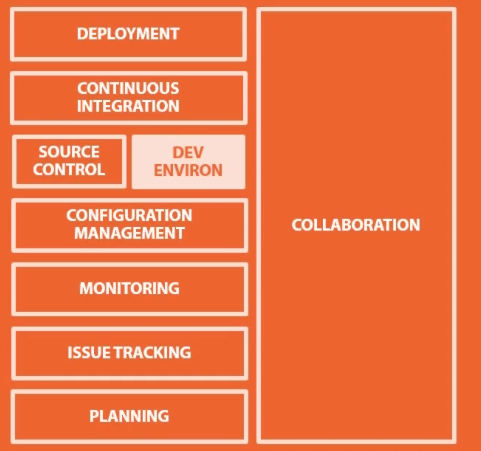
**DevOps-** It is a cultural changes in an organization where you need empowerment, trust, team work, accountability, learning. You need to identify bottlenecks, alter team structure, streamline procedures.



**Tools for various phases-**

**Collaboration** can be done by *real time chat, discussion rooms, knowledge repository, Campefire, Slack*.

Blogs to maintain discussions, decisions made (*wordpress*), *Github wiki* to keep codes, strategies.

**Planning**- *Kanban boards*- to share information online in real time, *feature review meeting* every week.

*Trello*- tool for creating boards for different functions. *Visual studio online.* These help to maintain transparency among team.

**Issue Tracking tools**- *Zendesk*- collective tool for knowledge, manage tickets. *JIRA*- tracking issues, also for continuous integration and deployment.

**Monitoring tools**- *Logstash-* Managing events and logs, collecting parsing, storing, exploring various logs. – free and open source.

*Microsoft system center*- to do group based collective management as well as monitoring of multiple systems in Microsoft environment.

*Kibana-*Create charts, get ranking to analyse information. Works well with logstash and elastic search.

Graphite, StatsD- to plot charts, trends.

Pure monitoring tool- *New relic*- monitor and ping services. Look at response time, health check, latency.

**Configuration management-** for maintaining consistency of configurations between multiple servers, to update policy, any configuration changes. Push and pull.

*Chef-* Manage lot of machines from central server. Has azure integration, loud integration.

*Salt-* create agent list model, centralized server, master server, uses SSH.

*Puppet*- creating declarative infrastructure definitions. Maintaining lifecycle from provisioning to runtime and deployment, reporting.

*Ansible*- automate configuration management,provisioning.

*Powershell DSC-* enforce configuration across windows, linux. Deploy, manage configuration information. Manage files, Start and stop services. Enable and disable roles and features. Manage registry. Deploy software.

**Source control-** for controlled access, guarding software assets**.** Track history of changes to ensure what is tested in deployed. Compare configuration in source control. We get audit trail of what happened, when and who did it, what changed.

*Github-* can use for pulling for configuration management tool, continuous deployment tool.

**Dev environment-** maintain consistency between environments.

*Codenvy-* let you do browser based development. Can build, test, deploy code all from a browser. can see real time code changes. Lead to collaborative development between teams.

*Vagrant-* treat infrastructure as code , as a config file to run in any environment. code on single environment and can run on any OS, can configure networks, set up multi- machine configuration. no need to configure environment for multiple servers, can just use vagrant box from source control.

**Continuous integration-** Integrating code multiple times a day to mainstream code without hassles. Having centralized builds coupled with automated suites for testing and automated security penetration and code inspection suites that check quality before deploying build.

**TeamCity-** good for .Net solutions, works with visual studio. Let u test before committing changes, shows progress report.

**Jenkins (earlier Hudson)-** monitor software builds, integrate changes

Travis CI- Integration capabilities, integrates with Github

**Deployment**- continuous deployment in DevOps- Every change goes right to production. Continuous delivery- software can be released to production at any time.

*Cloudformation*- handle environment deployments, not code deployment. Create multi-tier application by click of button.

*Packer*- create machine images for platforms. Image template creation provision is available. Images are present with latest code everytime and you just need to replace server instead of updating and patching.

*Docker*- Linus technology for creating isolated containers that are portable between machines. This is OS virtualization where we carve OS in individual containers with isolated resources, CPU, memory, network, virtual interfaces, content isolation. Can run on any hardware. Here we can integrate code, push it to different containers, test it, do performance testing

*Octopus*- great for ASP.Net, integrate with TeamCity. Take build artifacts and deploy on different environments .Provide audit trail. Manage application configurations, introduce approvals (manual process), self service deployments.

*Go*- For continuous delivery