Developers are responsible for new product features, bug fixes, security updates, code refactoring and much more.

Software products will eventually need to be deployed to the development environment to check integration and other aspects of the product.

Development environments can vary wildly to production environments. When deploying it from development to production, warnings and errors often occur.

Operations teams are responsible for managing servers, ensuring that services are running, and products are running correctly, that they are efficiently monitored and their server uptime is maintained, and growing capacity as the company/product scales and grows. Operations are met with similar errors and warnings as the development team when deploying development products.

Both teams work in silos, Operations think that developers are throwing the issues to them to deal with. Developers think that Operations are very slow and lacking the frequency that the development team need to deliver.

DevOps breaks the silos so that both teams are working together, sharing responsibilities, beginning to deploy the infrastructure development need as code, automate that pipeline throughout.

With the DevOps principles in place, greater efficiency can be found through working together, automating the pipeline, sharing responsibilities, and deploying infrastructure as code. However, these must be supported by good engineering practices, which are continuous integration, continuous delivery and continuous deployment (CI/CD).

CI/CD

Development team building products, writing tests to ensure the product works correctly, and storing that code in a source control system to allow other developers to work on it. Practice is known as CI.

Next is to engage with the Operations team and work together on deploying to either on premise or cloud infrastructure, utilising infrastructure as code and ensuring all tests pass here too. Infrastructure as code should be stored in Git to ensure that version control is continued throughout the product.

CI/CD server should be put in place to manage all these services (e.g. Jenkins).

Once the tests have passed, move to a pre-production environment, and complete the exact same practice to ensure quality is running throughout. Known as continuous delivery.

Finally, all the practices combined and running tests all the way through to production is known as continuous deployment.

Overall:

In turn, this will deliver faster time to market, improve team collaboration, continuous release cycles, automating scalable environments, and increase quality due to automated testing throughout.