**Synapsis**

**Scalable Continuous integration and Continuous deployment using DevOps**

**Introduction:**

Quality and Time-To-Market (TTM) are crucial factors for the success of any software product. This can be achieved with Continuous integration and Continuous deployment (CI-CD) by using various ‘Development-Operations’ (DevOps) tools. CI-CD pipeline enables organizations to develop high quality products and seamless delivery to the end customers. Tailoring various DevOps tools to construct CI-CD pipeline provides timely overview on product health status, brings different teams across the world together and delivers faster and reliable software to the stakeholders.

CI-CD is an emerging field in the industry where many companies lack this implementation. Only renowned organizations like Google, Microsoft and other major players are having full-fledged CI-CD setup at place. Rest of the organizations are still struggling go get there.

**Problem statement:**

First priority of any product is to bring it to the market before your competitors does. But before bringing any product to the market, below are the stages the product has to walk through.

1. Deliver the features developed into source archive, from different teams across the globe.
2. Test the quality of the code via unit tests, integration and automated tests. (Testability, maintainability, scalability and security).
3. Test across different environments (different Operating systems, browsers and software set combination).
4. Deploy it in test servers and do sanity check.
5. Promote to the production servers.

Assuming an average application with 1millian lines of code requires at least 15 days complete all above activities. Manual activities are always prone to error, which is also the risk this process.

Below is the typical pipeline of a software delivery:



How about bringing 15 days to one day? Proposed solution talks about it.

**Proposed solution:**

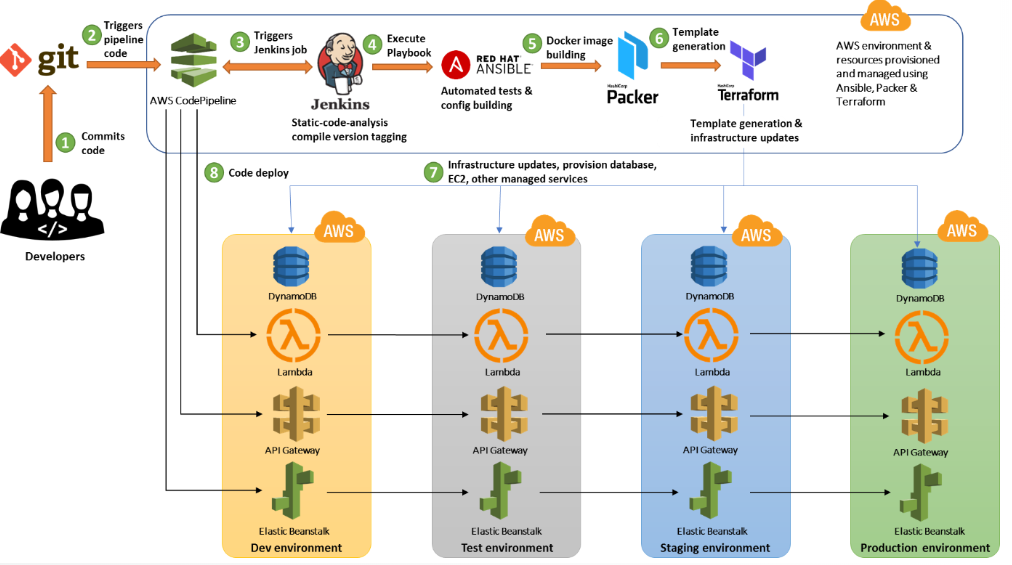
Bringing 15days of labor to 1 day is possible with the best use various DevOps technologies and tools available in the industry under one umbrella. Our ‘Scalable CI-CD using Devops’ addresses different stages of problem statement is described below:

|  |  |
| --- | --- |
| **Problem Statement** | **Proposed solution** |
| Deliver the features developed into source archive, from different teams across the globe. | Use ‘Git’ as source archive, where teams across the globe can deliver the changes at any time. |
| Test the quality of the code via unit tests, integration and automated tests. (Testability, maintainability, scalability and security aspects to be covered). | Use Nunit, Mstest, Jmeter to run unit/automated tests. Use ShoScar and FxCop to check for maintainability and vulnerability of the code being developed. |
| Test across different environments (different Operating systems, browsers and software set combination). | Use Virtual machines (VMs) with AWS-cloud combination to pull various operating systems along with the Docker-Container technology to host different sets. This technology is new to the industry. This is scalable on need basis, automatically pulls as many VM’s are required to complete activities in stipulated time. |
| Deploy it in test servers and do sanity check. | Use Ansible to perform product deployment in bulk (like 500 machines with one click) into different VM’s. |
| Promote to the production servers | Check the quality history of the product and automatically push to the production. |

Along with the above:

1. Product is being watched and tested around the clock.
2. Visibility and monitoring of the pipeline is made through different dashboards, which provides view on different stages the current product is going through, its health status and exactly identifies the reason behind if any failure occurs in pipeline.
3. NuGet packages being used to get various software set required.
4. Optional: Storing the product deliverables in the **Artifactory** for the reference of other teams.

Below image depicts the proposed solution.



In abstract view the CI-CD pipeline setup with DevOps tools brings in below into the development practices.

* **Infrastructure as a code:** Treat infrastructure the same way developer treats the code with all best practices and tests.
* **Continuous deployment:** Core concept of a DevOps strategy. Its primary goal is to enable the automated deployment of production-ready application code.
* **Automation:** Focuses on setup, configuration, deployment, and support of infrastructure and applications.
* **Monitoring:** Communication and collaboration is fundamental in a DevOps strategy. To facilitate this, feedback is critical.
* **Security:** Focus on security is still of paramount importance. Infrastructure and company assets need to be protected, and when issues arises, they need to be rapidly and effectively addressed.

**Conclusion:**

A software industry without DevOps setup is unimaginable. Setting up CI-CD pipeline is bringing in various heterogeneous tools together and getting the best out of it to achieve the high quality product with quickest possible time to the market. Our proposed solution is generic in nature, address the core issues in product development-delivery lifecycle and suits any organization at any scale.

DevOps is the way to make both developers and customers happy.

**References:**

<https://dzone.com/articles/focus-on-cicd?fromrel=true>

<https://cloudtweaks.com/2019/05/devops-secure-and-scalable-ci-cd-pipeline-with-aws/>