ & you need infrasture and cloud provides you.

[Plateform as Service]

- → database eng. & cloud, cloud offer plateform as a service (plateform) → host data for you.

→ create schema, @ table & everything [Get data & put on cloud]

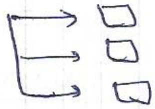
* Software as Service → monitoring, mailing or

* Function as a Service → (Serverless)

* AWS - Devops, machine learning, media person. {Network, media, services for all}

* EC2 (→ create image → convert as image → ansible in 100 another services } → take OS → install Ansible ↳ {create an Image.}

- Load Baranccer → (service)



* Developer tools: Elastic Container Service (ECS) (manage container better)

create cluster →

EC2 linux + Networking

- name: abcd

EC2 type: t3a.micro. (t2a.micro) + t3.micro

Number instances: 2

Storage: 22

Keypair: □

→ master node manged by AWS.
→ node manged by EC2.

create swarm for you. (manager node, you can't see)

→ you can see

→ task defination: how many container (Fargate) AWS (manages everything)

: ngins
Role:
task memory:
} → create container for you & adds to your cluster.

- auto memory increase (no need to worry about cluster)

* EKS (kubernetes serves). (elastic kubernetes)

- cluster ravi → (everything)

- host mano master

* Storage: S3 (simple storage service) (google drive) → object oriented

* Database:

- Developers tools
  - Code commit (git hub Repository)
  you can create a File)

* Amazon github Repository.

git clone & http:// .... } Code Commit ↳

(Amazon hosted service) - git Command

* Code Build (Build)
Deploy: Code deploy.

* Pipeline →

* Iam (create user)

* * Network: VPC (virtual private cloud)
- host appl. → how to get ip?
→ private ip?

→ subnets: to get IP address
- (purchase ip address.)

* Cloud Formation: infrastructure Developer
Ccloud architect.

- template perepcaration (whatever code you need)
- Ansible or terraform (AWS Cloud Formation)

* EKS(elastic Kubernetes Services) -

* Google Cloud *

→ EC2 ≈ VM Instances.
- Kubernetes Engine (create cluster)
↳ Node → how many → default (3)
→ OS
→ Space
→ how to login to master.

* VPC Rule → Firewall Rules
↳ all-traffics (security group)
↑ while accessing app to out side. → b allow every traffic
→ IP ranges, Service account

> kubectl get nodes.
◦ shows only Node } → Not master.
kubectl run ravipod --
image nginx --generator