**DEV-OPS**

**PLAN ==> CODE ==> BUILD ==> TEST ==> RELEASE ==> DEPLOY ==> OPERATE**

|<-------------------------------Dev-Ops Life Cycle--------------------------------->|

|<----------------Continues delivery-------------->|

|<-----Continues Integration----->|

|<-Agile Development->|

***DEVELOPMENT TOOLS***

CI & CD (Continuous Integration and Continuous Deployment)

--> Ant --> Maven

--> Jenkins

--+ Cucumber, Baboo, Cruise Control

--> Version control = Git

--+ Sub version, CVS

***VIRTUALIZATION***

--> Vagrant --> Docker

***TRADITIONS VIRTUALIZATION SYSTEM***

--> VMware --> Citrix --> HyperV --> Zen

***INFRASTUCTURE MANAGEMENT***

--> Puppet --> Chef --> Anisible --> Salt Stack

***WEB SERVICES***

--> AWS

***LEARN ANY SCRIPTING LANGUAGE***

--> C, Bash, Ruby, Python, Java, Shell, Perl,YAML

--+ Variable scope

--+ Data Types

>> Logic

(if, else, case, for, while, until)

***WHAT TO LEARN***

--> Architecture --> Installations --> Configurations

--> Real-time examples and codes by using tools

--> Troubleshooting (when problems occurred)

***Should have knowledge about the following.***

\* How do you provision virtual machines?

\* How do you configure network devices and servers?

\* How do you deploy applications?

\* How do you collect and aggregate logs?

\* How do you monitor services?

\* How do you monitor network performances?

\* How do you monitor application performances?

\* How do elect and remediate when there are problems?

**Important Dev-Ops Tips:**

* Configuration management, e.g. Puppet, Chef, ansible etc, write a sample playbook to perform x tasks, such as install redis cluster
* Explain and draw Release & Build process (CI)
* Explain and draw Deployment process (CD)
* Knowledge of Infrastructure & network & Linux, e.g. OSI layers, Linux concept and commands for day to day works and troubleshooting
* Performance troubleshooting in typical full stack, e.g. Explain performence metrics such as IO, CPU and memory and how to get and analyze them per process or instance
* AWS (router 53, CDN, EC2, EBS, S3, VPC, Cloudformtion etc), e.g. create an automation script (e.g using chef) to fully provision a cluster load balanced security hardened (e.g. SSL enabled and approriate firewall ports) Wordpress stack with auto scaling and self-healing
* Design high availability & performance pattern for infrastructure tool, such as docker registry or Artifactory or Jenkins
* Versioning such as Git, branching strategy for release, git commands, such as git log, git diff etc
* Scripting: Ruby, Python, GOLang or shell, able to whiteboarding basic algorithms and simple tooling use case etc
* Docker (Architecture overview, best practice, sample project with docker, how does it build and deploy with CICD pipeline, challenges working with Docker etc)
* Scheduler: mesos, kubernetes, rancher, Amazon ECS, GCS etc
* Service discovery tool e.g, consul, zookeeper, etcd
* Load balancer: HAProxy (level 4 + 7), Nginx (level 7 only), Netscaler, Amazon EBS
* Logging: elastic stack (ELK), best practice and design pattern for HP and HA
* Build and deploy tools: Jenkins, CircleCI and so many other ...
* 12 factor design for DevOps (Bonus)