**Level One Assessment DevOps Training**

1. Explain DevOps in your language in at least 500 words.

DevOps is a combination of two words, which are teams that are found in every organisation, **Development and Operations**. These two teams and words that during the waterfall period worked isolated from one another, barely having anything to do with each other. But with the coming of age of the agile method of managing a project, DevOps was created.

DevOps in an organisation allows for collaboration and alignment between the Development and Operations team. DevOps is a collection of methods and processes that allows organisations to service customers in faster and more efficient ways that produces better performance from team members and promotes effective synergy.

Sometime not long ago, development and Operations team were teams that worked on their own with no interaction or synergy, there was barely any communication between the departments and this resulted to long hours of unnecessary and delayed work and also inefficiency. But then, continuous and frequent releases of a project began to happen and organisations needed new approach.

DevOps doesn’t just combine both departments but according to Altexsoft, “It is a practice that aims at merging development, quality assurance and operations (deployment and integration) into a single, continuous set of processes.” It is a culture that fosters collaboration and synergy. The developers in an organisation prepare and build the software while the operations team, deploy and maintain the software.

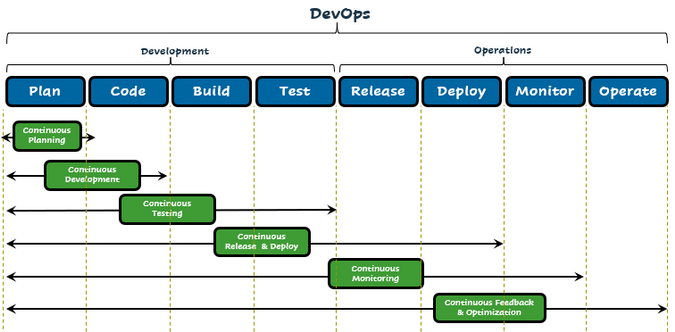
In an organisation without the DevOps culture a problem might be, the developers build the software on their computers and by the time the operators need to test the software in an environment different from the developers, there becomes a problem;

But with DevOps there are tools, principles, models, practices and so on that support the culture of proper communication, collaboration, effective synergy and others that comes with the culture of DevOps. They are what support the DevOps practices and cultures to effectively support the teams.

DevOps has some principles;

1. Develop and test against a production-like environment: The project or software or application being produced need to be tested in an environment that is close to the actual production environment before it is deployed and how the environment will behave when the application is released or deployed
2. Deploying with a repeatable, reliable process: Automation is critical in delivering continuous and an agile workflow. This principle allows the project to be deployed in an automated way and supports processes that allows the department to work in an iterative, reliable and continuous manner. With more deployment of the software, there is more testing of the deployment methods and prevents risks.
3. Monitor and validate Operational quality: This is one of the oldest in the books, quality control and awareness. Applications and projects deployed in an organisation need to be monitored early and continuously to pick up quality metrics, issues, its functionality rate to allow for analysis and validation to be run on them and be kept for posterity purposes.
4. Amplify feedback loops: Every business organisation requires feedback to grow and become better. With this principle, organisations can get early feedback and rapidly act on the feedback and teams should access these feedbacks easily through a communication channel and act on it.

The Development and Operations teams both have job descriptions and processes to meet their goals, with the help of DevOps, an integration has been formed that allows for smooth collaboration, according to opensource.com, these practices are called the DevOps lifecycle.



1. Continuous Planning
2. Continuous development
3. Continuous testing
4. Continuous release and deploy
5. Continuous monitoring
6. Continuous feedback and optimisation.

Also there are several tools that make devops possible, and these tools are integrated at every stage of the DevOps workflow. They include; Gitlab, Jenkins, Git, Gradle, Apache, Selenium, Bamboo, Splunk etc.

DevOps is important in any organisation, not only in the tech world. It helps to ensure smooth flow of work between teams that usually have no business with each other, in doing this, there is high productivity rate, less wastage, better customer service, better end product and my favourite; the best team work/team spirit

2. List and explain some IT tools needed to know to be a good DevOps Engineer

a) Docker

b) Ansible

c) Git

d) Puppet

e) Chef

f) Jenkins

g) Nagios

h) Gradle

i) Bamboo

j) Kubernetes

1. Docker;

It is a Linux form of open source platform that mainly focuses on container and has made it very popular. It packages up the software and all the dependencies into separates containers which makes them portable and more secure. And they are shipped together as one unit. It also integrates well with other tools like Jenkins and Bamboo.

1. Ansible;

Ansible’s selling point is the fact that it is easy to use. It can be used for configuration of infrastructure and automation deployment. It uses a syntax language written in YAML language. It is also agentless.

1. Git

This is one of the most popular devops tools in the world, well-liked by remotes and open- source contributors. It allows for tracking of progress in development work and coordinate work amongst team members. It is a great tool for experimenting as old versions of work can be reverted to and different versions of work can be saved. Also, separate branches of work can be created and when new features are ready to be added, they can be merged they are ready. We would also need to host repositories to integrate Git with workflow like GitHub where members can push their work.

1. Puppet

This is a cross-platform configuration management platform. It allows for management and automation of software inspection, delivery and inspection. It automates infrastructure management and hence allows for software to be delivered faster and more securely. It provides teams with an open-source free version for smaller projects, while Puppet Enterprise (paid version) allows for the management of multiple teams and hundreds of resources.

1. Chef

It is an open-source management configuration tool, also a puppet competitor. It allows for infrastructure to be converted into code. Regardless of the size of your infrastructure, it can automate infrastructure configuration, automate deployment and manage configurations across networks and support multiple platforms.

1. Jenkins

It is a free, open-source Continuous Integration and Continuous Delivery(CI/CD) tool that allows for automation of different stages of the delivery pipeline. It allows for testing and reporting of changes almost in real time. It offers a huge plugin ecosystem which allows it for integration with a lot of other devops tools.

1. Nagios

This is one of the most popular and free, open source devops tools. It allows for monitoring of infrastructure so problems can be found and fixed. It is a tool mostly used for monitoring of things like outages and failures. There are two types Nagios Core and Nagios XI; Nagios XI offers more features and more functionality.

1. Gradle

This is a reliable and versatile build tool that allows code to be written in Java, python, C++ and other languages. Gradle saves compile time and allows faster shipping with a lot of configuration possibilities.

1. Bamboo

This is a Jenkins look alike except it is not free. It offers almost if not all of what Jenkins offers, the difference is Bamboo comes with a lot of pre-built functionalities that have to be manually set up in Jenkins. It has fewer plugins, seamlessly integrates with other tools and products, provides access to built-in git and saves a lot of configuration time.

1. Kubernetes

This is a new container orchestra platform. With kubernetes, containerised apps can be deployed into a group of computers instead of one. It also automates the distribution and schedules of the containers. Kubernetes is better used for a large scale of containers. The cluster of kubernetes consist of one master node and worker nodes. The master nodes implement predefined rules and deploys the containers to the worker nodes. Kubernetes and dockers can be used together but while kubernetes is an orchestration platform, docker builds, distributes and runs containers.