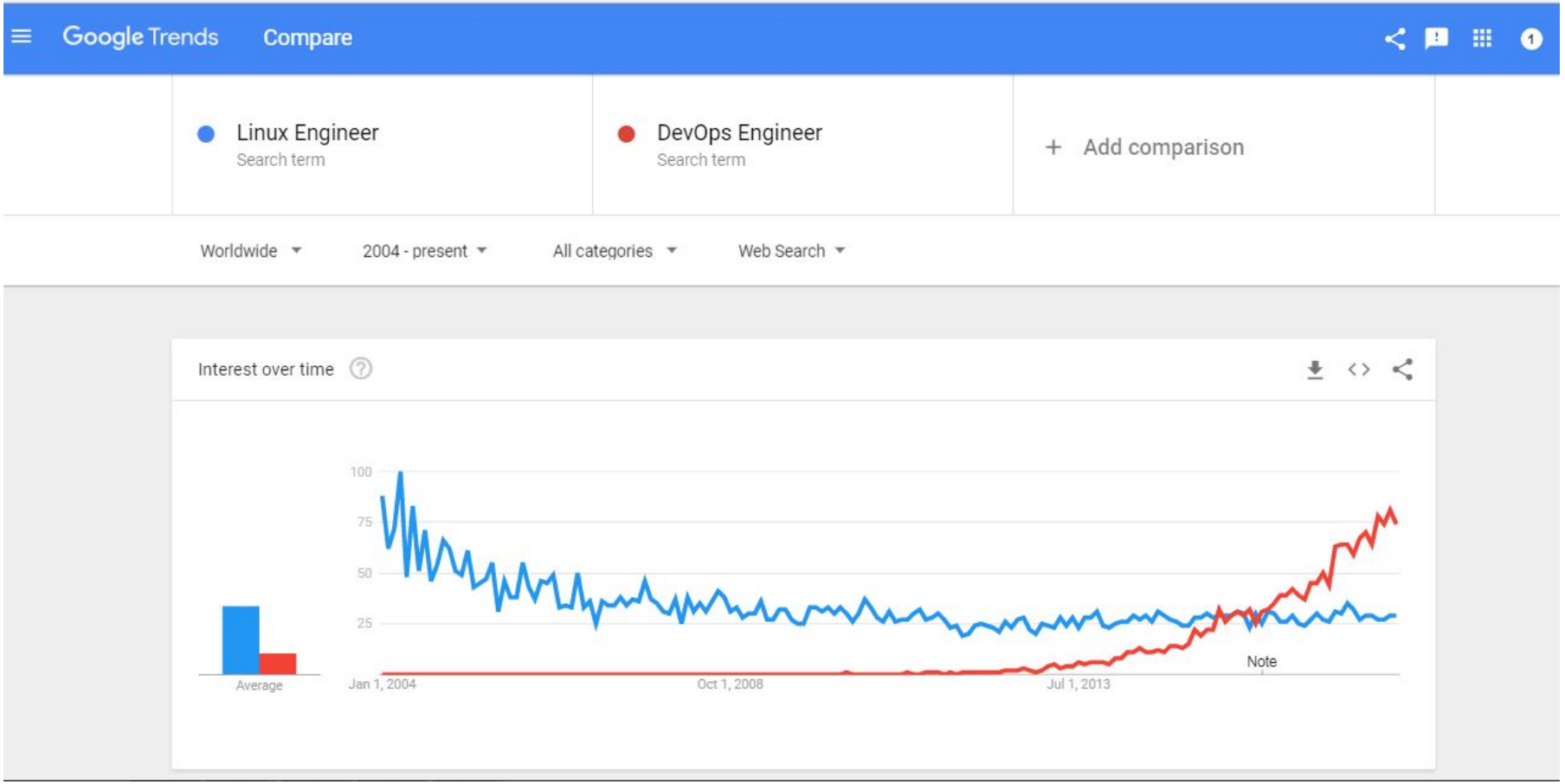
DevOps (CI, CD, GIT, Jenkins, Python)

Achieve single click code deployment on the multiple servers with automatic code quality testing and revert on failure

- Ashutosh Taiwal

Future of DevOps – Google Trends



What’s DevOps?

- “**DevOps**” as a term was first coined in 2009 by Patrick Debois

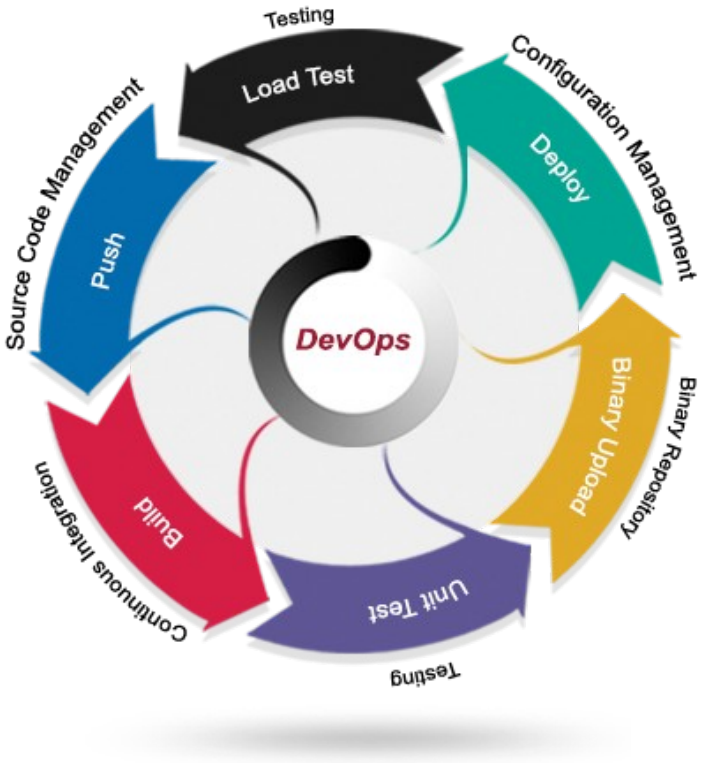
- A clipped compound of "**development**" and "**operations**")

- Software **dev**elopment **(Dev)** and software **op**eration **(Ops)**

DevOps Life Cycle

DEVOPS
LIFE CYCLE

- ✓ Push Code
- ✓ Fetch Changes
- ✓ Run Unit Tests
- ✓ Build Artifacts
- ✓ Store Artifacts
- ✓ Provision environment
- ✓ Deploy Your Build
- ✓ Run Load & Functional Tests
- ✓ Dev -> QA -> Staging -> Production



DevOps - Spanning across entire delivery pipeline

Continuous Integration | Continuous Delivery

CI (Continous Integration)

- The process of automating the build and testing of code every time a team member commits changes to version control.
- Encourages developers to share their code and unit tests by merging their changes into a shared version control repository after every small task completion.
- Committing code triggers an automated build system to grab the latest code from the shared repository and to build, test, and validate the full master branch

Benefits - CI (Continous Integration)

- If you're going to fail, then fail early
- Separate software builds from software deployments
- Enables automated testing
- Increases confidence in the software

CD (Continous Deployment)

- The process to build, test, configure and deploy from a build to a production environment.
- Multiple testing or staging environments create a Release Pipeline to automate the creation of infrastructure and deployment of a new.
- **Continuous Integration** starts the CD process and the pipeline stages each successive environment the next upon successful completion of tests.

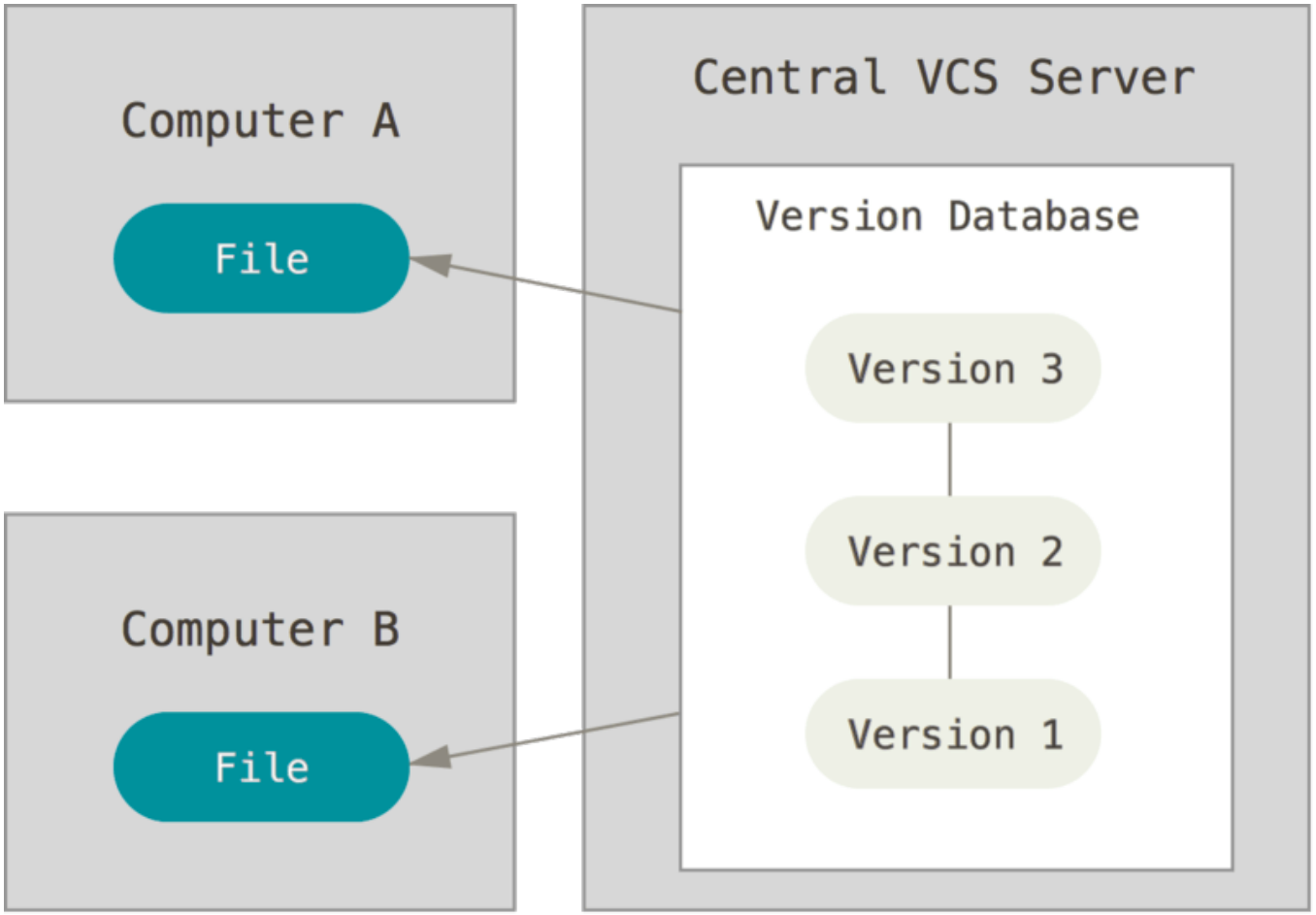
Benefits - CD (Continous Deployment)

- Automate the repetitive tasks and focus on actual testing.
- Make deployments frictionless without compromising security.
- Connect your existing tools and technologies (such as CI providers, DevOps tools, or scripts) into a harmonious workflow.
- Integrate teams and processes with a unified pipeline.
- Provide a single view across all applications and environments
- Improve overall productivity.

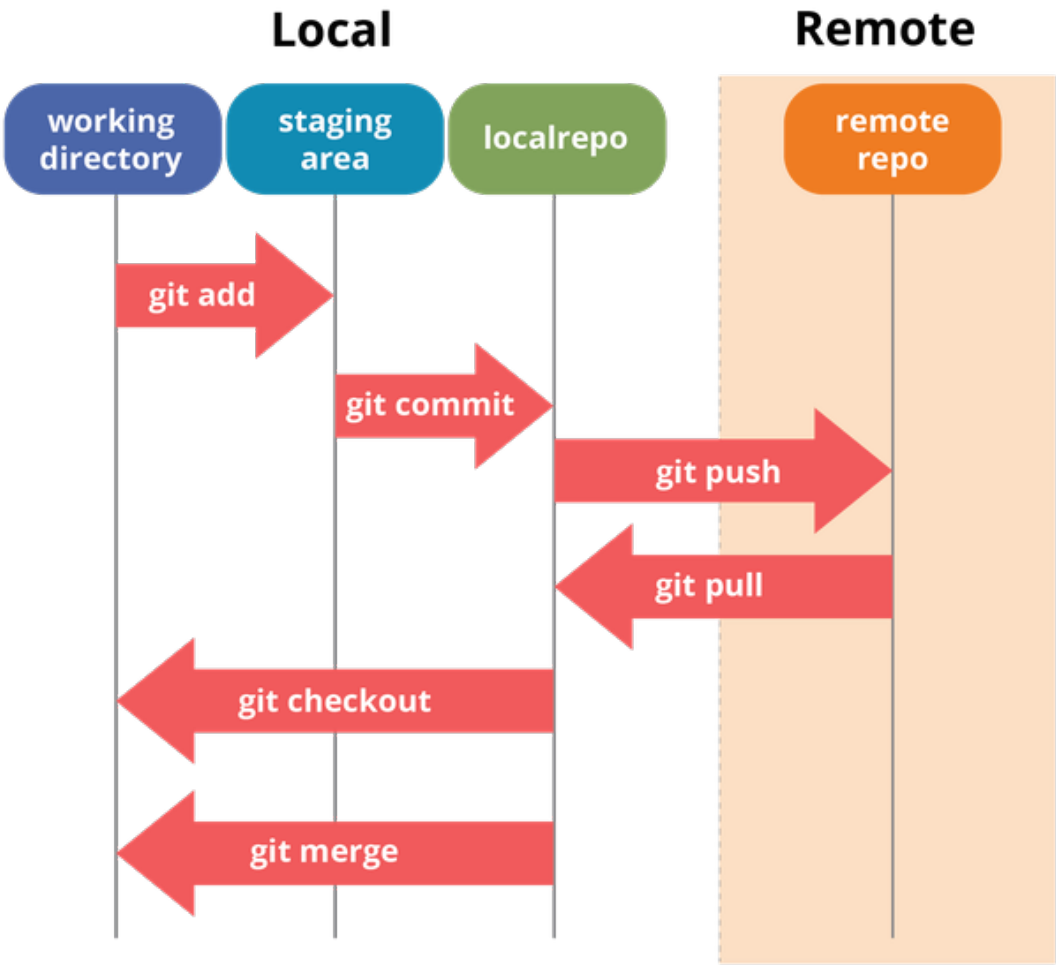
Git

- A “**version control system**”
- It can be used to keep track of changes in any set of files
- Git was created by **Linus Torvalds** in 2005 for development of the **Linux kernel**
- Git is free software distributed under the terms of the GNU General Public License version 2.

What’s “version control system”?



How VCS works – GIT?



Jenkins

- Jenkins is an open source automation server
- It’s written in **Java**
- It helps to automate the non-human part of software development process, with continuous integration and facilitating technical aspects of continuous delivery
- It supports version control tools
- Builds can be triggered by various means -
  - by commit in a version control system
  - by scheduling via a cron
  - by requesting a specific build URL
  - Many more..
- Jenkins functionality can be extended with plugins.

What’s “Python”?

- Python is a widely used high-level programming language
- Created by **Guido van Rossum** and first released in 1991
- An interpreted language
- It has a design philosophy that emphasizes code readability
- A syntax that allows programmers to express concepts in fewer lines of code

Thank you.