Myself Charan.Having overall 5 years of exp as a devops and aws solution architect.

Working for RSI.

Currently serving for Webpt project.

1. Planning
2. Coding
3. Building
4. Testing
5. Releasing
6. Deploying
7. Operating
8. Monitoring

The difficulty faced in large DevOps Team that maintain large huge IT infrastructure can be classified briefly into six different categories.

1. Infrastructure Automation
2. Configuration Management
3. Deployment Automation
4. Performance Management
5. Log Management
6. Monitoring.

DevOps is the culture of development. The goal is to bridge the gap between IT operations and development to improve communication and collaboration, create more seamless processes

The DevOps lifecycle is all about agility and automation.

1. Establish an agile development process
2. Adopt cloud computing
3. Adapt your processes to a CI and CD workflow
4. Automate your software deployment
5. Automate software testing
6. Implement continuous deployment
7. **Continuous Monitoring & Feedback**

**1. Continuous integration**

Developers regularly merge their code changes into a shared repository where those updates are automatically tested. Continuous integration ensures the most up-to-date and validated code is always readily available to developers.

### 2. Continuous delivery

Code changes are automatically built, tested, and packaged for release into production. The goal is to release updates to the users rapidly and sustainably. To do this, CD automates the release process (building on the automated testing in CI) so that new builds can be released at the click of a button.

### 3. Continuous deployment

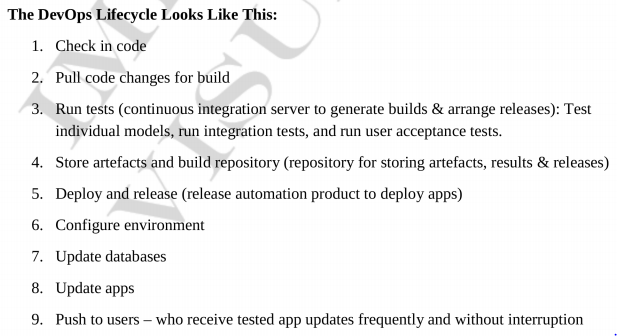
Continuous deployment is the fully automated version of CD with no human (i.e., manual) intervention necessary.

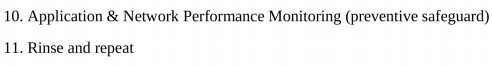
In a [continuous deployment process](https://www.lucidchart.com/blog/continuous-deployment-best-practices), every validated change is automatically released to users.

### 4. Continuous monitoring and feedback

Again, the majority of the monitoring process should be automated to provide continuous feedback.

This process allows IT operations to identify issues and notify developers in real time. Continuous feedback ensures higher security and system reliability as well as more agile responses when issues do arise.





Tools are used to enforce the entire devops life cycle would be

1. **SourceCodeMgmt (Or) Version Control systems : Git,GitHub,BitBucket,GitLab**



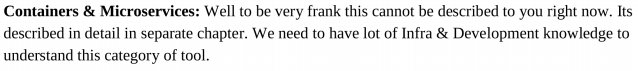
1. **CI Servers : Jenkins,GitLab**
2. **Build Tools :**

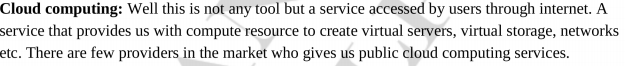


1. Artifact repo’s : Nexus,
2. Configuration Management : Ansible



1. Container and Microservices

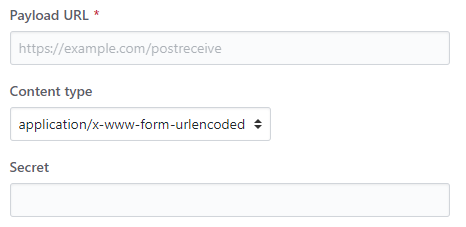


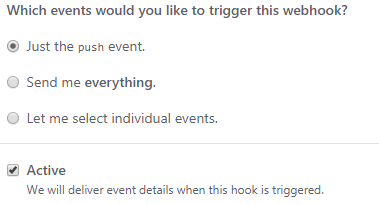
1. Cloud Computing
2. Monitoring Tools
3. Shell Scripting

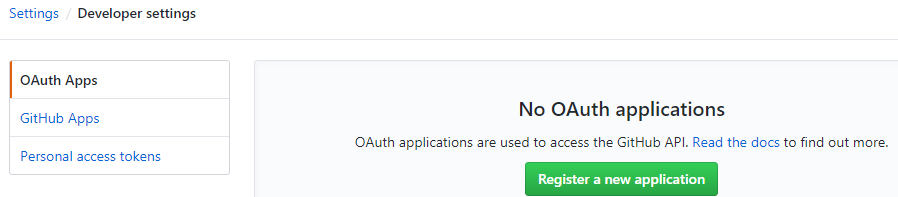
**configure jenkins with github**

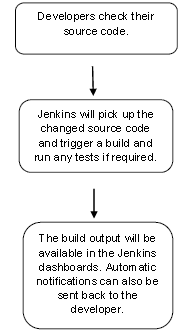
[GitHub Services are being deprecated](https://developer.github.com/changes/2018-04-25-github-services-deprecation/). Please contact your integrator for more information on how to migrate or replace a service with [webhooks](https://developer.github.com/webhooks/) or [GitHub Apps](https://developer.github.com/apps/differences-between-apps/#about-github-apps/).

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we’ll send a POST request to each of the URLs you provide.









JENKINS\_URL which can be accessed as ${JENKINS\_URL}.

Load Statistics

This pages displays graphical data on how busy the Jenkins instance is in terms of the number of concurrent builds and the length of the build queue which gives an idea of how long your builds need to wait before being executed.

Script Console

This screen lets you run Groovy scripts on the server.

Jenkins is capable of handling parallel and distributed builds. In this screen, you can configure how many builds you want. Jenkins runs simultaneously, and, if you are using distributed builds, set up build nodes. A build node is another machine that Jenkins can use to execute its builds.